

POLITECNICO DI TORINO

---



Faculty of Engineering  
Master in Mechatronic Engineering

Master Thesis

# Sensing leaf-springs for axle load monitoring

University tutor  
prof. Enrico Galvagno

Candidate:  
Siham El Makkaoui

Company tutors  
Mollebalestra SPA  
Ing. Marco Cucchi  
Ing. Spyros Papaspyrou

---

July 2022

To my family, friends, my professor and to my company tutor.

# Acknowledgements

I would like to express my deep gratitude to Professor Enrico Galvagno who guided me throughout this project providing active support and precious advice. The main part of this research was accomplished at the Mollebalestra company, so I would like to thank Dr. Spyros Papaspyrou and his team for this great opportunity that allowed me to grow incredibly both professionally and personally. I would also like to offer my sincere gratitude to my mentor Dr. Marco Cucchi for his constructive guidance, technical directions and for all his daily help during my research activity. I wish to express my special thanks to all the people of the Mollebalestra team for the support they provided me during these months clearing my doubts and explaining all new concepts. Finally, I would like to express my deepest and heartfelt thanks to my family and to all my friends for having always been by my side throughout this five-years journey.

# Summary

Nowadays human safety and comfort are the most important parameters in designing and producing a vehicle, that is why every organization ensures the quality of components used in the vehicle. Leaf spring is also a component of vehicles which plays an important role in human comfort during driving. It acts as a structural component of the suspension system and influences mostly the human safety. Accidents caused by heavy trucks have been increasing recently. This is mainly due to the fact that overloaded vehicle have steering difficulty and long braking distance. Therefore, this work aims to develop an automatic payload measurement for heavy vehicles, so that drivers, police officers and suppliers can monitor vehicle loads while on board the vehicle and remotely. For the ease of installation, high accuracy, and low cost, this work proposes to paste strain gauges onto each leaf spring in vehicle suspensions, instead of load cells. Based on measured output current in each suspension, the vehicle payload is calculated. Moreover, for promoting the accuracy of calculated payload, this work develops an electronic device able to analyze data and transfer them from the sensor to a well-equipped IoT platforms to build a proof-to-concept prototype for the future usage.

# Contents

|          |  |           |
|----------|--|-----------|
| <b>1</b> | <b>Introduction</b>                                  | <b>1</b>  |
| 1.1      | Motivation . . . . .                                 | 1         |
| 1.2      | State of art . . . . .                               | 1         |
| 1.3      | Objectives of the thesis . . . . .                   | 6         |
| <b>2</b> | <b>Leaf springs</b>                                  | <b>7</b>  |
| 2.1      | Functionality . . . . .                              | 7         |
| 2.2      | Fabrication process and thermal treatments . . . . . | 8         |
| 2.3      | Parabolic leaf-springs . . . . .                     | 9         |
| <b>3</b> | <b>Theoretical model</b>                             | <b>13</b> |
| 3.1      | Static analysis . . . . .                            | 13        |
| 3.2      | FEM analysis . . . . .                               | 23        |
| <b>4</b> | <b>Technologies</b>                                  | <b>29</b> |
| 4.1      | Selection of sensor technologies . . . . .           | 29        |
| 4.2      | Kimax 1 and 2 . . . . .                              | 40        |
| 4.3      | Data logger . . . . .                                | 42        |
| 4.4      | Arduino board . . . . .                              | 43        |
| <b>5</b> | <b>Migration of data inCloud</b>                     | <b>45</b> |
| 5.1      | Transmission chain . . . . .                         | 45        |
| 5.2      | IoT platform . . . . .                               | 46        |
| 5.3      | Arduino-IoT connection . . . . .                     | 48        |
| <b>6</b> | <b>Experiments and results</b>                       | <b>50</b> |
| 6.1      | Experiment's set up . . . . .                        | 50        |
| 6.2      | Calibration off-truck . . . . .                      | 51        |
| 6.3      | Sensor acquisition of the load . . . . .             | 54        |
| 6.4      | Uncertainty . . . . .                                | 67        |

|                                    |           |
|------------------------------------|-----------|
| 6.5 IoT analysis results . . . . . | 72        |
| <b>7 Conclusion</b>                | <b>82</b> |
| <b>A An appendix</b>               | <b>84</b> |
| <b>References</b>                  | <b>90</b> |

# Chapter 1

## Introduction

### 1.1 Motivation

With the rapid development of economy, the improvements of transportation industry continues to increase. The problem of overload or overrun in the heavy vehicle payload became a problem. Therefore, how quickly and simply to know the vehicle load and how to effectively limit overload inexpensively has become a key issue. The suspension system in a vehicle affects the behavior of the whole vehicle, i.e., vibration characteristics including the drive comfort or driving stability. Leaf springs are commonly used in the vehicle suspension system and are commonly designed to withstand the maximum anticipated loads. Furthermore, the high stiffness of the spring can lead to undesirable oscillations exerted on the vehicle potentially causing damage. In order to identify the payload, integrated weighting sensors are necessary. Installing a sensor directly on the leaf spring allows to see the effective load on trucks axles. Adding a complementary electronic device able to transfer useful data, could be the solution for suppliers to see real time data remotely and deliver any needed help to customers. Plus, since the acquisition of the load value changes with the ageing of the material, we can prevent accidents due to failure, thanks to an exhaustive analysis of real time data acquisition stored in-cloud.

### 1.2 State of art

During the last few years, different solutions have been developed in order to prevent the frequent traffic accidents and bridge collapses. The problem became even more serious in the last three years; this is why a law has been implemented to let everyone check the transportation freight regularly. The European Council Directive 2015 /719, amending (EC) 96/53 states that, by May 2021, all Heavy-Duty Vehicles (HDVs) including Heavy

Goods Vehicles (HGVs), buses and coaches using European roads must have taken measures to identify vehicles and vehicle combinations suspected of exceeding their maximum permissible weights in traffic, according to Article 10d [10]. This Directive establishes maximum common measures to ensure that road safety is not jeopardized and that degradation to roads, bridges and tunnels is minimal.

The evolution of weighting systems has highlighted two dominant technologies for measuring vehicles' payloads: weigh-in-motion (WIM) and on-board weighing (OBW). WIM measuring systems are external to the vehicle measured, and can be differentiated from OBW systems, which are installed into the vehicle itself [11]. On-Board Weighing (OBW) systems enable the weight data to be communicated at any time from a moving vehicle.

One example of WIM technology is the one certificated by the Australian National measurement institute, that uses the Central Weighing model named Supaweigh 4000 road vehicle weighing-in-motion instrument. The Supaweigh 4000 system uses a SUPA120 Weighing Platform and an indicator which receives signals from the weighing platform which it processes to determine the axle loads, and sums the axle loads to determine the total vehicle mass [14].



Figure 1.1: WIM SUPA120 Weighing Platform and indicator

In the other hand the sensors of OBW systems can be installed to measure individual wheel loads, axle loads or axle group loads. These measurements would allow the inspection authorities to filter the vehicles so that only overloaded vehicles are stopped for inspection. There are two general types of OBW systems: static On-Board weighing and dynamic On-Board weighing.

Static OBW weigh the vehicle when it is stationary, and it's the most reliable system with an accuracy of 2% within 95% of readings, or an error of  $\pm 500$  kg. In a dynamic OBW system, the weight values are monitored continuously with an inaccuracy of  $\pm 1-3\%$ . Many different sensors can be used as load cells, strain gauges, and air or hydraulic pressure transducers, but also other sensors, such as accelerometers and displacement sensors may be used. A successful example of the use of OBW systems to manage heavy vehicle access and compliance to the road network, is the implementation into Australia's Intelligent Access Program (IAP) [3].

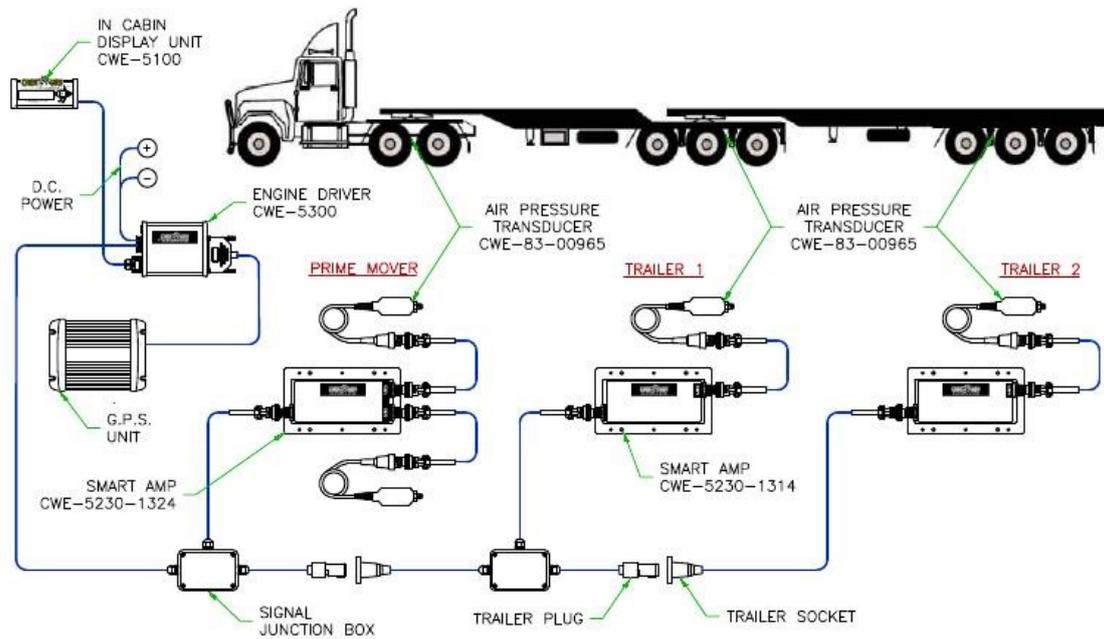


Figure 1.2: Example of dynamic OBW system

The most famous method to measure the weight is based on load cells that employ either strain gauges, where a measurable change in resistance occurs at every deformation of the material, or piezo materials that produce electrical charge when stressed. The strain sensor measures the strain of the axle body and thus enables the load to be calculated. The sum of the axle loads gives the vehicle weight including the load. A strain gauge converts the mechanical deformation into readable electrical value. Contrarily to a single strain gage, load cells comprise an array of strain gauges that convert the mechanical load into readable values. This makes load cells more precise and sensitive than strain gauges, but more expensive [9]. To attach on leaf springs, monoaxial strain gauges are usually preferred since the strain experienced by the suspension is mostly longitudinal, so we can neglect the one that is transversal. The sensor is bonded to the axle in which the strain is to be measured. The installation of the sensor can be done with an epoxy glue. Single strain gages are typically installed on steering front axles having leaf spring suspension or on leaf spring

suspended trailer axles. Strain gauges are the cheapest load sensors in the market and the simplest ones. A strain gauge depends on the electrical conductance principle. Whenever the sensor material is enlarged within the limits of its elasticity, it gets narrower and longer. When the strain gage sensor is compressed, it gets shorter and broader, changing its resistance. Strain gauges have been used to measure induced deflections on vehicle suspensions. For example, U.S. Patent No. 4,042,049 describes a vehicle weighing system in which respective strain measuring transducers are mounted on a pair of equalizing beams positioned on opposite sides of the vehicle [12]. The center of the leaf spring is usually attached to the vehicle, and their tandem axles are secured to their ends. The strain transducer gives the load induced deflection of the beams, providing an indication of vehicle weight. The objective is accomplished by measuring the load induced strain occurring in the springs' edges. In the case of a standard parabolic leaf spring with only one leaf a uniform tension is assumed, so the position of the strain gauge is unimportant. This can be proved numerically or via an FEM (finite element method). Anyway, the main concept is that in a cantilevered beam the bending moment varies from its maximum at fixed end to its minimum at free end. If we have beam of uniform cross section. Then the bending stress (strength) will be maximal at fixed end and minimum at free end. But if the cross section changes of size from fixed to free end (decreasing section modulus from max to free) then the stress can remain constant as the bending moment decreases; their ratio remains constant, measuring an uniform strength throughout the leaf spring. Here, in the figure 1.3, we have two types of leaf springs; the longitudinal ones in blue and transversal ones in green. For the experiments we will install the strain gauges in the two longitudinal suspensions located in the rear axles, since we want to measure a tension.

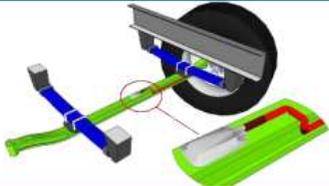
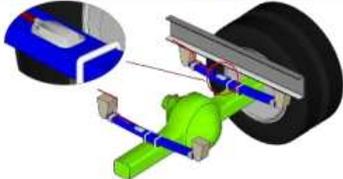
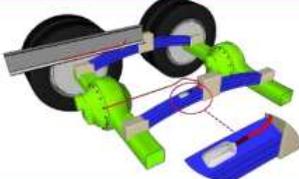
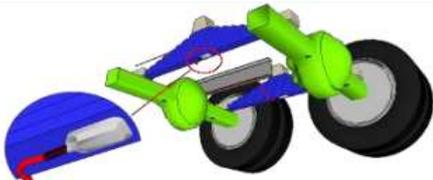
| Part number | Description                                  | Illustration  |
|-------------|--|---|
| 10205 (M)   | SG sensor intended for measuring compression |   |
| 10204 (L)   | SG sensor intended for measuring tension     |   |
| 10203 (L)   | SG sensor intended for measuring compression |   |
| 10204 (L)   | SG sensor intended for measuring tension     |  |



Figure 1.3: SG sensor on a leaf spring

Strain gauges are convenient because they can be used; their relatively low cost and the fact that they are reusable are an advantage. Electrical resistance of strain gauges compensates most of the disadvantages of mechanical gauges. Installing resistive sensors

like strain gauges on the heavy vehicles before selling them could be very convenient for both trucks suppliers and customers. Therefore, it is worth to rely on this type of sensor to develop a complete system that can give the right instruments to prevent most of the risks.

### 1.3 Objectives of the thesis

This experimental thesis is part of the research program of MOLLEBALESTRA company, a market leader in the production of leaf springs for road and rail freight vehicles. Being able to predict with sufficient accuracy the risk of failure of a leaf spring is the final objective that underlies a series of preparatory tasks for a digital architecture to support the collection of data from the vehicle, transmission in the cloud and their analysis with statistical and simulation tools in order to define a reliable digital twin with predictive models based on historical series and on the actual working condition and stress of the leaf spring. In particular, the thesis work exploits a MOLLEBALESTRA vehicle equipped with a strain gauge sensor applied to a leaf spring and connected to a data logger in the cabin. In order to overcome the limitations of using a data logger, MOLLEBALESTRA has designed a device based on Arduino to replace the data logger, offering also the possibility of on-board processing and wireless connection for data reading. The thesis work aims to:

- 1) Calibrate and validate the response of the sensor applied on the vehicle's leaf spring, first in the workshop on a dynamic measuring machine and then on the vehicle.
- 2) Validate the data collected by the electronics in terms of consistency with the attitude of the empty and fully loaded vehicle
- 3) Analyze the ability to interconnect with CLOUD systems for real-time collection of data from the stationary and moving vehicle
- 4) To analyze the reliability of using a strain gauge to weigh the load, compared to the use of load cells.

The result of the thesis work will then be used to identify digital services which, by exploiting the data collected and analyzed, will be able to:

- 1) Provide indications on the weight and optimal arrangement of the load
- 2) Provide information on the useful life of the leaf-spring based on its actual stresses
- 3) Identify the geographical point and the stress that caused the leaf-spring to break.

## Chapter 2

# Leaf springs

### 2.1 Functionality

The main function of leaf springs is to supply comfort to the passengers by minimizing the vertical vibration caused by the non uniformity of road irregularities. Generally, leaf springs have high strength and low modulus of elasticity within the longitudinal direction. High fatigue resistance is one of the foremost important properties for leaf springs since they carry the complete load of the vehicle and knowledge dynamic load during driving conditions. Leaf springs are guided by the SAE J1123 standard titled Leaf Springs for car Suspension, and might be full, semi, or quarter-elliptic with one or more leaves. There is a large variety leaf springs in the market and their characteristics differs looking on the ultimate usage of the vehicle. Leaf springs are subdivided into longitudinal and transverse leaf springs. Longitudinal leaf springs are used only on rigid axles, more precisely on heavy vehicles and trailers. Figure 2.1 contains a weight comparison between the previously exclusively used multi-layer leaf springs and modern parabolic springs. Springs with only one layer, named single-leaf springs, are usually used for light commercial vehicles. Transverse leaf springs, in contrast, can provide the springing on either side of the axle; they were previously utilized in independent wheel suspensions of passenger cars.

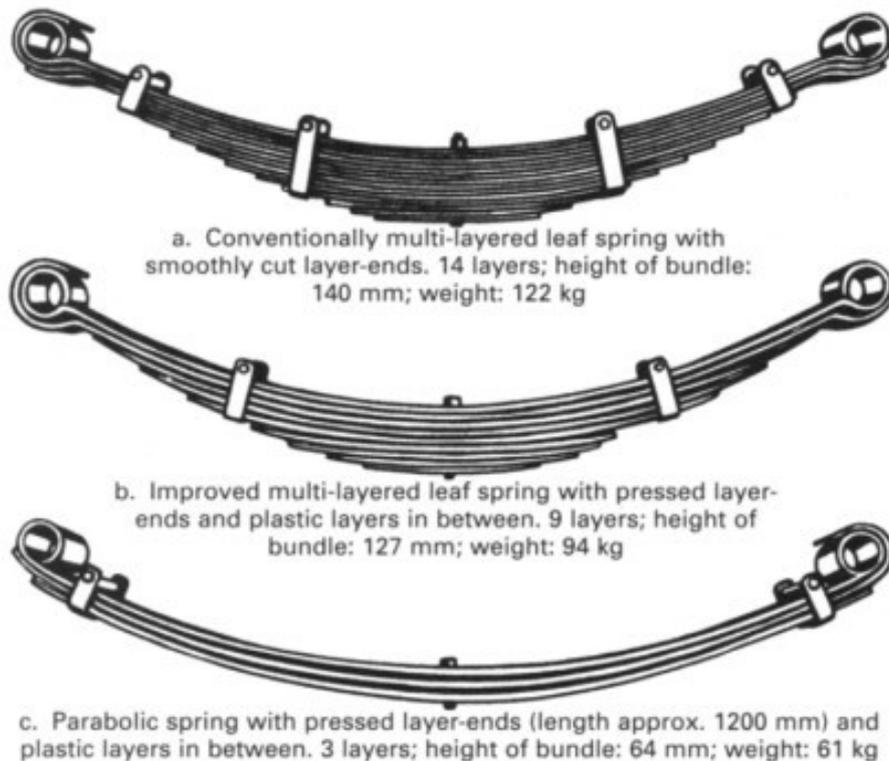


Figure 2.1: Types of leaf-springs

## 2.2 Fabrication process and thermal treatments

The leaves are usually characterized by an initial curvature or cambered so they're going to tend to rectify themselves under the load. The leaves remain by means of a band that surround them at the middle or by a bolt passing through the middle. Since, the band exerts an expansion effect, the effective length of the spring for bending is the overall length of the spring minus width of the band. Just in case of a middle bolt, two-third distance between centers of U-bolt should be subtracted from the length of the spring to search out the effective length. The spring is clamped to the axle housing by means of U-bolts. The top of the longest leaf called master looks like an eye fixed employed by the bolts to secure the spring to its supports. the extra leaves of the spring are called graduated leaves. so as to forestall friction between leaves, the ends of the graduated leaves are cropped in various forms. The clips are located at intermediate positions within the length of the spring, in order that the graduated leaves also share the constraints within the full-length leaves when the spring is stressed. The material used for leaf springs is sometimes a noticeable steel having 0.90 to 1.0% carbon. The leaves are submitted to a thermal process after the forming process. the warmth treatment of spring steel products higher robustness and thus greater carrying capacity, increased range of deflection and optimization in fatigue properties. The

heat treatment process is the most affecting procedure on the spring performances [1]:

1. Heating for Hardening: Any metal, or alloy with high ductility, is additionally used for springs, or any alloy which can be heat treated to high strength and good ductility before, or after forming could even be used. Leaves are heated to very high temperatures in an Oil-fired hardening furnace. Usually temperature is maintained between 850°C and 950°C.
2. Cambering: the longest leaf is called the master leaf. The quantity of bend on the spring is known as camber. The camber is provided so even at the utmost load the deflected spring shouldn't touch the machine member to which it's attached. The central fastener is required to carry the leaves of the spring. The machine used for this operation is that the press. Leaves are twisted to required radius employing a press. All the leaves are tested using cambering gauges.
3. Quenching: Hot leaves are kept in tray and quenched in oil bath to urge martensite structure. Martensite is that the hardest sort of steel crystalline structure. Martensite is made in carbon steels by rapid cooling. Machine used is carrying quench oil bath. After cooling down the structure of spring becomes rigid and this property isn't required. But this process is required to line the leaves to correct radius after cambering. To eliminate hardness tempering is completed.
4. Tempering: Tempering can be a process of heat treating, which is used to increase the pliability. Leaves are reheated to decrease the rigidity to required level. Electric heated temperature oven is used for this process. Hardness of the leaves is computed by Brinell hardness testing. This process is additionally done to alleviate stresses. The process is done at temperatures between 540 and 680°C. Tempering process involves heating of leaves below their re-crystallization temperature so cooling them using water or air.

## 2.3 Parabolic leaf-springs

A more recent implementation is the parabolic leaf spring. This design is characterized by fewer leaves whose thickness changes following a parabolic curve. During this design, friction between the leaves is unwanted, and for this reason the contact is kept only between the springs at the ends and at the center. Spacers are also used between the two leaf springs to avoid friction at different points. Additionally to the carrying capability, the main advantage of parabolic springs is their greater elasticity, that ensures higher driving quality and comfort. Also, the comfort characteristic of parabolic springs is better and not as "stiff" as conventional multi-leaf springs. The leaf spring used for our research is a parabolic leaf spring to which the strain gauge will be attached in order to measure the load on the truck.



Figure 2.2: Model of the parabolic leaf-spring with Solidworks

In particular, the leaf-spring considered during this study, is the 10.730.00 with section 70x18mm , as seen in the following figure taken from the Mollebalestra industrial document. This leaf spring is consisting of the main leaf (10.730.91) and second leaf (10.730.02). The first two digits usually refer to the main customer (10); the next three digits identify the leaf spring under consideration (730); the last two digits refer to the type of single leaf/leaf (00 a full leaf spring; 30 a leaf spring; 50 a main leaf spring; 91 a leaf spring with a silent block; 92 a main leaf with bushings; 01 a main leaf without silent block/bushings; 02 a second leaf; 03 a third leaf and so on; 71 to 79 a reinforcements; 80 to 99 à modified full leaf spring; 56 a shims); The spring under consideration belongs to the family of parabolic (two-leaf) leaf springs (the rolling profile follows a parabola in order to have a uniform state of tension along the span of the leaf spring without resorting to the trapezoidal plan profile characteristic of traditional, i.e. non-rolled, leaf springs).

2 – Leaf springs

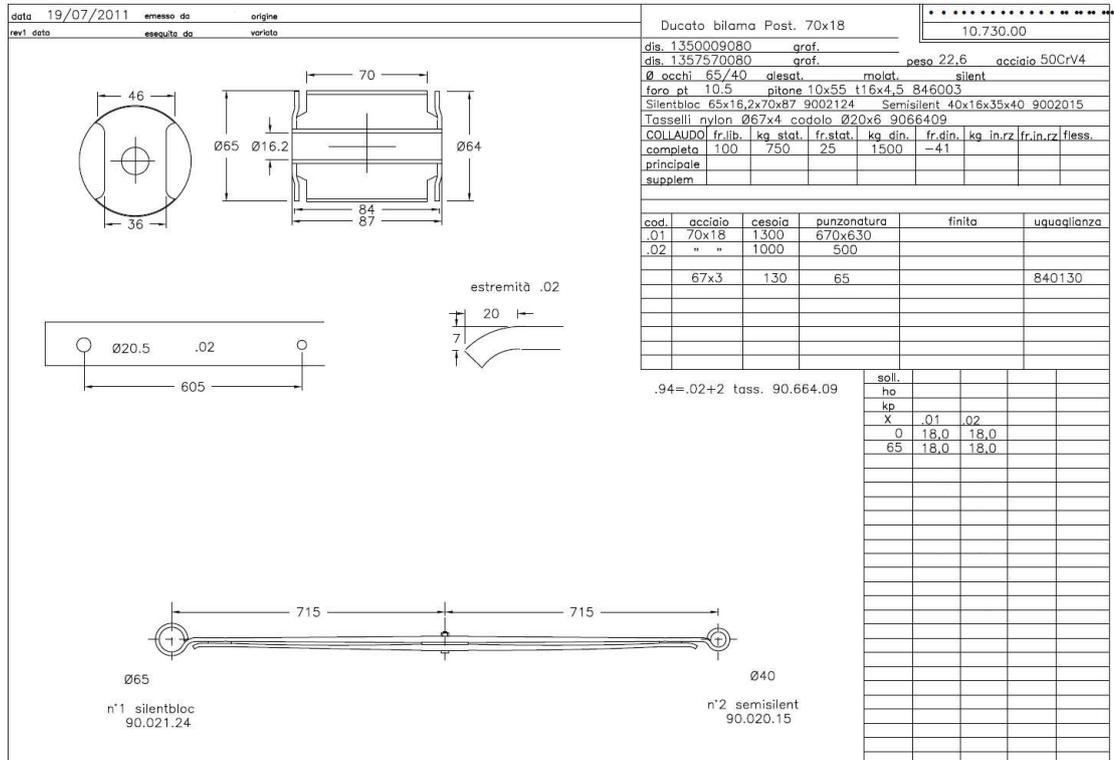
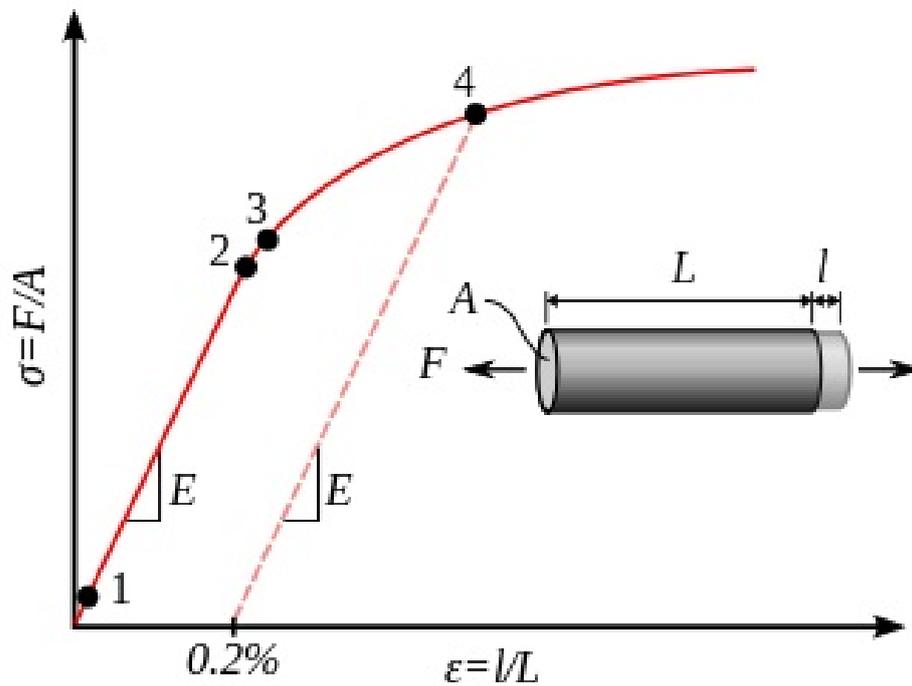


Figure 2.3: Technical data of the leaf-spring

Another important fact is that on the application of stress on an elastic body, the strain experienced by the body is directly proportional to the stress applied. A force applied provokes a strain or a deformation. If force does not exceed the elastic limit, according to Hook's law, the strain will increase with stress. But if the stress applied overpass the maximum strength allowed, permanent deformation will occur. In engineering and materials science, a stress-strain curve for a material gives the relationship between stress and strain also written as:

$$\sigma = E\epsilon$$

It is obtained by gradually applying load to a test coupon and measuring the deformation, from which the stress and strain can be determined as the following figure shows.



Typical yield behavior for non-ferrous alloys.

- 1: True elastic limit
- 2: Proportionality limit
- 3: Elastic limit
- 4: offset yield strength

Figure 2.4: Stress-strain curve

The yield strength point is the point in the stress-strain curve at which the curve levels off and plastic deformation begins to occur. When a yield point isn't easily computed depending on the shape of the stress-strain curve an offset yield point is applied. This value is commonly set at 0.1 or 0.2% of the strain. The offset value is given as  $R_{p0.2}=310\text{MPa}$ . High strength steel and aluminum alloys don't present a yield point, therefore the offset yield point is employed on these materials.

## Chapter 3

# Theoretical model

### 3.1 Static analysis

A light truck was selected as the research object. Its rear suspensions are equipped with the parabolic springs. Most precisely we used a small lorry , the BOXER CABINATO TELAIO of 1975kg made by Peugeot. This vehicle is able to carry 1525kg as weight as we can see in the registration document of the vehicle.

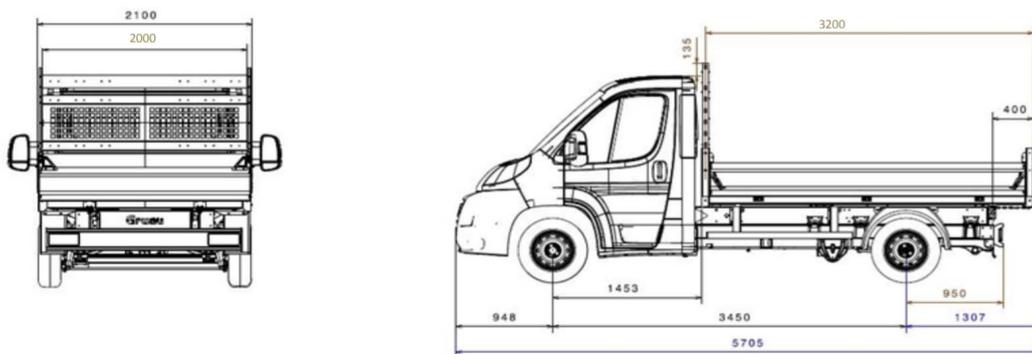


Figure 3.1: Boxer Cabinato telaio Peugeot truck

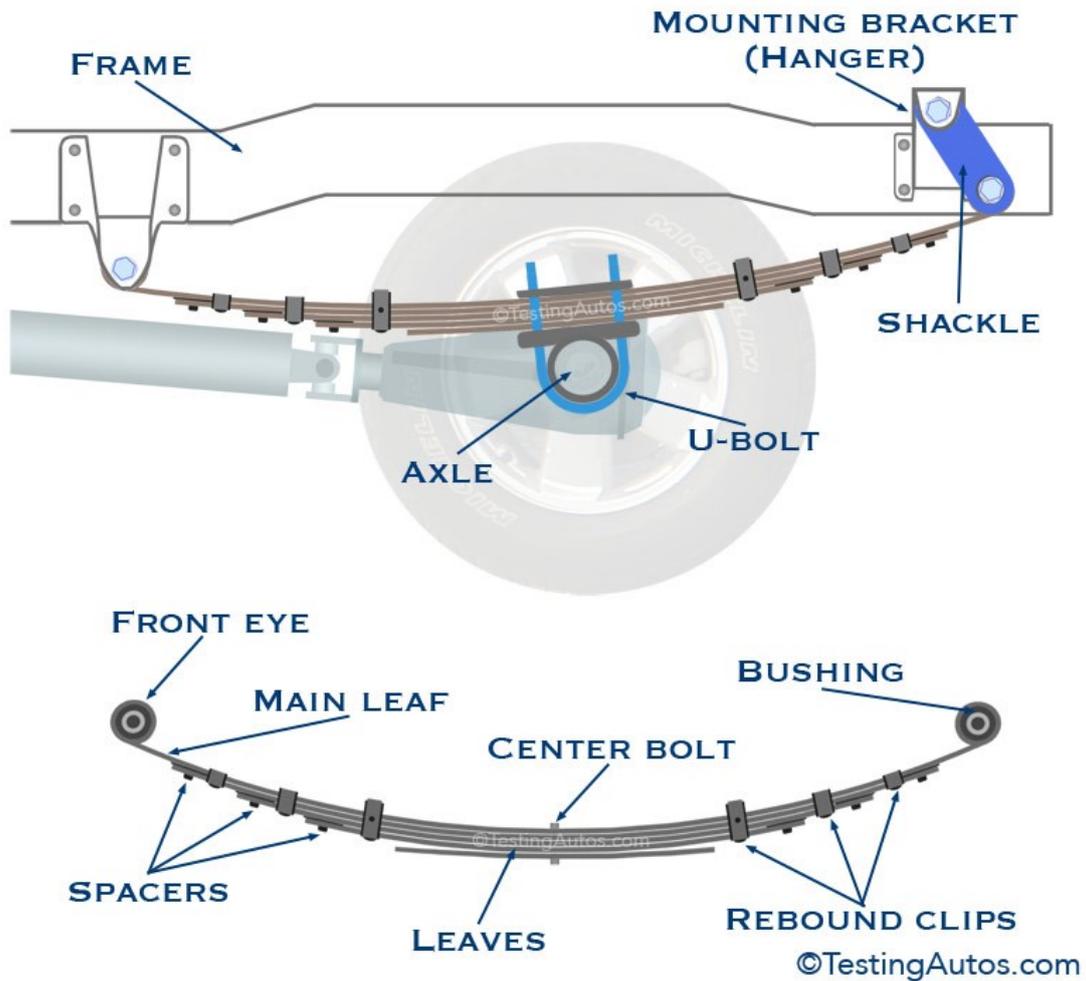
(D.1) PEUGEOT  
(D.2) Y C2MAU GY  
(D.3) BOXER 335 ONNICAR  
(E) VF3YC2MAU12D62138  
(F.1)  
(F.2) 3500 (F.3) 6000 (G)  
(I) 28.04.2017  
(J) N1  
(J.1) AUTOCARRO PER TRASPORTO DI COSE  
-USO PROPRIO  
(J.2) K0 (CASSONE)  
(K) LBNB073EST007GR  
(L) 2 (N.1) (N.2)  
(N.3) (N.4) (N.5)  
(O.1) 2500 (O.2)  
(P.1) 1997,00 (P.2) 096,00 (P.3) GASOL  
(P.5) AH03  
(Q) (S.1) 3 (S.2)  
(U.1) 80 (U.2) 2813  
(V.1) (V.2)  
(V.3) 0,1040 (V.5) 0,000400  
(V.6) 0,51 (V.7) 173,0  
(V.9) 2015/45

IMPOSTA  
 DI BOLLO  
 ASSOLTA  
 IN MODO  
 VIRTUALE

Figure 3.2: Registration document

It's important also to know how leaf springs are installed on the truck, to understand

its movement whenever we charge the lorry with a load. As we can see in the following figure, the front end of the leaf spring is connected to the mounting bracket attached to the frame. The axle is located to the leaf spring center bolt and secured by U-bolts. The rear eye of the leaf spring is connected to the frame through a shackle that can swing back and forth, allowing for spring expansion. In our case, contrarily to the figure, we have only two leaves.



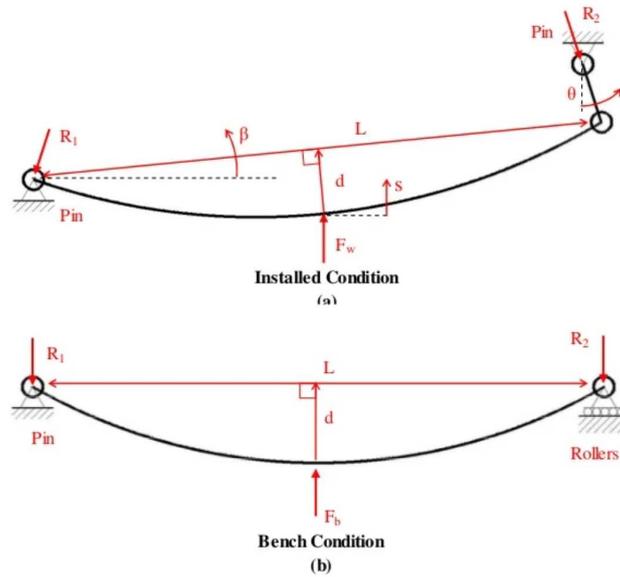


Figure 3.3: Leaf spring installation

Here we try to carry out a static analysis of the leaf spring behavior according to the charge of the lorry, because in dynamics things change slightly. For example, on dynamics, the leaf spring supports the double of the load supported during static analysis. Whenever we charge the lorry with a load, the chassis is subject to an inclination, that is more pronounced when we go closer to the front axle. This happens because the chassis is mostly supported by the front axle of the vehicle. This also means that if we measure the weight to which the leaf spring is subject, it will be more and more lower whenever we step away from the rear suspension. To better study the effect of the load distribution on the truck, a mathematical model of the truck was designed. For the static analysis, only one spring ( $k_1$ ) will represent the two rear leaf springs and another one for the two front suspensions ( $k_2$ ). The truck will be represented by a beam supported by two springs, and, then, the model will be studied with the load on it. The tracked vehicle weapon analysis is integrated over a half vehicle platform that is represented by the beam [2]. In the figure 3.4, the beam supported by 2 springs with different stiffness is shown. Friction can be neglected. The position and attitude of the system is defined by coordinates  $x_1$  and  $x_2$  and  $\sin(\theta) = \theta$ .

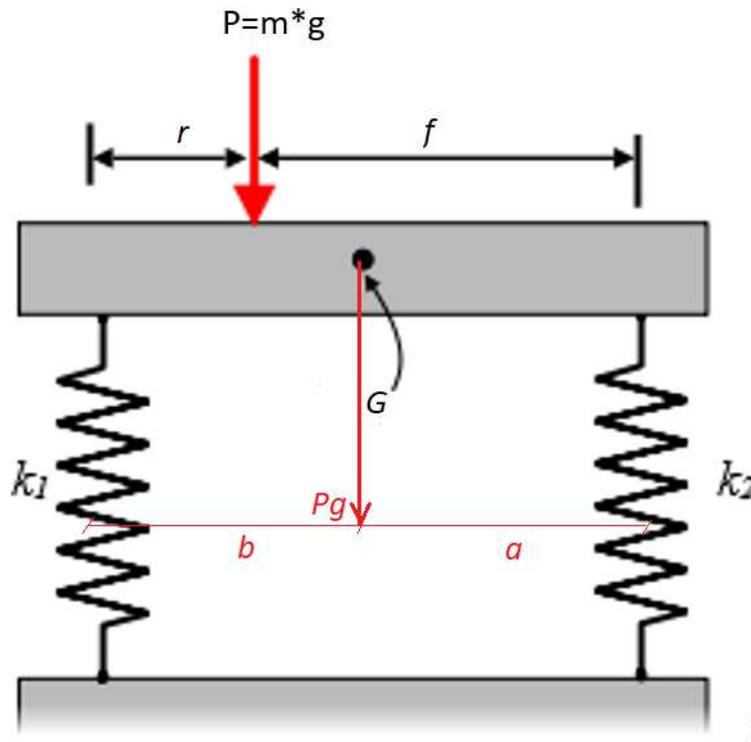


Figure 3.4: Real truck and model of the truck with leaf spring

Where  $R = m_{measured} * g = -k_1 * x_1$  and  $F = -k_2 * x_2$  are the forces on each spring

and  $P = m * g$  is the weight of the load on the vehicle. The equations of motion are derived from the sum of forces on body and sum of moments with respect to where the sensor is located. We will suppose that the rear and front suspensions are represented by springs with stiffness  $k_1$  and  $k_2$ , respectively. The aim of our study is to compute the mass

$$m_{real} = \frac{P}{g}$$

on the vehicle. Once we find the theoretical relationship between the mass and the displacement, it will be compared with the experimental one. At the static equilibrium applied on the vehicle carrying the load  $P$  we will have :

$$\sum F_i = R + F - Pg - P = 0$$

$$M_2 = -fP - aPg + (r + f)R = 0$$

From those equations we can get the force  $m_{measured}$  function of the distance from the front axle  $f$  and the weight  $P = m_{real} * g$ . Half of the mass measured is sensed by the right rear leaf-spring and the other half by the left rear leaf-spring. The two strain gauges measurements are finally summed up and displayed to the driver. So, we obtain,

$$m_{measured} = \frac{R}{g} = \frac{fP + aPg}{gL} = \frac{f * m_{real} + a * m_{truck}}{L}$$

Where

$$L = r + f = a + b$$

Since during calibration the sensor doesn't take into account the load of the vehicle, the sensor detects only the mass measured function of the real load, equal to:

$$m_{measured} = \frac{R}{g} = \frac{fP}{gL} = m_{real} * \frac{f}{L}$$

Where  $P = m_{real} * g$  is the real load weight that we charge in the vehicle. So, if  $R$  is proportional to the distance from the front axle  $f$ , we can observe that when  $f$  is at its maximum  $L=f$  so,  $R=P$  and the sensor on the rear axle can compute the real value of the load mass. Whenever,  $a$  decreases the weight experienced by the rear axle decreases too. If the real mass of a load that we want to put in the truck is  $m_{real} = 160kg$  we can plot the curve of the mass experienced by the rear axle function of  $f$ .

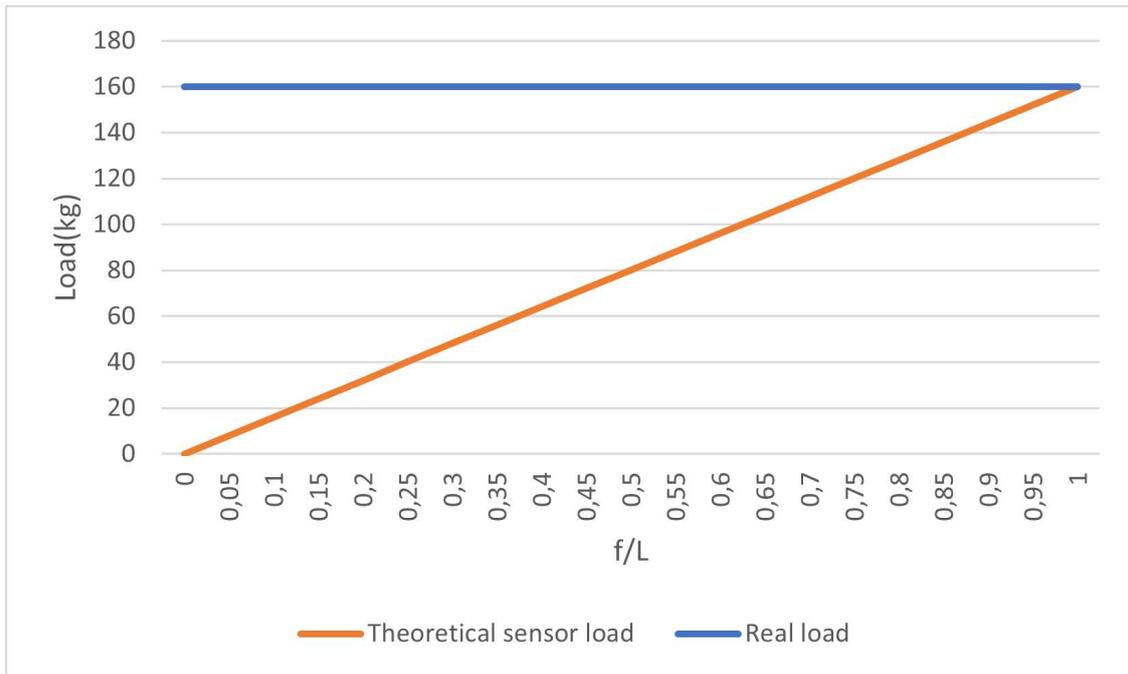


Figure 3.5: Mass function of the distance from the rear axle

In the figure 3.5 we can see the mass experienced by the rear axle function of  $f/L$ . Higher is the value of the distance  $f$ , lower is the change on the load  $m$  experienced by the sensor in the rear axle. Once we step away from the rear suspension the load decreases. The following graphic shows the error between the real load and the load measured by the sensor.



Figure 3.6: Error of the sensor measuring the load

The inclination of the road is important for the set up of the electronics. The road inclination will have an effect on the mass charged by the vehicle. In particular, the component related to the height of the load center of gravity will affect the measurements.

Using the same equilibrium equations of before and the following model in figure 3.7, we can conclude that,

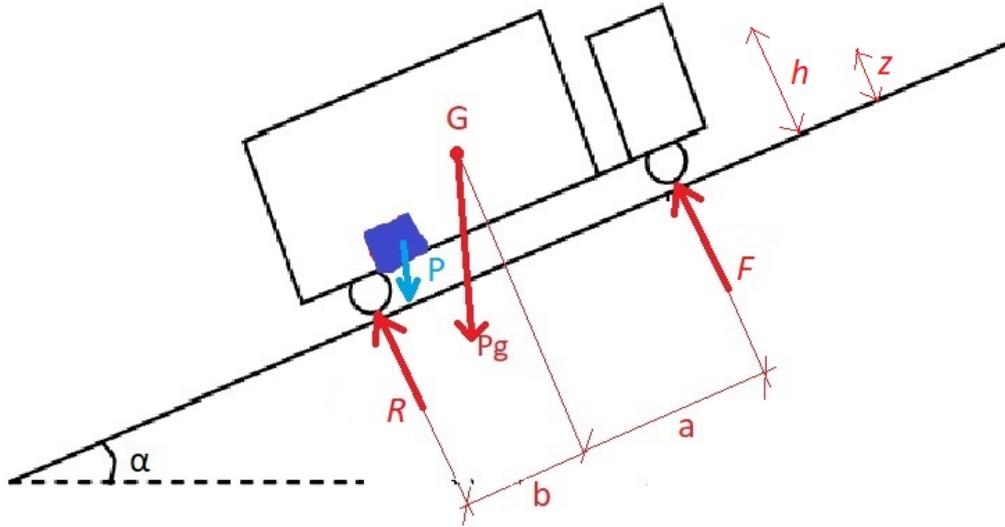
$$\sum F = R + F - Pg * \cos\alpha - P * \cos\alpha = 0$$

$$M_2 = -aPg * \cos\alpha - hPg * \sin\alpha - fP * \cos\alpha - zP * \sin\alpha + (f + r)R = 0$$

So the sensor will detect,

$$m_{measured} = \frac{m}{L} * (f\cos\alpha + z\sin\alpha)$$

where  $z = 950mm$  for a load of 160kg



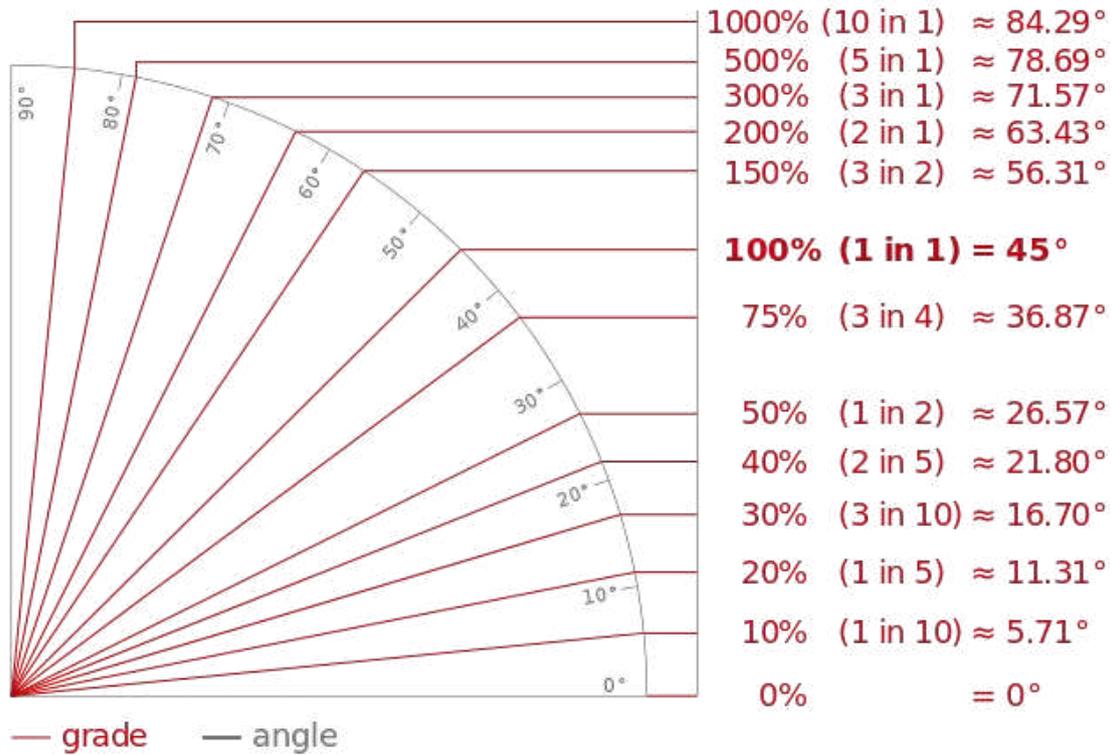


Figure 3.7: Road inclination model and angles information

In the following graph, it can be seen the curve representing the measured mass function of  $f/L$  and assuming the angle of inclination  $\alpha = 20^\circ$ . Compared to the curve without any inclination, we can observe that the load carried by the rear axle is lower if the inclination increases.

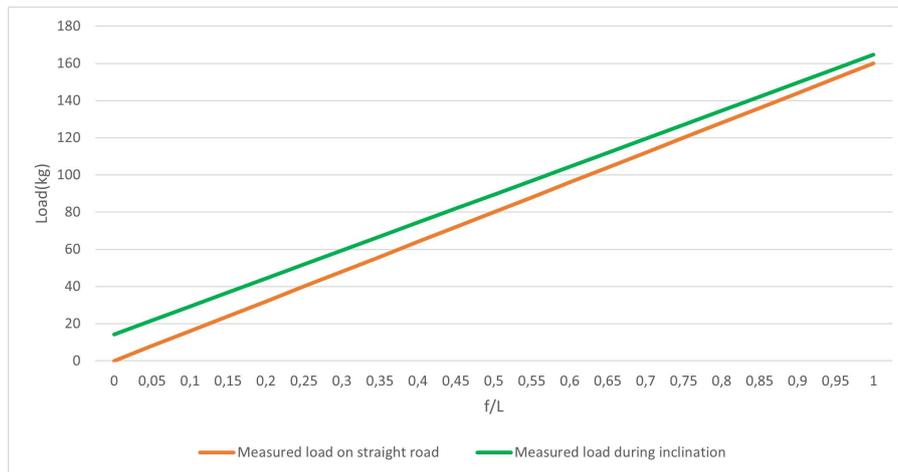


Figure 3.8: Measured load with respect to the distance from the front axle at  $\alpha = 20^\circ$

It's also important to consider the height of the load center of gravity "z" that affects the load measurements. In fact, in the following figure, it is shown that the load sensed by

the leaf springs increases with the height.

This means that the load measured will be equal to:

$$(P \pm 9)kg$$

The error equal to 9 kg will be calculated by computing an average between all errors measurements showed by the previous curve.

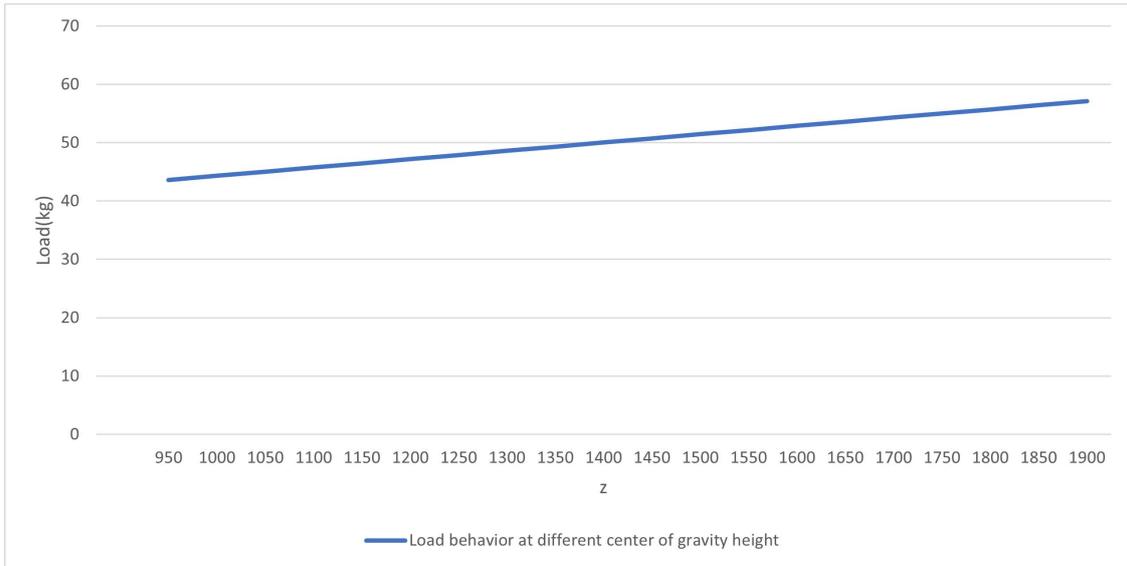


Figure 3.9: Measured load with respect to the load height

Usually for the road, in the extreme circumstance we have an inclination of  $45^\circ$  (a mountain road) as we can see in figure 3.7. In this case, the load sensed by the rear axle is only about the 60% (100kg) of the real load (160kg). However the maximum road for general roads is 30% that corresponds to an angle of  $16.70^\circ$ . In this case, the rear axle carry most of the real load (about 95%). In the following figure a graphic is shown where  $f/L$  is equal to 0.2 and the only variable is the angle of inclination of the road. It can be seen that for small angles, most of the load is carried by the rear axle.

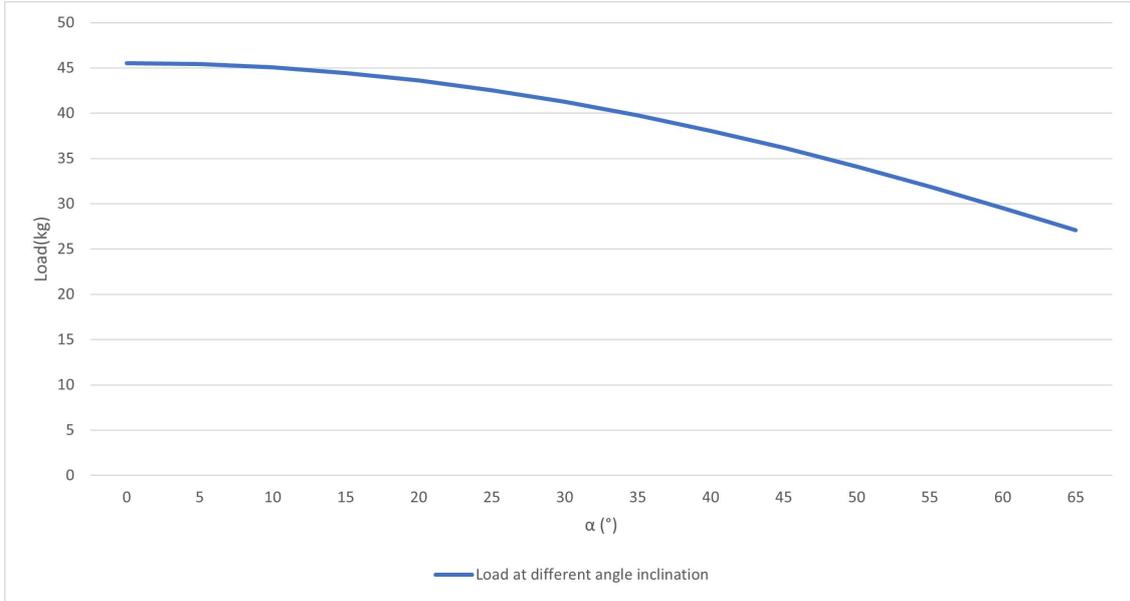


Figure 3.10: Measured load with respect to different roads inclinations

## 3.2 FEM analysis

The finite element method (FEM) is a technique used to perform finite element analysis (FEA) of any given natural phenomenon. To perform the computations, the model of a beam clamped on the left and free to slide on the right was taken into account. This type of analysis uses the correlation between load and bending stress, valid for laminated leaf-springs:

$$\delta_{FEM} = \frac{6Px}{n_f w t^2}$$

Where  $P$  is the load applied in the middle of the leaf-spring as shown in figure 3.11,  $x$  ( $x=485\text{mm}$ ) is the distance between the center of the silent block and where the sensor is placed,  $t$  ( $t=14\text{mm}$ ) is the thickness of the leaf spring where the sensor is located,  $w$  ( $w=66\text{mm}$ ) is the width of the leaf spring and  $n_f$  the number of leaves (in this case  $n_f = 2$  because two leaves are present). Of course, since the sensor has been calibrated, it will be able to detect the right load  $P$ , even if it is applied on the middle of the suspension and not where the sensor is located.

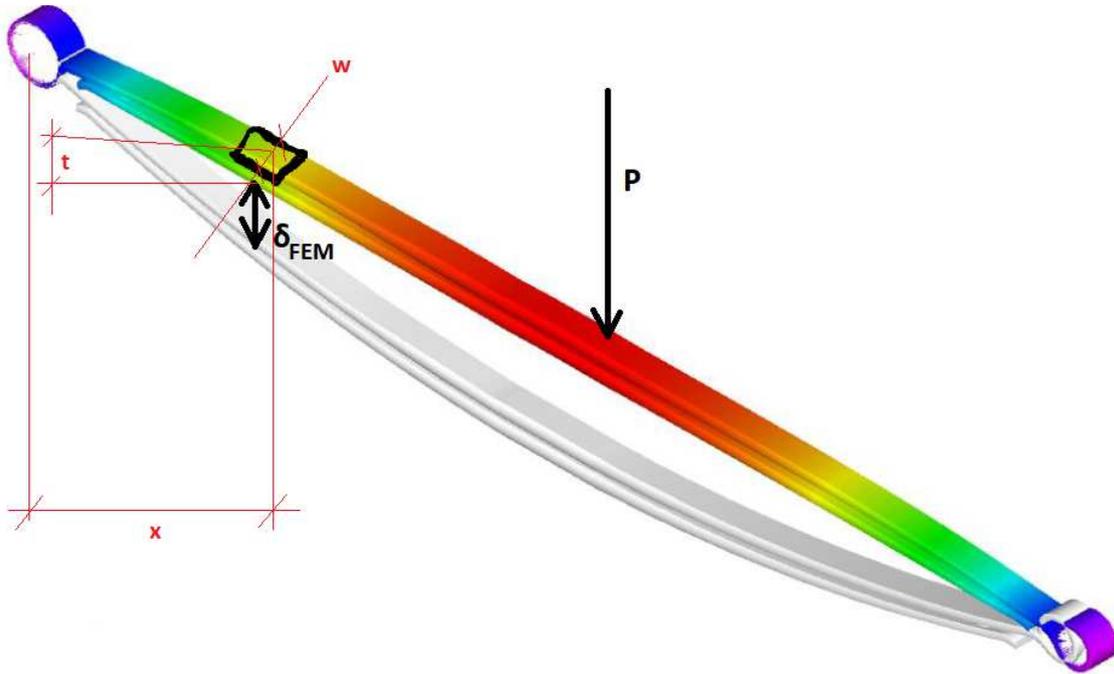


Figure 3.11: Bending stress deformation parameters

In this case the FEM analysis is necessary to justify the strain gauge position on the leaf spring. Generally, the sensor should be attached in a position with the highest strain value in order to have a strain gauge able to predict a right value of weight. The only problem is that the position with maximum strain or stress is also the most sensitive to rupture. Also, to install the strain gauge we need to grind the leaf spring, so we remove part of the material as we can see in figure. 3.12. This means that the material becomes more sensitive because the stress augments. This is why, attaching the sensor where the strain is maximum is not the best option. For all those reasons, we decide to attach the sensor where the strain is high but not maximum.



Figure 3.12: Grinding of the leaf spring material

To analyze a phenomenon, the FEM subdivides a wide area into smaller portions that are called finite elements. This is achieved by creating the « mesh » of the object: a mesh is a representation of a larger geometric space by small-scale cells. A mesh divides space into elements (or cells or zones) over which the equations can be solved, which then approximates the solution over the larger domain. Usually, the mesh becomes more dense for sensitive points like holes and notches, but in this case those elements were removed because the only aim is to understand which is the most stressed point of the beam to locate the sensor. The most stressed part, shown in the following figures in red cannot be chosen, since grinding the material is essential to install the sensor, otherwise it will become too much sensitive and more likely to break. For this reason the sensor was located on a part of the leaf-spring that is still highly stressed but more rigid, shown in the figure in orange. When stress is applied to a solid, a deformation occurs, which gives rise to internal inter-molecular forces in the solid that oppose the forces being applied. If the applied forces are reduced, then the stress can be carried by internal molecular forces, and the object will present a new equilibrium state. Once the force is removed, the object comes back to its previous equilibrium state. If the force is too high, a permanent deformation can occur or even a complete structural failure. That's why we need to study the response of the leaf spring material in sense of deformations, before analysing the stress at which the material is submitted, as shown in the following figures.

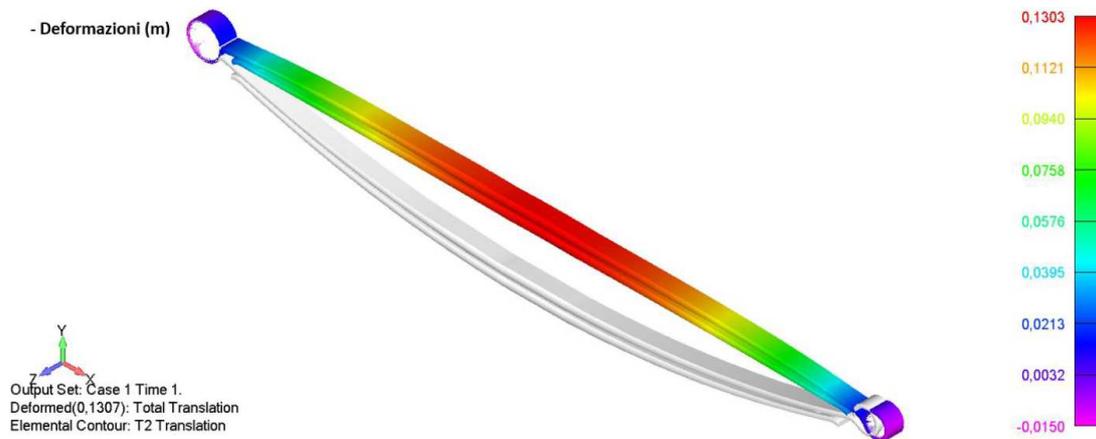


Figure 3.13: Leaf spring deformations

Two different load conditions, corresponding to the test conditions, were considered for the FEM: static load (750kg) and dynamic load (1500kg). Usually, the rule of thumb, to test leaf springs maximum loading is to submit them to a dynamic load that is double of the static one. But, generally, leaf springs are realized by manufacturers to carry 1.5 times the dynamic load during the leaf spring life-time, in order to keep a safety margin. For the material properties the nominal values were used (51 CrV4,  $E=210$  GPa,  $G=79$ GPa,

$\nu=0.29$ ). The most stressed points were identified by considering the values of the ideal stresses defined according to the von mises criterion. For both loading conditions, the most stressed areas were identified at the supported end and at the outer edge of the two blades (figure 3.14 and 3.15).

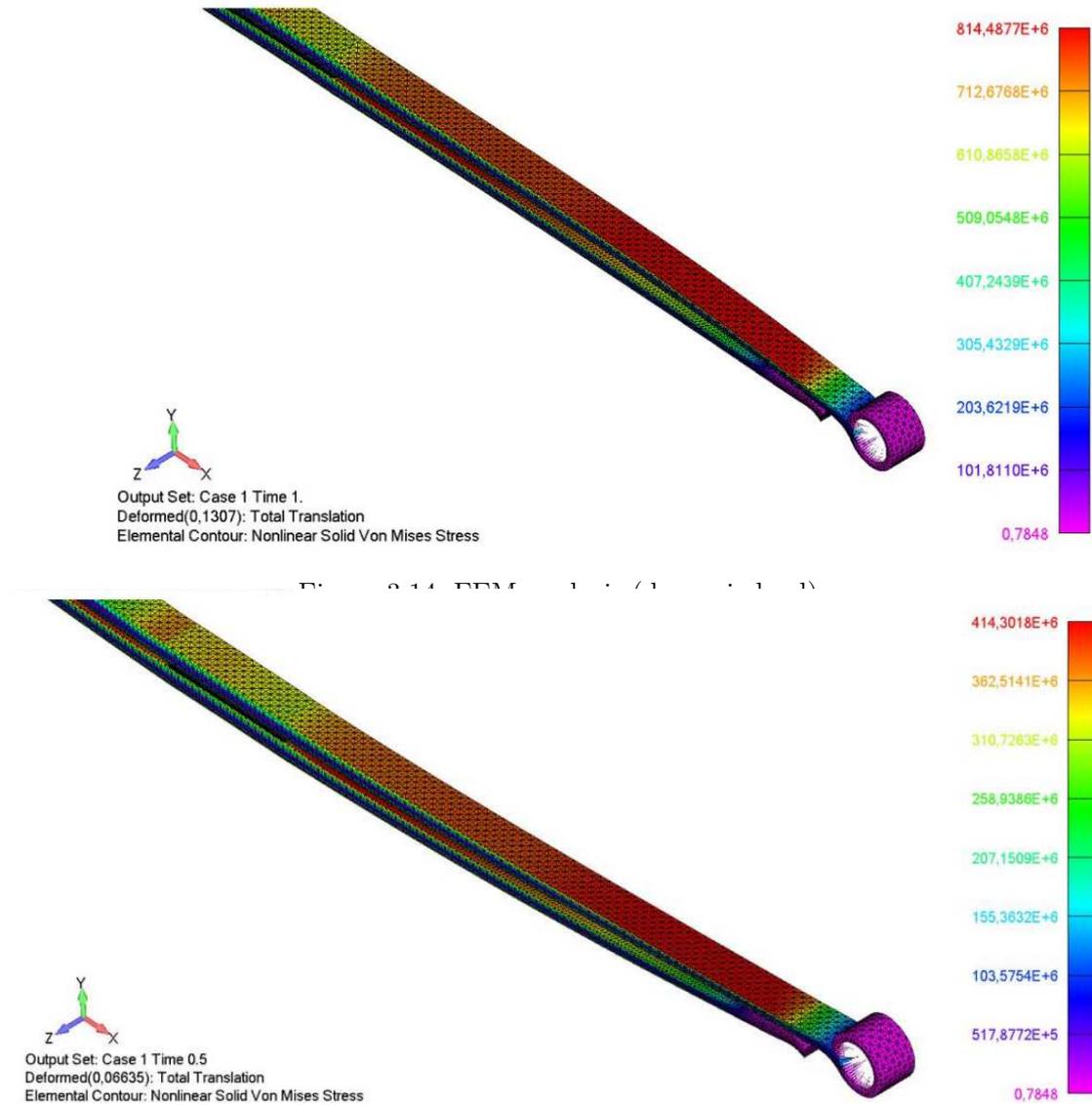


Figure 3.15: FEM analysis (static load)

To determine where to attach the sensor and, most importantly, which one to choose, it's important to see the behavior of the elongation with the FEM analysis. The goal is to find where the leaf spring presents the maximum elongation in order to past there the strain gage. The value of elongation is the distance measured from the center of the leaf spring to the center of the leaf spring silent bloc. So, as before, the model considered is clamped on the left and free to slide on the right. The simulation was made using the finite

element analysis on the model seen before, taking into account the elongation parallel to the leaf spring, defined as:

$$\epsilon = \frac{\Delta L}{L}$$

Using the values performed by the FEM simulation, a curve on excel has been realized, on figure 3.16. The elongation expressed on  $\frac{\mu m}{m}$  has been plotted. As shown in the following picture the elongation is higher on the edges, where two peaks can be noticed. So, where the elongation is higher the sensor will be attached. Only one sensor will be installed, on the right side of the beam, considering that the two elongations are symmetrical. The straight line represents the rigid plane on the middle of the leaf spring used for installation on the vehicles.

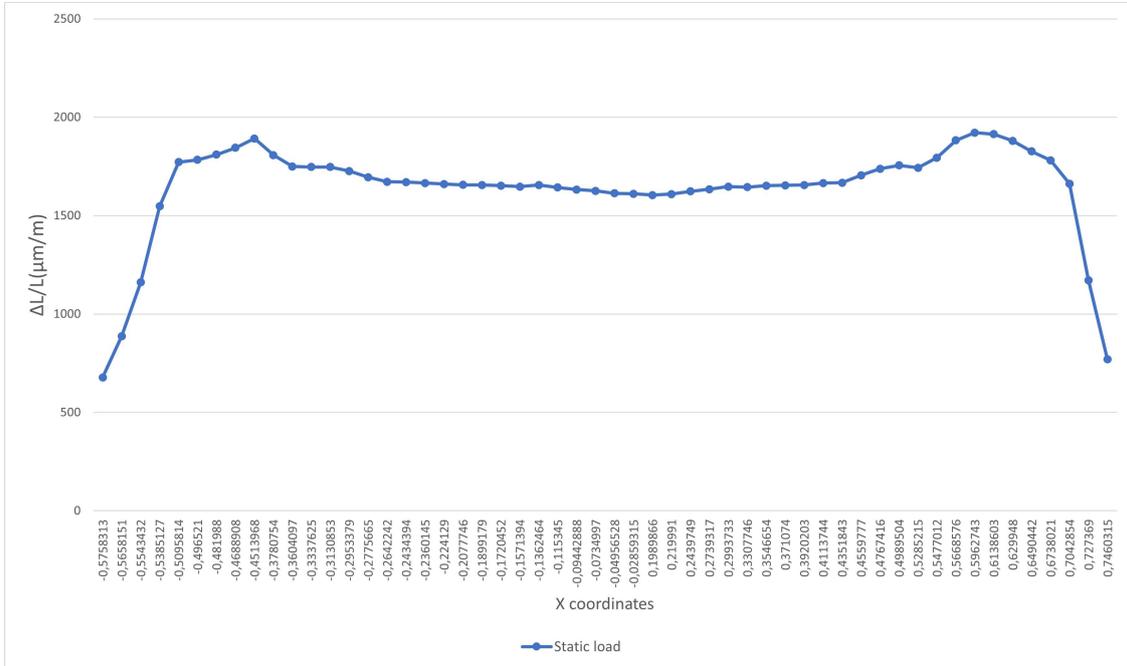


Figure 3.16: Elongation ( $\mu\epsilon$ ) of the leaf spring for the static and dynamic load

It can be seen, thanks to the previous graphic that the maximum elongation obtained is about  $1800\mu\epsilon$ . This value will be useful to determine the suitable sensor. Thanks to this last FEM analysis, the gain can be determined as follows:

$$K_{FEM} = \frac{F}{\epsilon} = 0.4kg/\mu\epsilon$$

Now that the elongation is known the next step is to choose the suitable sensor considering the sensor life duration. This relationship between the sensor life and the elongation is represented by the Wohler curve, explained in the following picture.

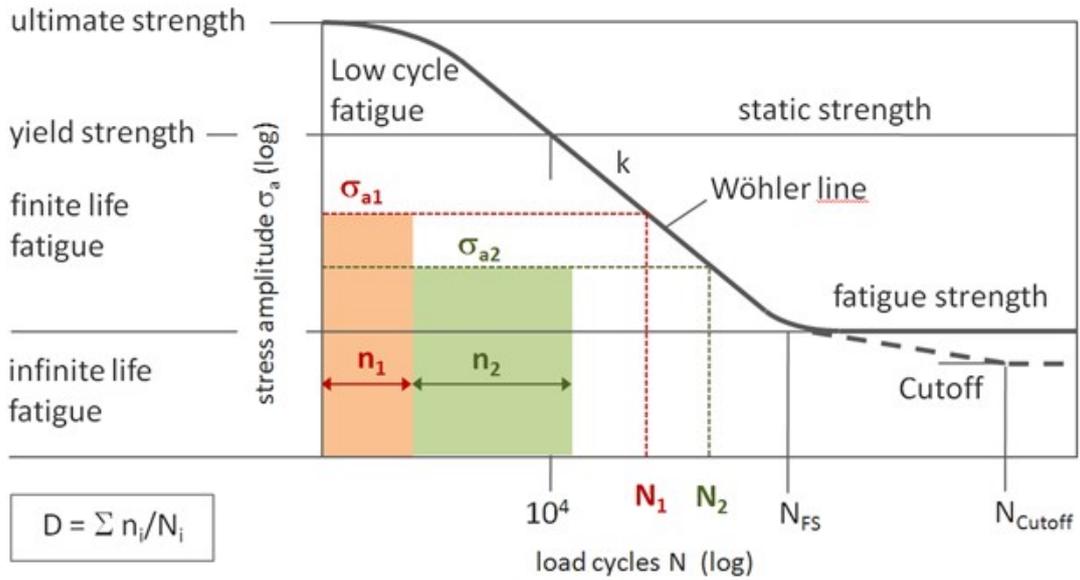


Figure 3.17: Wohler curve explanation

The main issue is that to have good sensibility of the weight on the truck the strain elongation should be as high as possible, but on the other hand, to have life long sensors the strain should be kept low. The goal is to find the right trad off between life duration and load sensibility

# Chapter 4

## Technologies

### 4.1 Selection of sensor technologies

The strain gauge converts the mechanical elongation and compression into its impedance value. The metallic strain gauge consists of a really fine wire in a grid. The grid allows to apply the strain in the parallel direction to the foil. The grid is attached to the carrier, that is connected to the test specimen. The strain experienced by the test specimen is proportional to the strain applied to the strain gauge, that will react with a variation of impedance.

The following picture and table present the strain gauge characteristics.

| Item                     | Linear strain gauge for stress measurements |
|--------------------------|---|
| Active length            | 3mm   |
| Temperature compensation | 6 /°K                                       |
| Grid material            | Konstantan alloy                            |
| Base material            | Phenolphthalein-oxygen                      |
| Nominal resistance       | (350±3) <i>ohms</i>                         |
| Series connection        | WA type                                     |
| Resistive load           | (250±1%) <i>ohms</i>                        |



Figure 4.1: Kimax sensor characteristics

The selection of the technologies was based on:

- Availability in the market
- Size
- Cost
- Efficiency

Another important factor was the maximum elongation found on the paragraph before. The strain gage needed should support an elongation of at least

$$\mu\epsilon = 1800\mu m/m$$

. A reliable and cheap method, that can be installed on leaf-springs by manufacturers, calibrated in advance and used to monitor leaf-springs' life and performances, is needed. Based on the following characteristics the strain gauge will be chosen.

|  |  |
|--|--|
| <b>1. Geometry:</b>                        | Number and position of grids (pattern)                         |
| <b>2. Strain gauge series:</b>             | Construction of strain gauge                                   |
| <b>3. Connections:</b>                     | Type and position  |
| <b>4. Temperature response adaptation:</b> | Material to which strain gauge temperature response is matched |
| <b>5. Active grid length:</b>              | in mm  |
| <b>6. Electrical resistance:</b>           | in Ohm   |

Figure 4.2: Strain gage criteria of selection

1. The first criteria is the geometry. On the following picture, it can be seen some of the geometries on the market. The choice of each geometry depends on which strain or stress the sensor should measure.

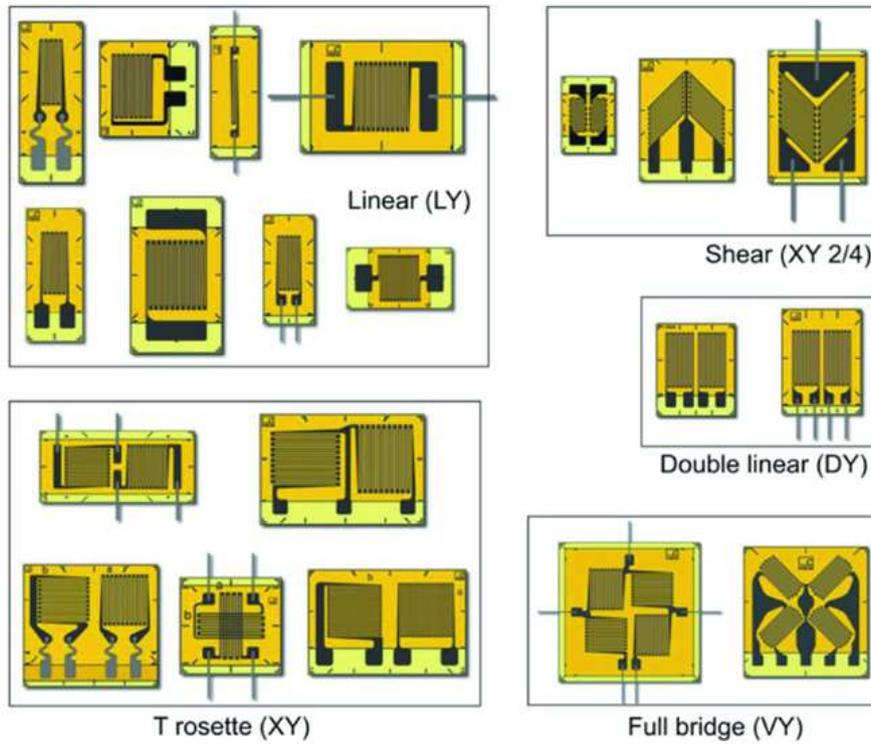


Figure 4.3: Strain gauge different geometries

For the strain gauge geometry the mono-axial needs to be selected since the elongation experienced by the leaf spring is mostly along the beam material. This elongation will be directly proportional to the weight applied on the truck. The strain gauge selected is the following one.

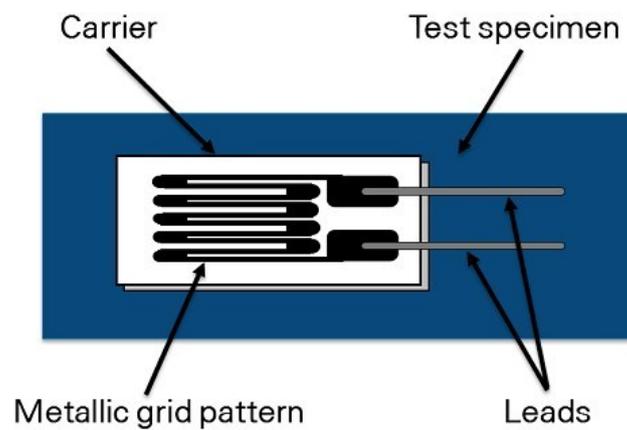


Figure 4.4: The bonded metallic strain gauge

The strain is actually the amount of deformation experienced due to an applied force. Strain can be positive, due to elongation, or negative, due to contraction.

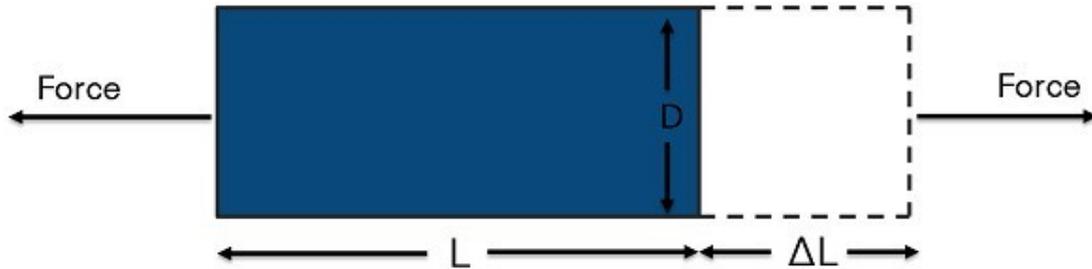


Figure 4.5: Model of the strain gauge

$$\epsilon = \frac{\Delta L}{L}$$

In our case, only axial strain is considered, that measures how a material stretches or compresses as a result of a linear force in the horizontal direction. Strain gauges measure the variation in impedance across a fine conductive foil. The gauge factor is the sensitivity of the strain gauge (usually 2). The change in resistance is converted into a variation of length.

$$\frac{\Delta R}{R} = K * \epsilon$$

Where K is the gauge factor, the strain (deformation), R the resistance variation due to strain and R the gauge resistance. In our case, since we use Quarter-Bridge strain gauge configuration, K or sensitivity is about 0.5 mV/V.

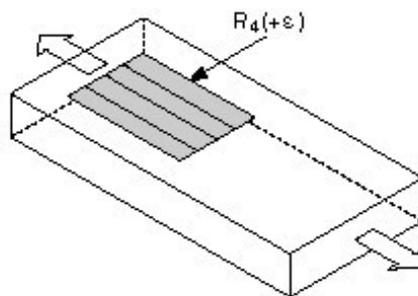


Figure 4.6: Quarter-Bridge strain gauge that measures axial strain

Strain gauge series are defined by the combination of strain gauge carrier and the measuring grid foil. The main factor determining the operating information is the alloy used for the grid. The strain gauge is supplied as a complete system that combines a particular

alloy with a carrier and includes other features, such as integral cables, encapsulation or solder pads. The following list is an example of different series materials:

A: Constantan in self-temperature-compensated form.

P: Annealed constantan.

D: Isoelastic.

K: Nickel-chromium self-temperature-compensated form.

The following is an example of table series selection for strain gauges.

| Gage Series           | Description and Primary Application  | Temperature Range  | Strain Range   | Fatigue Life                  |   |
|-----------------------|--|--|--|-------------------------------|---|
|                       |  |  |  | Strain Level in $\mu\epsilon$ | Number of Cycles                                      |
| EA                    | Constantan foil in combination with a tough, flexible, polyimide backing. Wide range of options available. Primarily intended for general-purpose static and dynamic stress analysis. Not recommended for highest accuracy transducers.  | Normal:<br>-100° to +350°F (-75° to +175°C)<br>Special or Short-Term:<br>-320° to +400°F (-195° to +205°C) | ±3% for gage lengths under 1/8 in (3.2 mm);<br>±5% for 1/8 in and over   | ±1000<br>±1500<br>±1200       | 10 <sup>6</sup><br>10 <sup>6</sup><br>10 <sup>6</sup> |
| CEA                   | Universal general-purpose strain gauges. Constantan grid completely encapsulated in polyimide, with large, rugged copper-coated tabs. Primarily used for general-purpose static and dynamic stress analysis. "C"-Feature gauges are specially highlighted throughout the gage listings of our Precision Strain Gages Data Book.                                | Normal:<br>-100° to +350°F (-75° to +175°C)<br>Stacked rosettes limited to<br>+150°F (+65°C)               | ±3% for gage lengths under 1/8 in (3.2 mm);<br>±5% for 1/8 in and over   | ±1500<br>±1500                | 10 <sup>6</sup><br>10 <sup>6</sup>                    |
| C2A                   | General purpose stress analysis strain gauges. Supplied with pre-attached cables for direct connection to instrumentation. RoHS compliant, lead-free solder.   | -80° to +180°F (-50° to +80°C)   | ±3%  | ±1700<br>±1500                | 10 <sup>6</sup><br>10 <sup>6</sup>                    |
| L2A                   | General-purpose stress analysis strain gauges. Supplied with pre-attached leadwire ribbons. RoHS compliant, lead-free solder.  | -100° to +250°F (-75° to +120°C)   | ±3%  | ±1700<br>±1500                | 10 <sup>6</sup><br>10 <sup>6</sup>                    |
| W2A<br>IPXBS<br>Rated | For water-exposure applications. Based on the CEA Series with Option P2 pre-attached cables, W2A strain gauges are fully enclosed with a silicone rubber coating and tested to 10 GΩ insulation resistance, 1 meter water depth, 30 minutes duration. Other requirements can be addressed on demand. RoHS compliant, lead-free solder.                         | -50° to +180°F (-50° to +80°C)   | ±3%  | ±1500                         | 10 <sup>6</sup>                                       |
| N2A                   | Open-faced constantan foil gauges with a thin, laminated, polyimide-film backing. Primarily recommended for use in precision transducers, the N2A Series is characterized by low and repeatable creep performance. Also recommended for stress analysis applications employing large gage patterns, where the especially flat surface eases mass installation. | Normal Static Transducer Service:<br>-100° to +200°F (-75° to +95°C)                                       | ±3%  | ±1700<br>±1500                | 10 <sup>6</sup><br>10 <sup>7</sup>                    |
| WA                    | Fully encapsulated constantan gauges with high-endurance leadwires. Useful over wider temperature ranges and in more extreme environments than EA Series. Option W available on some patterns, but restricts fatigue life to some extent.  | Normal:<br>-100° to +400°F (-75° to +205°C)<br>Special or Short-Term:<br>-320° to +500°F (-195° to +260°C) | ±2%  | ±2000<br>±1500<br>±1500       | 10 <sup>6</sup><br>10 <sup>6</sup><br>10 <sup>7</sup> |
| SA                    | Fully encapsulated constantan gauges with solder dots. Same matrix as WA Series. Same uses as WA Series but derated somewhat in maximum temperature and operating environment because of solder dots.  | Normal:<br>-100° to +400°F (-75° to +205°C)<br>Special or Short-Term:<br>-320° to +450°F (-195° to +235°C) | ±2%  | ±1000<br>±1500                | 10 <sup>6</sup><br>10 <sup>7</sup>                    |
| EP                    | Specially annealed constantan foil with tough, high-elongation polyimide backing. Used primarily for measurements of large post-yield strains. Available with Options E, L, and LE (may restrict elongation capability).   | -100° to +400°F (-75° to +205°C)   | ±10% for gage lengths under 1/8 in (3.2 mm);<br>±20% for 1/8 in and over | ±1000                         | 10 <sup>4</sup>                                       |

Figure 4.7: Standard strain gage series selection chart

In this case, the strain gauge selected has a constantan alloy for the grid and phenolphthalein-Oxygen material for the rest of the sensor.

2. The case under study uses a Constantan grid alloy for the strain gauge; more exactly the WA strain gauge. This type of strain gauge was selected based on a specific test "profile". The choice was made looking at the following table.

| Type of Test or Application                       | Operating Temperature Range       | Test Duration in Hours | Accuracy Required ** | Cyclic Endurance Req'd        |                  | Typical Selection      |                 |
|---|-----------------------------------|------------------------|----------------------|-------------------------------|------------------|------------------------|-----------------|
|   |                                   |                        |                      | Maximum Strain, $\mu\epsilon$ | Number Of Cycles | Gage Series            | M-Bond Adhesive |
| General Static or Static-Dynamic Stress Analysis* | -50° to +150°F (-45° to +65°C)    | <10 <sup>6</sup>       | Moderate             | ±1300                         | <10 <sup>6</sup> | C2A, L2A, W2A, CEA, EA | 200 or AE-10    |
|   |                                   | <10 <sup>6</sup>       | Moderate             | ±1300                         | <10 <sup>6</sup> | C2A, L2A, W2A, CEA, EA | AE-10 or AE-15  |
|   |                                   | <10 <sup>6</sup>       | Very High            | ±1600                         | <10 <sup>6</sup> | WA, SA                 | AE-15 or 610    |
|   |                                   | <10 <sup>6</sup>       | High                 | ±2000                         | <10 <sup>6</sup> | WK, SK                 | AE-15 or 610    |
|   | -50° to +400°F (-45° to +205°C)   | <10 <sup>5</sup>       | Moderate             | ±1600                         | <10 <sup>5</sup> | WA, SA                 | 600 or 610      |
|   |                                   | <10 <sup>5</sup>       | High                 | ±2000                         | <10 <sup>5</sup> | WK, SK                 | 600 or 610      |
|   | -452° to +450°F (-269° to +230°C) | <10 <sup>5</sup>       | Moderate             | ±2000                         | >10 <sup>6</sup> | WK, SK                 | 610             |
|   | +600°F (<315°C)                   | <10 <sup>2</sup>       | Moderate             | ±1800                         | <10 <sup>6</sup> | WK                     | 610             |
|   | <700°F (<370°C)                   | <10                    | Moderate             | ±1500                         | <10 <sup>5</sup> | WK                     | 610             |
| High-Elongation (Post-Yield)                      | -50° to +150°F (-45° to +65°C)    | <10                    | Moderate             | ±50 000                       | 1                | CEA, EA                | AE-10           |
|   |                                   | >10 <sup>3</sup>       | Moderate             | ±100 000                      | 1                | EP                     | AE-15           |
|   |                                   | >10 <sup>3</sup>       | Moderate             | ±200 000                      | 1                | EP                     | A-12            |
|   | 0° to +500°F (-20° to +260°C)     | <10 <sup>2</sup>       | Moderate             | ±15 000                       | 1                | SA, SK, WA, WK         | 610             |
|   | -452° to +500°F (-269° to +260°C) | <10 <sup>2</sup>       | Moderate             | ±10 000                       | 1                | SK, WK                 | 600 or 610      |
| Dynamic (Cyclic) Stress Analysis                  | -100° to +150°F (-75° to +65°C)   | <10 <sup>6</sup>       | Moderate             | ±2000                         | 10 <sup>7</sup>  | ED                     | 200 or AE-10    |
|   |                                   | <10 <sup>6</sup>       | Moderate             | ±2400                         | 10 <sup>7</sup>  | WD                     | AE-10 or AE-15  |
|   | -320° to +500°F (-195° to +260°C) | <10 <sup>6</sup>       | Moderate             | ±2000                         | 10 <sup>7</sup>  | WD                     | 600 or 610      |
|   |                                   | <10 <sup>6</sup>       | Moderate             | ±2300                         | 10 <sup>6</sup>  | WD                     | 600 or 610      |

Figure 4.8: Criteria selection based on the test profile

The experiments carried during this study do not require special temperature requirements. Plus, the tests that will be held are static analysis only. This is why the WA strain gauge was selected. Constantan has the best combination of properties needed for strain gauge applications. This alloy has good strain sensitivity, which is insensitive to strain level and temperature. The interesting thing is that this alloy can carry strains higher than 1800 $\mu\epsilon$  for a life time higher than 10<sup>5</sup> number of cycles. In fact, the two last columns of the table give information about the leaf spring life in sense of number of cycles; those two variables allow to construct the Wohler curve, as shown in the following figure. Those information mean that the strain gauge is robust enough for the weighting application under study.

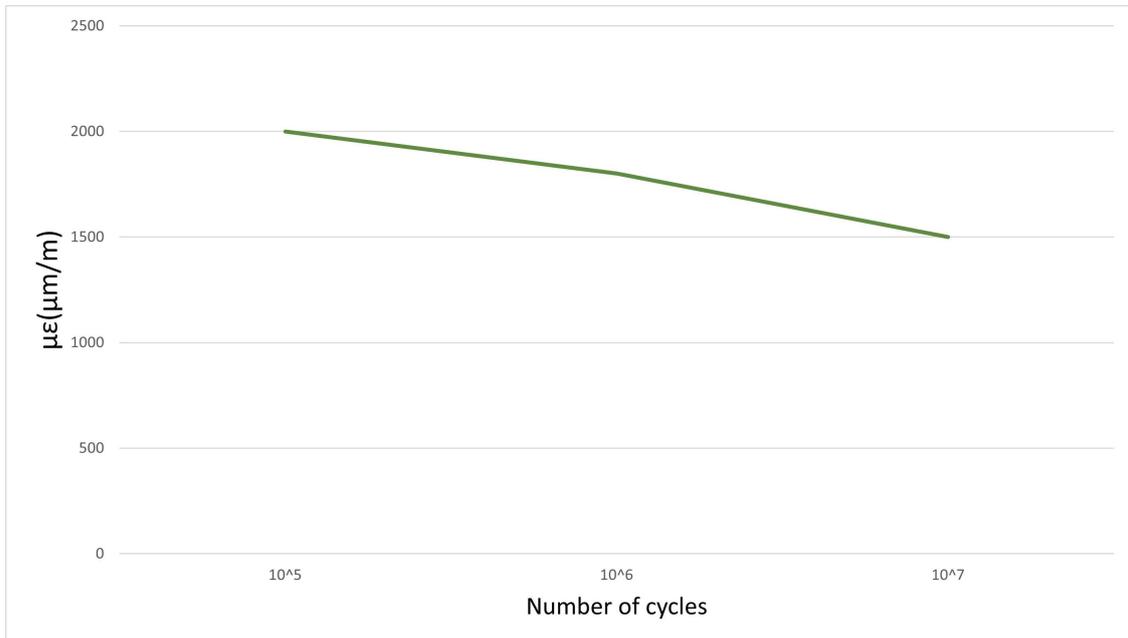


Figure 4.9: Wohler curve of the WA strain gauge

This curve shows that the strain gauge life increases when the elongation  $\epsilon$  decreases. This goes against the installation requirement, i.e. higher is the elongation, higher will be the sensibility to the load measurement. The goal is to find a good trade-off.

Another important property shared by constantan strain gage alloy is their self-temperature-compensation. Self-temperature-compensated strain gages are designed to produce minimum thermal output (temperature induced apparent strain) over the temperature range from about  $-50^\circ$  to  $+400^\circ\text{F}$ . This is another advantage of the sensor that has been selected.

3. Then, the connection should be completed to obtain a reliable measurement. Usually, to measure small changes in resistance, strain gauge configurations are based on the concept of a Wheatstone bridge. If the strain gauge is inserted inside the bridge, the deformation of the strain gauge can be detected through the variation of voltage. The general Wheatstone bridge, illustrated in figure 4.10, is a network of four resistive arms with an excitation voltage,  $V_{dc}$ , that is applied across the bridge. Usually the bridge gives an output voltage but in this study, there will be an output current, since the resistance  $R_m$  is present in the circuit, as shown in the following figure.

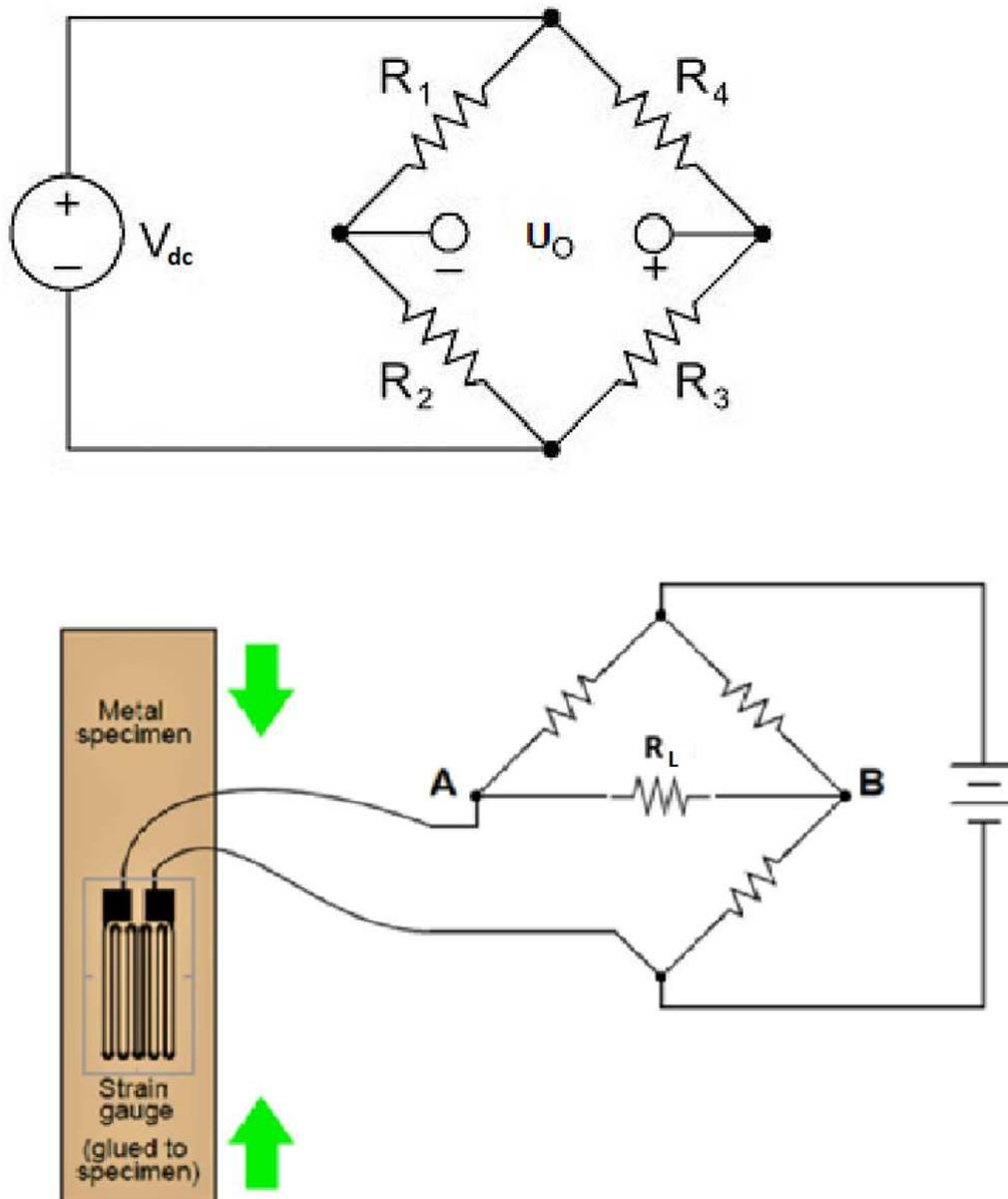


Figure 4.10: Wheatstone bridge configuration

If the strain gauge, that substitute the  $R_4$  resistance, changes its value, it will unbalance the bridge and produce a nonzero output voltage that is a function of strain. With the deviation method the bridge unbalance voltage is measured. If the instrument used to measure  $V_{EX}$  has an infinite impedance, the unbalance voltage is given by:

$$U_0 = V_{ex} \left( \frac{R_1}{R_1 + R_2} - \frac{R_4}{R_3 + R_4} \right)$$

If we suppose  $R=R_1=R_2=R_3=R_4$  and the resistance of the strain gauge changes to  $R + \Delta R$ , so the voltage is given by :

$$U_0 = V_{ex} \left( \frac{\Delta R}{4R + 2\Delta R} \right)$$

that becomes  $U_0 = \frac{V_{ex}}{4} K \epsilon$  only when  $R \ll \Delta R$

When the bridge is balanced  $V_0$  is equal to zero because  $R_1 \cdot R_3 = R_2 \cdot R_4$ . So strain gauges on adjacent sides of the bridge are subtracted, strain gauges on opposite sides of the bridge are added together. But since  $R_4$  is not constant but varies in time, the bridge will be unbalanced. Knowing the relationship between the mass and displacement we can obtain the gain of the measurement system :

$$\frac{U_0}{m} = \frac{\frac{V_{ex}}{4} K \epsilon}{\frac{F_1}{1_2 \cdot g \cdot L}} = \frac{V_{ex}}{4} K \frac{l_2 \cdot g}{k_1 \cdot L^2}$$

where the maximum elongation is given by:

$$\epsilon = \frac{\Delta L}{L} = \frac{F}{K_{FEM}} = 1800 \mu m / m$$

$$K_{FEM} = 0.4 kg / \mu \epsilon$$

This gain is obtained numerically thanks to the FEM (finite element method) analysis.

To conclude, comparing the single strain gauge and the bridge, for a Wheatstone bridge the sensitivity is much higher (about 10 mV/V). However, Wheatstone bridge type requires three more strain gauges than quarter-bridge type, so more expensive. It also requires access to both sides of the gauged structure. When choosing the type of configuration, we need to consider also the installation of the sensor, because the main aim is to realize a sensing leaf spring for a massive production so the procedure must be quick. The number of wires, and mounting location all can affect the level of effort required for the installation. Quarter-bridge type is the simplest one because it requires only one gauge and two or three wires.

4. The fourth point considers the temperature compensation of strain gauge. The temperature influence needs to be compensated in order to reduce most of the errors. The induced strain can be viewed as an error due to the temperature changes in strain measurements. The temperature compensation in the strain gauge is necessary to reduce the thermal effect on the measurements [5].

Temperature influences strain measurements in many ways:

- There will be a non-zero coefficient of thermal expansion. Unless compensated for changes in temperature, it will cause the sensor to expand or contract, which translated as a change in strain.

- The materials of the strain gauge itself have a non-zero coefficient of thermal expansion. Variation of temperature will cause expansion or contraction, independent of any strain in the part to which it is attached.
- The wiring and the strain gauge itself will have a non-zero temperature coefficient impedance. The resistance will change depending on the temperature.

To compensate those errors different methods can be used.

The strain material can be changed to compensate for the thermal variation of the item. The thermal sensitivity is reduced but not totally removed. The Coefficient of Linear Thermal Expansion (CLTE often referred as  $\alpha$ ) is a material property which characterizes the ability of a material to adjust the strain caused by temperature [6].

Another method is the one that uses a bridge circuit. By using a three-wire connection the bridge remains balanced as long as lead wire resistances  $R_L$  track with temperature, as shown in the following figure.

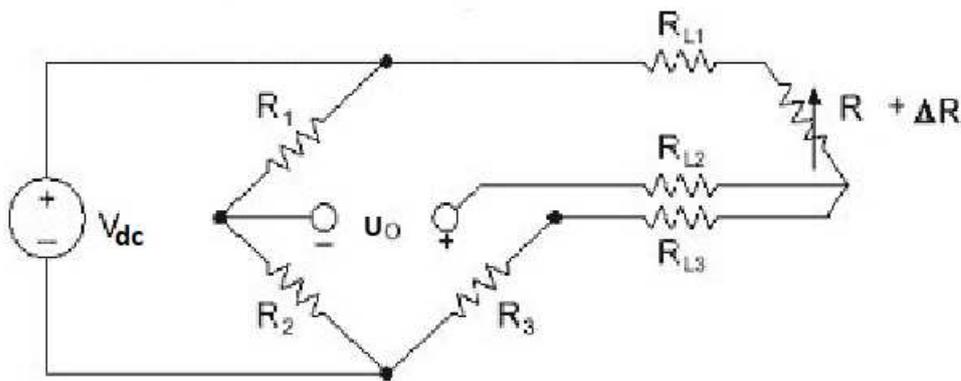


Figure 4.11: Three wire connection bridge

In the actual thesis a simplified structure is used by selecting a self-compensated strain gauge made available by KIMAX company, in order to carry out a first experiment. The kimax strain gauge is protected by a stainless steel cover and it transforms the strain of a beam into a measurable current signal. In the following figure we can see the electrical connection of the sensor and its temperature information.

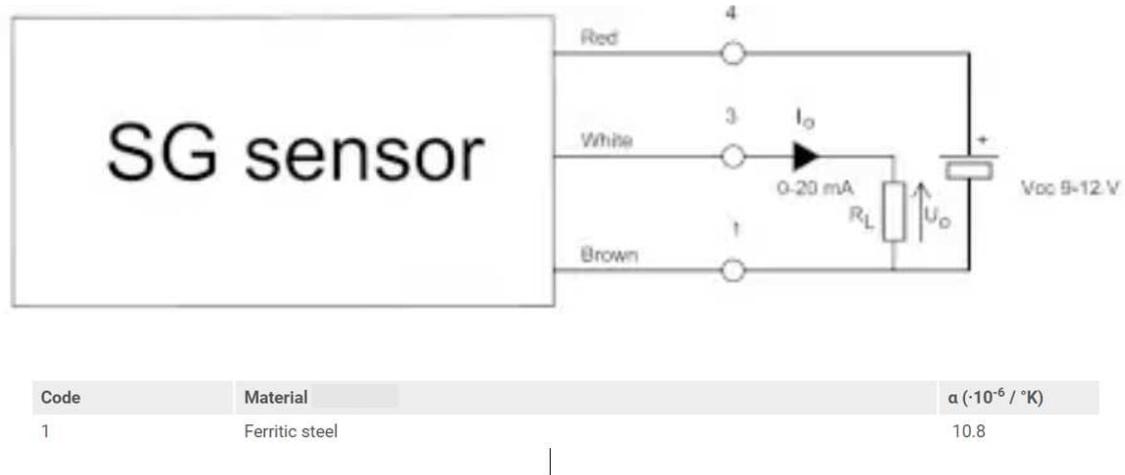


Figure 4.12: KIMAX SG specifications

The kimax sensor is made with steel and the coefficient reported in figure 4.5 is the elongation of linear expansion of the strain gauge material. It means that the temperature adjustment for the ferritic steel material is about 10 or 6  $/^\circ\text{K}$  [8]. However, we take into account the fact that this leads to errors. For example, if we consider a gauge compensated for aluminum that has a temperature coefficient of 23 ppm/C. With a nominal resistance of 1000  $\Omega$ , gauge factor of 2, the equivalent strain error is still 11.5  $\mu\epsilon/C$ . Therefore, additional temperature compensation is important. But for this first attempt we will consider only one strain gauge without any bridge.

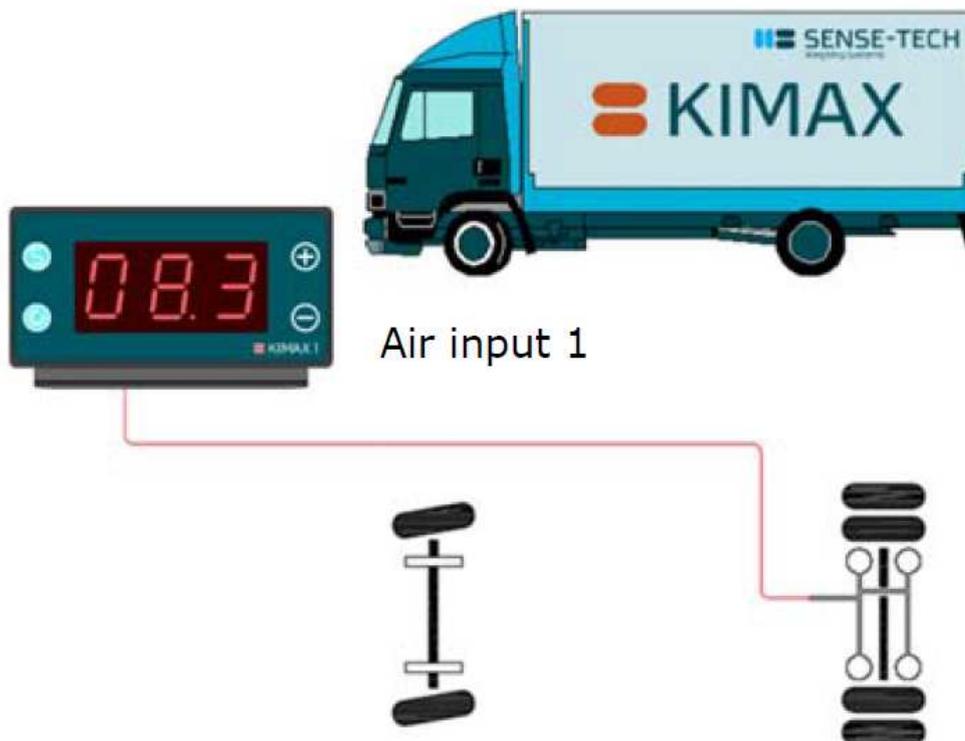
5. The strain gauge measuring grid length depends on the aim of measurements, as the result of the measurement using strain gauges will be the mean strain underneath the measuring grid. In general, measuring grid lengths of 3 or 6 mm (0.118 or 0.236 inches) represent a good solution. A length of 5mm was chosen as suitable.

6. The step to choose the right resistance is very important. In some cases the only difference between two configurations available in the same series is the electrical resistance, typically 120 and 350 ohms. If the choice is possible, the higher resistance is preferable as it reduces heat generation by a factor of 3. Higher electrical resistance minimises the effect of cables such as the sensitivity reduction induced by the resistance of the cables and undesirable variations caused by the variation of cable resistance to the temperature fluctuation. Since there are no temperature requirements in this specific case, a standard resistance of 120 ohms was chosen.

## 4.2 Kimax 1 and 2

For our project the kimax 1 is used to give an interface of the load to the driver and, instead, the kimax 2 is used to calibrate the sensor. Those instruments are given directly by the same company of the strain gauge manufacturer. The Kimax 1 and 2 are axle weighting gauges that the output current of the strain gauge to indicate the load, and to keep informed at all times about the present load value. Usually, Kimax 1 and 2, are used for air suspensions because it exploits the linearity between air pressure and load of the single axles and calculate the current axle load with an accuracy of 2% of the maximum load for each axle.

Kimax 2 is useful whenever we need to connect the strain gauges to more than two axles because it allows to see the weight on each axle (right figure 4.13).



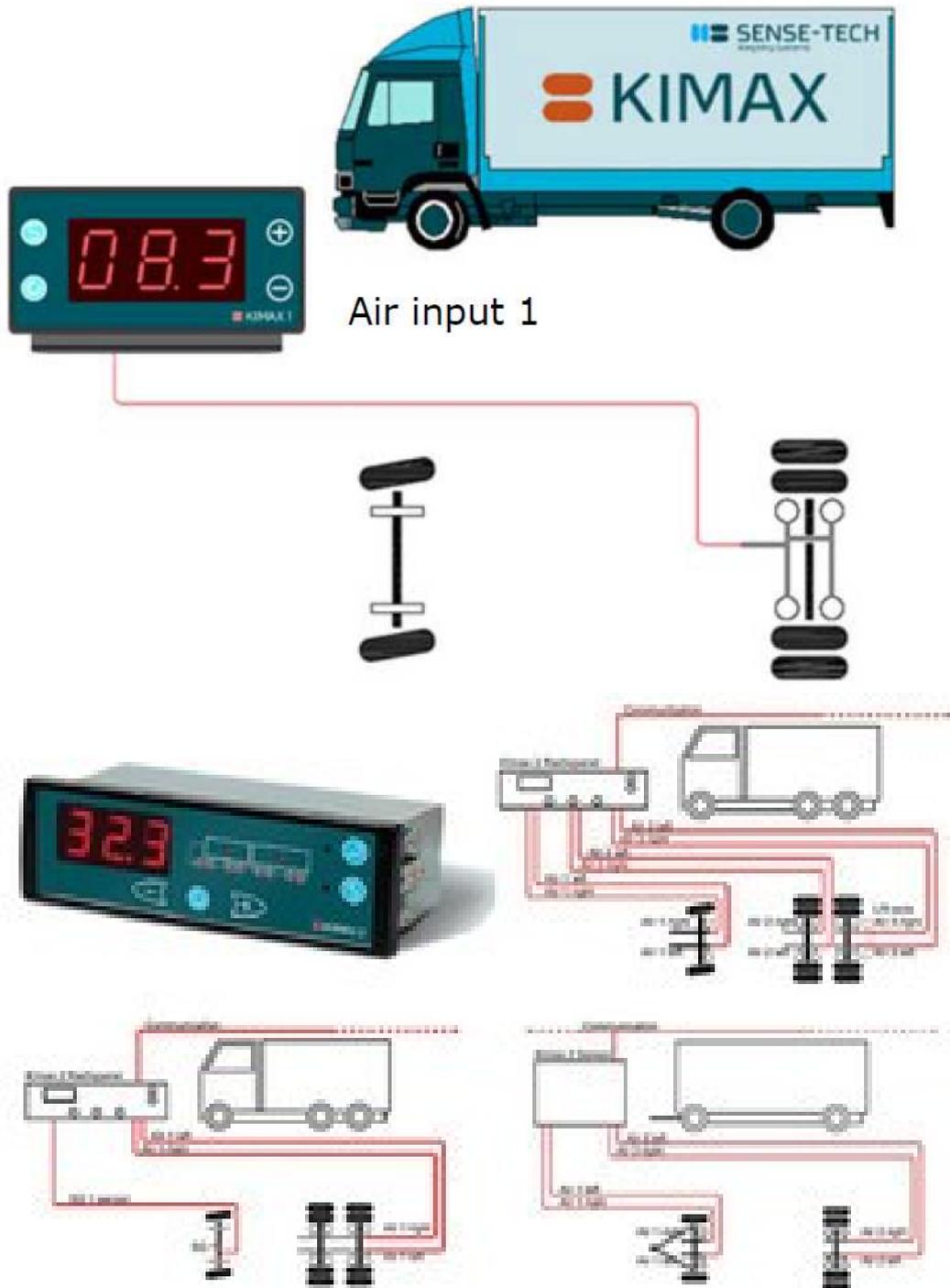


Figure 4.13: Kimax 1 and 2 installation

Since, for our experiments we only monitor the rear axle of the truck we can use the kimax

1 alone, but it's important to use multiple options given by the kimax 2 to settle the right gain of the curve between the weight and the current during calibration. To summarize, Kimax interfaces are optimal to give a real time monitoring to the driver, but also, it's essential to calibrate the sensor installed on the leaf spring before putting it on the truck.

### 4.3 Data logger

For sake of simplicity we connected the sensor to a data logger in order to read the exact current delivered by the strain gauge, and assure that the conversion given by the kimax 2 is correct. Here in figure 13, we can see the data-logger acquisition compared to the one of the kimax 2. Both of them give a value on mA, but in this case the value of the kimax 2 is wrong because the right gain wasn't settled. The data-logger help us to find the correct gain, in order to display the right load value.





Figure 4.14: Data-logger acquisition on the left and data-logger components on the right

The data-logger is also considered in this thesis to be an interface between the sensor and our future electronic board, in order to send data in Cloud. For a first moment, we used a data-logger without any ethernet output, so not connectable to a board. The data-logger in the figure 4.14 is a PCE-CR 10, a three channel current data-logger, where, two of them are connected to the two rear axle sensors. It is used to record up to 20 mA signals. The current data logger stores the measured current on an memory card in Excel format. In addition to storing them it shows on a display the current value. The maximum resolution here is about 0.01 mA. The device is powered by 6A 1.5V batteries or a power supply. To visualize the live-measured values, the data logger can be connected to a PC or the SD card can be recovered for offline analysis (our case). This technology is a quick and cheap method to acquire current values of the two rear axles thanks to the two channels available as input of the data-logger. We use the data-logger also to draw the curve of the current function the load during calibration.

## 4.4 Arduino board

A custom Arduino board, fabricated by an Arduino home settled in Italy, has been used. It works similarly to Arduino UNO microcontroller. The only difference is that this custom board that we are using for our project has everything embedded into it, so the WIFI module and the gyroscope is already in the main board. Arduino is the interface needed between our sensors and the IoT platform. So, we need to program Arduino in order to read all information of the sensors and send them in Cloud. Once we have data stored in our platform we can analyze them and also apply data cleaning. For this experiment, we

will need the following hardware:

- 1.Arduino board with the WiFi Module
- 2.Computer running Arduino IDE
- 3.Arduino USB cable

Arduino is to filter, adapt and transfer data after filtering them. The other calculations, such as translating current into weight and reorganize data, are computed offline by a script python. This script will also filter the duplicates and send data to the IoT platform.

The Arduino board wasn't ready in time this is why this thesis was based on offline analysis. So only the data logger memory card is used to analyze data.

## Chapter 5

# Migration of data inCloud

### 5.1 Transmission chain

Obviously, we need to go further on the study of weight monitoring and, most importantly we need to keep analyzing those data along the all leaf-spring life. For this goal, we need a safe way to store data and a good IoT platform to study them. We settled a transmission chain in order to propagate the information of the sensors till the IoT platform. As we can see in the figure 5.1, the chain starts from the sensors that, in this case are the strain gauges and the position sensors, then they are connected by cables to the Arduino board that is our gateway, to finally send the sensor data in Cloud to the IoT system in our computer by WIFI. The aim of the IoT platform is to store data and predict the future values of stress and load, in order to prevent the risks due to the leaf spring braking. Also the IoT platform needs to reorganize data in order to send useful data to the phone application used by the drivers.

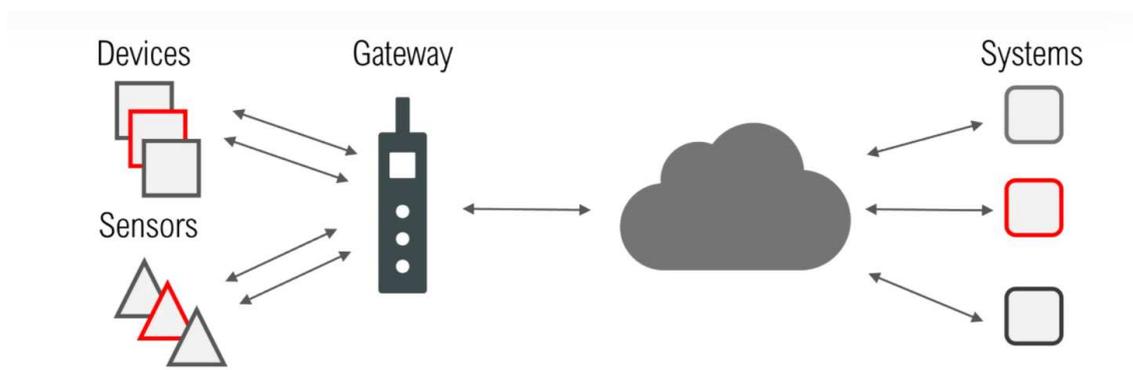


Figure 5.1: Transmission chain

In the next paragraphs we will discuss more deeply the Arduino block and the IoT connection.

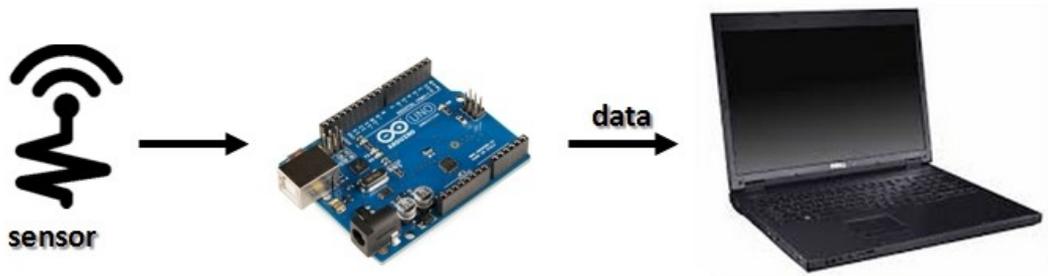


Figure 5.2: Arduino transmission chain

Also, our Arduino board contains the GPS and the gyroscope. The gyroscope is important in order to detect the beam truck inclination and give through a program the right value of the charged load. The GPS is useful for the manufacturer in order to know the road's condition in case of accidents or leaf spring wear.

## 5.2 IoT platform

An IoT platform is the software that makes an Internet of Thing solution work by providing data access, monitoring and storage. An IoT platform is designed to reduce development time by providing reusable technologies to analyze and order data information. Many storage services sellers are available on the web, and there are different companies owners of separate storage facilities that can be chosen. Thanks to the in cloud storage the user can access data using internet service, at the remote location. Those services are convenient because data can be stored for a long life-time. Custom IoT platforms can be created from scratch but it's a little bit hard. It can be a multi-year effort that involves a large development with different kind of skills. This is why it's better to use an affordable IoT platform already developed on the market. In this case we can choose MindSphere, the Siemens IoT platform [7]. Thanks to the MindConnect IoT extension on the launchpad we can log in and see all the devices connected to the MindSphere broker. The devices are normally called clients. Actually, it has been decided to develop an independent application with node-red since the platform is a little bit expensive. It's better to pay for a service when the analysis to do on each bunch of data is clear.

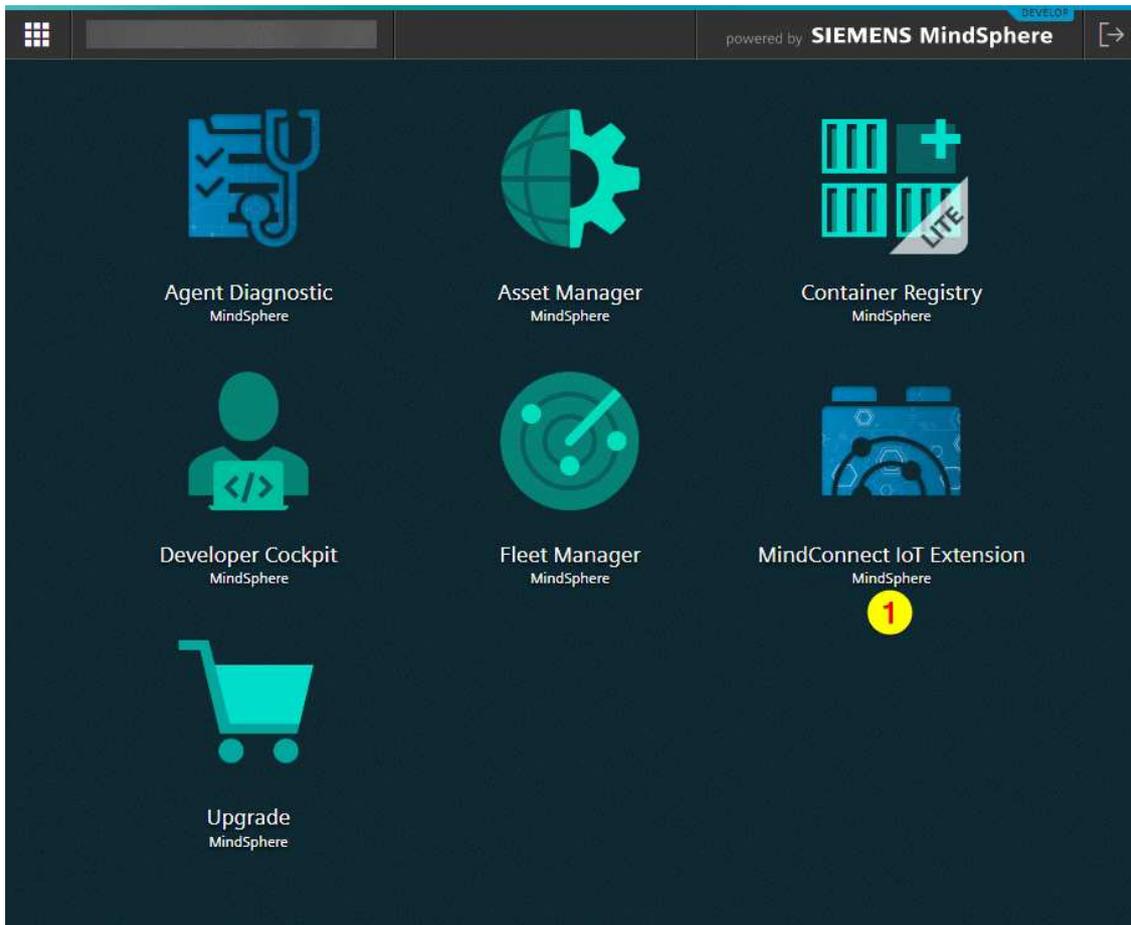


Figure 5.3: MindSphere launchpad

The clients appear in the platform when they connect to the broker with right information of IP address, host and port.

| STATUS                | NAME | MODEL   | SERIAL NUMBER | GROUP                 | REGISTRATION DATE | SYSTEM ID       | IMEI       | ALARMS |
|-----------------------|------|---------|---------------|-----------------------|-------------------|-----------------|------------|--------|
| MI 6                  | MI 6 | unknown | Phones        | 29 January 2020 15:38 | 82246968          | 866822034462069 |            |        |
| opcua-demo            |      |         |               | 5 February 2020 14:10 | 83165984          |                 |            |        |
| My First MQTT Device  | X    | 1234    | Simulators    | 16 March 2020 20:18   | 87686501          |                 | 1 ⚠️ 🔴 🟡 🟢 |        |
| AdvancedSimulator3 #2 | X    | 1234    | Simulators    | 1 July 2020 12:26     | 92001781          |                 | 1 ⚠️ 🔴 🟡 🟢 |        |
| NSDeviceRiekerJuly    |      |         | *test         | 8 July 2020 15:33     | 92418374          |                 | 1 ⚠️       |        |
| NS Opcaa Test Gateway |      |         |               | 6 August 2020 17:43   | 94221224          |                 |            |        |

Figure 5.4: MindSphere Clients examples

Other free platforms are also available on the web, such as the Eclipse IoT. To simplify our experiment and also for economic reasons we started with a free platform that works

as a server only, so it doesn't give any interface to see our data and manipulate them. In the following figure we see the only interface of this free mqtt server.



Figure 5.5: Eclipse IoT server

### 5.3 Arduino-IoT connection

So how do we do to transfer our data from the electronic board to the cloud? We will configure a setup composed from an Arduino and a WiFi module as an IoT Thing and make it ready to communicate with our cloud platform. First, the WiFi module will be used to connect Arduino to the internet, then we will send sensor data to our Cloud platform channel by using the MQTT protocol [13].

MQTT stands for Message Queuing Telemetry Transport. MQTT is a quite easy language protocol, designed for devices with low bandwidth. So, it's a suitable tool to exchange data between various IoT machines. It is based on publisher/subscriber principle and it uses a central broker that in our case is our IoT platform. In MQTT there are a few basic concepts that are important to understand. First, in a publish and subscribe principle, a device can publish a message on a topic, or it can be subscribed to a particular topic to receive messages :

- For example, Device 1 publishes on a topic.
- Device 2 subscribes to the same topic .
- So, device 2 receives the message.

As we can see in figure 5.6, the three subscribers receive the publisher messages because they are subscribed to the same topic.

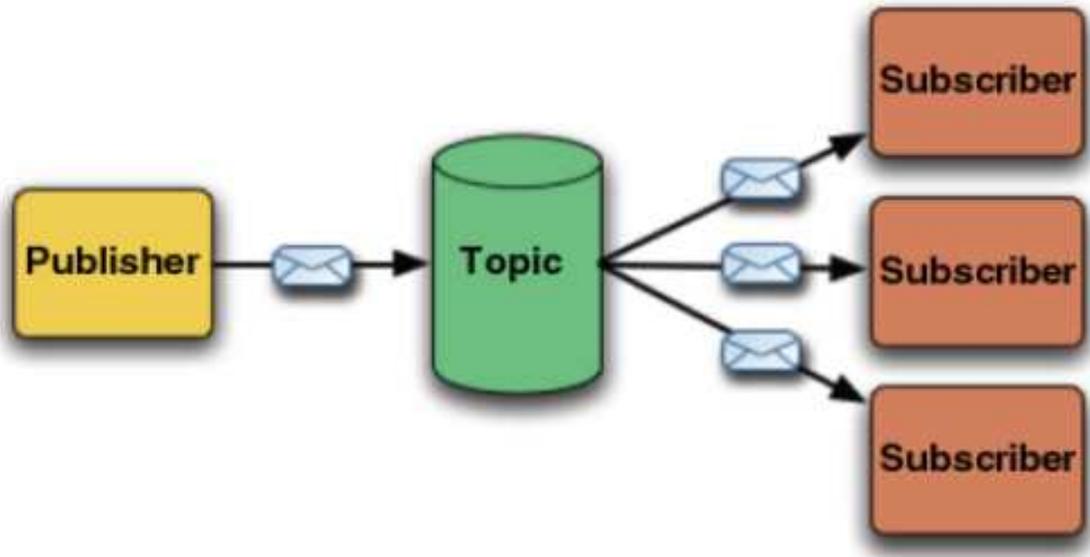


Figure 5.6: MQTT principle

Second, another important concept is the topic. Topics are a way to subscribe to a device in order to read incoming messages or a way to specify where to publish the message. A topic can be simply a word.

Finally, another important concept is the broker. The MQTT broker receives all messages, filtering the messages, deciding which device can see them, and then writing the message to all subscribers. In home automation projects, the Mosquitto Broker is used, but it can be installed also on the PC. In our case a public broker is used, for a first attempt, then the mindsphere broker will do the job.

# Chapter 6

## Experiments and results

### 6.1 Experiment's set up

First, the vehicle characteristics need to be specified. As written before, for the experiments, a small lorry will be used, the BOXER CABINATO TELAIO VF3YC2MAU12D62138 made by Peugeot. This light truck presents only two leaf springs in the rear axles and two springs' suspensions in the front axles. So, for the rear axles we can surely use the strain gauges but for the front ones, it's really difficult to attach a sensor because of the springs and the sensor availability in the market. This means that if the load on the truck is not located exactly between the two rear leaf springs the load acquired by the sensor is not correct because far from the rear suspensions, the weight is partly supported by the front suspensions. For an accurate measurement of the load we should add two additional sensors on the front suspensions, but since it's complicated we realize the acquisition only with the rear sensors. The correct acquisition can be achieved by computing, with the help of the Arduino board, the angle of inclination of the chassis, that is proportional to the front suspensions' stiffness. In this way, it will be easier to compute the load carried by the front suspensions and, finally have the total load displayed by the phone application. Subsequently, we connected the rear strain gauge sensors to the two data-logger channel and to the kimax 1 radio, that both are positioned close to the driver. The kimax 1 will display the addition between the right and left leaf spring current value. For what concerns the data-logger, the first value represents the rear right leaf spring, while the second one represents the rear left leaf spring. In a first moment we do the acquisition without an electronic board to simplify the installation and read the current values and the weight from the displays available.

## 6.2 Calibration off-truck

The calibration of the sensor attached to the leaf spring is realized off-truck, positioning the leaf spring into the testing machine in the Mollebalestra workshop. Leaf Spring Testing Machines are used to test the leaf spring flexibility, meant for the production line. Usually this machine is used to know the life duration of a leaf spring, submitting it to a compression for multiple months till its break. The spring is loaded till the maximum deflection and back to zero. The force is applied on the middle of the spring. The testing machine measures the beam deflection by computing the bending force. For leaf spring tests, contrarily to the structure on the truck, the suspension rests on the machine rollers on two flat supports fixed to the support beam. The testing machine measures the leaf spring deflection that is the result to the applied compressive force. Thanks to the kimax 2, depending on the gain, we obtain different curves of the current function of the load. Then, we need to choose the suitable curve that can represent our leaf spring during all its life.

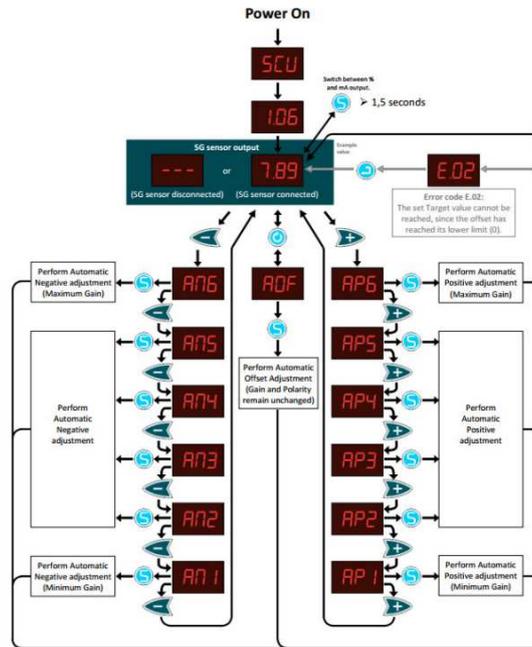


Figure 6.1: Kimax 2 gains options

As we can see in figure 6.1 with the kimax 2, we select different gains (AN1, AN2, AN3 ecc..), so for the same load we obtain different values of current. We do the same thing with different weights, in order to draw the curves with the respective gains and select the best one for our parabolic leaf spring.

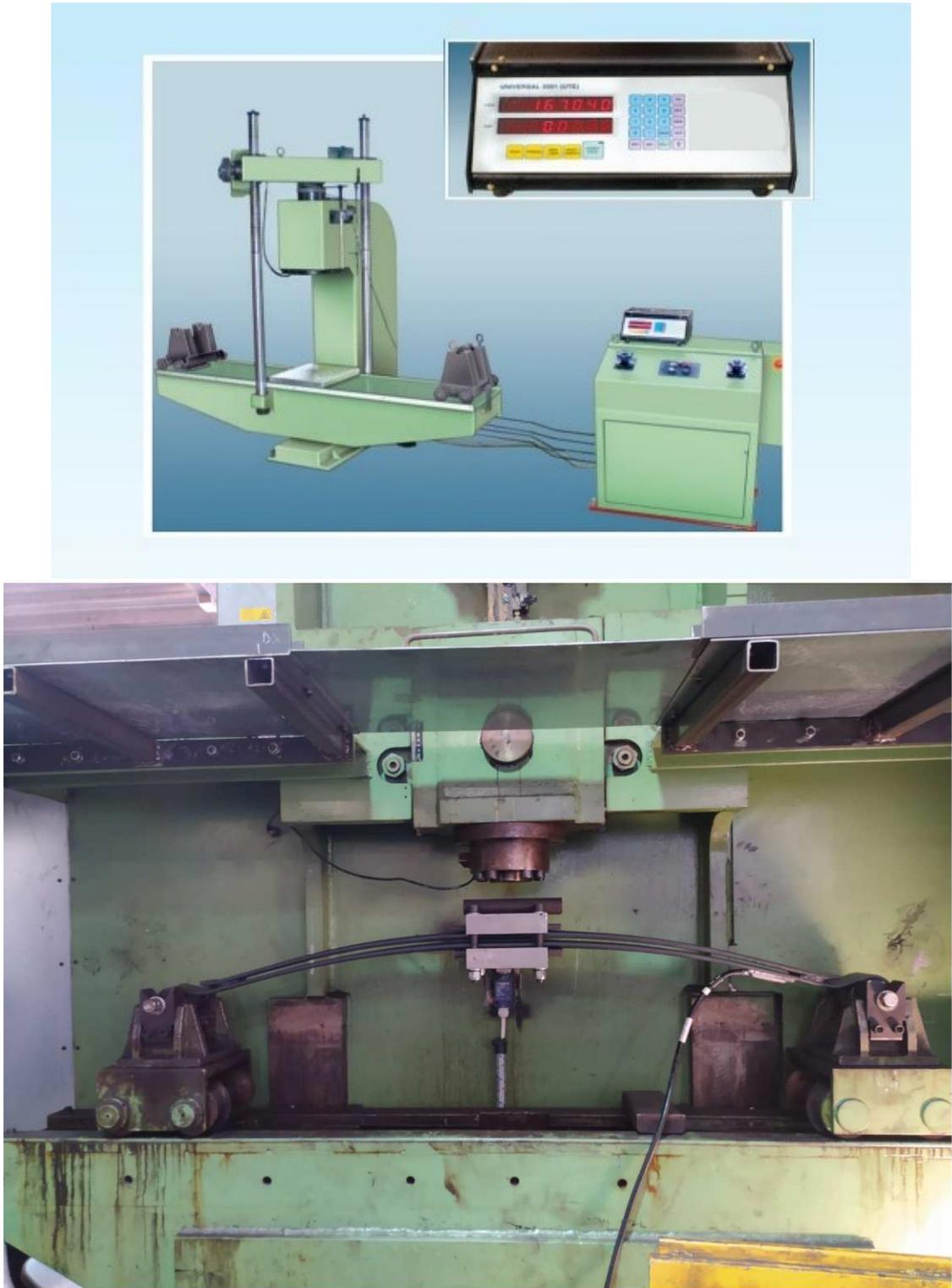


Figure 6.2: Testing machine for leaf springs  
The calibration procedure requires to apply different forces to the leaf spring and, thanks

to the current acquired by the kimax 2, assign the value of weight to the respective value of current. This procedure, also, allows to justify the fact that the position of the strain gauge on the leaf spring gives the right value of weight. To know which value of weight we applied to the leaf spring we can use a kind of platform as we can see in figure 6.2. Once we repeated the procedure for different gains we can draw the curves of figure 6.3.

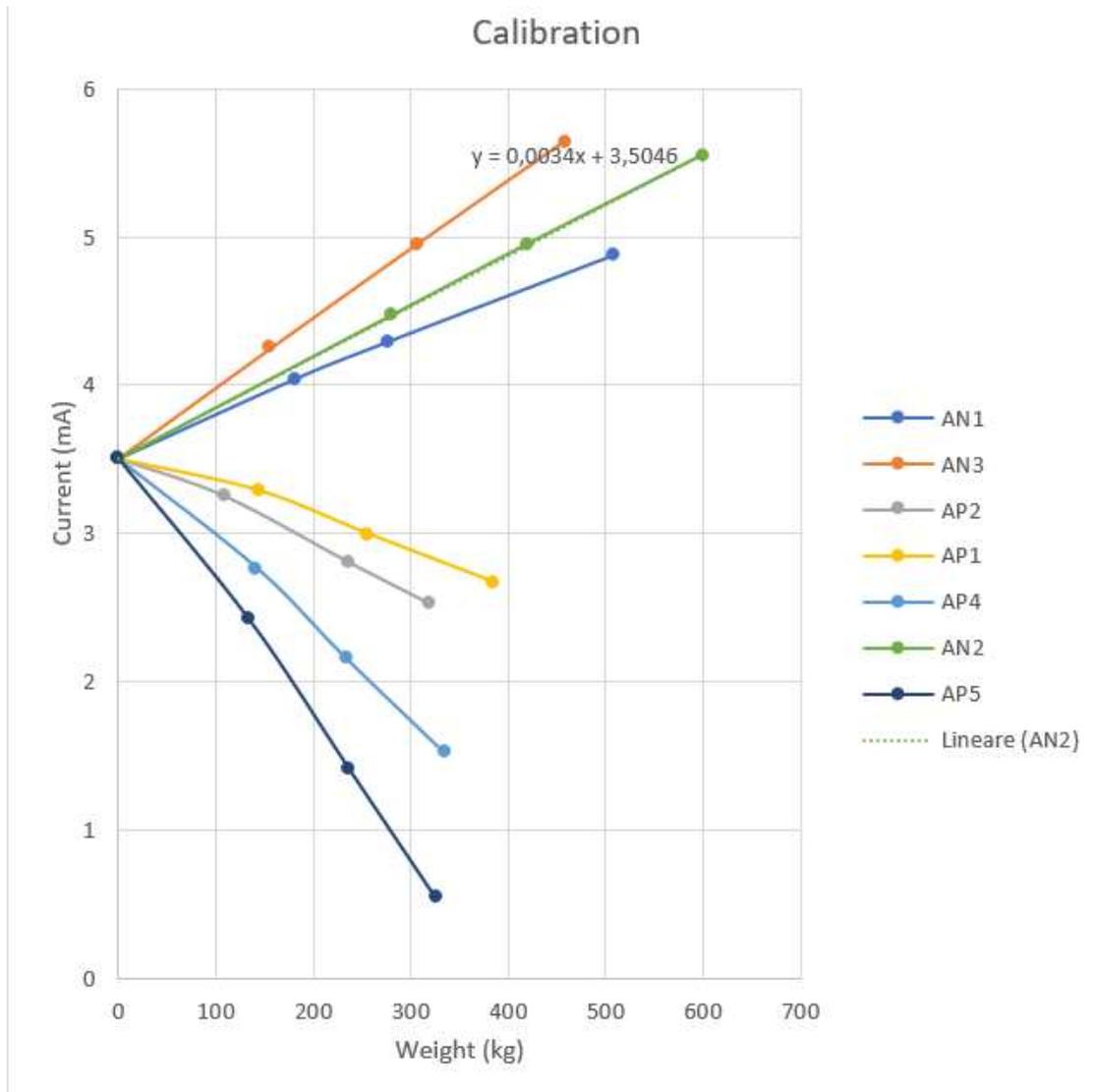


Figure 6.3: Calibration curves with different factor gains

In our case, the AN2 (on green) gain was selected to trace the curve that is represented by the equation  $y = 0,0034x + 3,5046$ , where  $y$  is the current and  $x$  the load. This curve is necessary to display the right value of the load on the kimax 2 sensor.

### 6.3 Sensor acquisition of the load

Once the calibration completed it is needed to settle out an experiment to prove the theoretical relationship between the position of the load on the truck and the weight. To do this a metallic box was used of a load of 160 Kg (figure 6.4) and moving it along the truck platform, a relationship can be computed between the load and the position on the truck of the load.





Figure 6.4: Load of 160 kg on the truck platform

Thanks to the data logger on the truck cabin, connected to the two rear sensors, the value of the current can be translated into weight using the AN2 calibration curve on figure 6.3. The data logger is equipped with a memory card in which the current values in the excel file can be found. Those values will be studied off-line to compute the experimental curve showing the relationship between load and distance from the front axle. Once the Arduino board will be ready the values can be filtered and sent in cloud directly. At the moment of the experiment the Arduino board wasn't available so the values was taken by hand from the data logger in the following figure.



Figure 6.5: Data logger installation on the truck

To simplify the experiment, the platform truck was divided in 12 square areas as in figure 6.6; so, the length of 3840 mm is divided in four equal parts and the width of 1940 mm in three equal parts. As seen in the previous figure, the origin corresponds to the center of the rear axle. The experiment has been held by moving the load along the three axes of the truck's platform and storing the values of right and left currents corresponding to each position of the load on the platform, given by the data logger. Then, the values of current of each sensor have been summed up together and finally, have been translated into mass values thanks to the function computed during the calibration of the sensor;  $y = 0,0034x + 3,5046$ , where  $y$  is the current and  $x$  the load.

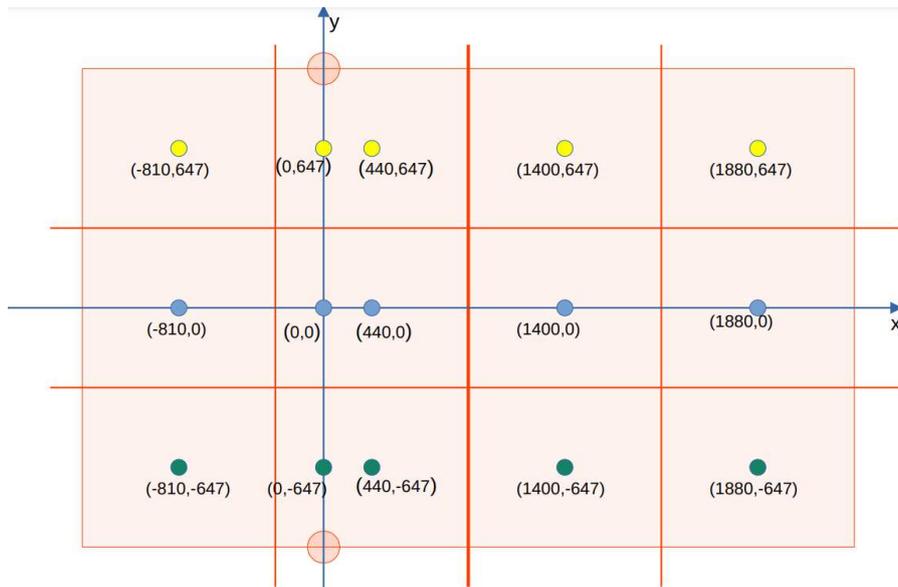


Figure 6.6: Truck platform divided in 12 areas

The aim is to plot the values of mass measured experimentally, function of the distance from the front axle of suspensions, "f". It is expected to have an high value of weight on the right axis of the platform, because the offset of the right sensor is higher than the left one, as seen in the following figure.

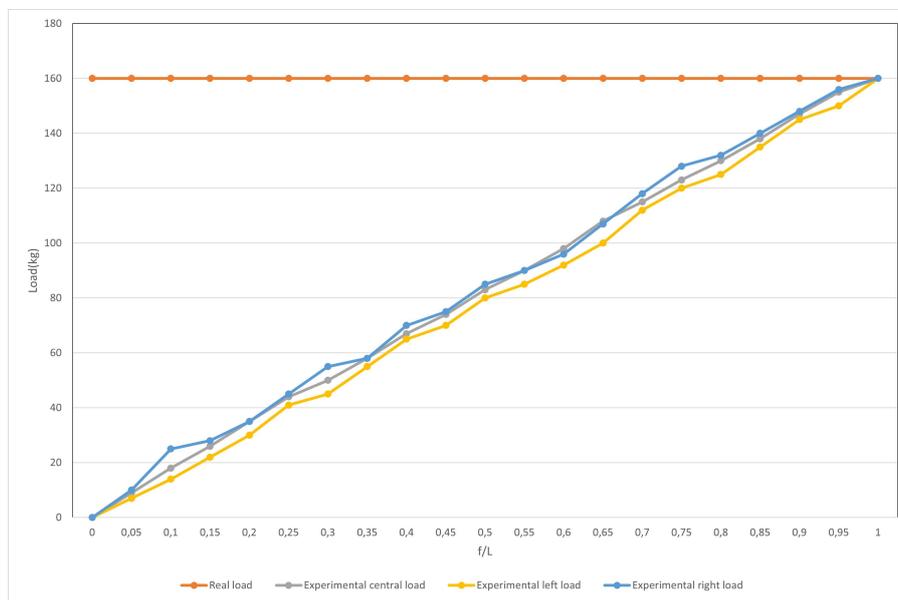


Figure 6.7: Behavior of the load along the three axis of the truck's platform

Generally, if the load is always kept in the centered axle of the truck the errors are minimized with respect to the theoretical curve, as shown in the following figure.

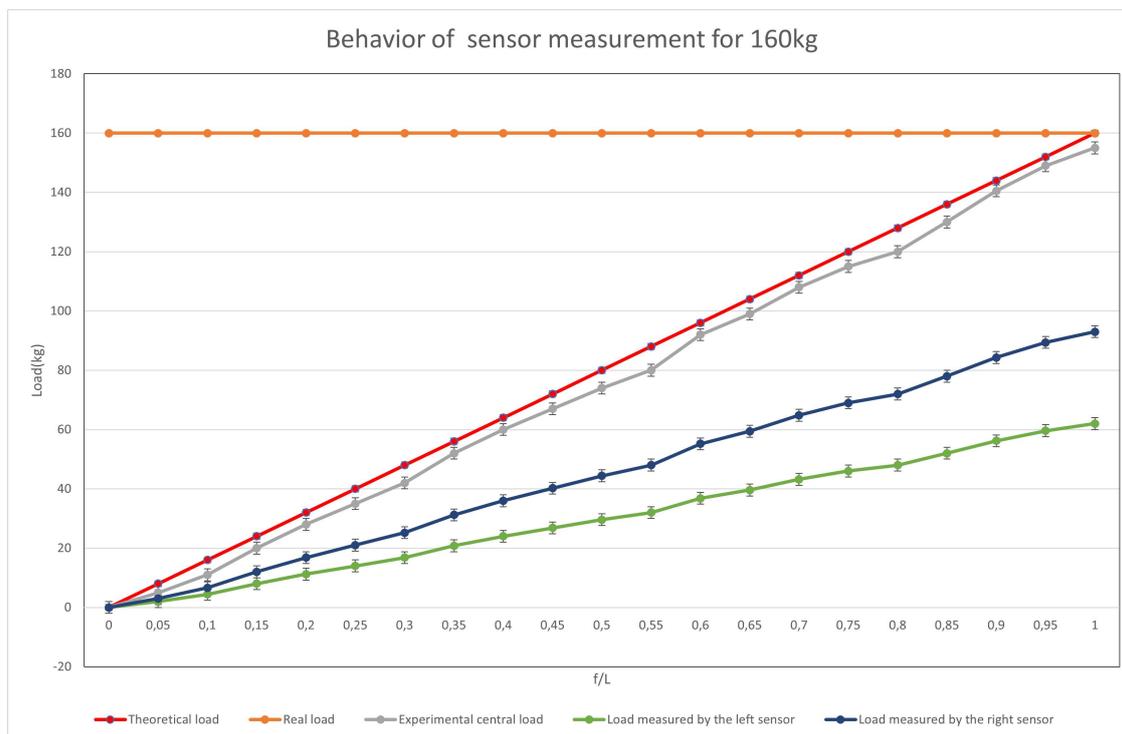


Figure 6.8: Behavior of the load carried on the centered axle of the truck’s platform, experimentally and theoretically

The curve in the figure 6.8 shows the behavior of the weight experimentally and theoretically. The error of the load given experimentally is about 2kg and theoretically it’s about 1,3 kg. Generally, the curves obtained after the experiment are consistent with the theoretical one. It can be seen that the value of the weight decreases with the displacement from the front axle. The errors are caused by many factors, like the inclination of the road or the fact that part of the load is carried by the front suspensions. If the same experiment is held, adding more boxes of 160kg, the error propagation can be shown. First, the load of 160kg has been doubled and, then, tripled as shown in figure 6.9.





Figure 6.9: The load of 16kg doubled and tripled

The error for a load of 160kg with respect to the theoretical curve is an average of 5 kg, that is acceptable. For the double and the triple of the load, the error becomes higher as shown in the following figure.

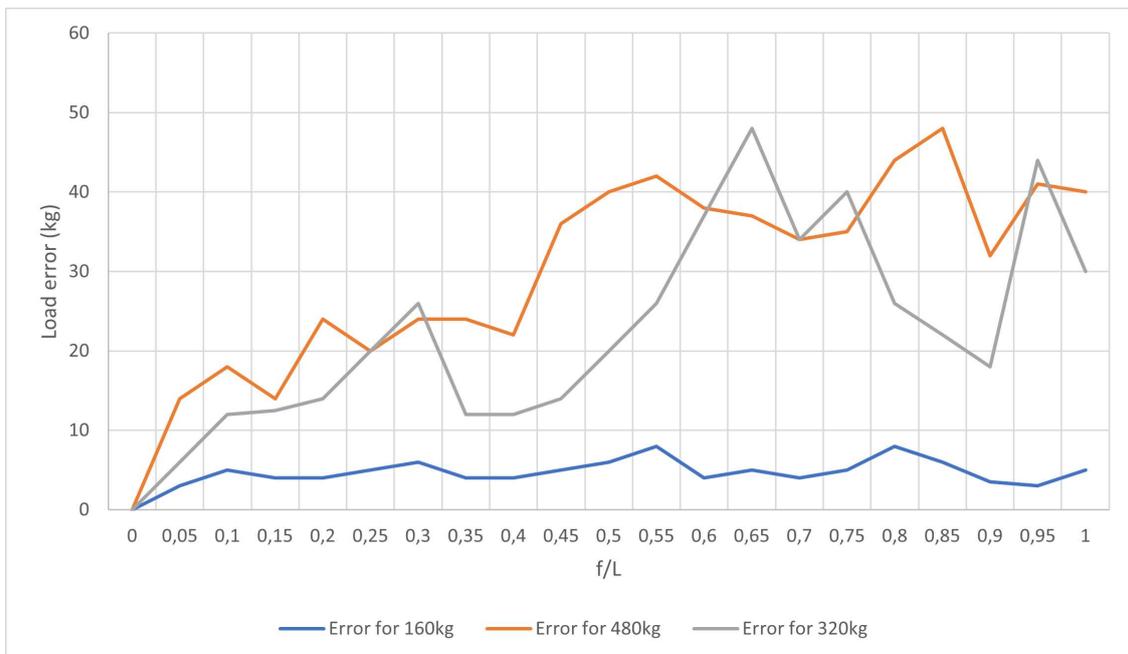
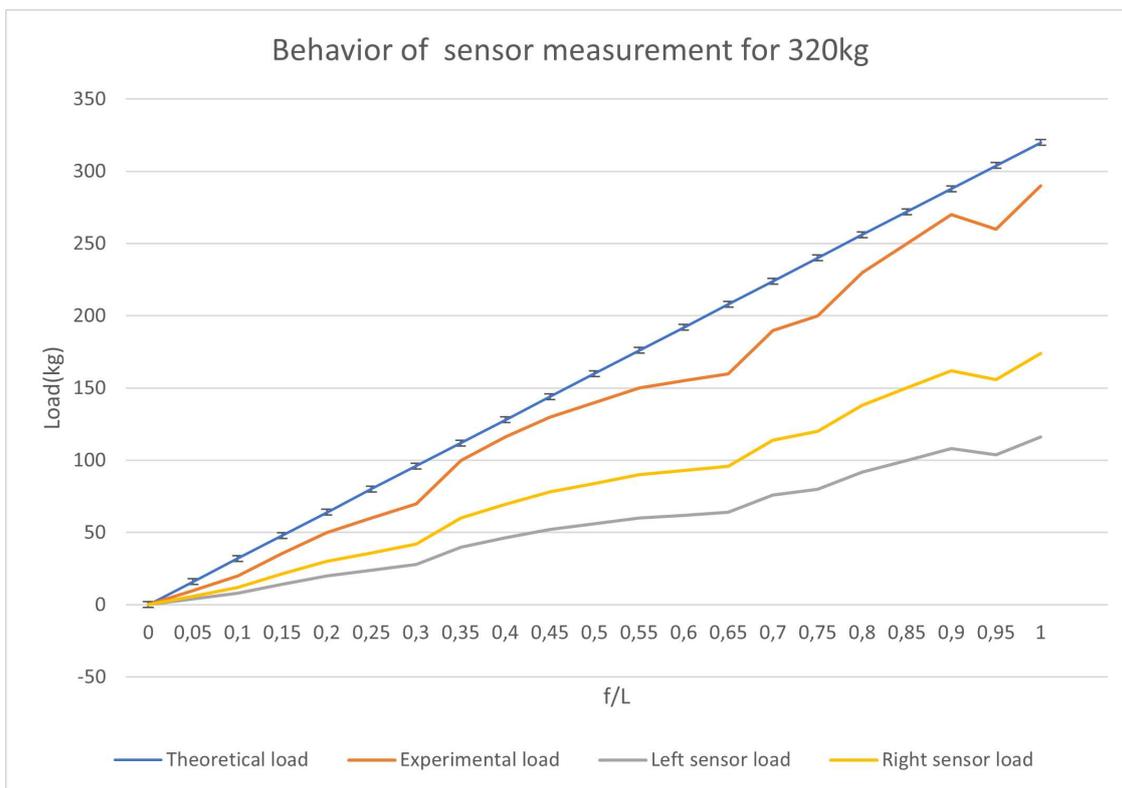


Figure 6.10: Behavior of the error on the acquisition depending on the weight and distance from the front axle

For double and triple the weight the curves of load carried by each rear sensor and the sum of those two are reported in following figures.



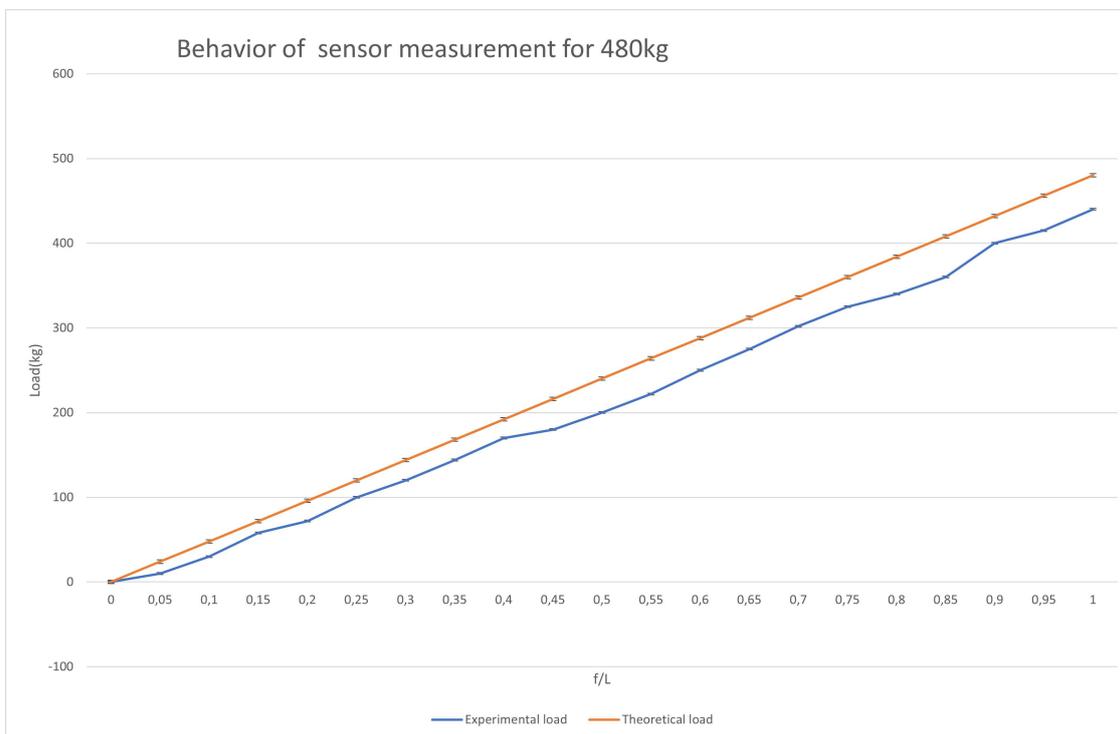


Figure 6.11: Behavior of the sensor measurement for a load of 320kg and 480kg

Generally, it can be said that the experimental curves of the load are similar to the theoretical ones. The load on the right and left axis are a little bit distant from the theoretical curve because of the offset (4.08mA for the right suspension and 3.77mA for the left one) present on the strain gauges measurements. This offset can be caused by a tension appeared when installing the strain gauge. However, it's clear that the theoretical and experimental curves are consistent with each other.

However, the error between the experimental curves and the real load value makes an important difference with respect to the theoretical curve and real one. In the following curves, it can be seen the error between the experimental curves of different values of weight and the real one.

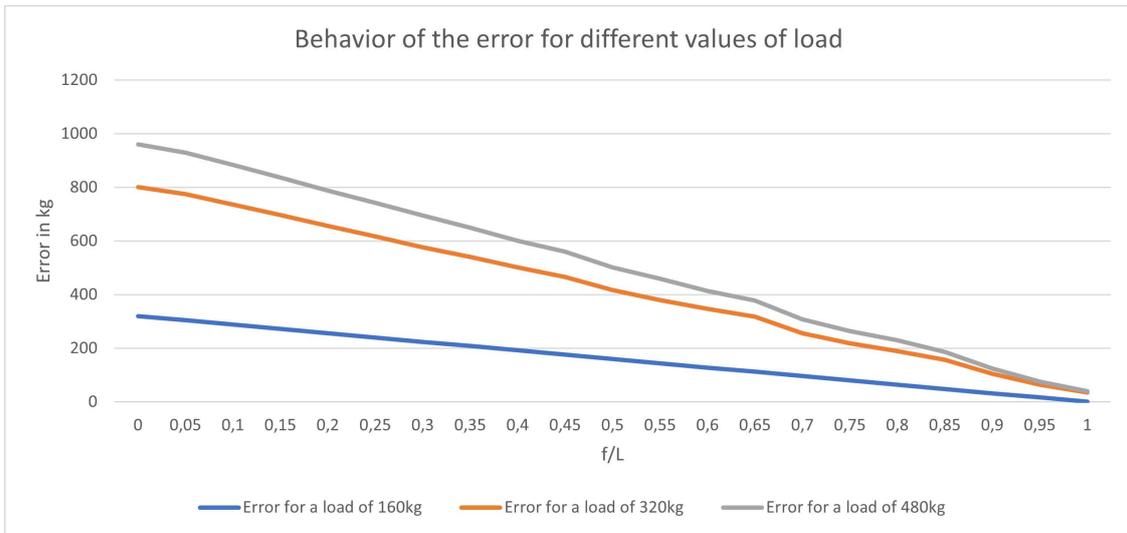


Figure 6.12: Errors of the experimental curves with respect to the value of weight on the lorry

It can be noticed that the error increases with the value of weight. For the theory of error propagation, whenever a load is added, the respective error is added too. Also, depending on the height of the center of gravity, another factor such as load measurement on inclined road can be effected, as shown in the following figure.

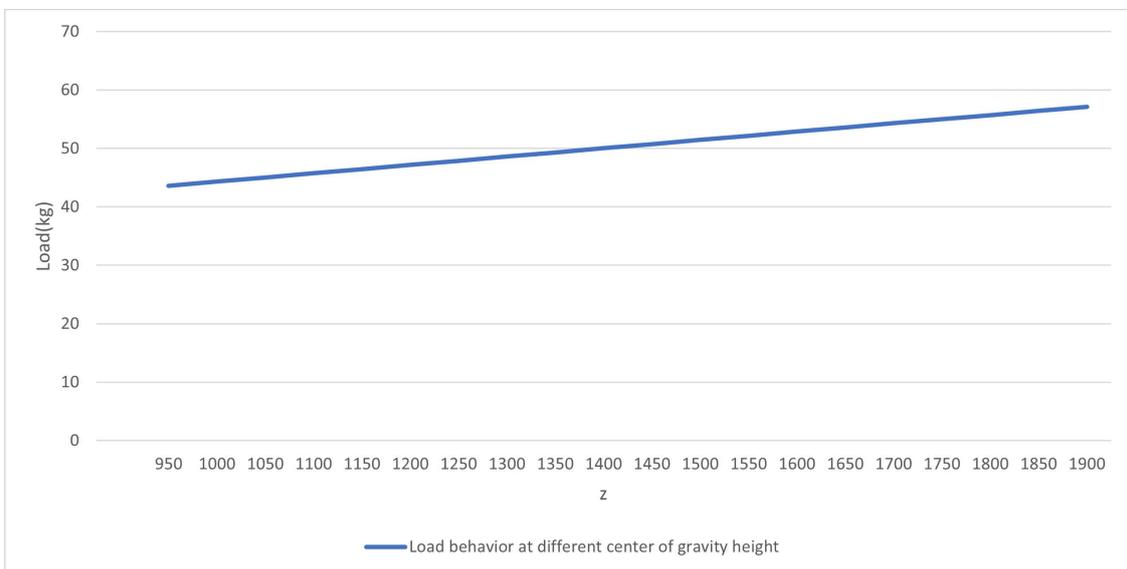


Figure 6.13: Measured load with respect to the load height

The center of gravity starts from 950mm for a weight of 160kg. Then, with two boxes of 160kg the center of gravity was equal to 1640 mm and, with three of them it became 2045 mm.

Another interesting experiment is the one that includes a distributed load on the truck. The same experiment as before has been held, but this time the load on the truck has been distributed uniformly. As shown in the following figure, even if the load is doubled the errors have been a little bit reduced.



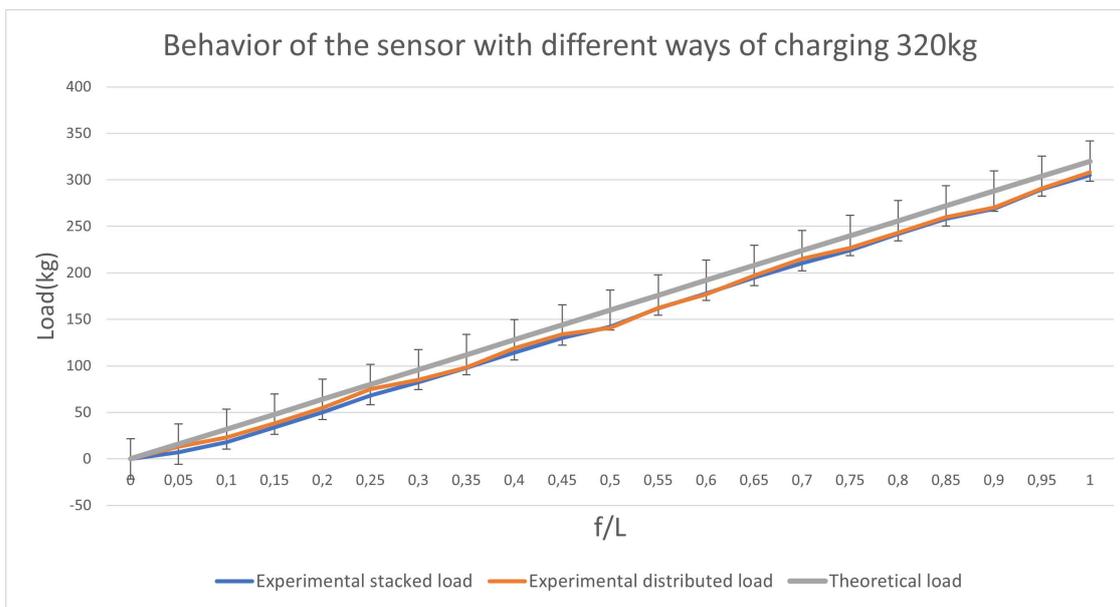


Figure 6.14: Behavior of the sensor measurement for a distributed load of 320kg

In the figure 6.14 the same experiment as before has been held but with a distributed load of 320kg; so we charged the lorry with two boxes without stacking them on top of each other. This experiment is promising since usually the load charged in a truck is distributed and also, trucks are generally fully filled up. The error due to the distributed load is a little bit lower than the error caused by the stacked boxes load, but the two curves are still similar. In fact, it can be noticed in the following graphic that the error due to the distributed load is lower with respect to the stacked one. The following first picture shows the errors with respect to the theoretical one and the second one, with respect to the real load value.

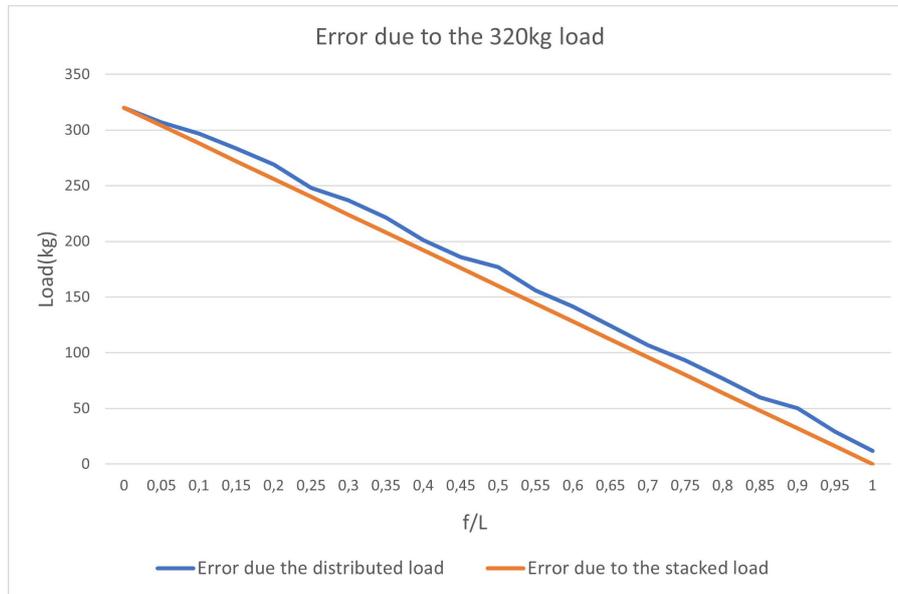


Figure 6.15: Behavior of the error for different ways of charging the 320kg load

However, to correct the errors, more elements are needed in order to program correctly the electronic board. The aim is to program the future board (Arduino for example) calibrating the electronic component in order to add the error to the load acquired by the sensor in order to display the total value of the weight. This can be done with the information related to the angle of inclination of the road but also of the platform's truck given by the accelerometer angle added on the Arduino board. The error as expected increases with the value of weight; this is why it's worthy to realize another solution, including sensors for suspension springs, on the front axle.

## 6.4 Uncertainty

On this paragraph, the goal is to find with which uncertainty, the value of weight will be measured. In our case, the position of the weight on the truck needs to be known by the electronics in order to give the right value of payload. So, the goal is to print the errors based on the position of the load and then, see which errors will influence the load measurement. The errors will be used by the Arduino board to evaluate the right value of weight. There are, as seen before, different factors that influence the measurements and so, this weighting system. The experimental and theoretical error will be computed.

First of all, the theoretical uncertainty will be determined. It can be seen in the first paragraph of the section "Theoretical model" that the first source of error is the strain gage. A sensor of an uncertainty equal to  $\Delta U_o = 24mV$  has been used. To compute the error on the load measurement, it will be considered that the position of the weight on the

truck will be measured manually. So, an error about  $\Delta f = \Delta L = \pm 10mm$  will be taken into account.

The output of the strain gauge is given by the following scheme:



Figure 6.16: Resistive load connection to the strain gauge

Taking into account the fact that,  $K_{FEM} = 3.92N/\mu\epsilon$ ,  $\overline{R}_L = (120 \pm 3\%)\Omega = (120 \pm 3.6)ohms$ ,  $\overline{R} = (350 \pm 3)\Omega$

$$\overline{I}_o \pm \Delta I_o = \frac{\overline{U}_o \pm \Delta U_o}{\overline{R}_L \pm \Delta R_L} = \frac{V_{dc}}{R_L} * \frac{\Delta R}{4 * (\overline{R} \pm \Delta R)} = (\overline{I}_o \pm 0,0016)mA$$

where,

$$\epsilon = \frac{m_{measurement} * g}{K_{FEM}}$$

So,

$$\overline{I}_o \pm \Delta I_o = \frac{\overline{U}_o \pm 1,2}{\overline{R}_L \pm 3,6} = \frac{V_{dc}}{\overline{R}_L \pm \Delta R_L} * \frac{(\overline{m}_{measurement} \pm \Delta m_{measurement}) * g}{4 * K_{FEM}}$$

This relationship defines the mass measured by the sensor function of the output current of the sensor.

$$(\overline{m}_{measurement} \pm \Delta m_{measurement}) = \frac{\Delta R * 4K_{FEM}}{(\overline{R} \pm \Delta R) * g}$$

The following value of error will be obtained:

$$(\overline{m}_{measurement} \pm 0,8)kg$$

The last step will be to translate the measure load into the real load value, that will depend on the position of payload on the truck, as seen before.

$$m_{measured} = m_{real} * \frac{f}{L}$$

So,

$$(\overline{m_{real}} \pm \Delta m_{real}) = \frac{\Delta R * 4K_{FEM}}{\overline{R} * g} * \frac{\overline{L}}{\overline{f}} \pm \frac{\frac{\Delta R * 4K_{FEM}}{\overline{R} * g} * \overline{L}}{\overline{f}} * \left( \frac{\Delta f}{\overline{f}} + \frac{\Delta L}{\overline{L}} + \frac{\Delta R}{\overline{R}} \right)$$

Where,

$$\overline{L} = (3840 \pm 10)mm$$

$$\overline{f} = (1920 \pm 10)mm$$

The results will give the mean value of the real possible load on the truck with its respective error. So, in this case it will be:

$$(\overline{m_{real}} \pm \Delta m_{real}) = (m_{real} \pm 1.3)kg$$

This means that the relative error will be of about 0.8% on the load measurement.

If the box of 160 kg is taken into account, since the equation of the error is known with respect to the position on the truck, the electronic board will calculate the right load using the following curve characteristic.

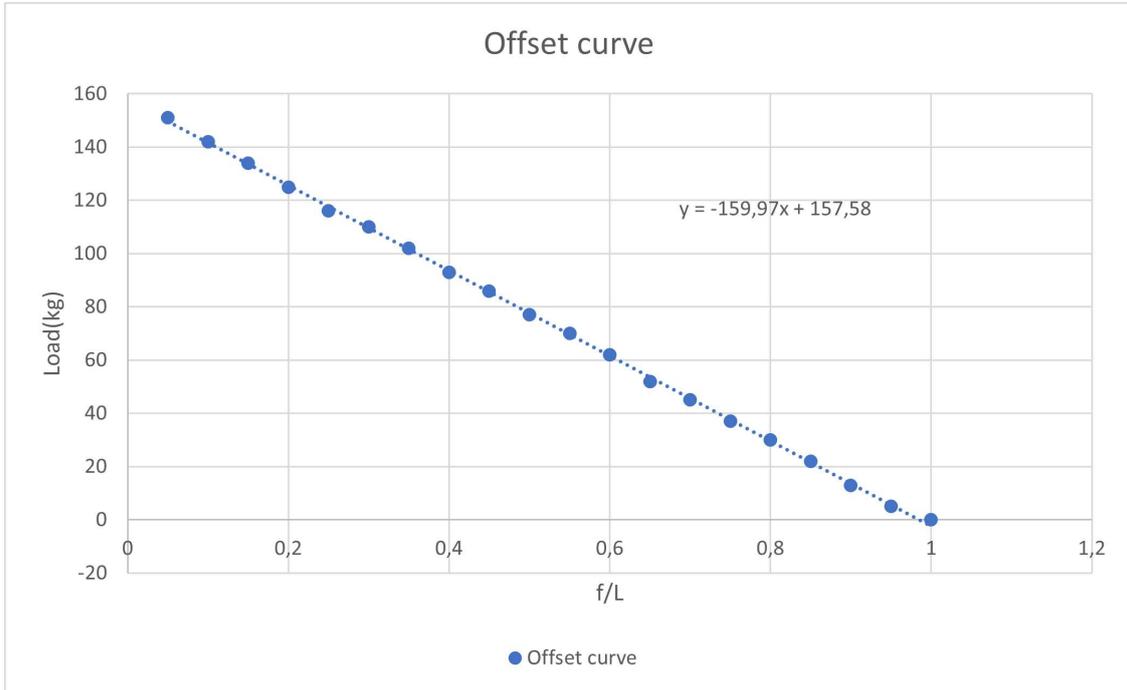


Figure 6.17: Error based on the position of the load

This means that the offset added to the load measure will be equal to:

$$offset = -159,97 * x + 157,58$$

Where  $x=f/L$  is the position and the offset is the error on kg. The load will be displayed by the electronic board with an error of the 0.8%.

Then, another error that needs to be taken into account is the inclination of the road. In fact, repeating the same experiment for different inclination of the road, the value of the same weight in the same position changed.

$$\overline{m_{real}} \pm \Delta m_{real} = \frac{\overline{m_{measured}} \pm \Delta m_{measured}}{(\overline{f} \pm \Delta f)\cos\alpha + (\overline{z} \pm \Delta z)\sin\alpha} * (\overline{L} \pm \Delta L)$$

where,

$$\overline{I_0} = (0,2 \pm 0,016)mA$$

$$\overline{z} = (950 \pm 10)mm$$

$$\overline{L} = (3840 \pm 10)mm$$

$$\overline{f} = (1920 \pm 10)mm$$

The angle of inclination will be considered measured manually. In the future the Arduino will be able to measure it automatically. For now, it can be considered an angle measured with an assumption of uncertainty equal to  $2^\circ$  and the nominal value is equal to  $10^\circ$ .

First, the variable  $(\overline{r} \pm \Delta r) = \overline{f}\cos(\overline{\alpha}) + \overline{z}\sin(\overline{\alpha}) \pm [\frac{\Delta f}{f} + \frac{\delta z}{z} + \Delta\alpha * \cos(\alpha) + \Delta\alpha * \sin(\alpha)]$  will be calculated.

$$(\overline{r} \pm \Delta r) = (1923 \pm 10)mm$$

$$(\overline{m_{real}} \pm \Delta m_{real}) = \frac{\Delta R * 4K_{FEM}}{\overline{R} * g} * \frac{\overline{L}}{\overline{r}} \pm \frac{\frac{\Delta R * 4K_{FEM}}{\overline{R} * g} * \overline{L}}{\overline{r}} * (\frac{\Delta r}{\overline{r}} + \frac{\Delta L}{\overline{L}} + \frac{\Delta R}{\overline{R}})$$

The error given by the inclination is too high. In fact, the value of the measured weight will be equal to

$$\overline{m_{real}} \pm \Delta m_{real} = (m_{real} \pm 2)kg$$

i.e.

$$\frac{\Delta m_{real}}{m_{real}} = 1.3\%$$

The error will grow with the angle. If the angle is kept low such as in this case the error can be assumed the same as before. But whenever a great inclination is experienced by the heavy vehicle the load measurement can be mistaken.

Plus, whenever boxes are added and the load increases, the error increases too, because

the height increases as seen before.

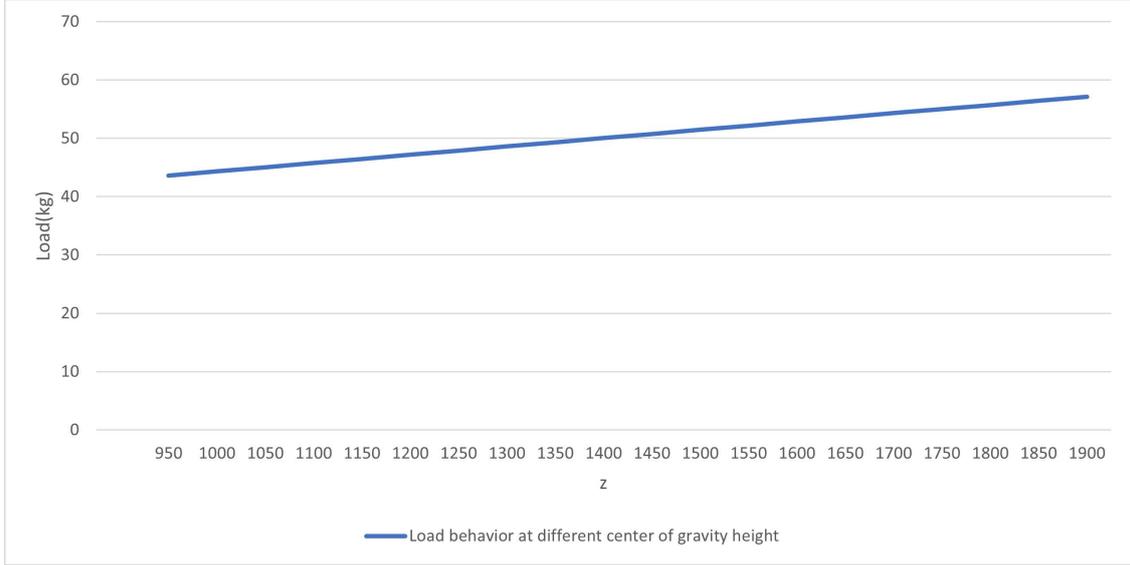


Figure 6.18: Measured load with respect to the load height

The error given by the height will be added since the inclination is known and the height of the center of gravity too.

It is clear that the inclination and the height of the center of gravity will influence the measurement heavily. This is why it's important to realize, in the future, a new electronic solution with a inclinometer able to take into account the inclination of the road.

It can be concluded that the the total error on the weight acquisition will be about  $\pm 1,3kg$ ; it can be written that the value measured by the sensor, in normal conditions, such as on a flat road, will be equal to  $(P \pm 1,3)kg$ .

Now, for what concerns the experimental error, since it can be considered that the sensor measures the same value of mass (in this case,  $m_{real} = 160kg$ ) in different positions, the mean value of all the measurement can be computed to determine the error. More precisely, the relationships that allows to compute the experimental error are the following ones.

$$\hat{m} = \sqrt{\frac{\sum_{i=1}^n (m_{measured_i} - \bar{m})^2}{n - 1}}$$

Where  $\bar{m}$  is the mean observed value of mass, and  $\hat{m}$  the mass experimental error. In other words each value of  $m_{measured}$  will be subtracted to the mean value that is defined as follows:

$$\bar{m} = \frac{m_{measured_1} + m_{measured_2} + m_{measured_3} + \dots + m_{measured_n}}{n}$$

Since the relationship between position and load is known, the offset given by the position is added to each measurement depending on the location of the load on the truck. It means that for a load of 160kg the mass  $m_{measured_i}$  will be translated into:

$$m_i = m_{measured_i} + offset$$

Taking into account that,

$$offset = -159,97 * x + 157,58$$

In the case where the mass charged into the truck is 160 kg the mean value is equal to

$$\bar{m} = 160kg$$

| position(f/L) | $m_{measured}$ | $m_i$ | $(m_i - \bar{m})^2$ |
|---------------|----------------|-------|---------------------|
| 1             | 160            | 160   | $(160-160)^2$       |
| 0.95          | 155            | 159   | $(159-160)^2$       |
| ...           | ...            | ...   | ...                 |
| ...           | ...            | ...   | ...                 |
| 0             | 9              | 158   | $(158-160)^2$       |

The experimental error will be simply the summation of the last column of the table divided by the number of measurements that in this case is equal to (21-1), since the load is installed in 21 different positions in order to carry out the experiment. Finally, it can be concluded that the experimental error is equal to:

$$\hat{m} = \sqrt{\frac{\sum_{i=1}^n (m_i - \bar{m})^2}{n - 1}} = 0,9kg$$

The weight measured experimentally will be equal to:

$$(P \pm 0,9)kg$$

It means that the relative error is about 0,5 %

## 6.5 IoT analysis results

Since the MQTT principle was explained, the analysis of data can be explained. First, the code the arduino uses to connect via MQTT to the IoT system that in this case is the eclipse IoT broker, explained in the following paragraph. The goal is to create an application useful for the driver and the manufacturers to see the load carried by the vehicle and prevent

whenever the maximum limits of load are over-passed.

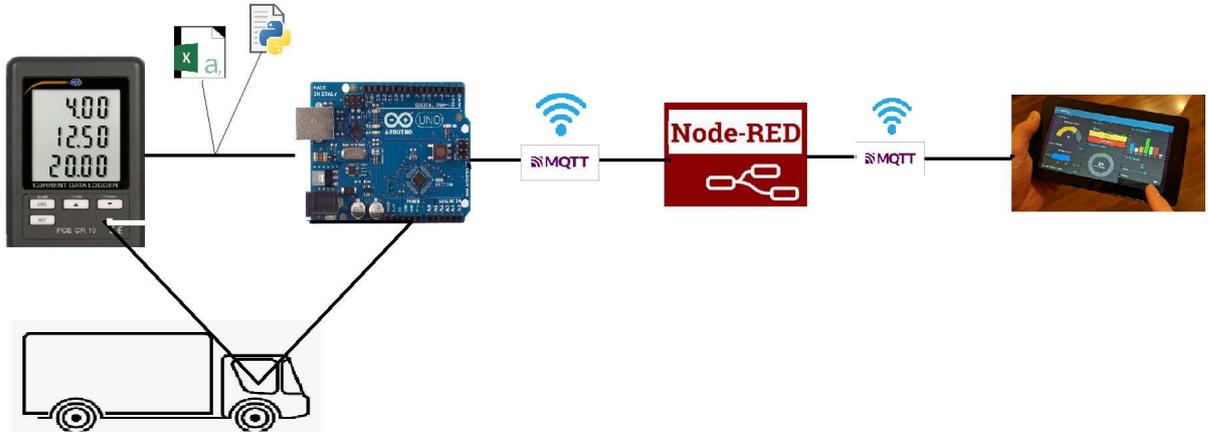


Figure 6.19: Transmission chain starting from the output of the sensor  
 First, we need to specify that the output of the sensor will be connected to a datalogger, that is connected to the arduino board. So, the arduino board will receive a csv file with all values of current of the right and left wheel, that are respectively in the third and fifth column of the following file example.

|       | A     | B          | C        | D    | E  | F    | G  | H | I  | J | K | L | M | N | O | P | Q |
|-------|-------|------------|----------|------|----|------|----|---|----|---|---|---|---|---|---|---|---|
| 23771 | 23770 | 27/04/2021 | 07:43:38 | 4,1  | mA | 3,94 | mA | 0 | mA |   |   |   |   |   |   |   |   |
| 23772 | 23771 | 27/04/2021 | 07:43:39 | 4,06 | mA | 4,14 | mA | 0 | mA |   |   |   |   |   |   |   |   |
| 23773 | 23772 | 27/04/2021 | 07:43:40 | 4,24 | mA | 4,03 | mA | 0 | mA |   |   |   |   |   |   |   |   |
| 23774 | 23773 | 27/04/2021 | 07:43:41 | 4,26 | mA | 4,04 | mA | 0 | mA |   |   |   |   |   |   |   |   |
| 23775 | 23774 | 27/04/2021 | 07:43:42 | 4,1  | mA | 3,94 | mA | 0 | mA |   |   |   |   |   |   |   |   |
| 23776 | 23775 | 27/04/2021 | 07:43:43 | 4,02 | mA | 4,05 | mA | 0 | mA |   |   |   |   |   |   |   |   |
| 23777 | 23776 | 27/04/2021 | 07:43:44 | 4,08 | mA | 4,02 | mA | 0 | mA |   |   |   |   |   |   |   |   |
| 23778 | 23777 | 27/04/2021 | 07:43:45 | 4,18 | mA | 4,08 | mA | 0 | mA |   |   |   |   |   |   |   |   |
| 23779 | 23778 | 27/04/2021 | 07:43:46 | 4,18 | mA | 4,13 | mA | 0 | mA |   |   |   |   |   |   |   |   |
| 23780 | 23779 | 27/04/2021 | 07:43:47 | 4,26 | mA | 4,24 | mA | 0 | mA |   |   |   |   |   |   |   |   |
| 23781 | 23780 | 27/04/2021 | 07:43:48 | 4,26 | mA | 4,24 | mA | 0 | mA |   |   |   |   |   |   |   |   |
| 23782 | 23781 | 27/04/2021 | 07:43:49 | 4,3  | mA | 4,15 | mA | 0 | mA |   |   |   |   |   |   |   |   |
| 23783 | 23782 | 27/04/2021 | 07:43:50 | 4,25 | mA | 4,15 | mA | 0 | mA |   |   |   |   |   |   |   |   |
| 23784 | 23783 | 27/04/2021 | 07:43:51 | 4,32 | mA | 4,17 | mA | 0 | mA |   |   |   |   |   |   |   |   |
| 23785 | 23784 | 27/04/2021 | 07:43:52 | 4,29 | mA | 4,21 | mA | 0 | mA |   |   |   |   |   |   |   |   |
| 23786 | 23785 | 27/04/2021 | 07:43:53 | 4,05 | mA | 4,39 | mA | 0 | mA |   |   |   |   |   |   |   |   |
| 23787 | 23786 | 27/04/2021 | 07:43:54 | 4,02 | mA | 4,46 | mA | 0 | mA |   |   |   |   |   |   |   |   |
| 23788 | 23787 | 27/04/2021 | 07:43:55 | 4,15 | mA | 4,5  | mA | 0 | mA |   |   |   |   |   |   |   |   |
| 23789 | 23788 | 27/04/2021 | 07:43:56 | 4,34 | mA | 4,3  | mA | 0 | mA |   |   |   |   |   |   |   |   |
| 23790 | 23789 | 27/04/2021 | 07:43:57 | 4,54 | mA | 4,1  | mA | 0 | mA |   |   |   |   |   |   |   |   |
| 23791 | 23790 | 27/04/2021 | 07:43:58 | 4,56 | mA | 4,13 | mA | 0 | mA |   |   |   |   |   |   |   |   |
| 23792 | 23791 | 27/04/2021 | 07:43:59 | 4,39 | mA | 4,26 | mA | 0 | mA |   |   |   |   |   |   |   |   |
| 23793 | 23792 | 27/04/2021 | 07:44:00 | 4,42 | mA | 4,21 | mA | 0 | mA |   |   |   |   |   |   |   |   |
| 23794 | 23793 | 27/04/2021 | 07:44:01 | 4,24 | mA | 4,04 | mA | 0 | mA |   |   |   |   |   |   |   |   |
| 23795 | 23794 | 27/04/2021 | 07:44:02 | 4,28 | mA | 4,13 | mA | 0 | mA |   |   |   |   |   |   |   |   |
| 23796 | 23795 | 27/04/2021 | 07:44:03 | 4,39 | mA | 4,22 | mA | 0 | mA |   |   |   |   |   |   |   |   |
| 23797 | 23796 | 27/04/2021 | 07:44:04 | 4,27 | mA | 4,2  | mA | 0 | mA |   |   |   |   |   |   |   |   |

Figure 6.20: Output csv file given by the data logger

Those values need to be analysed through the python code that will be translated into Arduino IDE code, in order to remove all duplicates due to the device interference. So, the goal of the python script is to filter data and transfer them in Cloud. Each value of the csv file can be sent one by one and stored in the IoT dashboard, where we can manipulate data. In the python code the cleaning and the transfer by MQTT is implemented. Pycharm application was used to write the codes. We create a publisher by, first specifying the broker address, "mqtt.eclipseprojects.io", and giving a name to the client, "dati-corrente".

Then, the csv file is opened by the python code and with a loop the duplicates are removed. In the loop we read the lines one by one and while we do this we also send them to the broker using the topic "CORRENTE". In outFile we can also find all values without the duplicates. Then, every client willing to see data needs to subscribe to the topic "CORRENTE". We can either see data from an IoT interface or connect with an external client.

In the following pseudo code in figure 6.21, the python code is better explained.

---

**Algorithm 1** An algorithm with caption

---

```

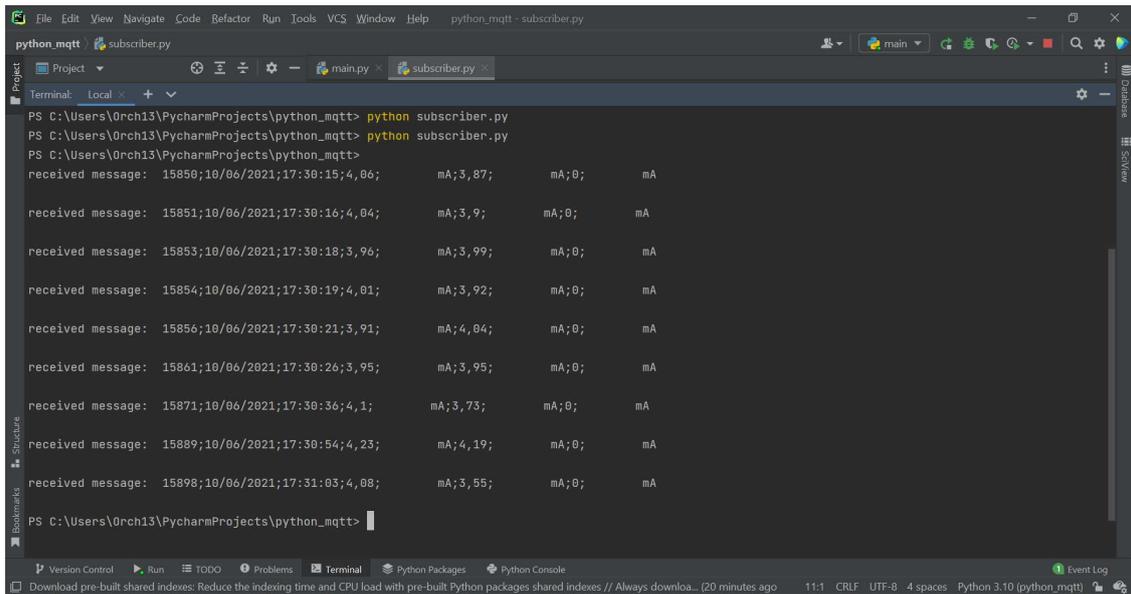
0: File ← CSV file from data logger memory card
Require: File ≠ Empty
0: LeftArray ← []
0: RightArray ← []
0: LineArray ← []
for every line in File do
    line ← split the line using character ;
    Right ← split the third column using character ;
    Left ← split the third column using character ;
    if Right is not inside RightArray then
        Add Right in RightArray
        Add left in LeftArray
        Add line in LineArray
        i ← i + 1
    if i = lim then
        for i from 0 to lim-1 do
            Convert the values of third and fifth column of LineArray into float
            [MAX, Date, Time] ← Give the local maxima and minima, the time and date
            client.publish ← CORRENTE, str(MAX)
            client.publish ← CORRENTE, str(Date)
            client.publish ← CORRENTE, str(Time)
            Diff ← RightArray[j] − LeftArray[j]
            if Diff is inferior to 0 then
                CountRigth ← CountRigth + 1
                Total ← Total + 1 {This is a comment}
            else
                CountLeft ← CountLeft + 1 {This is a comment}
                Total ← Total + 1 {This is a comment}
            end if
            client.publish ← CORRENTE, str(Rightperc)
            client.publish ← CORRENTE, str(Leftperc)
        end for
    end if
    else if Left is not inside LeftArray then
        same code
    end if
end for

```

---

Figure 6.21: Pseudo code representing the python code

Once the script completed, data sent have been printed in the pyCharm terminal in order to see consistency between transmitter (python) and receiver (IoT application).



The screenshot shows a PyCharm IDE window titled 'python\_mqtt - subscriber.py'. The terminal output displays a series of received MQTT messages. Each message is a JSON object with four fields: 'time', 'topic', 'payload', and 'qos'. The messages are as follows:

```
received message: {"time": "15850;10/06/2021;17:30:15;4,06;", "topic": "mA;3,87;", "payload": "mA;0;", "qos": "mA"}
received message: {"time": "15851;10/06/2021;17:30:16;4,04;", "topic": "mA;3,9;", "payload": "mA;0;", "qos": "mA"}
received message: {"time": "15853;10/06/2021;17:30:18;3,96;", "topic": "mA;3,99;", "payload": "mA;0;", "qos": "mA"}
received message: {"time": "15854;10/06/2021;17:30:19;4,01;", "topic": "mA;3,92;", "payload": "mA;0;", "qos": "mA"}
received message: {"time": "15856;10/06/2021;17:30:21;3,91;", "topic": "mA;4,04;", "payload": "mA;0;", "qos": "mA"}
received message: {"time": "15861;10/06/2021;17:30:26;3,95;", "topic": "mA;3,95;", "payload": "mA;0;", "qos": "mA"}
received message: {"time": "15871;10/06/2021;17:30:36;4,1;", "topic": "mA;3,73;", "payload": "mA;0;", "qos": "mA"}
received message: {"time": "15889;10/06/2021;17:30:54;4,23;", "topic": "mA;4,19;", "payload": "mA;0;", "qos": "mA"}
received message: {"time": "15898;10/06/2021;17:31:03;4,08;", "topic": "mA;3,55;", "payload": "mA;0;", "qos": "mA"}
```

Figure 6.22: Received data on the pyCharm terminal

Another way to see data without MindSphere is to create an application that connects to the Eclipse free server and organize data in a dashboard. This last was our first option before knowing about the Mindsphere platform. Since, creating an application is difficult for someone unexperienced and time consuming we decided to use a software called node-red in order to create a dashboard.

Node-RED can be used as a programming tool for connecting together hardware devices, APIs and online services. It makes it easy to code new applications using the nodes (that can be seen as programmable functions) in the palette that can be deployed in a single-click. With node-red a custom dashboard can be created to which it is possible to connect via the web. Node-RED works using JavaScript functions. The programmed nodes can be saved and exported for re-use. The runtime is built on Node.js. The flows created in Node-RED are programmed using the JSON language.

The function "client.publish" used in the pseudo code reported previously, allows to send data to the Node-red application. When creating the application the JSON functions nodes have been used. As shown in the following figure three main groups of nodes have been created.

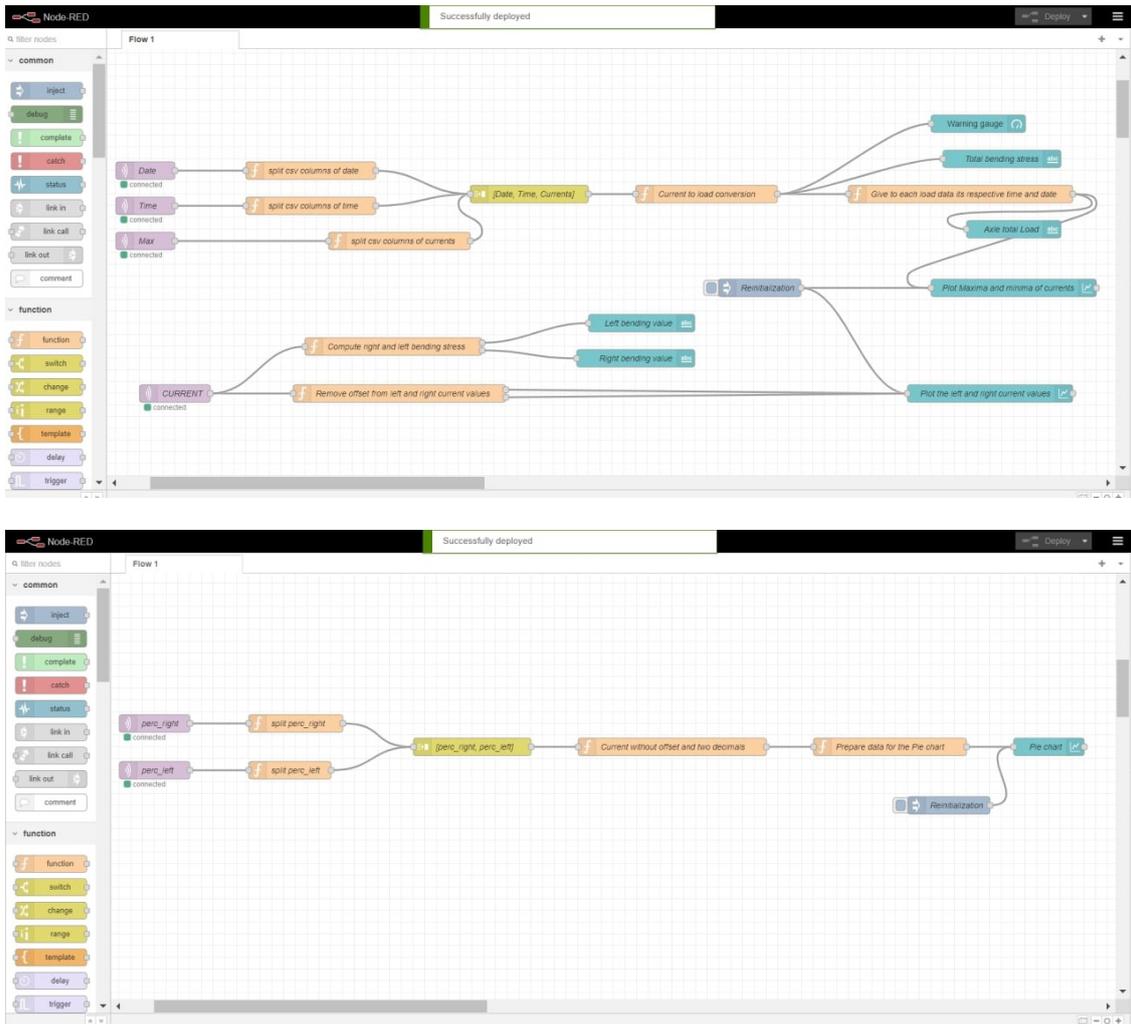


Figure 6.23: Node-red analysis of data sent by MQTT broker

In the first group in figure 6.23, data sent by MAX are converted into load values in order to plot the following curve in the dashboard node-red application and the total axle load, taking into account the data and time at which the acquisition have been held. Also, the total bending stress is calculated as explained in the following paragraphs.

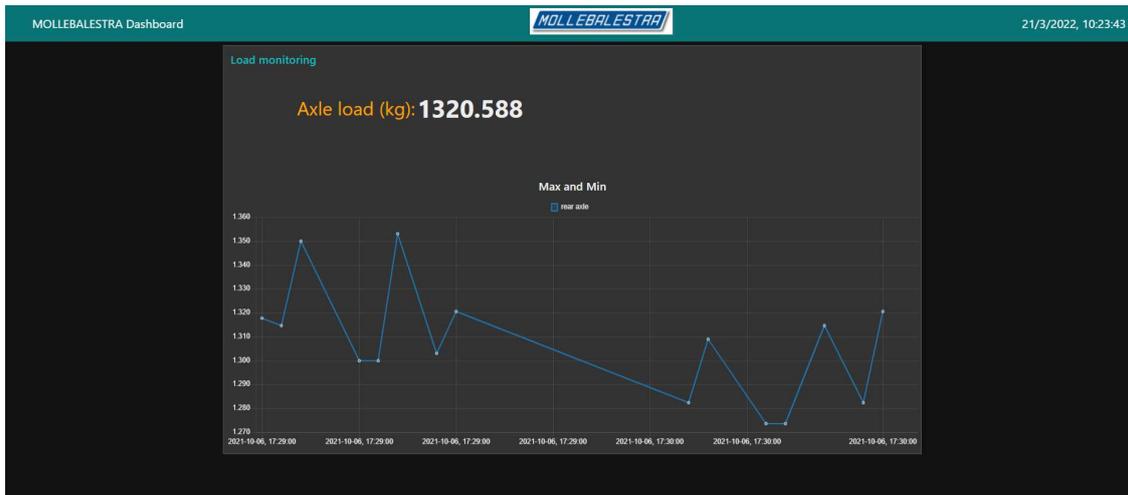


Figure 6.24: Maxima and minima plot

This can be useful to know the maximum value of load carried by the truck during a certain amount of time. In the figure 6.24 we can see plotted the local maxima and minima on 6th October 2021. The same study can be done for other dates. This can be meaningful, for example, to know the wear and tear of the leaf-spring.

Then, in the second group of nodes in figure 6.23, the current values have been translated into bending stress values in order to warn the customer and the manufacturer whenever the maximum load has been reached. This was done by creating a gauge in the dashboard. In this way we can compare the total bending stress reported in the gauge with the left and right values (figure 6.25).

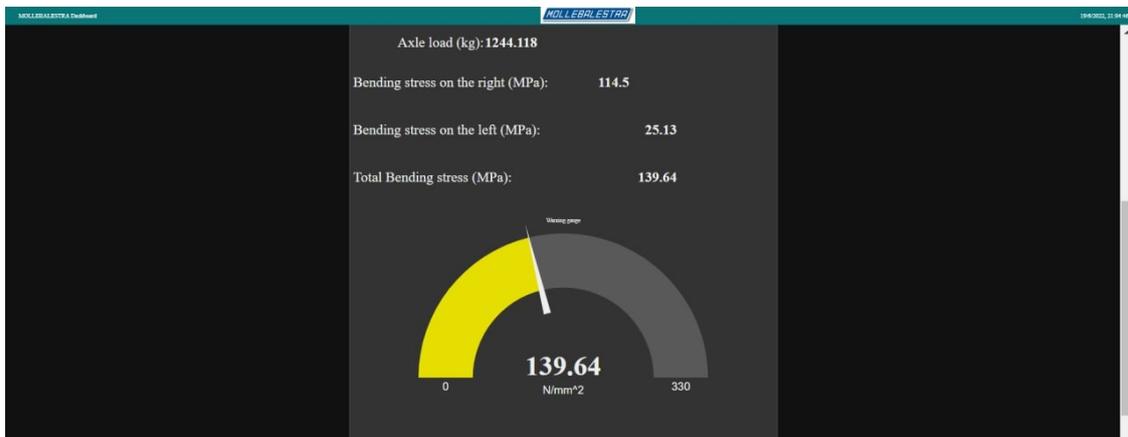


Figure 6.25: Values of Bending stress

It's important to notify the client whenever an axle is carrying too much load or a leaf spring is stressed more than usual. This is why the analysis on Node-red presents a warning system whenever the leaf spring is carrying more than 1450kg, a value near to the maximum dynamic load (1500kg). This warning is also important to prevent the breaking

of the suspension and, if each warning is connected to a time and GPS coordinates, it will be easier to know the reason why leaf springs break up. In the following gauges the warning is shown from the point of view of the bending stress computed thanks to the following relationship;

$$\delta_{FEM} = \frac{6Px}{n_f w t^2}$$

Where the bending stress is  $\delta_{FEM}$  and the load detected by the sensor is P. The other parameters are constant, as stated in paragraph 3.3:

$$n_f = 2$$

$$x = 485\text{mm}$$

$$t = 14\text{mm}$$

$$w = 66\text{mm}$$

This means that the warning is sent to the driver and manufacturer whenever the value of  $162.7\text{N/mm}^2$  (equivalent to 1450kg) is over-passed.

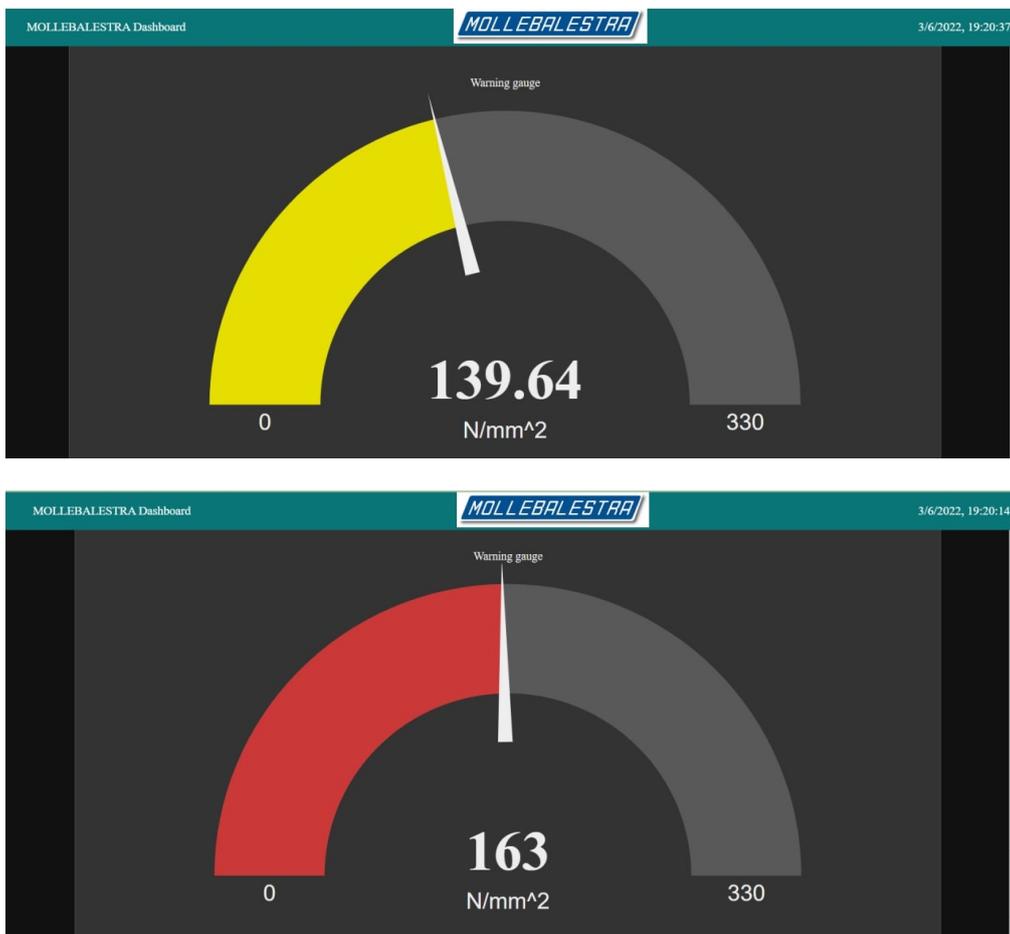


Figure 6.26: Warning gauges with respect to bending stress value

In the third part, shown in the second figure of figure 6.23, the pie chart was created by computing which between the right and left spring was the most stressed every acquisition time. The difference of load carried in time by the right leaf-spring with respect to the left spring was calculated, in order to consider when a leaf spring is more stressed with respect to the other one.

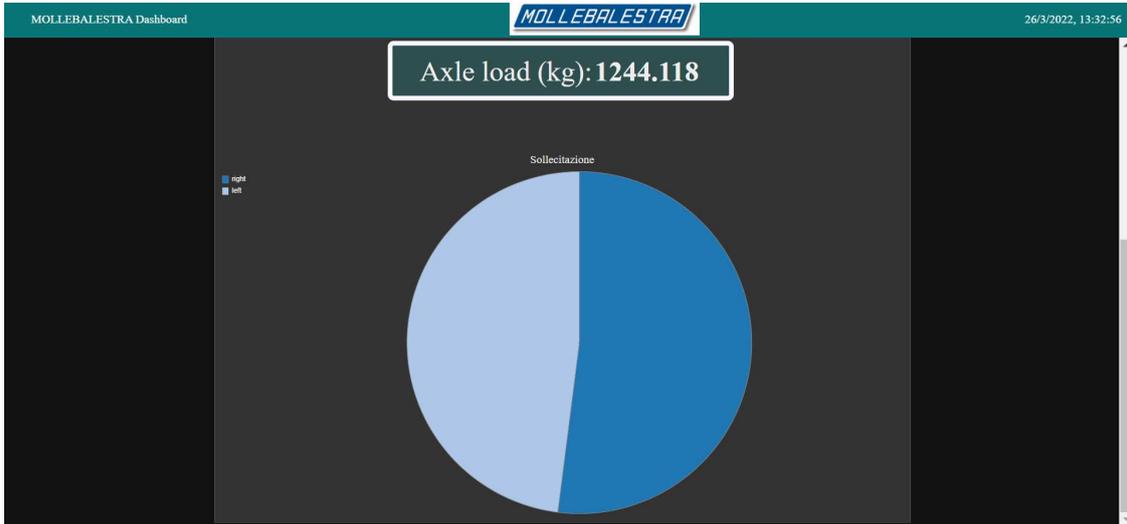


Figure 6.27: Pie chart showing difference of load between left and right leaf spring

In the figure 6.28, for example the right wheel (204.5 kg) is carrying more with respect to the left wheel(69.2 kg). Those plots appeared when turning left with the small lorry. We can say that the peaks appear in a turn and that the right leaf-spring will be teared before the left one, because during all experiments the right suspension carry most of the whole load. This means that the right leaf spring is more sensible, so more flexible in general. Even when the offset is not present, the right part of the vehicle presents more stress and carry most of the load as shown in the following chart.

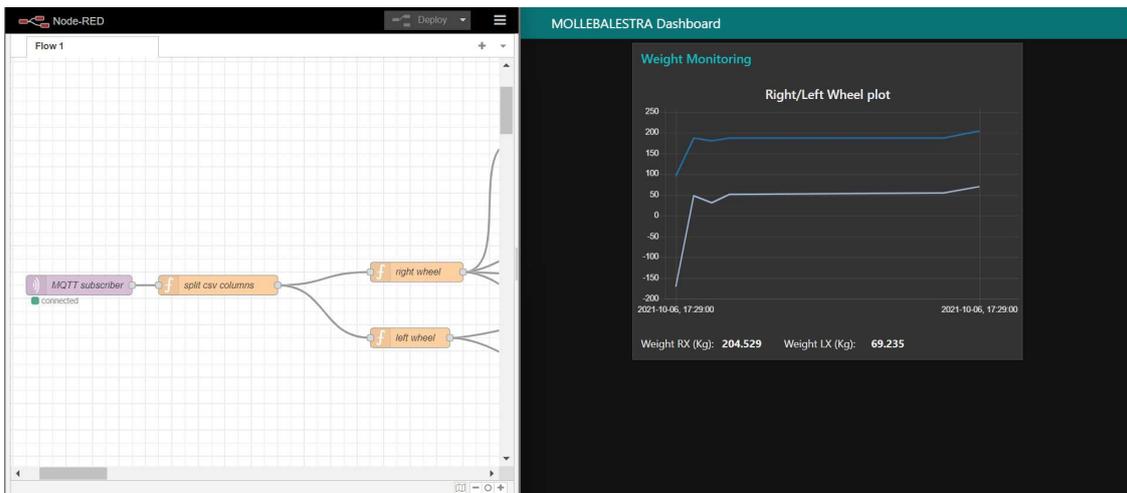


Figure 6.28: Right and left offset on the strain gauges current measurements

On the left in Figure 6.28, we can see the dashboard with the plot and the values of weights (Weight rx and Weight lx) supported by the right and left leaf spring changing in time with the curve. It can be observed that there's an offset with respect to the zero axis; that's why the curves don't start from the load equal to zero. This is due to the fact that during installation when attaching the strain gauge to the leaf spring parasitic tensions appears. At  $P=0\text{kg}$ , the initial current condition is  $3.77\text{mA}$  for the left leaf spring and  $4.08\text{mA}$  for the right one, as shown in the following figure. The previous figure shows the load carried by the left and right suspensions, so the whole weight is not equally distributed because of the offset on the load.

Once the offset is known, the plots without offset can be realised to have consistency with the theory and show that the whole load is equally distributed between the right and left suspension. In the following figure the left and right current are plotted without offset and the two leaf springs carry approximately the same load as shown by the pie chart in figure 6.27.

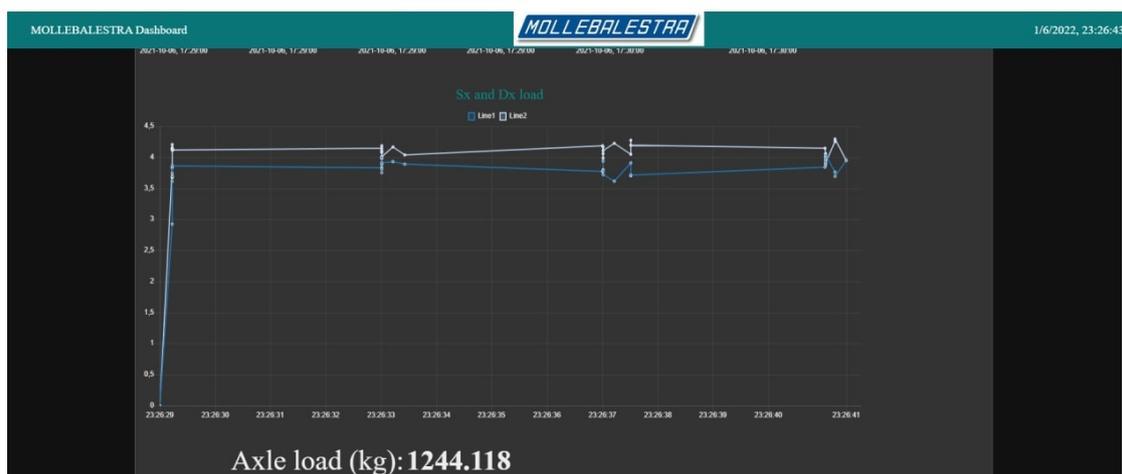


Figure 6.29: Right and left strain gauges current measurements without offset

# Chapter 7

## Conclusion

The aim of this thesis was to realize a weighting system able to give to the user the weight charged by the vehicle. Most importantly the goal was to, first, calibrate and validate the response of the sensor and check for the consistency in its behavior whenever the sensor is installed on the vehicle. The calibration was consistent with the behavior of the sensor as shown during experiments. However, the experiments carried during this work thesis showed that the system is incomplete because, since the sensor is installed only in the rear axle of the lorry, only a part of the load carried by the vehicle is detected. To fully achieve our objectives, the future work will be to find a method to install sensor in springs in order to measure the real load on the vehicle. However, this is not a problem for heavy vehicles with more axles because in that case a strain gage can be used in each axle of the vehicle and no monitoring is needed in the front axle. Also, another aim was to collect data and send them in cloud to an IoT platform in order to prevent any risk of break of the leaf-spring and this was successfully achieved in the last part of this document thesis, explaining the migration in cloud via the MQTT protocol. Obviously the analysis of data can be done easily with a platform like MindSphere, where no coding is needed, but it's more expensive. This is why NodeRed, a free platform using Java script language, was used. With NodeRed, the current values detected by the two strain gages were used to calculate the maximum value of weight reached by the left and right leaf-springs in order to predict the life time of each leaf-spring. This was done thanks to the support of a python script. The life time calculation was not treated in this thesis, but it will be subject of a future work with the same company. In a second moment we also analyzed the difference of forces carried by the right leaf-spring with respect to the left one. This can be useful to know that leaf-springs installed on the right wheel are more subject to fatigue. To conclude, this system can be used as weighting system, but it must be taken into account that the uncertainty of measurement will be about  $\pm 90kg$ . For this kind of company this error can be neglecting since most of the heavy vehicles can carry between 6000kg and 8200 kg. In

the study carried during this thesis, a small lorry was used, able to transport only 1525kg as weight, so this kind of uncertainty could be not desired. However, most of customers use heavy vehicles that are larger and equipped with more axles where leaf-springs can be installed and also sensors. To have a precise measurement as stated in the chapter 4, at least four strain gages are needed. Finally it can be anticipate that this solution can be used in other customers' vehicles with lower uncertainty since more sensors can be installed.

All the study done can be deeply developed in order to have more information about the life of a leaf-spring taking into account the behavior of the driver. The fact that information are stored in Cloud is very valuable to continue this study in the future and predict the geographical point of break in a leaf spring according to past results. The information that will be stored in the next years will give more hints to the manufacturers about what are the factors for which the leaf-springs break and when they need to warn the driver to change its suspensions.

# Appendix A

## An appendix

The following python code is the one used to filter the data logger data

---

```
1 import paho.mqtt.client as mqtt
2 import time
3 import keyboard
4
5
6 mqttBroker = "mqtt.eclipseprojects.io" #free mqtt broker cloud on
   the web https://mqtt.eclipseprojects.io/
7
8 client = mqtt.Client("datiii") #name of the client
9 client.connect(mqttBroker) #connection of the client to the cloud
10
11
12 inFile = open('C:\\Users\\Orch13\\Desktop\\Mollebalestra_
   Tesi\\TESI\\SIHAM\\DATI\\MAA02002.csv','r') #open the csv file
13
14 #outFile = open('C:\\Users\\Orch13\\Desktop\\Mollebalestra
   Tesi\\TESI SIHAM\\DATI\\my_file.csv','a+')
15
16 new=[] #data without duplicates
17 sum=[] # sum of right and left current
18 subtracted=[]
19 current = []
20 new_test=[] #data of the left wheel without duplicates
21 new_test2=[] #data of the right wheel without duplicates
22
23 lim =10 #number of data sent each time
24 count=0 #number of data sent in total
```

```
25 time_sum=[]
26 date_sum=[]
27
28 maximas_sum = [] #local maxima and minima of the current
29
30 perc_right=0
31 perc_left=0
32 count_total =0
33 count_left =0
34 count_right =0
35 i = 0
36 j=0
37 a=0
38 def wait():
39     while True:
40         if keyboard.is_pressed("space"):
41             break
42
43 for line in inFile:
44
45     #split of right and left values by ;
46     value_test = line.split(';')[5]
47     value_test2 = line.split(';')[3]
48     value = line.split(';')
49
50     #loop to remove the duplicates by checking left values
51     if not value_test in new_test:
52
53         new_test.append(value_test)
54         # outFile.write(line)
55         new.append(value)
56
57
58
59     new_test2.append(value_test2)
60
61     #remove the headers of the csv file
62     if a == 0:
63         new.pop(0)
64         a = 1
65     count = count + lim
66     i += 1
```

```
67
68     #send package of lim each time
69     if i == lim:
70
71         i = 0
72
73         #convert values in float before summing them
74         for j in range(lim - 1):
75             new[j][3] = float(new[j][3].replace(",", "."))#
76                 right
77             new[j][5] = float(new[j][5].replace(",", "."))#
78                 left
79             current.append([new[j][3], new[j][5]])
80             # client.publish("CORRENT11", str(new[j][3]))
81             client.publish("CORRENT22", str(current))
82             # print(current)
83             item = (new[j][3]) - (new[j][5])
84
85             if item >= 0:
86                 count_right = count_right+1
87                 count_total = count_total+1
88             else:
89                 count_left = count_left + 1
90                 count_total = count_total + 1
91
92             perc_right = count_right/count_total
93             perc_left = count_left / count_total
94             client.publish("CORRENT1", str(perc_right))
95             client.publish("CORRENT2", str(perc_left))
96
97             subtracted.append(item)
98             sum.append(new[j][5] + new[j][3])
99             # print(new)
100
101     #find local maxima and minima
102     for j in range(lim - 3):
103
104         # print(sum[j])
105         # print(sum[j+1])
106         # print(sum[j+2])
```

```
107
108         if sum[j + 1] <= sum[j + 2] and sum[j + 1] <=
           sum[j]:
109             maximas_sum.append(sum[j + 1])
110             date_sum.append(new[j + 1][1])
111             time_sum.append(new[j + 1][2])
112         elif sum[j + 1] >= sum[j + 2] and sum[j + 1] >=
           sum[j]:
113             maximas_sum.append(sum[j + 1])
114             date_sum.append(new[j + 1][1])
115             time_sum.append(new[j + 1][2])
116         # print(maximas_sum)
117         # wait()
118
119
120         #send values by MQTT one by one
121         for k in range(len(date_sum)):
122             client.publish("CORRENTE4", str(date_sum[k]))
123             client.publish("CORRENTE3", str(time_sum[k]))
124             client.publish("CORRENTE1", str(maximas_sum[k]))
125             #print(maximas_sum[k])
126             time.sleep(2)
127
128
129         new.clear()
130         maximas_sum.clear()
131         date_sum.clear()
132         time_sum.clear()
133
134         # time.sleep(2)
135
136         #if consecutive left values are the same check right values
137         elif not value_test2 in new_test2:
138             #outFile.write(line)
139             new.append(value)
140             new_test2.append(value_test2)
141
142         if a == 0:
143             new.pop(0)
144             a=1
145         count=count+lim
146         i += 1
```

```
147
148
149     if i == lim:
150
151         i=0
152
153
154     for j in range(lim-1):
155
156         new[j][3] = float(new[j][3].replace(",",".",
157                                # right
158                                "."))
159         new[j][5] = float(new[j][5].replace(",",".",
160                                # left
161                                "."))
162         current.append([new[j][3], new[j][5]])
163         # client.publish("CORRENT11",
164             str(new[j][3]))
165         client.publish("CORRENT22", str(current))
166         item = (new[j][3]) - (new[j][5])
167         if item > 0:
168             count_right = count_right + 1
169             count_total = count_total + 1
170         else:
171             count_left = count_left + 1
172             count_total = count_total + 1
173
174         perc_right = count_right / count_total
175         perc_left = count_left / count_total
176         client.publish("CORRENT1", str(perc_right))
177         client.publish("CORRENT2", str(perc_left))
178
179         subtracted.append(item)
180         sum.append(new[j][5]+new[j][3])
181         #print(new)
182
183     for j in range(lim-3):
184         #print(sum[j])
185         #print(sum[j+1])
186         #print(sum[j+2])
187
188     if sum[j+1] <= sum[j+2] and sum[j+1] <=
189         sum[j]:
```

```
185         maximas_sum.append(sum[j+1])
186         date_sum.append(new[j+1][1])
187         time_sum.append(new[j+1][2])
188     elif sum[j+1] >= sum[j+2] and sum[j+1] >=
        sum[j]:
189         maximas_sum.append(sum[j + 1])
190         date_sum.append(new[j + 1][1])
191         time_sum.append(new[j + 1][2])
192     #print(maximas_sum)
193     #wait()
194
195
196     for k in range(len(date_sum)):
197         client.publish("CORRENTE4", str(date_sum[k]))
198         client.publish("CORRENTE3", str(time_sum[k]))
199         client.publish("CORRENTE1",
                str(maximas_sum[k]))
200         #print(maximas_sum[k])
201         time.sleep(2)
202
203
204     new.clear()
205     maximas_sum.clear()
206     date_sum.clear()
207     time_sum.clear()
208 inFile.close()
```

---

# Bibliography

- [1] Michihiko Ayada et al. *Leaf spring material and manufacturing method thereof*. US Patent App. 12/324,586. May 2009.
- [2] AE Baumal, JJ McPhee, and PH Calamai. “Application of genetic algorithms to the design optimization of an active vehicle suspension system”. In: *Computer methods in applied mechanics and engineering* 163.1-4 (1998), pp. 87–94.
- [3] D Cai et al. “Australia’s intelligent access program (IAP): Enabling improved road safety outcomes”. In: *Proceedings of the Australasian road safety research, policing and education conference*. Vol. 14. Monash University. 2010.
- [4] Ramin S Esfandiari and Bei Lu. *Modeling and analysis of dynamic systems*. CRC press, 2018.
- [5] Karl Hoffmann. *Applying the wheatstone bridge circuit*. HBM Germany, 1974.
- [6] F Lanza di Scalea. “Measurement of thermal expansion coefficients of composites using strain gages”. In: *Experimental mechanics* 38.4 (1998), pp. 233–241.
- [7] Fábio Lima, Alexandre Augusto Massote, and Rodrigo Filev Maia. “IoT energy retrofit and the connection of legacy machines inside the Industry 4.0 concept”. In: *IECON 2019-45th Annual Conference of the IEEE Industrial Electronics Society*. Vol. 1. IEEE. 2019, pp. 5499–5504.
- [8] Vishay Micro-Measurements. “Strain gage thermal output and gage factor variation with temperature”. In: *Strain Gauges and Instruments, Vishay Micro-Measurements* (2010).
- [9] Ivan Muller et al. “Load cells in force sensing analysis—theory and a novel application”. In: *IEEE Instrumentation & Measurement Magazine* 13.1 (2010), pp. 15–19.
- [10] European Parliament and the Council. “Directive (EU) 2015/720 of the European Parliament and of the Council of 29 April 2015 amending Directive 94/62/EC as regards reducing the consumption of lightweight plastic carrier bags”. In: *Off. J. Eur. Communities L115* 11 (2015), pp. 4–8.

- [11] Goran Radoičić, Miomir Jovanović, and Miodrag Arsić. “Experience with an On-board Weighing System Solution for Heavy Vehicles”. In: *Etri Journal* 38.4 (2016), pp. 787–797.
- [12] Keith W Reichow, David C English, and Jerry L McCauley. *Vehicle load measuring system*. US Patent 4,042,049. Aug. 1977.
- [13] Andy Stanford-Clark and Hong Linh Truong. “Mqtt for sensor networks (mqtt-sn) protocol specification”. In: *International business machines (IBM) Corporation version 1.2* (2013), pp. 1–28.
- [14] Richard Stokes. “Weigh to go”. In: *Traffic Technology International* (2005).