ABSTRACT

Title of dissertation: MODELING AND IMPLEMENTING
A SELF-DRIVING GOLF CART

Jacob García Palau
Master of Science, 2019

Dissertation directed by: Dr. Yasser Shoukry
Department of Electrical Engineering

A new project about designing and implementing an autonomous golf cart is introduced with the ultimate goal of running tests and cyber-attacks on it to develop new techniques and improve existing ones for the improvement of resilience and security of cyber-physical systems.

The current thesis tackles the two of the aspects of the cyber-physical system that is the autonomous golf cart. In the physical part of the system the steering mechanism and the braking mechanism are designed and well defined and an overview of a solution for the mechanism of the accelerator is also covered.

On the cybernetic aspect of the project the algorithm of YOLO has been implemented for the computer vision of the golf cart. This algorithm relies on a deep convolutional neural network which architecture has been changed from the standard DarkNet to the MobilenetV2. MobileNetV2 provides good results for object classification despite the fact that it has much less parameters than other architectures, so this change has been done to test how good can it perform with respect to DarkNet, which is the architecture which the algorithm was created with.

After training the network, it was able to make good detections of easy cases but struggled with cases in where more elements to detect were present which could be caused by the inability of MobileNetV2 to capture the complexity of the problem or by an encounter of a local minimum during training.

MODELING AND IMPLEMENTING A SELF-DRIVING GOLF CART

by

Jacob García Palau

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Master of Science
2019

Advisory Committee:
Dr. Yasser Shoukry, Advisor

Dedication

Os lo dedico a vosotros, yayos. Por hacer posible mi viaje. Por transmitirme siempre tanta felicidad. Por ser como sois. Por eso y por mucho mas, gracias. Os quiero.

Te dedico mi trabajo a ti, papá. Por siempre estar ahí cuando lo he necesitado. Por transmitirme la tranquilidad que necesitaba en los momentos difíciles. Por ayudarme a convertirme en la persona que soy hoy. Siempre estaré agradecido por todo lo que has hecho por mí. Gracias por ser mi padre. Te quiero papá.

Te dedico mi trabajo a ti, mamá. Por haber apostado siempre por mí sin importar las circunstancias. Por hacer de mí la persona que soy hoy. Por estar ahí en los momentos difíciles o simplemente cuando he necesitado tu compañía. Todo lo que escriba aquí siempre será insuficiente para mostrarte toda mi gratitud. Gracias por ser mi madre. Te quiero mamá.

Acknowledgments

First and foremost I'd like to thank my advisor, Dr. Yasser Shoukry for giving me the great opportunity to work on this amazing project. He has always made himself available for help and advice, and also has created a great work environment.

I'm also grateful to the University of Maryland and the Polytechnic University of Valencia for giving me the opportunity to work on this project, teaching me about many interesting subjects and showing me other cultures and experiences.

I'd also like to thank all my friends for making this journey unforgettable and I am grateful that we were able to share one of the most amazing experiences together.

Finally, I'd like to show my gratitude to my family, who has given me support to take this opportunity and I couldn't have reached this far without them.

Table of Contents

List of Figures

Chapter 1: Introduction

The population has been starting to imagine more clearly the idea of having the streets filled with autonomous cars. This can be explained by the technological advances that the media has shown, such as the new commercial and autonomous Tesla cars, the research that's taking place in the Mcity of the University of Michigan [13] or the Google self-driving cars [12].

To reach the point at which the state of the art is now, there have been numerous events in history that date back to much before that one could expect. The first evidence that the human being had in mind the idea of autonomous driving was much before than the invention of the car itself and dates back to 1478, when Leonardo da Vinci invented the self-propelled car [5]. The self-propelled car was a very complex mechanism by the time. It consisted of 3 wheels and a set of gears packed in a wooden body that sized around 1x1 meters (Figure 1.1). The way that it worked was that by turning a specific gear, a set of springs was compressed allowing the storage of energy that would propel the mechanism. Then, the front wheel could be adjusted so the car could follow a fixed trajectory along a desired path.

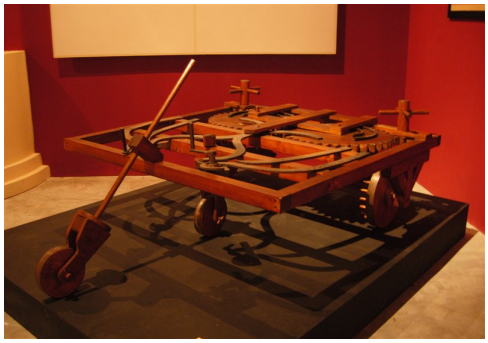


Figure 1.1: Da Vinci's self-propelled car.

Much later in 1939 we find the first presentation of an autonomous car by Norman Bel Geddes in an exhibition at the fair Futurama. The project consisted in an electrical autonomous car that could follow an electrical circuit that was embedded in the pavement [2]. This was a huge advance in the field, but the car had the big limitation that it required the electrical circuit to drive autonomously.

In 1980, a Mercedes Benz van modified by Ernst Dickmanns and his team in the University of Munich was able to reach top speeds of 100 km/h in an empty highway [4]. However, it could only drive in one lane until in the 1990's they managed to make it change lanes autonomously. The European Commission found this study very promising and made an 800 million Euros investment in the project EUREKA Prometheus with the aim of developing an autonomous vehicle.

Another very impactful event was the the DARPA challenge organized by the U.S Defense Advanced Research Projects Administration [4], which first took place in 2004 and continued for the following years where major leaps in the field were made. The first challenge was held in the desert and the contestants had to reach the finish line with autonomous vehicles within a specific time limit. The result of this event was unfortunate for the participants, as none of them were able to make their autonomous vehicles to reach the finish line.

Although the next challenge presented more obstacles and turns, five out of the twenty-three contenders were successful and reached the finish line, showing a big improvement with respect to the last event. Lastly, in 2007 the Grand Challenge III took place in a prepared urban environment where 40% of the participants made it to the finish line within the 6-hour limit.

Another more recent milestone in the field (that is already available to customers) has been the Tesla Autopilot, which hardware was introduced in September 2014 and later complemented with it's software in October 2015 with a simple update. This is a feature of the car that allows for semi-autonomous driving, but that still is far from being fully autonomous.

Up to this date, the Waymo vehicles [Figure 1.2] are the closest to being fully autonomous, being able to provide testing services as taxis in the city of Phoenix in Arizona [3].

Figure 1.2: Waymo self-driving car.

While these technological advances are very big and important, the current state of the art in the field of autonomous driving is far from perfect in all the aspects.

In order to describe what the word "perfect" would mean in terms of effective autonomous driving, it is necessary to define some distinction in the levels of automation that can exist [7]:

- **Level 0** : As the starting point, this level represents no automation.
- **Level 1** : The system provides some driving assistance by performing small steering, braking or acceleration tasks, but never more than one of these tasks

simultaneously. One example of this would be the adaptive cruise control, in which the system can regulate its velocity according to the velocity of other vehicles. In this level of automation, the driver still is the main responsible for driving.

- **Level 2** : The system can combine 2 or more of the small tasks mentioned before that allow to perform very easy maneuvers such as following the trajectory of the lane. For slightly more complicated maneuvers, it requires the driver's intervention. Thus, the driver has to be aware of their environment all the time.

- **Level 3** : There is a conditional automation provided by the system. This means that the system is able to drive autonomously under specific conditions, but requires the driver to take over in other situations for which the system has not been programmed. Thus, the vehicle still requires the driver's supervision all the time.

- **Level 4** : At this level, there is a high automation, so the driver doesn't need to to intervene almost at any time. However, it's still not able to drive through areas that don't figure in the system's database and severe weather may interfere with the sensors. Then, the system is still not considered to be fully autonomous.

- **Level 5** : At this level, the vehicle would be fully autonomous under any conditions.

At this point in history, the most that we have reached has been level 4. This achievement was reached by Waymo, a company that was originated as a project of

Google responsible for the widely known Google self-driving cars.

It's important to note that this classification is only accounting for the ability of the vehicle to drive autonomously, but doesn't consider other very important aspects, such as the resilience against cyber-attacks. As it is known from experience, with the implementation of technology in every aspect of society thus far, has come with its consequent safety issues that the designers of the system have to account for.

Some predictions say that in 2030 autonomous vehicles will be commercially available [7], so it is crucial to improve the current state of the art techniques in cybersecurity in order to provide autonomous vehicles that are completely resilient and safe.

1.1 Motivation

With the enormous leaps that the automation of processes and the internet of things are having, cyber-physical systems become more efficient and reliable, but also more vulnerable to cyber attacks.

Cyber-physical systems are composed of physical processes that strongly rely on cyber components. As shown in Figure 1.3, the components in a cyber-physical system rely on a network to synchronize and communicate all the different components of the system [10]. Thus, cyber-attacks in the cyber domain can have physical consequences.

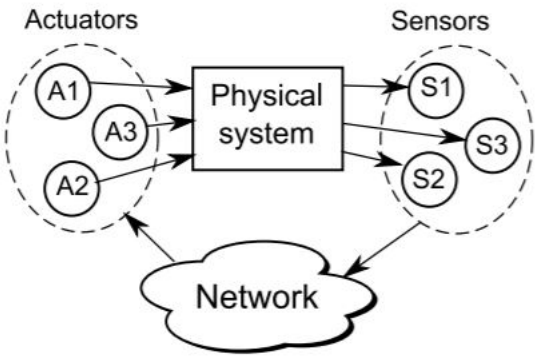


Figure 1.3: Communication of the different components in a cyber-physical system.

For this reason it is important to develop methods and techniques to make the system resilient to cyber attacks.

This project will initiate the design and implementation process of an autonomous golf cart which will be used by researchers in the University of Maryland to run tests and perform attacks on it with the ultimate goal of developing new techniques and improving existing ones in order to improve safety in autonomous driving and other cyber-physical systems.

Chapter 2: Methodology

An autonomous car is a cyber-physical system because it combines its physical components (motors, actuators, transmission systems, etc.) with software components, that serve as the brain of the whole system by connecting and coordinating all its elements. As such, the thesis at hand will divide the project into these two fields, and each field will be broken down into different modules with the aim of analyzing each of them independently.

In the physical aspect of the autonomous driving system the scope of this thesis will reach the following modules:

- **Steering system:** In order to give the car the ability to steer by itself, it will be a requirement to be able to impose motion with the electronic signals provided by the software. In order to achieve this, a motor and some mechanism to attach it to the steering column will be needed.
- **Brakes:** The way to impose motion for the brake will be by using a servomotor. Then with the help of a brake cable attached to the motor it will be possible transform its rotational motion into the linear pulling motion that the pedal achieves.
- **Accelerator:** In order to actuate on the throttle, the electrical system will be modified. Thus, the instructions to transmit the appropriate amount of power to the motor can be given by software.

The golf cart that will be used is electric with a 48 volts DC motor and a power of 2.5

kW. The aim will be to use the six 8-volt batteries of the golf cart to provide enough power to the all the components of the system. In case it can't provide enough power to keep the system functional under any load conditions, additional batteries can be connected in series with the current batteries or be used independently (having 2 sets of batteries to run the car and the components separately).

In terms of software this thesis will create a computer vision algorithm that will rely on cameras mounted around the golf cart to detect elements of interest on the road. This algorithm will be programmed in python due to the great number of libraries for machine learning that it has. In particular, keras with a tensorflow backend will be used to create and train the convolutional neural network.

Finally, the computational power required to train the neural network will be obtained by using a virtual machine from the Google Cloud Platform.

The remaining modules necessary to build the autonomous golf cart will be addressed in future work by other students of the University of Maryland.

Chapter 3: Theoretical framework

This section will introduce all the required theoretical concepts that will be used in further sections. As the project consists of a hardware part (in where theory is needed to understand the design of the gears) and software part (for the computer vision algorithm), the theoretical framework will be divided accordingly.

3.1 Design of the gears

First, it is necessary to introduce some definitions of the elements of he gears. Figure 3.1 shows the main elements of a gear.

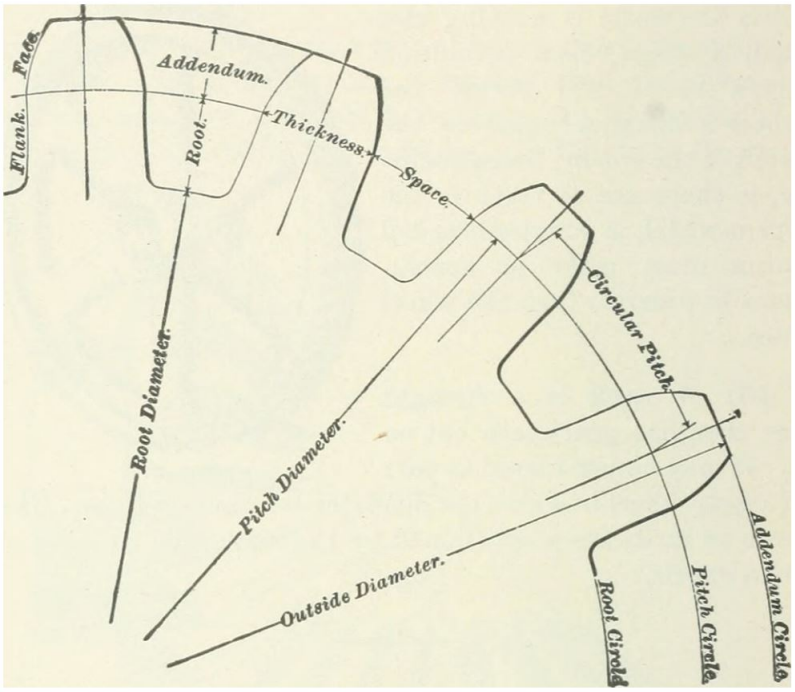


Figure 3.1: Elements of a gear.

From this figure some important definitions must be introduced:

- **Module (m):** It is defined as the reference diameter divided by the number of teeth ($m = \frac{D}{z}$). Thus, it gives a reference of how big the gear is with respect to the number of teeth.

- **Pressure angle (ϕ):** It is the angle that is formed between the shared tangent line of the two pitch circles and the line of action of the force that one gear exerts on the other.

- **Gear ratio (G):** It is defined as the relation between the input velocity and the output velocity of the gear train. Due to the properties of the transmission it can also be defined as: $G = \dfrac{w_{in}}{w_{out}} = \dfrac{R_{out}}{R_{in}} = \dfrac{T_{out}}{T_{in}} = \dfrac{z_{out}}{z_{in}}$.

Then, the number of teeth for both of the gears must be determined in order to have the desired gear ratio and thus, the desired torque and velocity. For this, it is necessary to be aware of the interference that could occur if the number of teeth is not sufficient. This phenomenon can produce abnormal wearing of the teeth of the gears, produce undesired noise and vibrations and even could cause the gear train to not work at all. Therefore, in order to determine the minimum number of teeth for the wheel gear the following equation can be used:

$$z_{w_{min}} = \frac{2\,A_w}{\sqrt{1 + \frac{1}{G}(\frac{1}{G} + 2)\sin^2\phi} - 1} \tag{3.1}$$

Where A_w is the factor by which the module should be multiplied to obtain the addendum of the gear and ϕ is the pressure angle.

Finally, the gears have to be designed to withstand the mechanical requirements that the task will demand. For this, the bending stress equations established by the

American Gear Manufacturers Association (AGMA) will be used in order to relate the width of the gears (known as the face width) to a particular material in order to fulfill the specifications.

The AGMA equations are:

$$\left.\begin{aligned} \sigma &= W^t K_0 K_v K_s \frac{K_H K_B}{b\, m_t\, Y} \\ \sigma_{all} &= \frac{\sigma_{FP}\, Y_N}{S_F Y_0 Y_z} \end{aligned}\right\} \tag{3.2}$$

The purpose of the first equation is to compute the bending stress produced by a certain task given a set of gears. The second equation relates the AGMA bending stress (σ_{FP}) with the allowable bending stress for the application given specifications of temperature, reliability and number of cycles with a specified security factor. Therefore, by setting the bending stress of the first equation equal to the the allowable bending stress of the second equation (σ_{all}) one can determine the requirements of the material (σ_{FP}).

All the physical phenomena that has been taken into consideration through the different terms of the equations is explained below:

- **Tangential force (W^t):** It is the force that one gear produces on the other in the tangential direction of the pitch circle.
- **Overload factor (K_0):** It is based on experience for particular applications and accounts for variations in torque from the mean value.

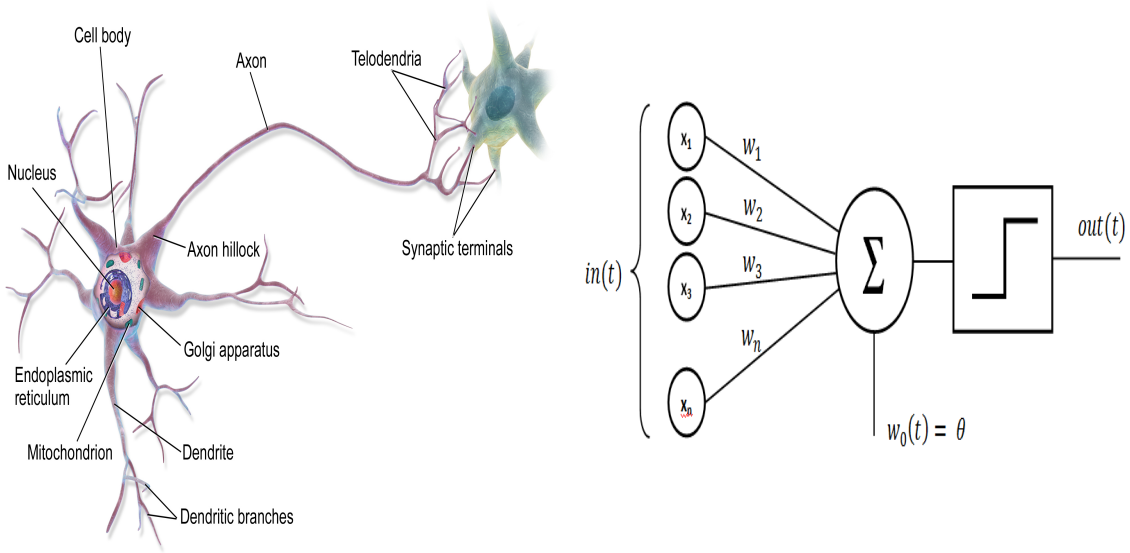
- **Dynamic factor (K_v):** It accounts for the fact that the operation of the gears is not static and also for the manufacturing quality of the gears.
- **Size factor (K_s):** Reflects the heterogeneity of the material's properties due to size.
- **Load distribution factor (K_H):** This factor considers the misalignment of the gears.
- **Rim factor (K_B):** If the gear has rim, this factor allows for the support that it is able to provide to the teeth.
- **Face width (b):** It is the width of the gear.
- **Tangential module (m_t):** It considers different orientations of the teeth by projecting the model to the case of spur gears.
- **Geometry factor (Y):** It considers the shape of the teeth.
- **Bending stress cycle factor (Y_N):** Accounts for the number of cycles that the gears must withstand.
- **Reliability factor (Y_z):** Considers the consistency in which the gears must perform optimally.
- **Temperature factor (Y_0):** Accounts for the effect of the temperature on the lifespan of the gears.
- **Safety factor (S_F):** It determines how stronger the material is with respect to what the intended load is. Thus, it is an indicator of how safe the system is.

3.2 Computer vision with deep learning

For the task of computer vision, a deep convolutional neural network will be used along with the YOLO algorithm. In the following sections the basic theory needed to understand its implementation and training will be introduced.

3.2.1 Concept of neural network

Neural networks are based on how brains work. Brains contain a great number of neurons that are connected to each other and transmit nerve impulses to each other with the final goal to produce an output on the body. Each of the neurons receives the outputs of other neurons through its dendrites, and if the overall signal reaches a certain threshold the neuron fires and transmits an output to the next neurons.



(a) Neuron. (b) Perceptron.

Figure 3.2: Comparison between neuron and perceptron.

Figure 3.2b shows a model called perceptron, which aims to replicate this behavior of the neurons (Figure 3.2a). Note that the dendrites are modeled with connections

where each x_i is the output of a previous perceptron and w_i are the weights that are given to each of the outputs. Then, the perceptron combines all the input signals with a weighted sum.

Note that at this point, the weighted sum would output the function $\hat{y} = \sum w_i\, x_i$, which passes through the origin of the coordinate system input (x_i) vs output ($\hat{y}$). Therefore, the model adds a term called bias (θ), to give the output that extra degree of freedom in the input vs output plane.

Finally, the model of the perceptron adds a non-linearity called activation function, with the purpose of providing the model with the ability of producing non-linear outputs. Otherwise, the combination of many linear function would lead to another linear function, which doesn't add complexity and limits the model to the linear domain.

By adding more neurons in layers and organizing the parameters in tensors, it is possible to create artificial neural networks with a certain depth and width. As an example, a neural network with depth 2 would produce an output like the one shown in Equation 3.3, where f_i are the activation functions of each layer.

$$\hat{y} = f_2(w_2^T f_1(w_1^T\, x_i + \theta_1) + \theta_2) \tag{3.3}$$

Therefore, given an input x_i and a desired output y neural networks can act as function approximator, which by changing their parameters can produce an output $\hat{y}$ similar to y.

In the task of detecting objects from images, the neural network must be able to take into consideration the spatial distribution of the input. This means that given

an object in an image, the neural network must be able to detect it and classify it regardless the position, orientation and scaling of the object.

In order to consider the spatial distribution of images, convolutional neural networks use filters where the parameters have a spatial layout that extracts the underlying features of images by using the convolution operation.

Therefore, in the same way as fully connected layers take an input and transform it with a weighted sum, convolutional neural networks take an input and transform it into a subspace by using the convolution operation.

3.2.2 Training a neural network

In this project, the supervised learning paradigm will be used for training. This means that the way to make the neural network learn will be by providing a ground truth annotation/label to every input image. Therefore, given an input image (x), the neural network will make a prediction ($\hat{y}$) that will try to resemble the ground truth label (y).

In order to determine how close the output ($\hat{y}$) is from the label (y), an objective/loss function has to be defined. This function must be convex and must measure a distance between $\hat{y}$ and y , i.e., the cost from $\hat{y}$ to y is the same as the cost from y to $\hat{y}$.

Therefore, the training process can be conceived as an optimization problem in which the weights and biases of the neural network have to be modified for it to make predictions that resemble as much as possible the labels, i.e., the loss is minimum.

For the purpose of optimizing the parameters of the network, one can utilize many

optimizer algorithms. As an example, gradient descent will be explained as it's the base for the rest of optimization algorithms. For this project, the ADAM optimizer will be used, which has the same working principle as gradient descent with improvements that make it better.

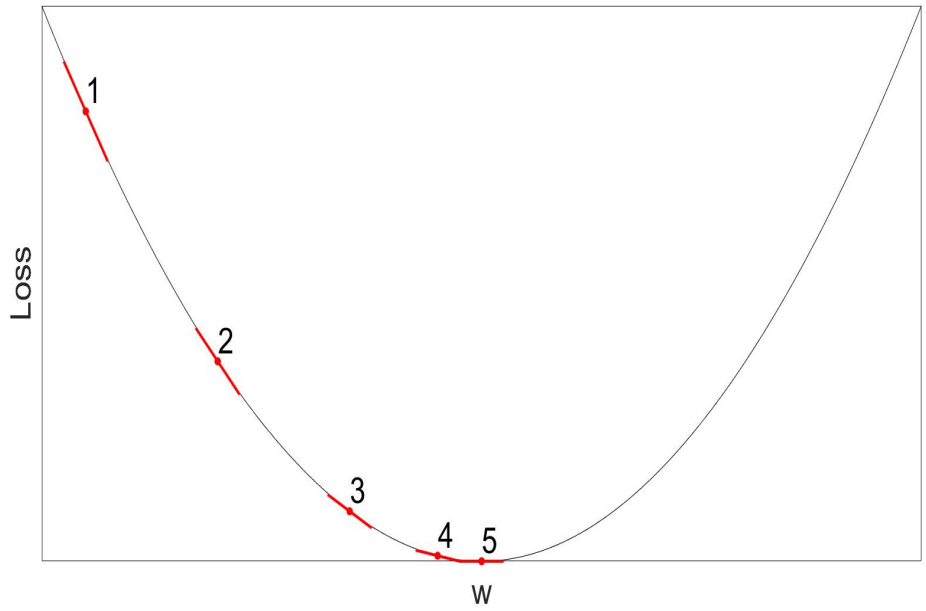


Figure 3.3: Gradient descent.

In the loss function shown in 3.3, the initial weights of the neural network lead to the loss value represented in point 1. Gradient descent consists in taking steps towards the direction in which the slope is negative to reach a minimum of the function. The size of the steps is determined by a hyperparameter called learning rate (α). Every time a step is taken, the weights of the network are updated and a new value for the loss is achieved. In Figure 3.3, points 2, 3, 4 and 5 represent consequent steps that lead to the minimum of the function, which in this case is global.

This algorithm becomes unfeasible when dealing with large datasets, because would require a lot of RAM to work with it all at once. Therefore, Stochastic Gradient De-

scent is often used, which consists in taking small batches of the dataset to compute the gradients. With this, the direction of the steps given towards the minimum is also a little stochastic in the sense that they have little deviations from the optimum direction. Thus, the smaller the batch size, the faster the training iterations, but also the more stochastic the steps are.

3.2.3 YOLO algorithm

The YOLO (You Only Look Once) algorithm aims to perform object detection on images by feeding the entire image into the convolutional neural network only once.

This algorithm consists in taking the input images and dividing them into grids. Then, the labels are put into a tensor of size (SxSxAx(5+C)), where S is the number of grid cells, A is the number of anchor boxes and C is the number of classes to predict. Figure 3.4 shows an example of how an image containing a cat would be labeled if the object detector was trying to detect cats and dogs.

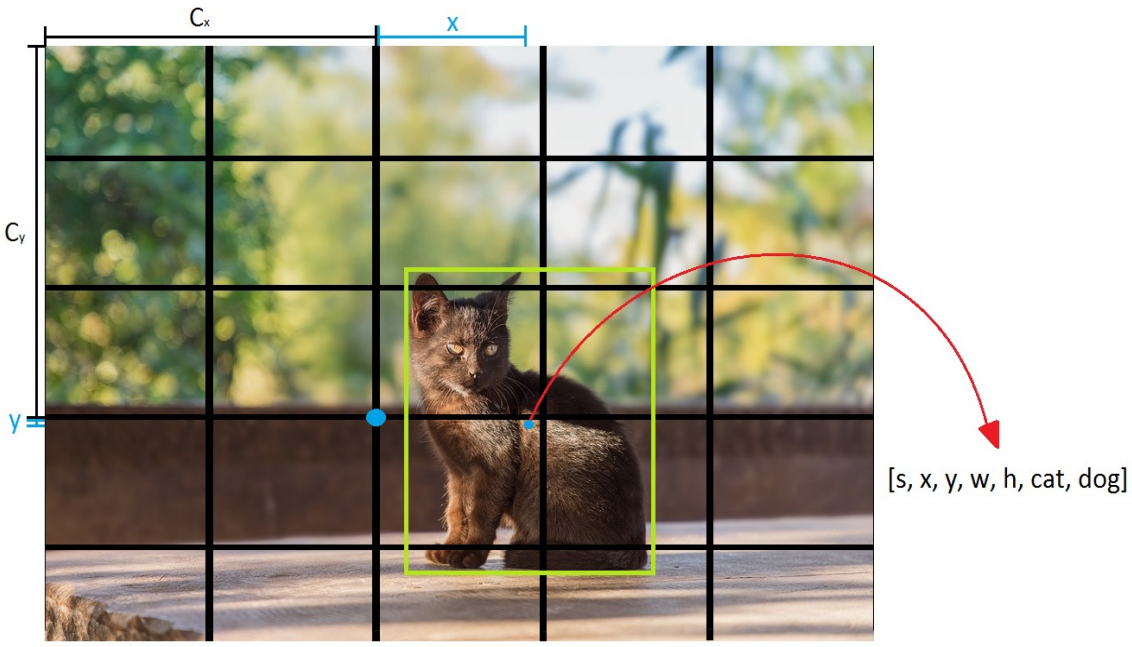


Figure 3.4: Example of labeled image in the YOLO format.

In this example, each cell in the image would contain the information of a score that represents the presence of an object (s), the relative position of the center of the box that contains the object with respect to the cell (x, y), the width and height of the box (w, h) and a one hot vector determining the class that is contained in the box. The values of x and w are normalized by dividing by the width of the cell in pixels, whereas y and h are normalized with the height of the cell in pixels. Thus, the values of x and y are always between 0 and 1 and the values of w and h can be greater than 1.

Therefore, the label for the cell that contains the center of the bounding box shown in Figure 3.4 would be [1, 0.95, 0.05, 1.5, 2.2, 1, 0] and the rest of the cells would contain zeros.

Finally, the neural network would have to output a tensor of the same shape as the labels that will be used for training. Then, to train the neural network it is necessary to define a loss function that will determine how similar is the output to the labels. This loss function [8] is defined in Equation 3.4 and will be the one to optimize during training.

$$
\begin{aligned}
Loss = {} & \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{obj} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right] \\
& + \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{obj} \left[\left(\sqrt{w_i} - \sqrt{\hat{w}_i} \right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i} \right)^2 \right] \\
& + \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{obj} \left(C_i - \hat{C}_i \right) + \lambda_{noobj} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{noobj} \left(C_i - \hat{C}_i \right) \\
& + \sum_{i=0}^{S^2} \mathbb{1}_{i}^{obj} \sum_{c \in classes} \left(p_i(c) - \hat{p}_i(c) \right)^2
\end{aligned}
\tag{3.4}
$$

Chapter 4: Design and implementation

4.1 Steering system

For the autonomous car to be able to actuate on the steering wheel, a motor needs to be mounted in such a way in which it is able to transmit the rotation motion in the axis of the steering shaft. In this section, all the considered configurations for this attachment are explained and discussed with the objective of selecting the most appropriate and efficient mechanism possible.

Figure 4.1: The steering wheel of the golf car.

The steering shaft of the golf cart is shown in Figure 4.1. As shown in the figure, the cover of the steering shaft and the steering wheel are accessible, which allows for the consideration of several configurations for the coupling of the motor and the steering system. The steering shaft forms 45º with respect to the floor of the car and measures about 50 cm long and 42.5 mm of diameter.

In order to explain and analyze the available options a model of the golf cart has been created in Autodesk Inventor. Only the components that are relevant to determine

the configuration of the components to be installed has been represented in the model. This model is shown in Figure 4.2.

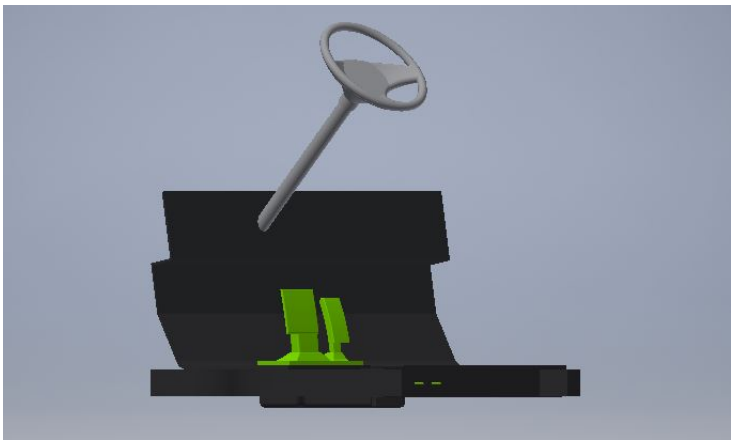


Figure 4.2: 3D model of the golf cart.

4.1.1 First option

The first configuration that was considered was to directly attach the motor to the intermediate steering shaft (Figure 4.3). This way the motor is hidden from the users leaving more space for the passengers, as the steering column along with the driving wheel would be eliminated.

The proposed initial attachment of the motor to the steering shaft was by substituting the original attachment made with part number 7, achieving a direct transmission of torque from the motor to the rack and pinion system though the same axis. This configuration was quickly disregarded as it had a very hard installation process due to the inaccessibility of the steering shaft and the lack of space for the motor.

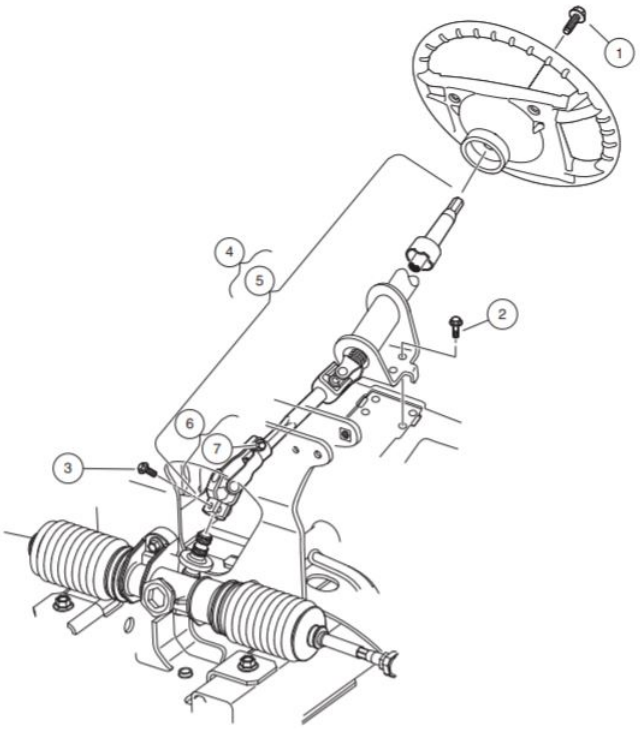


Figure 4.3: The original configuration of the steering shaft.

4.1.2 Second option

The next configuration that was considered aimed to solve the issues presented in the previous configuration. This one consisted in cutting the steering column and attaching it to the motor by using an axial coupling. Then the motor would be mounted on the dashboard of the cart and thus the space that the previous configuration provided would be maintained while having the motor in an accessible position for maintenance and installation.

However, in this second configuration new installation problems appear. Cutting the shaft of the motor is a destructive process that if performed incorrectly could compromise the usability of the golf cart. Moreover, the cutting process would need to be performed very precisely and at different points for the cover of the shaft

and the shaft so that there's access for installing the coupling. For this it would be necessary to disassemble the steering column to cut it, as the cutting points wouldn't be directly accessible if the motor is desired to be mounted on the dashboard.

Finally, and also for the reason that the motor would be mounted on the dashboard, the installation of the coupling would also be a hard task as it would be placed inside the dashboard, which is not accessible without making any access holes on the plastic. The placement of the motor for this configuration is shown in Figure 4.4.

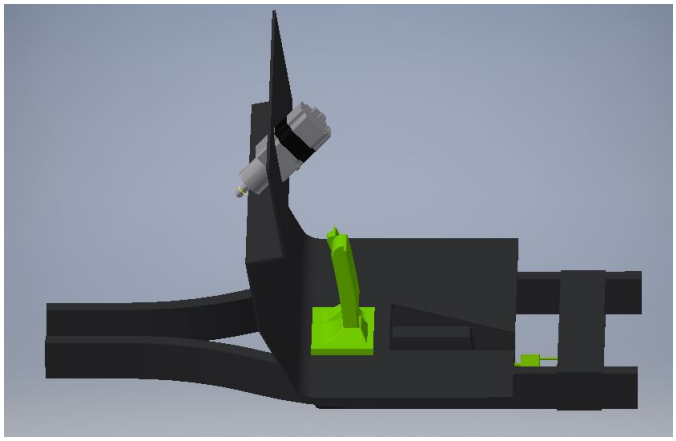


Figure 4.4: Placement of the motor for the second configuration.

This configuration still can be done, but more possible configurations were explored for the sake of reducing costs, potential installation problems and efforts.

4.1.3 Third option

The last configuration is proposed in Figure 4.5. This last configuration consists in placing the motor parallel to the steering shaft and then making use of spur gears to transfer the power from the motor to the steering wheel. This has the advantage that cutting the steering shaft or disassembling major parts is no longer needed.

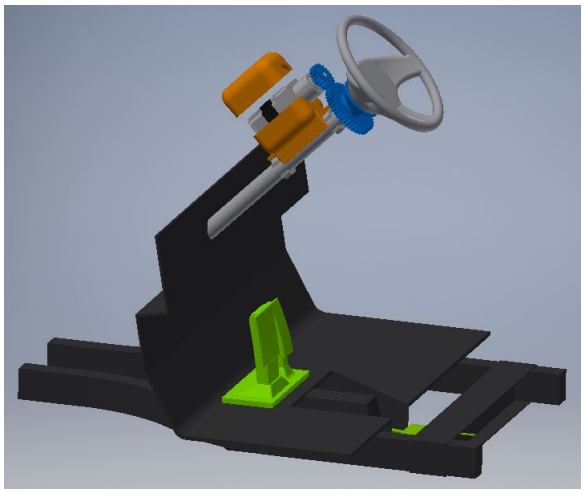


Figure 4.5: Placement of the motor for the final configuration.

In addition, the motor rests in an accessible position where the problem of space is no longer present and the maintenance of the motor and its electronic connections can be made easily.

Moreover, it is possible to take advantage of the spur gears by using a gear ratio (G_{spur_gears}) greater than 1 so that the transferred torque is increased by that factor and the velocity is decreased also by that factor. Servomotors output high speeds at a relatively low torque for this task, so a gearbox will be needed to meet the specifications. Therefore, the spur gears will require a gearbox with G_{spur_gears} times less gear ratio which will reduce the cost of the gearbox, whereas both of the previous proposed configurations relied only on one gearbox to increase the torque to meet the specifications.

Finally, this setup also allows switching from autonomous driving mode to manual driving mode and vice versa, as it preserves the steering wheel.

4.1.4 Final selection

All things considered, due to all the advantages that the last setup presented over the others, it's the one that has been selected for the steering system. Now it's necessary to determine the specifications that the task will demand to select the motor and design the gears.

In particular, steering the driving wheel will demand an input torque and a velocity at which the torque has to be applied. The most unfavorable combination of the two will establish the power of the motor to be used.

The measurements of the torque were made by attaching a hand scale to the rim of the steering wheel and pulling in the tangent direction. Then, the force obtained was multiplied by the radius of the steering wheel to obtain the torque. The radius of the steering wheel is 15 cm and the force measured with the scale ranged from 4 to 13 kg depending on whether the measurements were made close to the neutral position of the steering wheel or close to the most right and left positions and whether the cart was resting on a smooth or a rough floor. However, the force required doesn't surpass the 8 kg in about 90% of the range of motion, reaching 13 kg only at both ends. This corresponds to a maximum torque of 19.3 Nm but the maximum for 90% of the range of motion the maximum torque is 11.8 Nm.

For the velocity at which the steering needs to be done, the requirement that the steering wheel must be able to go from the neutral position until the most right or left positions in at least 1 second has been set. Considering that from the neutral position until the most right or left positions the steering wheel has to make 1.75 revolutions, the speed at which the turn has to be made is 105 rpm.

Consequently, the motor must have at least a power of 210.34 W. Now, considering that it is desirable that the motor is powered with batteries of the car and that they output 48 VDC, it is necessary to select a DC servo-motor that can be operated also at 48 VDC. The motor will also need to have an encoder in order to make it possible to control it. In particular, it is preferable that the encoder is an absolute encoder so that the autonomous cart can determine its initial steering angle every time it's turned on.

The motor that has been selected to satisfy these requirements has been the motor "PD6-CB87S048030-E-09" from `https://us.nanotec.com`, which has a rated torque of 0.7 Nm, a peak torque of 2.1 Nm and a rated speed of 3000 rpm, which corresponds to a rated power of 220 W. In addition, it has the advantage of having an integrated controller.

The total transmission ratio that will be needed to reach both of the specifications of 11.8 Nm and 105 rpm must be between 16.86 and 28.57, which with the maximum torque of 2.1 that the motor can output would be sufficient to reach the required torque of 19.3 Nm. The final transmission ratio of the gearbox to purchase will be as low as possible within that range in order to reduce its cost and will depend on the gear ratio of the gears that will connect the motor to the steering wheel.

As discussed in the next subsection, the transmission ratio for the gears will be 2.2 so the gearbox to purchase will need at least a ratio of 7.66. The selected gearbox for this matter is the "GPLE60-1S-8-F87" also from `https://us.nanotec.com` which has a gear ratio of 8. Therefore, the final transmission ratio of the combination of the gearbox and the gears will be 17.6.

4.1.4.1 Design of the gears

First, the distance between the axis of the motor and axis of the steering shaft has to be established. Considering that the diameter of the cover of the steering column is 42.5 mm and that the distance from the axis of the motor to one of its sides is 43 mm, the axis can't be separated by less than 64.25 mm. Considering that for the motor to be mounted on the cover of the steering shaft a 35.75 mm tall base will be needed, it can be stated that the final separation between both axes will be 100 mm (Figure 4.6).

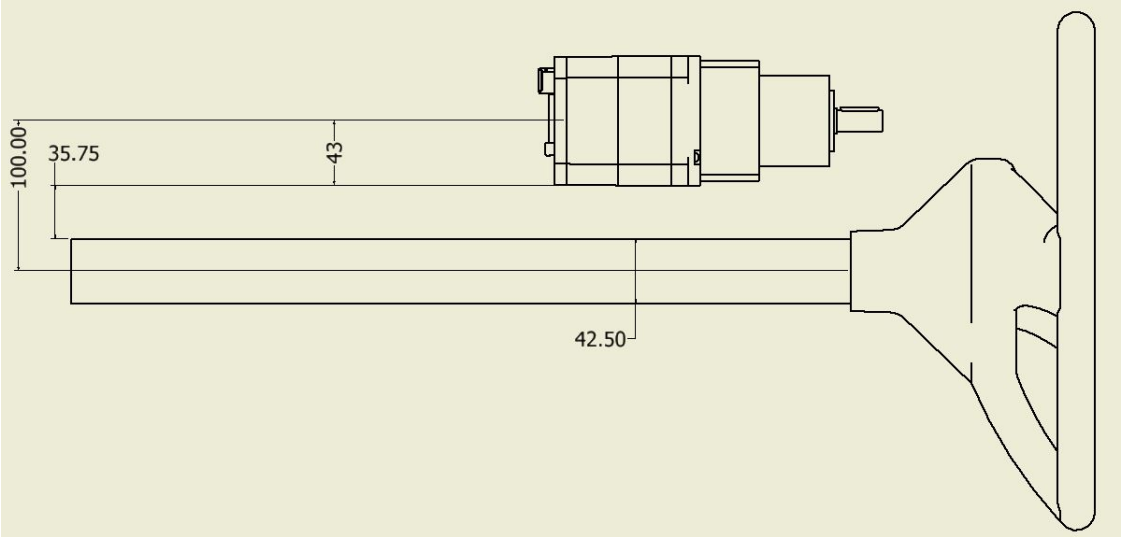


Figure 4.6: Distances considered for the design of the gears and the coupling of the motor.

As stated in previous sections, it is desired that the gear ratio is greater than 1 in order to increase the torque and reduce the speed. This is achieved if of the 2, the input gear is the small one (pinion) and the output gear is the big one (wheel).

Let the gear ratio be 2.2. Then:

$$\left.\begin{array}{ll} R_p + R_w & = 100 \\ \dfrac{R_w}{R_p} & = 2.2 \end{array}\right\} \iff \begin{cases} R_p = 31.25mm \\ \\ R_w = 68.75mm \end{cases} \tag{4.1}$$

In order to determine the number of teeth for each of the spur gears, the formula for the minimum number of teeth ($z_{w_{min}}$) is used for the wheel. Assuming $A_w = 1$ and a pressure angle (ϕ) of 20 degrees:

$$z_{w_{min}} = \frac{2\,A_w}{\sqrt{1 + \frac{1}{G}(\frac{1}{G} + 2)\sin^2\phi - 1}} = \frac{2 \cdot 1}{\sqrt{1 + \frac{1}{2.2}(\frac{1}{2.2} + 2)\sin^2 20 - 1}} = 31.61 \tag{4.2}$$

Next, to calculate the number of teeth for the pinion, $z_{w_{min}}$ has to be divided by the transmission ratio. As an integer value for z_p is desired and the next divisible value by 2.2 is 33, which is very close to 31.61, the next divisible value will be chosen, which is 44. Therefore, the final number of teeth for both of the spur gears will be:

$$\left.\begin{array}{ll} z_w & = 44 \\ \dfrac{z_w}{z_p} & = 2.2 \end{array}\right\} \iff \begin{cases} z_w = 44 \\ \\ z_p = 20 \end{cases} \tag{4.3}$$

Now it's only left to determine the face width (b) of the gears. For this the AGMA bending equations will be used (Equations 3.2).

To do so, all the different factors have to be computed of both of the equations and the bending stress that the task will produce given a face width b. Then, it will be ensured that the allowed stress given by the chosen material is greater than the computed stress for the task.

Geometry factor (Y)

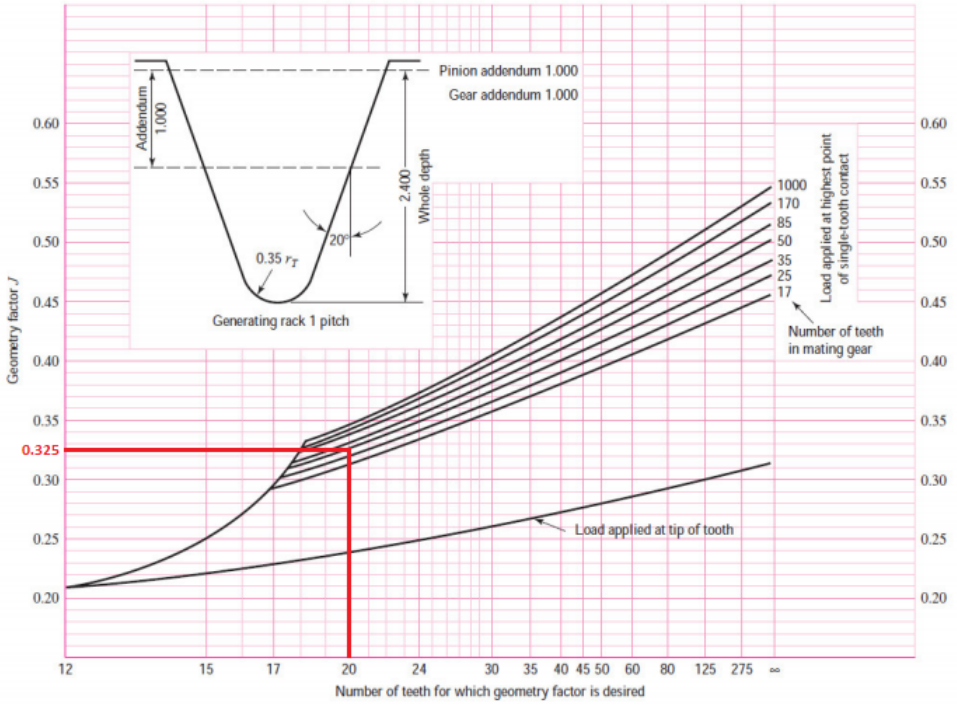


Figure 4.7: Graph of the geometry factor for spur gears with a pressure angle of 20º.

From the graph of the geometry factor for spur gears with a pressure angle of 20º the geometry factor is: Y = 0.325.

Tangential force (W^t)

For the tangential force, first calculate the tangential velocity has to be calculated and then divide the power by the tangential velocity as shown in Equation 4.4.

$$\left.\begin{aligned} W^t V^t &= Power \\ V^t &= \frac{2\pi w_w^t \, R_w}{60} \end{aligned}\right\} \iff \begin{cases} W^t = \dfrac{Power}{V^t} = \dfrac{220}{0.76} = 291N \\ V^t = \dfrac{2\,\pi\,105 \cdot 68.75 \cdot 10^{-3}}{60} = 0.76\ m/s \end{cases} \tag{4.4}$$

Note that the force has been calculated with the values of the wheel but using the values for the pinion leads to the same result: $W^t = 291\,N$.

Tangential module (m_t)

Multiplying the module by the cosine of the angle that the teeth form with the axis of the gears, the tangential module is obtained:

$$m_t = 3.125\ \cos(0) = 3.125\ mm \tag{4.5}$$

In this case, $\alpha = 0$ because they are spur gears.

Overload factor (K_0)

	Load on driven machine		
Source of power	**Uniform**	**Moderate shock**	**Heavy shock**
Uniform	1.00	1.25	1.75
Light shock	1.25	1.50	2.00
Medium shock	1.50	1.75	2.25

Figure 4.8: Table for the overload factor.

As there are no shocks in this application, the assumption will be that the transmitted power is uniform at worst conditions. Therefore, $K_0 = 1$.

Dynamic factor (K_v)

$$\left.\begin{array}{l} B = 0.25(12 - Q_v)^{2/3} \\ A = 50 + 56(1 - B) \\ K_v = \left(\frac{A+\sqrt{200V^t}}{A}\right)^B \end{array}\right\} \Longleftrightarrow \left\{\begin{array}{l} B = 0.25\,(12-5)^{2/3} = 0.9148 \\ A = 50 + 56\,(1 - 0.9148) = 54.7697 \\ K_v = \left(\dfrac{54.7697 + \sqrt{200\,0.76}}{54.7697}\right)^{0.9148} = 1.2 \end{array}\right. \tag{4.6}$$

As shown in equation 4.6, the AGMA quality factor Q_v has been considered to be 5 because it is the worst case scenario. As the way the gears will be manufactured is still unknown this is a safe consideration.

Size factor (K_s)

This is considered an optional factor by the AGMA association. Therefore: $K_s = 1$.

Rim factor (K_B)

As the gears won't have rims this factor won't be taken into consideration. Therefore: $K_B = 1$.

Load distribution factor (K_H)

The formula to compute this factor is the following:

$$K_H = 1 + C_{mc}\,(C_{pf}\,C_{pm} + C_{ma}\,C_e) \tag{4.7}$$

C_{mc} can be 1 or 0.8 depending on weather the gears have uncrowned teeth or crowned teeth respectively. In this case, the gears will be manufactured with uncrowned teeth, so $C_{mc} = 1$.

For the factor C_{pf} equation 4.8 is used, where $\frac{b}{10d} = 0.05$ if $\frac{b}{10d} < 0.05$.

$$C_{pf} = \begin{Bmatrix} \dfrac{b}{10d} - 0.025 & b \leq 25mm \\ \dfrac{b}{10d} - 0.0375 + (4.92 \cdot 10^{-4})\,b & 25 < b \leq 425mm \\ \dfrac{b}{10d} - 0.1109 + (8.15 \cdot 10^{-4})\,b - (3.53 \cdot 10^{-7})\,b^2 & 425 < b \leq 1000mm \end{Bmatrix} \tag{4.8}$$

As the shaft of the gearbox where the pinion will be attached has a keyway modification that is 25 mm long, b won't be considered to be greater than 25 mm. Therefore, as $\dfrac{b}{10 \cdot 2 \cdot 31.25} < 0.05$ for all $b \leq 25mm$:

$$C_{pf} = 0.05 - 0.025 = 0.025 \tag{4.9}$$

The C_{pm} factor can be 1 for straddle-mounted pinion with $S_1/S < 0.175$ or 1.1 for straddle-mounted pinion with $S_1/S \geq 0.175$. The values of S and S_1 are shown in Figure 4.9. As in this case the pinion won't be supported on one of the sides: $C_{pm} = 1.1$.

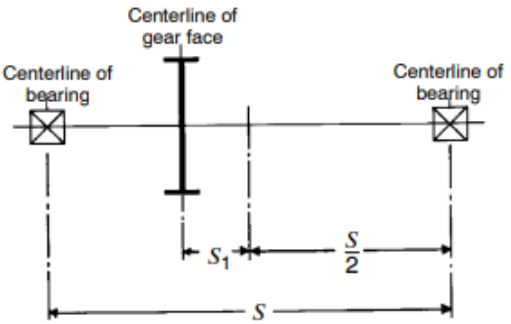


Figure 4.9: Dimensions of S and S_1 [1].

Next, with Equation 4.10, C_{ma} will be computed by using the values for A, B and C shown in Figure 4.10.

Curve	A	B	C
Curve 1 Open gearing	2.47×10^{-1}	0.167×10^{-1}	-0.765×10^{-4}
Curve 2 Commercial enclosed gear units	1.27×10^{-1}	0.158×10^{-1}	-1.093×10^{-4}
Curve 3 Precision enclosed gear units	0.675×10^{-1}	0.128×10^{-1}	-0.926×10^{-4}
Curve 4 Extra precision enclosed gear units	0.380×10^{-1}	0.102×10^{-1}	-0.822×10^{-4}

Figure 4.10: Table for the mesh alignment factor [1]. The values are in the imperial system.

$$C_{ma} = A + B\,b + C\,b^2 \tag{4.10}$$

As the values of A, B and C are in the imperial system b will be used in inches for this factor. The gearing will be open and the face width (b) will be 20 mm as a first

iteration. If more face width is needed for the gears to fulfill the requirements this factor will be recomputed. Therefore, the mesh alignment factor is:

$$C_{ma} = 0.247 + 0.0167\left(\frac{20}{25.4}\right) - 0.765 \cdot 10^{-4}\left(\frac{20}{25.4}\right)^2 = 0.26 \tag{4.11}$$

The last factor to compute K_H is C_e, which can have the values of 1 (if there is adjustment at assembly or compatibility is improved by lapping) or 0.8 (for all the other conditions). Considering that there won't be any special mounting, $C_e = 1$.

Finally, the load distribution factor will be:

$$K_H = 1 + 1\,(0.025 \cdot 1.1 + 0.26 \cdot 1) = 1.28 \tag{4.12}$$

The bending stress for this task will then be:

$$\sigma = 291 \cdot 1 \cdot 1.2 \cdot 1\,\frac{1.28 \cdot 1}{20 \cdot 3.125 \cdot 0.325} = 22\,MPa \tag{4.13}$$

On the other hand, the bending stress that the material will need has to be calculated in order to determine the material to use for the gears. Consulting the charts and tables found in the AGMA standards [1] the following coefficients are found:

- **Bending stress cycle factor (Y_N)**: Considering a life of 10^7 cycles, $Y_N = 1$.
- **Reliability factor (Y_z)**: Considering a 99% reliability, $Y_z = 1$.
- **Temperature factor (Y_0)**: As the temperature will always be less than 120º, $Y_0 = 1$.

And lastly, the bending stress that the material has to withstand (σ_{FP}) can be computed with Equation 3.2 by setting the allowable bending stress equal to the computed bending stress. For this, it has been decided to have a safety factor of 2. Therefore:

$$\sigma_{FP} = \frac{\sigma_{all} S_F Y_0 Y_z}{Y_N} = \frac{22 \cdot 2 \cdot 1 \cdot 1}{1} = 44\, MPa \tag{4.14}$$

With this information, the selected material has been Poly-Lactic-Acid plastic, which can be printed in 3D. Thus, the specifications can be met while still being able to easily make shapes that will adapt to the steering shaft and keeping the cost of manufacturing the gears low with respect to other manufacturing techniques.

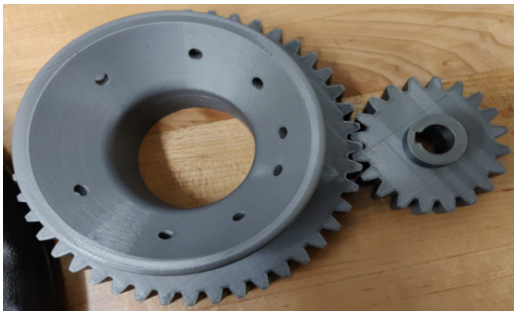
4.1.4.2 Assembly of gears and motor

In order to assemble the gears and the motor new 3D printed parts will be made. In the case of the gears, they will have a special design to fit the steering wheel and the motor shaft.

On the other hand, the motor will rest inside of a 3D printed box that will both protect the motor and clamp it to the cover of the steering shaft.

Starting with the design of the gear that will be mounted on the steering wheel, the objective is to have the gear rest on the flat part of the steering wheel and attached to its conic surface (Figure 4.12a). For this, the gears were printed as shown in Figure 4.12b.

(a) Inner view of the conic surface of the steering wheel.



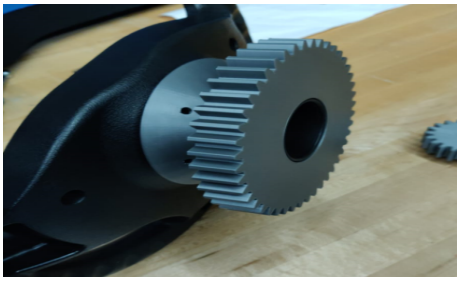
(b) The 3D printed spur gears.

Figure 4.11: View of the gears (left) and the steering wheel (right).

In order to attach both of the elements together, 8 holes will be drilled on the conic surface of the steering wheel, matching the holes of the conic surface of the gear. Then, they will be firmly attached with the help of 8 bolts and 8 nuts.

The measurements for the big gear were made with the help of a caliper for the cylindrical surface of the steering wheel while the angle of the conic surface was measured with image processing with the help of a computer.

(a)



(b)

Figure 4.12: Big gear attached to the steering wheel.

Finally, the result of the attachment of the two elements is shown in Figure 4.12.

On the other hand, the motor will rest in the 3D printed box. Observe that the box will be clamped onto the cover of the steering shaft with the help of two 42 mm clamps that will be fastened to the box with bolts. The final result of the entire assembly is shown in Figure 4.13.

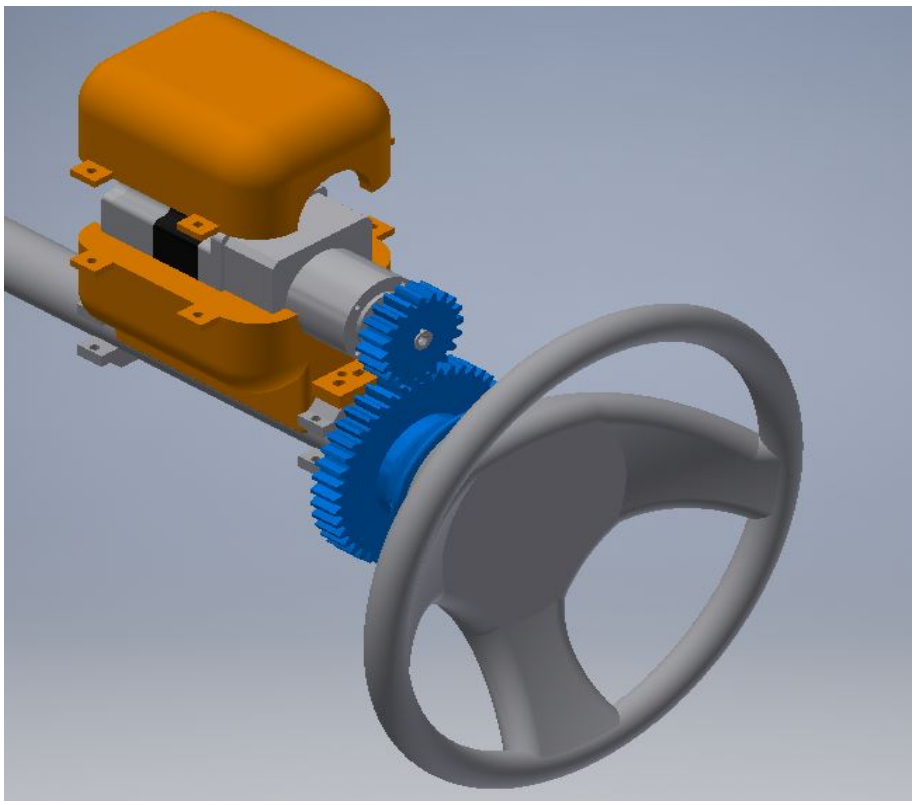


Figure 4.13: Final assembly of the motor and gears.

4.2 Braking system

In the same way that the steering system needs a motor to actuate on the steering wheel, the braking system will also need one to actuate on the brakes. The brakes of the golf cart consist in a metallic rod attached to the pedal and to a bracket that at the same time is attached to the 2 brake cables that actuate on the 2 rare wheels.

By pressing the brake pedal, the metallic road is pulled by means of a class 2 lever,

i.e., the rod (which is where the load is applied) is attached between where the pedal's fulcrum is and the tip of the pedal (where the force is applied). Therefore, in order to replicate the braking process with a motor, it will be necessary to make it pull from the bracket that holds the two other cables. This pull has to be made from the middle of the bracket, as it's purpose is to distribute the braking force between the 2 cables appropriately.

These cables, the metallic rod and the class 2 lever are underneath the floor of the golf cart, inside of a box. This configuration is shown in Figure 4.14.

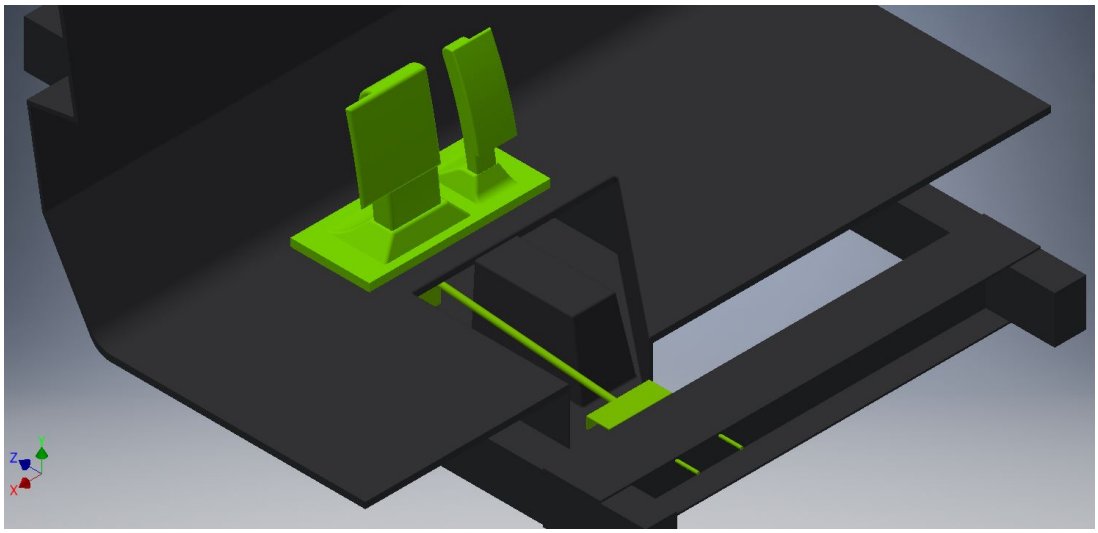


Figure 4.14: Brakes of the golf cart.

Considering that a force of 35 kg applied at the tip of the pedal is needed to brake effectively according to the measurements, the force with which the metallic rod will pull from the bracket can be calculated setting the summation of moments equal to zero. The class 2 lever to consider is shown in Figure 4.15.

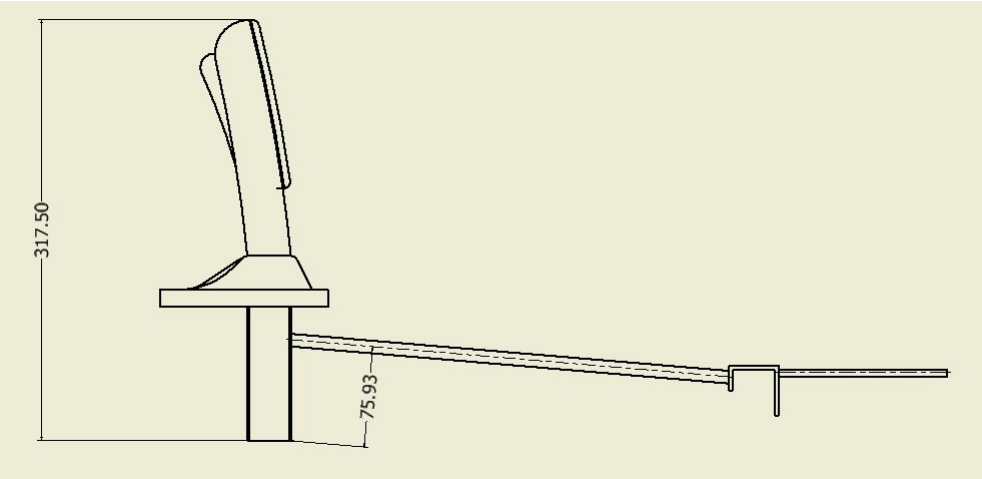


Figure 4.15: Class 2 lever of the brakes of the golf cart. The fulcrum of the lever is where the starting point of the annotations is.

Therefore, the force that the motor will need to pull from the bracket will be:

$$F = \frac{317.5 \cdot 35}{75.93} = 146.4\, kg \tag{4.15}$$

And in order to provide that force, a brake cable will be attached to the motor by means of the 3D printed pulley shown in Figure 4.16.

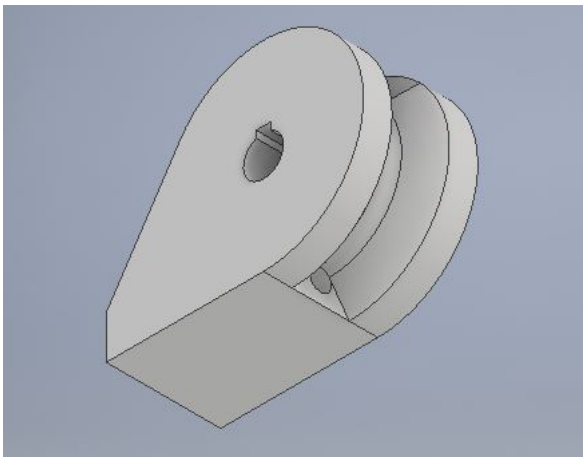


Figure 4.16: 3D printed pulley for the motor of the brakes.

The shown pulley will be attached to the motor. Then its flat part will hold the brake cable, which upon rotation will pull from the bracket, which will be attached

to the other end of the cable. The rail of the pulley where the brake cable will rest is 20 mm away from the motor shaft. Thus, the force that the motor will transmit to the cable will be this distance times its torque.

The selected motor for this task has been the "PD6-CB80M048030-E-09" and it has been purchased from the same company as the motor used for the steering system. In fact, it is of the same lineup, but in this case the motor is able to provide a torque of 1.7 Nm.

Then, the needed gear ratio for the gearbox can be calculated knowing that the torque of the motor times the transmission ratio divided by the distance between the shaft of the motor and the cable has to be equal to a force of 146.4 kg. This calculation is shown in Equation 4.16.

$$G = \frac{F_{brake}\, R_{cable}}{T_{motor}} = \frac{146.4 \cdot 9.81 \cdot 0.02}{1.7} = 16.9 \tag{4.16}$$

Therefore, the selected gearbox is the "GPLE60-2S-20-F87" which has a gear ratio of 20.

Finally, it has to be ensured that the requirement of speed is also fulfilled. To do so, the displacement of the cable to actuate the brakes effectively has been measured. This displacement is of 20 mm, and it is desired that the braking process takes place in half a second. Consequently, the brake cable has to be pulled at 40 mm/s, which corresponds to an angular velocity at the output of the gearbox of 19.1 rpm and 382 rpm at the shaft of the motor. Thus, considering that the rated speed of the motor is 3000 rpm, the specification of the speed is also fulfilled.

As for the placement of the motor, it has to be placed in such a way that the perpen-

dicular plane of the shaft of the gearbox (where the brake cable will be attached) is aligned with the bracket to which the cable will be attached. Therefore, the motor has been placed behind the box that contains the brakes and under the dashboard. The final position of the motor is shown in Figure 4.17. This will be done by welding a metallic board to the lower face of the structural beams. This board will act as a floor that will support the motor and fix it with the help of a 3D printed container.

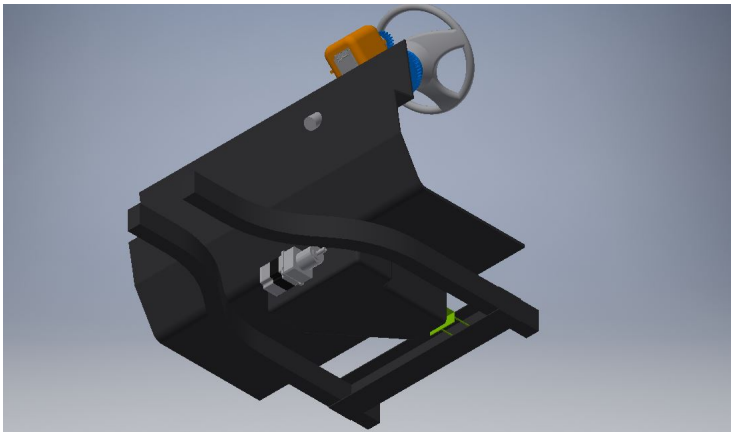


Figure 4.17: Final position of the motor for the brakes.

It is important to note that the setup for the brakes presented in this section doesn't substitute the original braking system. Thus, the possibility of braking by pressing the pedal is still present, which allows switching the driving mode from autonomous to manual and vice versa very easily and also represents an important safety feature when testing the autonomous golf cart, because the passenger can press the brake at any time upon unexpected behavior.

4.3 Accelerator

Unlike the brake pedal, the accelerator is not entirely a mechanical system. The way it works is that when the pedal is pressed, an MCOR (Motor Controller Output

Regulator) is rotated, which acts like a potentiometer that regulates the amount of power that the electric motor of the golf cart will demand.

In order to be able to control the amount of power that is transmitted to the motor, it will suffice to use a PWM (Pulse-Width Modulation) controller instead of the MCOR. This has the advantage that it not only can control the amount of power that wants to be transmitted to the motor but also its polarity, so it is possible to drive the golf cart forward or in reverse.

Note that fully substituting the MCOR by the PWM controller would eliminate the possibility of controlling the golf cart manually. Therefore, both elements must coexist by having a manual switch that will allow the selection of the desired driving mode.

4.4 Object detector

For the golf cart to be autonomous, it must be able to replace all the tasks that a human driver would carry out when driving. Therefore, the cart must sense its environment and manage this information efficiently. For this purpose, the car will have lidars installed that will allow the cart to make a map of its surroundings, and also cameras that will help identify the elements that surround the cart and thus, allow it to make decisions.

In this project, the attention will be put into the computer vision algorithm that will be used to identify and localize different objects that can be found on the road. The cameras of the golf cart will sample images from the road and send feed them into a neural network that will be trained to do the task of identification and detection.

In particular, the neural network will be a convolutional neural network (CNN). This selection is based on the fact that CNNs are widely used for image classification tasks, as they use filters where the parameters have a spatial layout that extracts the underlying features of images by using the convolution operation. This feature extraction allows the CNN to deal with translation, scaling and other operations on an object in an image and still detect it.

The dataset that will be used to train the CNN will be the 2014 release of the COCO dataset [6], which has 80 different categories. This dataset has clean labels with no noise, so it can be classified into the supervised learning paradigm. The distribution of the number of annotations per category is shown in Figure 4.18.

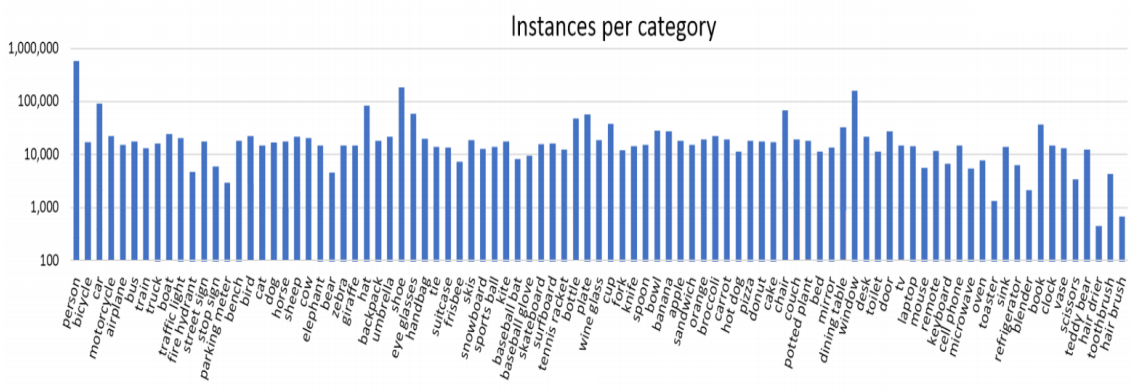


Figure 4.18: Number of annotations per category in the COCO dataset [6].

From this dataset, the classes that will be relevant to the task at hand will only be: person, car, truck, bicycle, motorcycle, cat, dog, bus, stop sign, street sign, traffic light and train.

The training process was initially made removing from the dataset the non relevant classes, but this led to unsatisfactory results in where most of the predictions were false positives for the class "person". The reason for this is that if most of the categories are removed, the dataset becomes imbalanced with predominance of the

category ”person”. Note that in Figure 4.18 is shown how the number of annotations for the class ”person” is well above the rest of the categories, so removing most of the categories accentuates the predominance of the class ”person”. Thus, as the CNN trains over much more instances of ”person” than the rest, it tends to make predictions for ”person” than for the other categories.

Therefore, despite the fact that most of the classes in the dataset are not relevant to the task, the CNN has been finally trained considering all of the 80 classes. Then, at inference time every detection of a non relevant class will simply be disregarded.

As mentioned before CNNs are good at classification tasks, but for the purpose of object detection they must rely on algorithms. There are many algorithms that researchers are still improving and using, but among all the options that can be used, these are the ones that have been considered for this project:

- **R-CNNs:** It uses selective search which consists in using color segmentation at different scales to create region proposals around the image by grouping together adjacent pixels by texture, color or intensity to identify objects [11]. Thus, this region proposals become bounding boxes that potentially contain objects. Then, each of these bounding boxes are fed separately to the convolutional neural network which classifies each of the boxes into a category.

  If the CNN predicts that there is no object of interest in the proposed region it is simply disregarded, but if it predicts that it contains an object of interest the proposed bounding box is highlighted as a detection.

- **YOLO:** As its name suggests (You Only Look Once), the image is only fed once into the neural network to detect all of the elements of interest. This

is achieved by dividing the images into grids, and in every cell of the grid, a prediction is made. This predictions consist of a set of coordinates for the boxes that want to be predicted and for each box, the score for the presence of an object and a one hot encoded array to determine the class of the object. Once the network outputs the predictions, an algorithm called Non-max suppression is performed in order to suppress the boxes that have low confidence scores as well as the ones that have high IOU with other boxes that have better scores for the same class.The fact that this method only sees the image once, makes it faster than the rest of the methods at inference time.

The chosen algorithm for this project has been YOLO for the speed and good accuracy that it can provide. In particular, YOLO allows the model to work with videos at 45 frames per second [8]. The algorithm is less accurate than other algorithms such as R-CNN, but still gives good accuracy, and due top the fact that the task will require fast detection as the car will move at a relatively high speed, the fast inference time that it provides will be very helpful for this application. This method has the versions YOLOv1, YOLOv2, YOLO9000, YOLOv3 and Fast YOLO, which have small differences between each other that represent improvements in accuracy or speed in every new version. The YOLO that has been used for this project is between YOLOv1 and YOLOv2, because it introduces the usage of anchor boxes, but doesn't use some of the preprocessing methods that YOLOv2 utilizes.

Therefore, it is necessary to convert the COCO dataset into the right format for YOLO. This means that for every image in the dataset, its corresponding label has to be a volume of shape (SxSxAx(C+5)) where S is the grid size into which the image is divided, A is the number of anchor boxes that will be used, C is the number of

classes to learn and 5 refers to the 4 coordinates of the predicted bounding box plus 1 score for the presence of an object in the box.

Usually, the YOLO algorithm is used along with an architecture for the convolutional neural network called Darknet. However, in this project the architecture that has been used is MobileNetV2 instead. This is to provide the fields of machine learning and computer vision with some experimentation on other architectures in the YOLO algorithm.

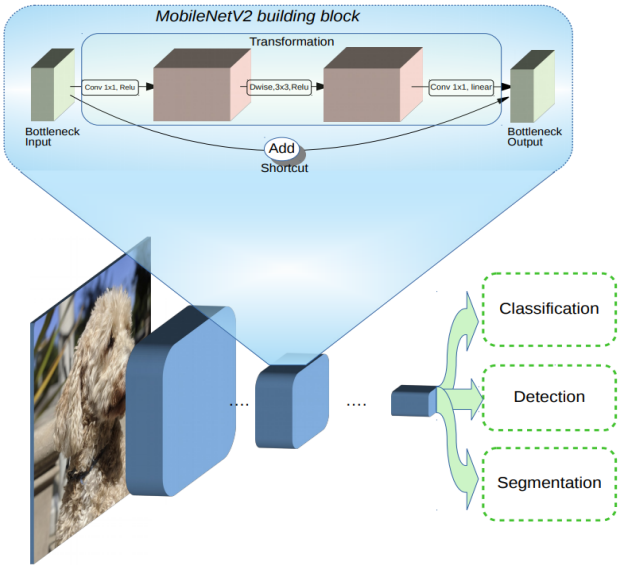


Figure 4.19: MobilenetV2 building block.

The MobileNetV2 architecture consists of 17 of the blocks shown in Figure 4.19 put one after the other, followed by a 1x1 convolution, a global average pooling layer, and a classification layer. In order to adapt the architecture to the YOLO algorithm, it is necessary to change the output to the desired shape, which is the shape of the labels (SxSxAx(C+5)). For this, the last layers were substituted by the block shown in Figure 4.20.

Layer (type)	Output Shape	Param #	Connected to
Conv_1_bn (BatchNormalization)	(None, 14, 14, 1280)	5120	Conv_1[0][0]
leaky_re_lu_1 (LeakyReLU)	(None, 14, 14, 1280)	0	Conv_1_bn[0][0]
conv2d_1 (Conv2D)	(None, 7, 7, 20)	230420	leaky_re_lu_1[0][0]
batch_normalization_1 (BatchNor	(None, 7, 7, 20)	80	conv2d_1[0][0]
leaky_re_lu_2 (LeakyReLU)	(None, 7, 7, 20)	0	batch_normalization_1[0][0]
conv2d_2 (Conv2D)	(None, 7, 7, 425)	136425	leaky_re_lu_2[0][0]
batch_normalization_2 (BatchNor	(None, 7, 7, 425)	1700	conv2d_2[0][0]
leaky_re_lu_3 (LeakyReLU)	(None, 7, 7, 425)	0	batch_normalization_2[0][0]
reshape_1 (Reshape)	(None, 7, 7, 5, 85)	0	leaky_re_lu_3[0][0]
input_1 (InputLayer)	(None, 1, 1, 1, 50,	0	
lambda_1 (Lambda)	(None, 7, 7, 5, 85)	0	reshape_1[0][0] input_1[0][0]

Total params: 2,626,609
Trainable params: 2,591,607
Non-trainable params: 35,002

Figure 4.20: New final layers of the CNN's architecture.

Note that the first layer of the added block is connected to a previous convolutional layer of the original MobileNetV2. Then, 2 convolutional layers have been added:

- The first convolutional layer has a stride of 2 in order to reduce the shape of the output feature map from 14x14 to 7x7. This is because the grid size in which the images will be divided is of shape 7x7.
- The second convolutional layer has 425 filters. This is because the algorithm will be using 5 anchor boxes and will be predicting 80 classes. Thus, 5x(80+5) is equal to 425.

Finally, the 425 feature maps produced by the 425 filters are reshaped into the desired shape of (5x85). Thus, the final output of the CNN is (7x7x5x85).

The ReLU (Rectified linear unit) activations have been substituted by Leaky ReLU activations due to the fact that ReLU killed the gradients in the training process.

Figure 4.21 shows that when the input of a ReLU activation is negative the gradient is zero, which impedes updating the parameters of the CNN in backpropagation.

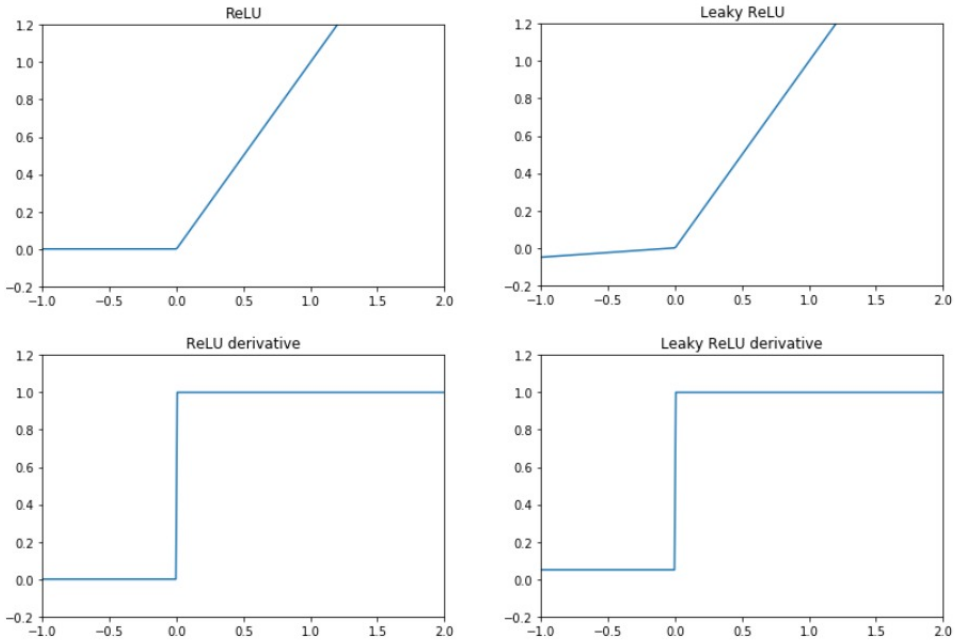


Figure 4.21: ReLU and Leaky ReLU activations and their derivatives.

Once the architecture of the neural network has been built, it must be trained. For this task it is necessary to have a great amount of computational power, which can't be provided by a regular computer. Thus, a virtual machine has been created with the service of Google Cloud. The virtual machine has 40 GB of storage, 4 vCPUs, 15 GB of RAM and one GPU NVIDIA Tesla K80.

Then all the files containing the model and the dataset have been uploaded to this virtual machine and all the dependencies necessary to run the code have been installed.

Finally, the training was performed with the following configuration:

- **Grid size:** The grid size has been set to 7x7. This number has been set to an odd number because as it has been found that having a central cell helps

the model make better predictions as usually objects tend to appear in central positions of the images [9].

Scale factors for the YOLO loss: They have been set to the most standard values used except for the no-object scale, which has been increased from 0.5 to 1 due to previous results leading to false positive predictions. The standard values for the rest of the factors are 1, 5 and 1 for the scales of "presence of object", "box coordinates" and "classification" losses respectively.

- **Batch size**: The batch size has been set to 16 as in previous experiences with the Google Cloud, CUDA ran out of memory with batch sizes of 24. This value for the batch size seems to work well.

- **Anchor boxes**: For determining this hyperparameter a k-means clustering algorithm has been performed on the dataset in order to determine the number of anchor boxes and their width and height. The algorithm has been run for k = 5 clusters and the resulting average IOU value for the 5 anchor boxes was 0.52, while the final values for the height and width of the anchor boxes were [0.62,0.38 , 1.87,1.04 , 3.96,2.52 , 10.15,3.6 , 14.12,8.52].

- **Learning Rate and decay**: The learning rate has been set to $1 \cdot 10^{-4}$ with a decay of zero in the ADAM optimizer.

- **Data augmentation**: The method to avoid overfitting that has been used is data augmentation. This consists in scaling, translating, rotating and mirroring the images of the dataset to create more data points that are similar to the original data point. This makes it hard for the model to overfit to all the data points, so the variance of the model is reduced.

4.4.1 Results

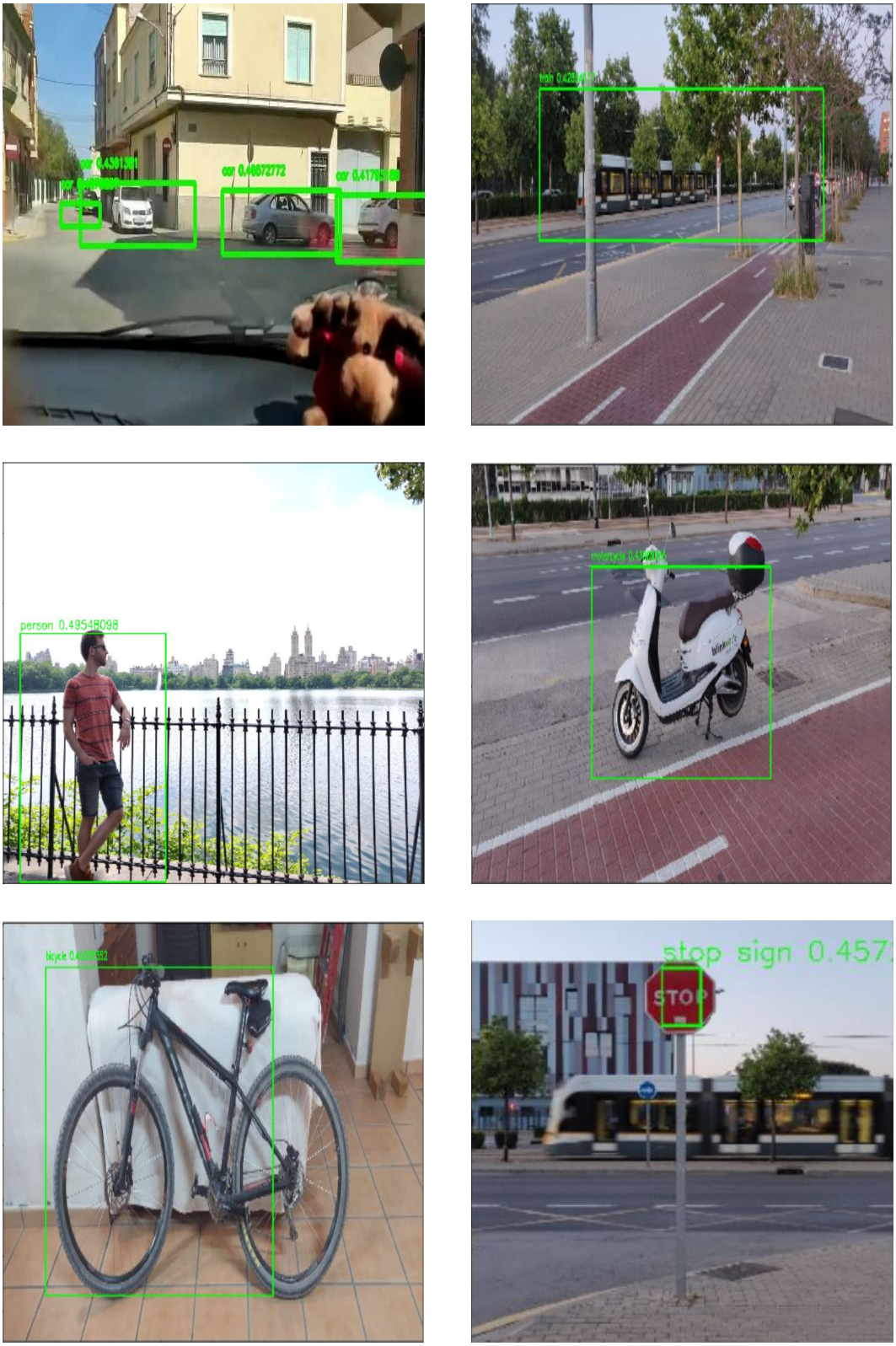


Figure 4.22: Results for detections of 6 of the categories.

Figure 4.22 shows some pictures with detections that the convolutional neural network has output after the training. Note that the predictions are good in most cases, even when the objects are covered by other elements.

However, the objects are detected with lower confidence scores than what the DarkNet architecture achieves in state of the art YOLO object detectors, so the detection threshold has to be lowered to 0.4 to start detecting objects. This means that the model is not separating as much the cases where there is not an object from when there is, causing false negative errors and false positive errors more often than with DarkNet. One example of this is the false negative error in the bottom right picture, where the train behind the stop sign is not detected.

Therefore, one can conclude that the MobileNetV2 architecture achieves decent results, but DarkNet is much more suited for the object detection task using the YOLO algorithm.

Chapter 5: Discussion

In this document the advances on the autonomous golf cart project have been covered in the aspects of hardware and software. In the hardware aspect, all the mechanical components necessary to actuate on the elements that allow driving the car with software commands has been designed and implemented successfully.

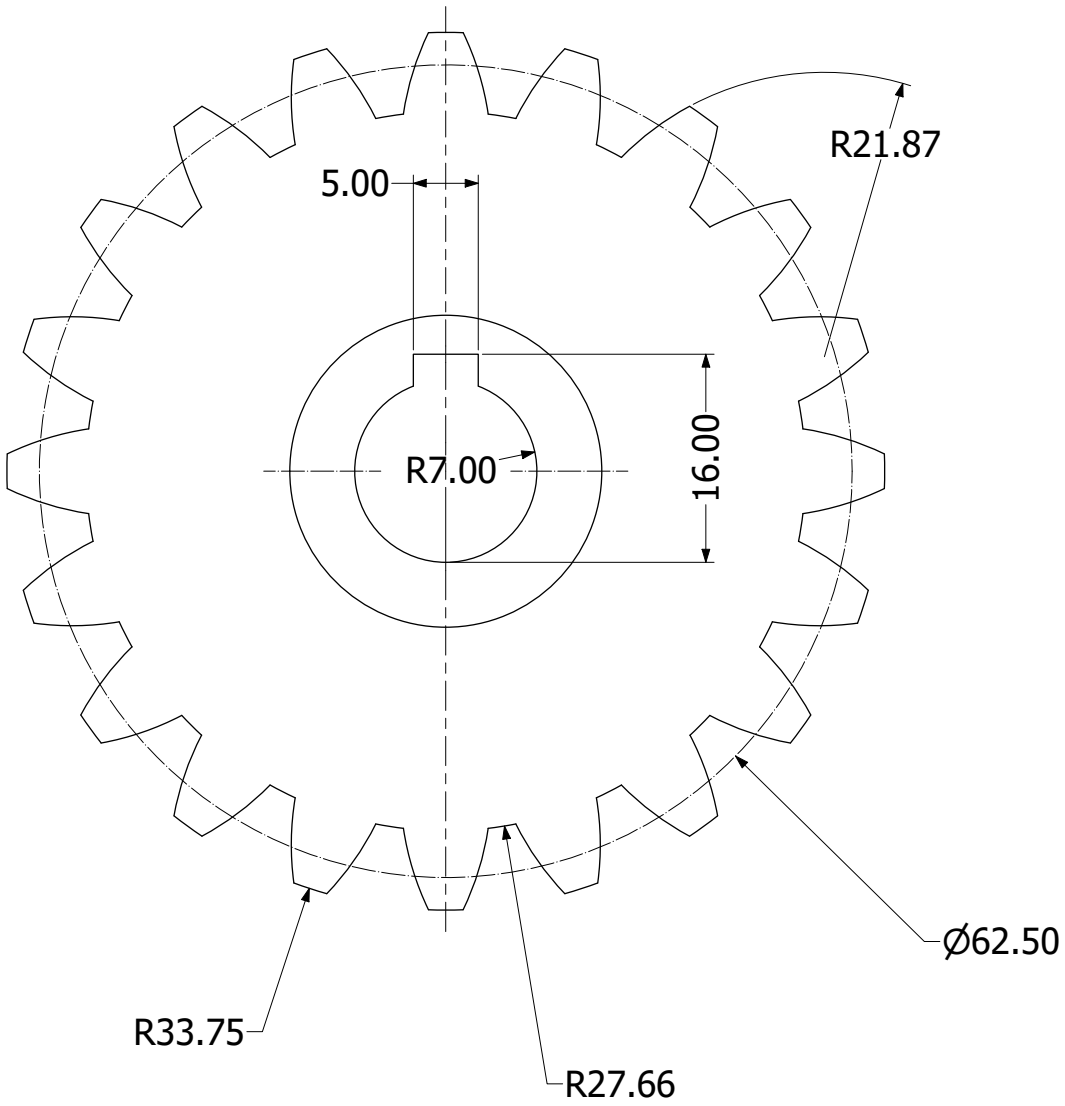
On the other hand, the software aspect of the autonomous driving was covered with the training of a convolutional neural network that uses the YOLO algorithm to detects elements on the road. The architecture for the neural netwrk was MobileNetV2 instead of DarkNet, which is the most commonly used architecture for this algorithm. The resulting object detector was able make accurate detections in most of the cases, but the overall performance is still significantly higher with the DarkNet architecture. The reason for this can be that instead of a global minimum, a local minimum was found in the objective function during training, or simply that the MobileNetV2 architecture doesn't have the ability to capture the complexity of the problem.

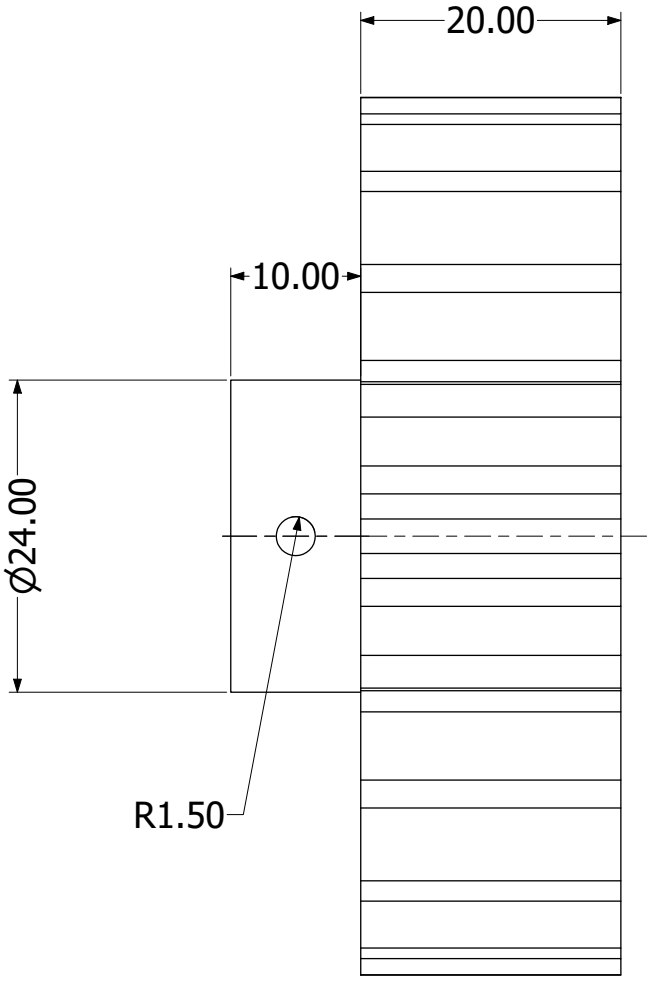
If it is the former, this could be solved by using random restart during training to find other minima in the function, which requires more training time to explore all the domain and thus, more resources.

The remaining efforts to complete the project will have to focus on providing the car with sensors and cameras that will gather information of the surroundings of the golf cart, and also on powering and programming all the mechanical components

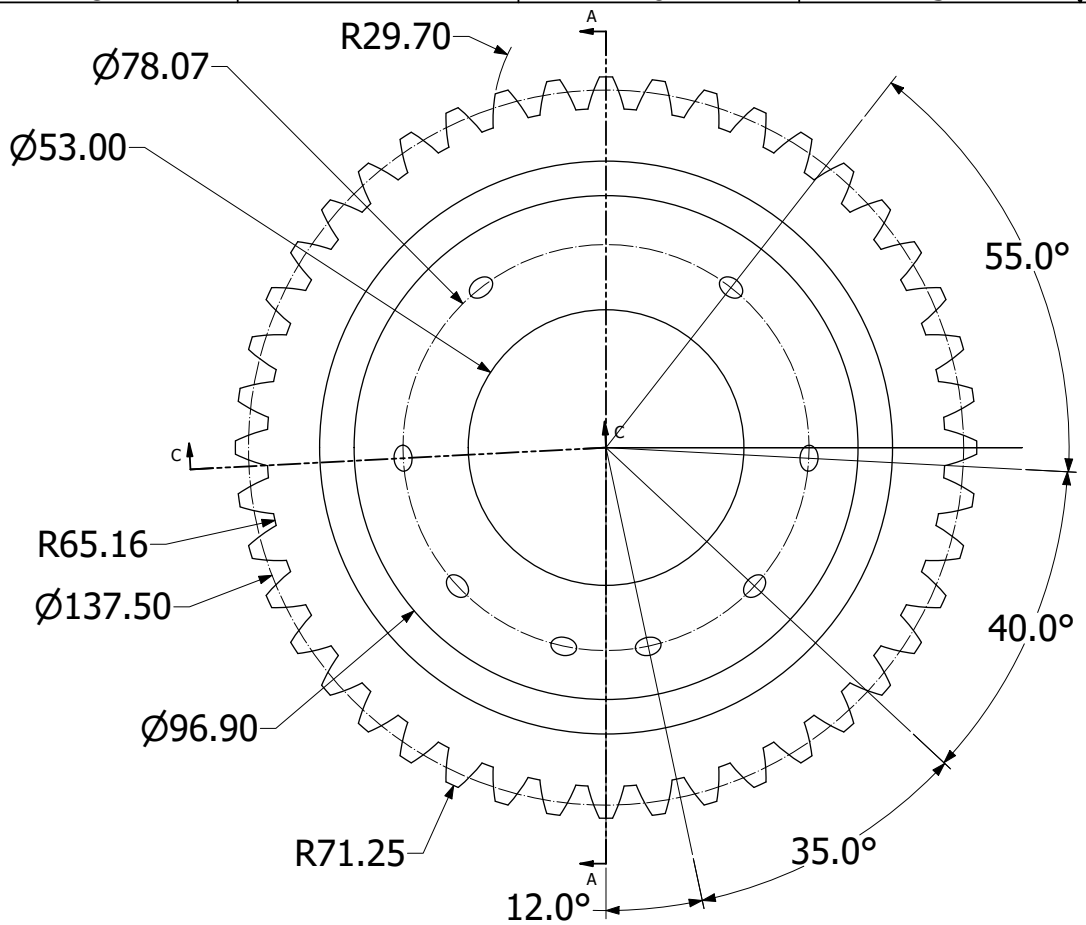
implemented in this thesis and setting up a SCADA system to synchronize all the components into a functional unit.

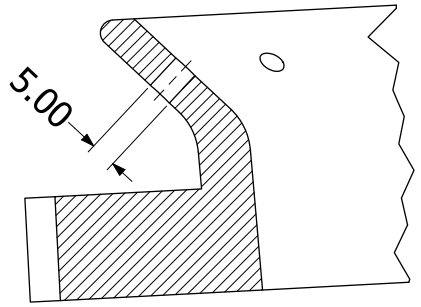
Appendices



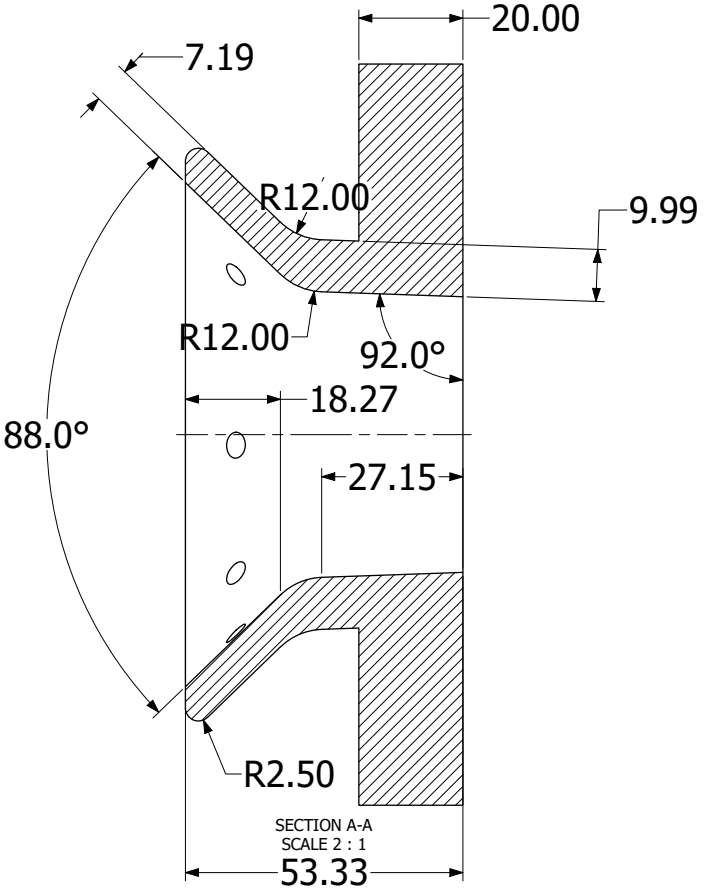


DRAWN jacob	6/27/2019	
CHECKED		TITLE: PINION GEAR
QA		
MFG		
APPROVED		SIZE D · DWG NO 1 · REV
		SCALE 5 : 1 · SHEET 1 OF 5



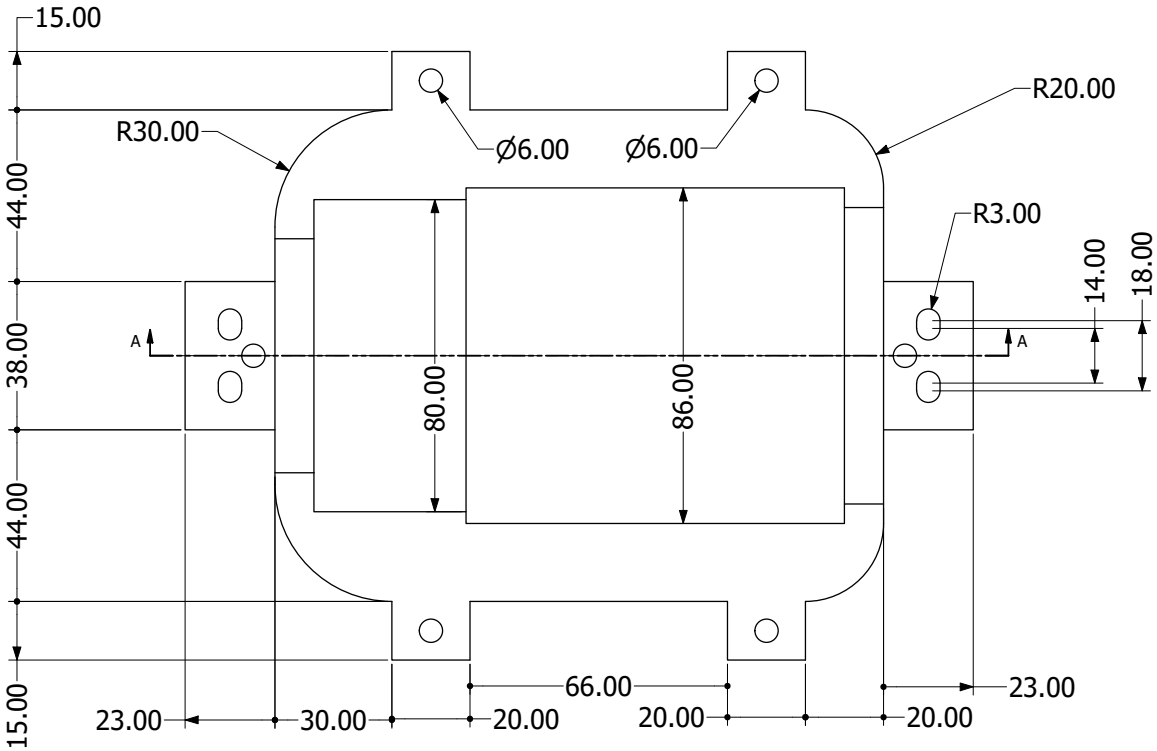


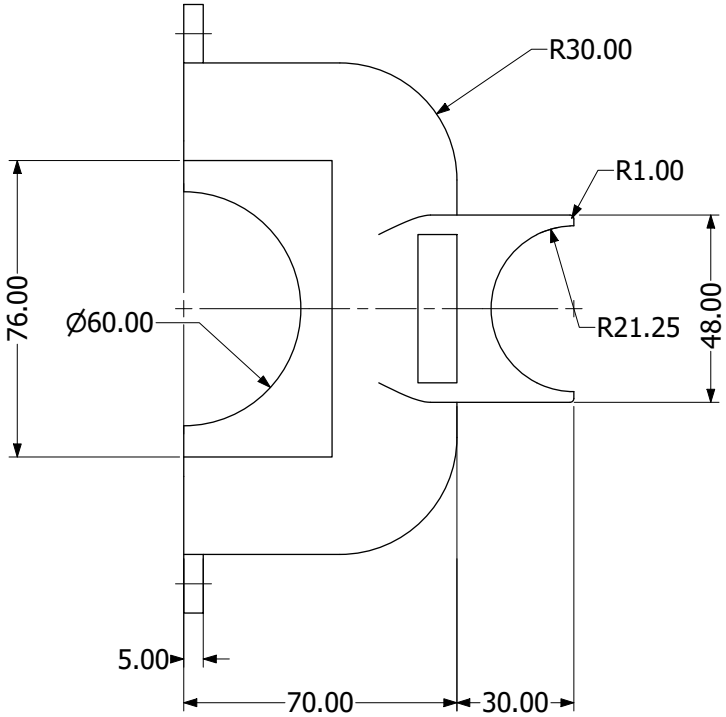
SECTION C-C
SCALE 2 : 1

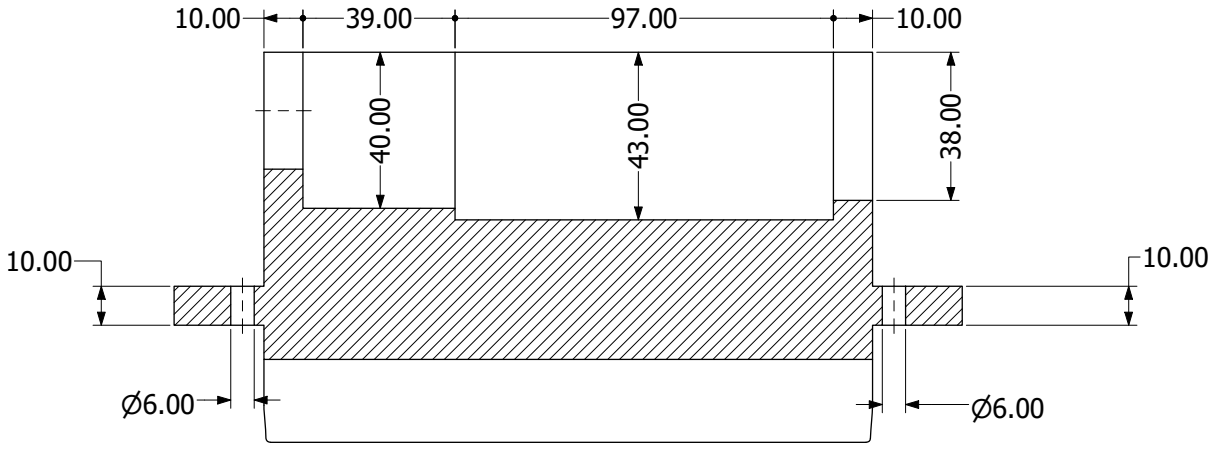


SECTION A-A
SCALE 2 : 1

DRAWN jacob	6/27/2019	
CHECKED		TITLE: WHEEL GEAR
QA		
MFG		
APPROVED		SIZE D · DWG NO 2 · REV
		SCALE 2 : 1 · SHEET 2 OF 5

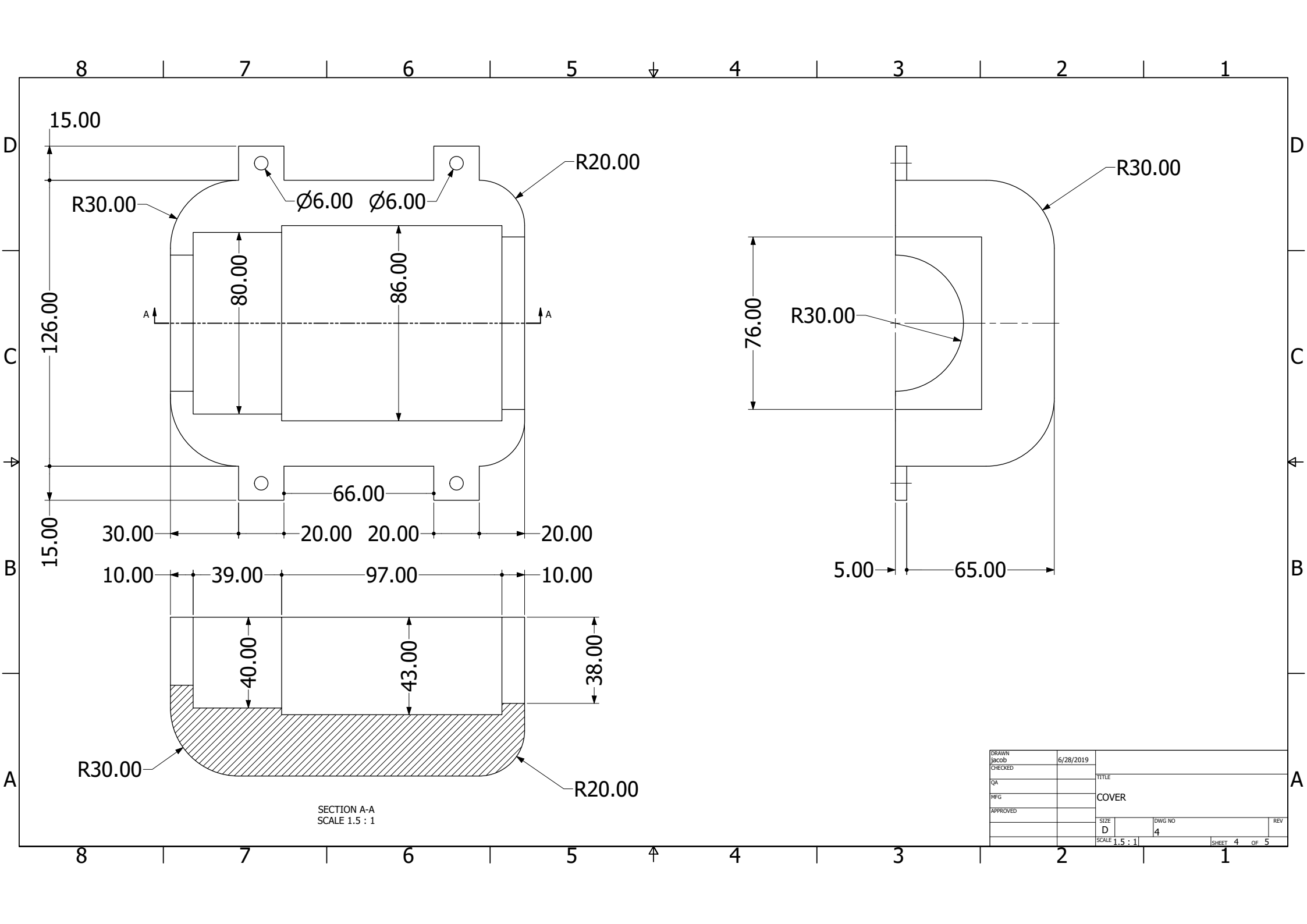


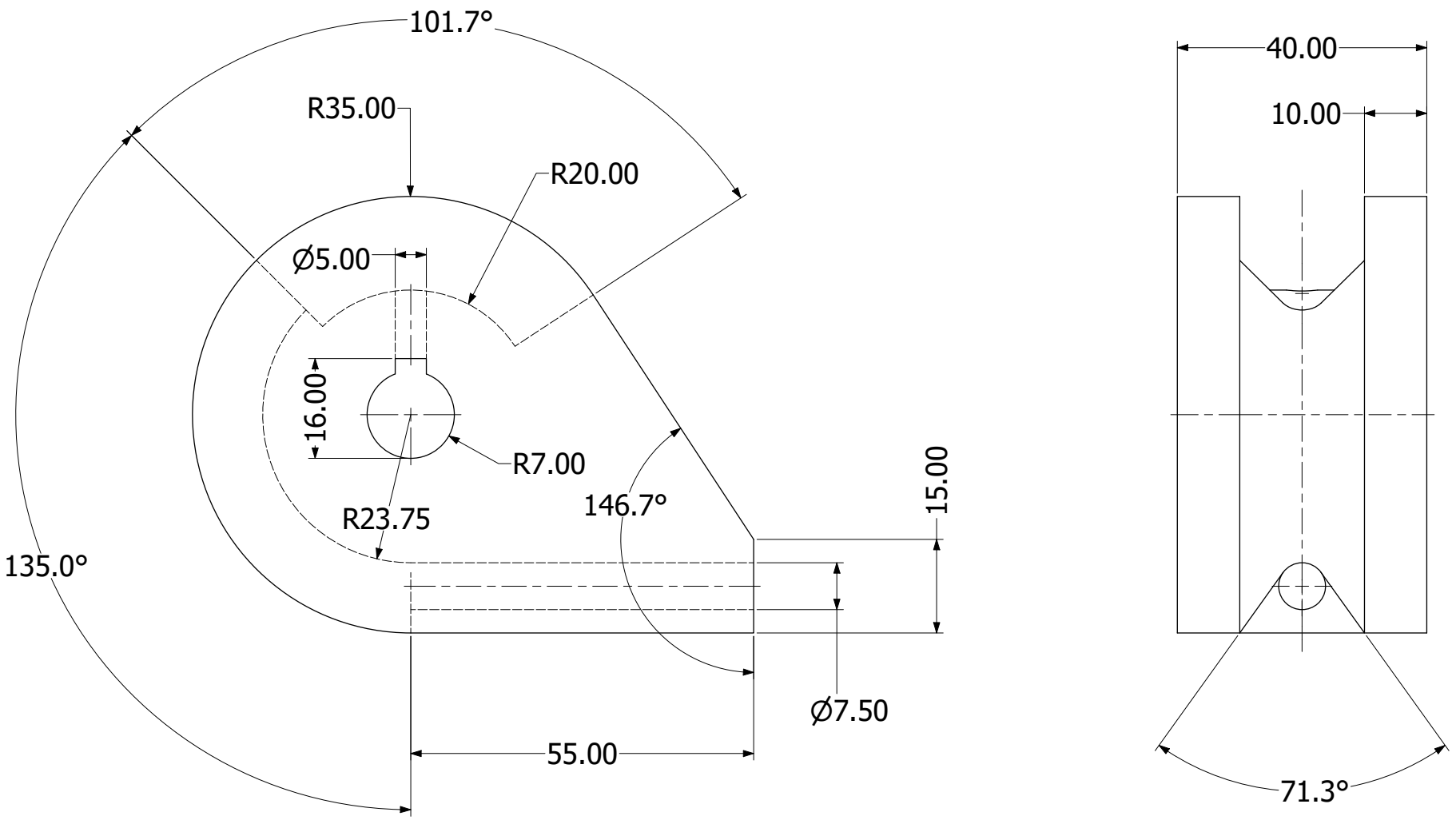




SECTION A-A
SCALE 1.5 : 1

DRAWN jacob	6/28/2019	
CHECKED		TITLE: BASE
QA		
MFG		
APPROVED		SIZE D · DWG NO 3 · REV
		SCALE 1.5 : 1 · SHEET 3 OF 5


15.00
R20.00
R30.00
Ø6.00
Ø6.00
80.00
86.00
126.00
A
A
66.00
15.00
30.00
20.00
20.00
20.00
10.00
39.00
97.00
10.00
40.00
43.00
38.00
R30.00
R20.00
R30.00
76.00
R30.00
5.00
65.00
SECTION A-A
SCALE 1.5 : 1
DRAWN
jacob
6/28/2019
CHECKED
QA
MFG
APPROVED
TITLE
COVER
SIZE
D
DWG NO
4
REV
SCALE 1.5 : 1
SHEET 4 OF 5



DRAWN jacob	6/28/2019	
CHECKED		TITLE: PULLEY FOR BRAKES
QA		
MFG		
APPROVED		

SIZE D | DWG NO 5 | REV

SCALE 3 : 1 | SHEET 5 OF 5

Bibliography

[1] American Gear Manufacturers Association et al. Ansi/agma 2101-d04,". *Fundamental Rating Factors and Calculation Methods for Involute Spur and Helical Gear Teeth. F net face width (mm) F 0*, 1, 2004.

[2] Keshav Bimbraw. Autonomous cars: Past, present and future a review of the developments in the last century, the present scenario and the expected future of autonomous vehicle technology. In *2015 12th International Conference on Informatics in Control, Automation and Robotics (ICINCO)*, volume 1, pages 191–198. IEEE, 2015.

[3] Joschka Bischoff, Caroline J Rodier, Elham Pourrahmani, Miguel Jaller, Anmol Pahwa, and Michal Maciejewski. Competition among automated taxis, transit, and conventional passenger vehicles: Traffic effects in the san francisco bay area. Technical report, 2019.

[4] Peter Davidson and Anabelle Spinoulas. Autonomous vehicles: what could this mean for the future of transport. In *Australian Institute of Traffic Planning and Management (AITPM) National Conference, Brisbane, Queensland*, 2015.

[5] Massimo Guarnieri. Looking back to electric cars. In *2012 Third IEEE HISTory of ELectro-technology CONference (HISTELCON)*, pages 1–6. IEEE, 2012.

[6] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014.

[7] Todd Litman. *Autonomous vehicle implementation predictions*. Victoria Transport Policy Institute Victoria, Canada, 2017.

[8] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[9] Joseph Redmon and Ali Farhadi. Yolo9000: better, faster, stronger. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7263–7271, 2017.

[10] Yasser Shoukry, Pierluigi Nuzzo, Alberto Puggelli, Alberto L Sangiovanni-Vincentelli, Sanjit A Seshia, and Paulo Tabuada. Secure state estimation for cyber-physical systems under sensor attacks: A satisfiability modulo theory approach. *IEEE Transactions on Automatic Control*, 62(10):4917–4932, 2017.

[11] Jasper RR Uijlings, Koen EA Van De Sande, Theo Gevers, and Arnold WM Smeulders. Selective search for object recognition. *International journal of computer vision*, 104(2):154–171, 2013.

[12] M Mitchell Waldrop. Autonomous vehicles: No drivers required. *Nature News*, 518(7537):20, 2015.

[13] Ding Zhao and Huei Peng. From the lab to the street: Solving the challenge of accelerating automated vehicle testing. *arXiv preprint arXiv:1707.04792*, 2017.