Automated Wheel Loader

Spring Term Report 2021

TESS ANTONSSON
LUCAS JACOBSSON FALCON
DANIEL GRÖNÅS
HOMING KONG
FREDRIK MAZUR
JOACHIM OTTOSSON
MARCUS RUDERER
JAN SIWEK
CHIEH-JU WU

SOTA report at ITM
Supervisor: Tobias Vahlne

Abstract

This project is in collaboration with Volvo Construction Equipment. The aim of this project is to construct a scaled autonomous wheel loader to tackle the existing problems in the plastic bale handling process. In this paper, the state of the art analysis, the proposed design of the autonomous wheel loader, and the future development plan of this project are introduced. The proposed solution provides a customised scaled wheel loader design with fully autonomous capabilities to find and perform a loading, carry, and unloading cycle in a controlled environment. This concept model will be built and realised during the autumn of 2021.

Acknowledgements

We would like to thank Volvo CE for this opportunity, especially Joakim and Bobbie for their professional inputs and feedback. We would also like to thank our team coach Tobias for his guidance and support during the project development.

Contents

List of Figures

List of Tables

List of Abbreviations

3D Three-Dimensional, page 12

Adam Adaptive Moment Estimation, page 37

ANN Artificial Neural Network, page 17

CAD Computer Aided Design, page 7

CE Construction Equipment, page 1

CNN Convolutional Neural Network, page 21

DC Direct Current, page 12

DNN Deep Neural Network, page 21

FoV Field of View, page 13

GPU Graphics Processing Unit, page 3

IMU Inertial Measurement Unit, page 13

IR Infrared, page 13

LiDAR Light Detection and Ranging, page 13

MSE Mean Squared Error, page 20

NN Neural Network, page 17

PMSM Permanent Magnet Synchronous Motor, page 12

RGB Red, Green, Blue, page 13

RGB-D Red, Green, Blue colour model with Depth, page 13

RNN Recurrent Neural Network, page 21

ROS Robot Operating System, page 13

SGD Stochastic Gradient Descent, page 19

SLAM Simultaneous Localisation and Mapping, page 13

SR Stakeholder Requirement, page 3

ToF Time of Flight, page 13

TR Technical Requirement, page 3

VO Visual Odometry, page 14

WD Wheel Drive, page 29

Chapter 1

Introduction

In this chapter, the background of the project, its scope, and the stakeholder and technical requirements will be presented.

1.1 Background

In 2017, over 2.4 million tons of plastic waste was introduced to the Swedish market, by means of either importation or production. Unfortunately, only a small percentage of this is recycled [1]. When the plastic waste is reprocessed, the plastics are compressed into bales and transported in this form to unloading stations. Because of the heavy machinery that is utilised in this handling process, the work environment is potentially hazardous. Due to the influence of human operators, the overall process can also be time inefficient and incur high labour costs [2].

Today, Volvo CE (Construction Equipment) manages plastic waste handling primarily with manual, fuel-driven wheel loaders. While this is a reliable and customisable solution, a new concept has been created in collaboration with LEGO with the goal of designing the loader of the future - namely, the Volvo Zeux. This conceptual model is fully autonomous and electric, introducing an ”eye” for human interaction and a drone to increase environment awareness. Other key features include four-wheel drive with double Ackermann steering for traction and stability, sensored wheels to maintain balance, a scissor frame for increased maneuverability, and a moving counterweight to automatically alter the vehicle's centre of gravity [3]. Because this concept (seen in Figure 1.1) is powered with electricity instead of fuel, environmental impact is reduced, and since it operates autonomously, both safety and efficiency are improved.

This project aims to implement this concept, with support from and in collaboration with the stakeholders (Volvo CE), by developing an autonomous system on a scaled, prototypic loading device. This prototype will require functioning actuation and control systems to drive and perform load cycles in a controlled manner, a vis-

Figure 1.1: Volvo Zeux concept [3]

ion system with robust image recognition to detect bales of recycled material, and a positioning system to navigate between loading/unloading points. Reliable machine communication will also be needed to ensure all systems act in harmony. Once operational, the concept will be tested on a scaled model site (containing plastic bale replicas, loading/unloading points, and obstacles) with the goal of successfully and autonomously performing a load and carry cycle.

This report will further outline the project requirements, a theoretical background of the *state of the art* alternatives that exist for the prototype's mechanical and software interfaces, as well as the initial design concepts generated for the prototype. A summary of the project group's organisation, a budget of planned purchases to construct the model, and a time plan of future work will also be assessed.

1.2 Scope

The main scope of this project is to build a scaled prototype that can be tested to prove the Volvo Zeux concept. The scaled prototype will be able to detect plastic bales using image recognition and safely load, carry, and unload the plastic bales at a designated unloading spot using a navigation system. If time allows, the developed prototype will include autonomous path planning capabilities and an obstacle detection and avoidance system. The total cost of the developed prototype should not exceed the budget of 50 000 SEK provided by the stakeholders.

The prototype will be a stand-alone system, meaning that all necessary components will be placed onto the prototype and function independently from remote hardware. The ratio of the model will be determined by a minimum scale, limited by the space required to fit all necessary components within the prototype, and a maximum scale, defined in terms of material cost and manufacturability.

On the real operating site, as described by the stakeholder, confined spaces can

occur. Therefore, the steering mechanism will be designed to allow for accurate and high mobility in order to operate in these cramped areas. The lifting mechanism will be designed based on the same characteristics in order to safely load, carry, and unload the plastic bales.

For the prototype to be able to detect the material, namely the plastic bales, a vision system with robust image recognition will be developed. The main limitation in this area is firstly to procure a GPU (Graphics Processing Unit) that is powerful enough to handle the computations needed to perform the image recognition, whilst remaining in accordance with our budget. Therefore, this component will be considered carefully together with the stakeholders. The second constraint within this area is to create a large enough dataset and construct a robust network model to achieve a satisfactory accuracy in testing.

The site model will be created to give considerably ideal conditions regarding the testing environment. That is, the surface can be assumed to be flat and the surroundings, such as walls and obstacles, can be assumed as well defined.

1.3 Requirements

To define the problems that need to be solved in order to meet the stakeholders' expectations, a set of requirements was devised. These stakeholder requirements were further evaluated to construct more extensive and detailed technical requirements. Initially, a first set of stakeholder requirements was constructed to, in agreement with the stakeholders, be classified as either a primary or secondary requirement. The secondary requirements can be seen as lower priority, meaning that these requirements should be satisfied if all primary requirements are fulfilled. The primary requirements are listed in priority order, meaning that the first requirement has a higher priority that the second one, and so on. The stakeholder requirements are denoted as **SR** and technical requirements are denoted as **TR**.

Primary Requirements

1. **SR** The software shall be able to recognise plastic bales.
    a) **TR** A trained neural network should be able to perform image recognition and detection of the plastic bales.
    b) **TR** A dataset of sufficient size of plastic bale images shall be created to train the network model.
    c) **TR** The prototype should have a LiDAR and/or camera embedded.
2. **SR** The prototype shall be able to pick up plastic bales and unload them at a designated spot.

   a) **TR** The prototype must have the mechanical strength and stability to pick up and carry the weight of the bales.
   b) **TR** The tool should be able to lift to a height of at least 180 mm and tilt at an angle of 60°.

3. **SR** The prototype shall be tested and verified.
   a) **TR** The prototype should have a scale of approximately 1:10.
   b) **TR** The prototype should be tested on a scaled site model that includes model bales, obstacles, a loading point, and an unloading point.
   c) **TR** The plastic bales should have a scale of approximately 1:10 with a weight of approximately 1-2 kg.
4. **SR** All the functionality should be deployed on the prototype itself, or with flexibility and modularity in mind.
   a) **TR** The prototype must carry its own rechargeable battery.
5. **SR** The prototype should be automated.
   a) **TR** The prototype must have the ability to know its orientation and if it is driving in a straight line.
6. **SR** The given budget of 50 000 SEK should not be exceeded.

Secondary Requirements

1. **SR** The prototype shall be able to navigate itself in the environment.
   a) **TR** The prototype should be able to navigate autonomously.
2. **SR** The prototype shall have an obstacle detection and avoidance system.

Chapter 2

State of the Art

The state of the art analysis provided in this chapter will cover the mechanical and software concepts that are relevant for this project's development.

2.1 Mechanical Structure

The mechanical structure of the vehicle has to satisfy many roles to ensure the structural integrity of the vehicle, as described by SR2. The mechanical system is responsible for fulfilling the needs of a propulsion system, steering to navigate in narrow spaces, a chassis as a base for specialised equipment, and a lifting mechanism.

2.1.1 Chassis

The chassis is a complex assembly of components that supports a vehicle's structure. It should be designed to withstand the loads that occur during usage, such as lifting the load, driving to a designated point with the load, and unloading in a safe manner. The chassis is obligated to be robust enough to withstand the forces that can appear during these actions. Some of the loads that can impact the chassis are:

- The weight of the body and cargo
- Vertical and twisting loads as a result of uneven surfaces
- Lateral forces caused by the road camber, side wind, and steering of the vehicle
- Torque transmitted from the engine and transmission

The rigidness of the chassis can be described by two major characteristics: torsion stiffness and lateral stiffness. Torsion stiffness refers to a cross section of the chassis frame. It describes how resistant the chassis is to reluctant momentum, which twists the construction. Lateral stiffness specifies the rigidity of the overall body to the lateral forces [4].

The shape and size of the vehicle chassis is correlated to the dimensions of the components mounted on it. Some of the parts that need to be taken into account when designing this structure are the sensors, steering system, propulsion system, and, especially in this project, lifting mechanism.

Scissor Frame

The scissor frame presented in the Zeux concept is a construction that allows for vertical movement of the whole chassis, whether while immobile or in motion. The choice of lifting system responsible for moving the frame depends on parameters such as lifting height, lifting weight, and the stiffness of the structure. Typically, a scissor construction is used in hydraulic lifts where one (or multiple) actuator is able to lift the entire structure. An example of a scissor frame construction is presented in Figure 2.1.

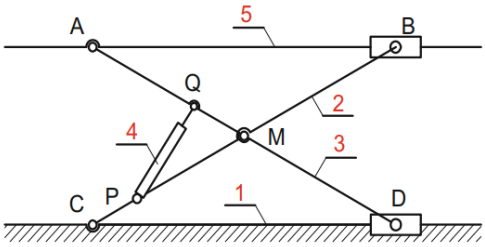

Figure 2.1: Sketch of the scissor frame mechanism: 1. Ground frame, 2. Inner frame, 3. Outer frame, 4. Cylinder, 5. Platform (hoisting terrace) [5]

In the Zeux concept, a scissor movement is accomplished by one actuator and transitional movement of the rear axle, as presented in Figure 2.2.

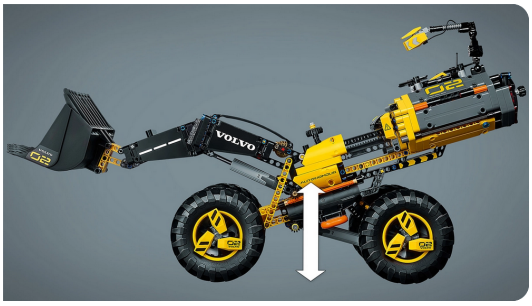

Figure 2.2: Scissor frame of LEGO Zeux concept [6]

Stiff Chassis

A stiff chassis is a concept that simplifies the construction of a vehicle. Torsion and bending stiffness calculations are simplified due to the reduced number of moving

components. Therefore, the model can easily be simplified and imported to CAD (Computer Aided Design) software or other simulation environments. However, a stiff chassis limits the options of applicable steering systems, the type of suspension, as well as method for damping the vibrations [4].

2.1.2 Lifting Mechanisms

In order to transfer, load, and unload the plastic bales, a lifting mechanism is essential, as mentioned in SR2. The movement and functions of real-scale construction vehicles are often accomplished through the usage of hydraulic fluid. In this project, hydraulic actuators were not an option, by virtue of complexity in control and mechanical design in the simplified model. The lifting mechanisms used in modern construction equipment will be investigated in this section as a reference for mechanical design.

Arm Joint System (Kinematic Linkages)

The arm joint system is a mechanism that prevails in the construction vehicle industry. It consists of a kinematic chain with linear actuators and rigid bars. The linkages of vehicles varies with their corresponding application. Figure 2.3 illustrates a sample of kinematic design of a wheel loader. The arm joint system consists of two cylinders which enable the wheel loader to drive the bucket in a two-dimensional plane.

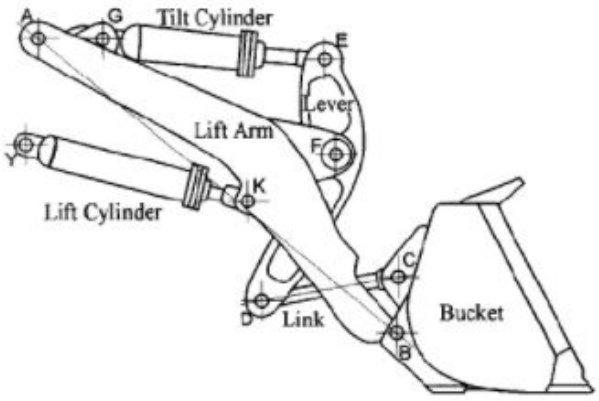

Figure 2.3: Kinematic linkage of a wheel loader [7]

Sliding System

The sliding system is used in forklift trucks to lift and move materials. A carriage with forks is mounted to a vertical mast. The lifting chain is driven by the hydraulic system to elevate and lower the carriage, moving the carriage by sliding it on the rail. A pair of hydraulic cylinders are anchored to the bottom of the mast and pivot

the mast forwards and backwards to tilt the fork carriage. By tilting or lowering the forks, the target object can be transferred [8].

2.1.3 Tools

Various tools could be attached to the lifting mechanism. The configuration of the tools alter the performance and stability of the lifting motion. Choosing an appropriate tool would simplify a complicated motion path and increase the success rate of picking up the plastic bales, as described by TR2a and TR2b.

Loader Bucket

The loader bucket is one of the most common construction vehicle attachments. The general purpose of a loader bucket is to move materials, for instance soil and sand in a construction site. The bucket in Figure 2.4a is a low-profile bucket which has a long and flat bottom compared to normal buckets. A large, sealed bottom prevents the construction materials from being dropped. The bucket scoops material up via the motion of the loader vehicle.

Tynes

Tynes are the forks on the forklift which are used to make direct contact with a load for transport. Tynes are attached to the carriage of a forklift. A pair of tilt cylinders adjust the tilt movement of the carriage and the angle of the tynes relative to the ground. The forks of the machine are altered to an appropriate position and when the vehicle approaches the objective, the forks slide between the ground and the base of the objective.

Bale grabber

A bale grabber, shown in Figure 2.4c, is an attachment on a loader that is typically used to handle bales. A pair of arms extend from the grab mast, actuated by hydraulic cylinders. The target object is held in position rigidly by the sidewise friction force. The simple mechanism of the grabber results in uncomplicated control and motion of the attachment [9].

(a) Bucket [10]

(b) Tynes [11]

(c) Bale Grabber [12]

Figure 2.4: Various lifting tools

2.1.4 Steering

The steering is a system of components that allow a vehicle to follow a desired course. The manoeuvrability of a wheel-based ground vehicle is highly dependent on its steering system. Steering can be achieved in many different ways. Most commonly, four-wheeled ground vehicles steer by applying an angle to the front wheels, which forces the vehicle to move in a curved trajectory. Another method of steering is to control the speed of the inner and outer wheels to make the vehicle turn, called differential steering [13].

Articulated Joint Steering

An articulated vehicle has a vertical pivot joint in its construction, allowing the vehicle to apply an angle between the wheel axles. Any vehicle with a towing trailer could be described as articulated. For a four-wheeled vehicle, an articulated joint can be enough to achieve a balanced steering system with tight cornering capabilities, see Figure 2.5 [13].

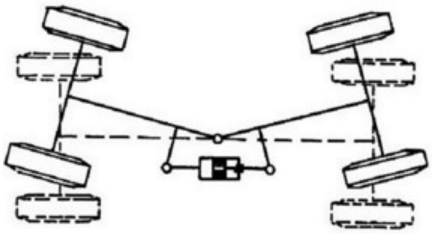

Figure 2.5: Geometry of an articulated vehicle [13]

Ackermann Steering

The Ackermann steering geometry consists of an arrangement of linkages designed to solve the problem of the inner and outer wheels of a vehicle turning with different radii. With a perfect Ackermann geometry, the inner and outer wheels turn with the ideal radius, pivoting around the center of turning, see Figure 2.6. A good approximation of a perfect Ackermann geometry can be achieved with a four-bar linkage on the steering wheels [13][14].

The drawing in Figure 2.6 shows an example of Ackermann steering when implemented on a front-wheel steered vehicle, but solutions exist for all wheel steering and vehicles with more than four wheels. Double Ackermann, for instance, is an implementation of Ackermann steering geometry for four-wheel steered vehicles [15].

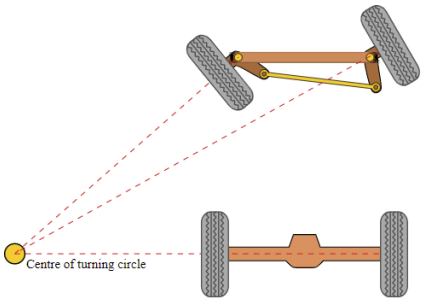

Figure 2.6: Geometry of Ackermann steering [16]

Mecanum wheels

Omni-directional control of a vehicle can be achieved through the use of mecanum wheels. These wheels have rollers that are skewed with respect to the wheel axle, allowing them to roll in two dimensions. A platform with four of these wheels, individually actuated, results in a vehicle that can move in the longitudinal, lateral, and yaw directions. Controlling such a vehicle does not require excessive computations, not even when including wheel slips in the calculations [17]. An example of an application using mecanum wheels is shown in Figure 2.7.

Figure 2.7: An example of an omni-directional platform using mecanum wheels [18]

2.1.5 Drive Train

The drive train of a vehicle is a set of actuators and mechanical components that together make the vehicle move. This section covers well-known solutions of drive train systems and explains their principles.

Drive Types

A wheel-based ground vehicle commonly has a set of wheels that are actuated. The choice of wheels to actuate directly affect the vehicle's performance, agility, and complexity. All-wheel drive, for instance, is beneficial for terrain vehicles, since it allows for more thrust, regardless of the weight balance, and has a lower chance of

getting stuck in difficult terrain. But actuating all the wheels of a vehicle is complex and costly, especially for a vehicle with a suspension system and steering [19].

The Differential is a device that is often used for distributing the torque of a single actuator to two wheels of a single axle during cornering. As the vehicle turns, the inner and outer wheels will have different turning radii and therefore rotate at different speeds. Using a differential to solve this problem works well as long as both wheels on the axle have traction. In other cases, the differential will distribute the torque evenly to both wheels, so in the case where one of the wheels loses traction, the other wheel loses torque [20].

Front/Rear Wheel Drive is an alternative to all-wheel drive, which has reduced cost and complexity. On road conditions, this solution is sufficient for most consumers, and therefore the most common design choice. During hard forward acceleration however, the load balance of the front and rear wheels change and more of the vehicle's weight pushes against the rear tires and less against the front. This increases the maximum thrust that the rear wheels can achieve and reduces the maximum thrust of the front wheels. For a performance vehicle, rear-wheel drive is therefore beneficial for greater acceleration and all-wheel drive can increase the acceleration performance even more [19].

Wheel/Hub Motors are a less common solution for cars and transport vehicles, but more common for robots. The reason why this solution is less common for vehicles are several, but in particular, there's one major problem affecting the stability and comfort. As a motor is introduced into the wheel hub or wheel itself, the unsprung mass is increased. This puts higher loads on the tires and suspension on a bumpy road, making the ride less comfortable for a passenger [21]. This is not a problem for all applications though. Individual-wheel actuation comes with the benefit of controlability, which can benefit the performance of a vehicle by dynamically balancing the torque distribution [19]. In some robotic applications, such as omni-directional and differential steered platforms, individual-wheel actuation is a requirement in order to achieve the intended functionality [17].

2.1.6 Actuators

Actuators are components that are responsible for the controlled movements of mechanical systems. An actuator requires a source of energy and a control signal and outputs a force, moment, linear movement, rotation, or a combination of these. Examples of common actuators are electric motors, servos, combustion engines, hydraulic pistons, and solenoids. Motors and engines generally output continuous rotation and torque, and are therefore the common choice for wheel actuation of ground vehicles. Pistons and linear actuators are responsible for outputting linear motion and force [22].

Electric Motors

An electric motor can be divided into two main parts: the rotor and the stator. The rotor is the part that rotates and the stator is the part which contains the rotor, attaches the motor to the machine of its implementation, and contains the electrical connections. Most electric motors generate torque and angular speed using rotating magnetic fields. Hence, an electric motor requires electromagnets that can be toggled on and off [22].

Brushed DC Motors are one of the simplest types of motors. It has permanent magnets in the stator and electromagnets on the rotor. The coil windings on the rotor are connected to a commutator which rotates with the rotor. Two brushes, that are mechanically connected to the stator, are "brushing" against the commutator, connecting the electrical source to the coils. The commutator has two or more terminal plates that, depending on the angle of rotation, supply the coils in such a way that the magnetic field of the rotor is at an offset angle to the magnetic field of the stator. This ensures that, for a constant direct current (DC) voltage source, regardless of the rotor angle, there is a moment with a constant direction generated [22].

Brushless Motors use electrical switches (such as transistors) to control the magnetic field of the stator. The rotor on some types of brushless motors contain permanent magnets. These are called "permanent magnet synchronous motor", or PMSM in short. They are termed "synchronous" because the rotor needs to be in synchronisation with the electric signal in order to work properly. This requires a properly tuned feedback control system [22].

PMSMs can have any even number of magnetic poles on the rotor and any number of phases on the stator coils. Single-phase motors are used in some applications, but three-phase motors are a more common choice when the power requirements are higher. Two-phase motors, commonly known as stepper motors, are widely used in 3D (three-dimensional) printers and industrial machines [22].

2.2 Navigation

The vehicle is required to locate points of interest and navigate within the environment, as described by the Primary Requirements SR2 and TR5a and Secondary Requirements SR1 and SR2. Thus, a navigation system is needed.

2.2.1 Operating System

The Robot Operating System (ROS) is a framework for writing robotic software. ROS contains tools, libraries, and conventions for creating complex and robust robot behaviour across robotic platforms [23]. ROS offers a variety of packages in which the ROS software is organised [24], including packages with Simultaneous Localization and Mapping (SLAM) functionalities which can be used to construct maps for navigation with collected data [25].

2.2.2 Perception

The ability to process a 3D environment defines an autonomous vehicle's capability to position itself in space. There are several hardware solutions for robotic vision, such as light detection and ranging (LiDAR) and RGB-D (Red, Green, Blue colour model with Depth) cameras. When combined with inertial measurement units (IMU) and odometry systems, the accuracy of the estimated position is increased [26].

RGB-D sensor

RGB-D sensors commonly consist of a RGB (Red, Green, Blue) colour camera and infrared (IR) technology to measure depth and the distance to objects through time-of-flight (ToF) or stereo [27]. Depth calculations by ToF are based on the time it takes for the emitted light to reach an object and return [28], while stereo depth measures are based on the coordinate disparity of an object between two readings with different viewpoints [29]. By utilising a projected IR pattern with known variation with respect to distance, the depth can be calculated through triangulation and the surroundings mapped into point clouds, a cluster of points representing a 3D object. The use of IR projection results in dense and robust information, and improves the vision in conditions with varying and/or bad lighting [27].

LiDAR

A common option for perception and localisation is LiDAR systems. LiDARs use laser beams to scan the surroundings through laser rangefinder principles, where the laser beams' ToF is measured and the distance to the object is calculated, outputting a 3D point cloud of the scene. By rotating the base, a 360 degree field of view (FoV) can be achieved, using different solutions to cover the vertical scanning,

such as mechanically moving mirrors.

The LiDAR provides very accurate ranging measurements but offers poor object recognition, and is therefore typically combined with a camera [30]. Another downside of the LiDAR system is the laser projections' ability to accurately map the surroundings in challenging weather conditions, such as fog and rain [31].

IMU

An inertial measurement unit is a device that measures force, velocity, and orientation of a body through a combination of sub-units, such as gyroscopes and accelerometers. While gyroscopes and accelerometers can provide information about the angular rotation and inertial acceleration respectively, more advanced IMU systems also include magnetometers [32]. Magnetometers are mainly used for computing the direction through changes in magnetic fields and may provide a minor increase in position accuracy. IMU systems, with or without magnetometers, provide most of the information needed for navigation, but both are subject to drift errors that increase with operating time [33].

Odometry

Odometry is commonly used for positioning and estimating location with respect to an initial location, where the odometry data may be acquired from a sensor such as LiDAR, encoder, or camera. Each solution has its weaknesses, but can be combined to provide an increased accuracy of the odometry measures. A common combination consists of a LiDAR and wheel encoder or visual odometry (VO), where the secondary data source (Encoder/VO) allows a point cloud to be mapped with respect to a fixed coordinate system [34][35].

2.2.3 Path Planning

There are several different algorithms and methods for an autonomous vehicle to determine a route with obstacle avoidance between two points. In this subsection a few different methods are explored. The path planning functionality is a secondary project requirement, mentioned in SR1.

The Dijkstra Algorithm

The Dijkstra algorithm is a computer algorithm that can be used to find a solution to the single-source shortest paths problem between two points, s and v. It is based on creating an array of the available vertices $d[\,]$. This array is initially set to infinity (or a number sufficiently larger than any possible path length) for all vertices except $d[s] = 0$. It then performs n iterations (where n is available vertices) and selects an unmarked vertex with the lowest value, initially the starting position $d[s] = 0$,

and updates the close-by vertices and marks the already updated vertices [36]. An illustration of this can be seen in Figure 2.8.

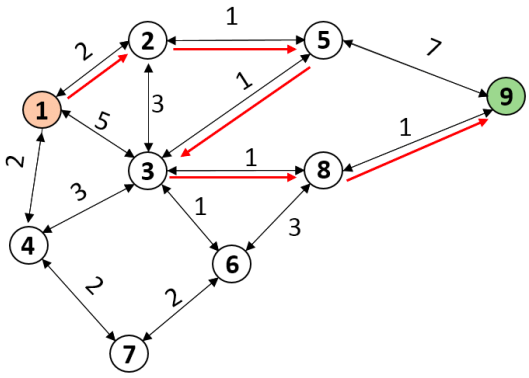

Figure 2.8: Illustration of Dijkstra's Algorithm [37]

The A* Algorithm

The A* Algorithm is an optimisation of the Dijkstra algorithm but instead of investigating all possible vertices and thus losing much time expanding outwards, it focuses on the ones closer to the actual target. The Dijkstra algorithm can be described as shown in Equation (2.1), where the function $g(n)$ is the cost of the path from the start point to any vertex n [38].

$$f(n) = g(n) \tag{2.1}$$

The A* algorithm is expanded with a heuristic estimated cost, $h(n)$, which estimates the cost from the vertex n to the goal. This can be seen in Equation (2.2).

$$f(n) = g(n) + h(n) \tag{2.2}$$

This heuristic cost can be estimated with multiple methods such as euclidean distance (real distance), Manhattan distance (only moving in four directions), and Chebyshev distance (moving in eight directions).

Artificial Potential Field

The artificial potential field is a path-finding method inspired by the concept of electrical charges [39]. The target is assumed to generate an attracting force field while obstacles generate a repulsive force field. These fields result in a total summed potential map which is used to obtain a path. However, it is common for the generated path to oscillate in constrained spaces due to opposing force fields, which can be solved by a low-pass filter.

2.3 Computer Vision

Computer vision is necessary for the prototype to recognise plastic bales, and therefore fulfill SR1. In this section, the main underlying theories for computer vision are discussed.

2.3.1 Image Processing

In order to implement computer vision, the computer has to process the images taken by the camera. Image processing as a computational technique has existed for a long time and has varied applications, such as digitisation, restoration, and segmentation [40].

Image processing is the method which transforms a camera image to a data representation. Moreover, working with real-time image processing creates constraints on image processing speed, since each frame has to be processed within a specific time frame [41].

Edge Detection

Edge detection is a well-known approach for recognising objects in an environment. By scaling down the data needed for processing, edge detection simplifies the image analysis [42]. As the name implies, it finds the edges or boundaries of objects inside images and distinguishes these. It is common to use edge detection for image segmentation in computer vision [43].

In edge detection, algorithms use the gradients of pixels in order to distinguish edges. The regions with the largest differences in gradient magnitude are then determined to be edges. A visual representation of edge detection can be seen in Figure 2.9.

Figure 2.9: Canny Edge Detection [43]

2.3.2 Image Recognition

Image recognition is another method by which to achieve computer vision. It represents a set of techniques for detecting and analysing images to enable the automation of a specific task. The three main tasks within the field of image recognition are [44]:

- Classification

  It is the identification of the ”class”, i.e. the category to which an image or the objects inside the image belong.

- Detection

  It is to perform classification and localisation. Detection is a technique to distinguish between objects in an image or video and use bounding boxes to show their location within that image/video.

- Segmentation

  It is to perform classification on each pixel in an image. The whole image is divided into pixels which can then be classified and labeled.

2.3.3 Artificial Neural Networks

An Artificial Neural Network (ANN), usually simply called Neural Network (NN), is a computing system architecture, seen in Figure 2.10, that mimics the biological neural networks of the human brain and its operation. In other words, a NN is a set of neurons organised in layers. Each neuron is a mathematical operation that takes its input, multiplies it by its weights, plus its biases, and then passes the sum to the other neurons with or without activation functions. Equation 2.3 shows a simple linear transformation of a neuron. A neural network learns how to classify an input by adjusting its weights and biases based on previous computations.

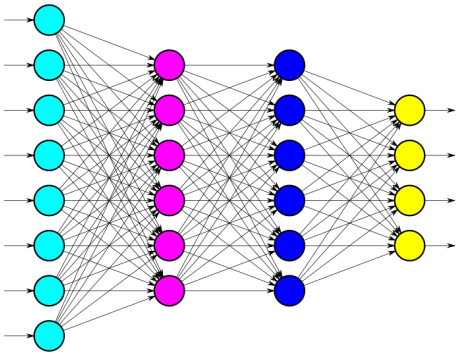

Figure 2.10: A fully connected artificial neural network [45]

Neurons

Neurons are elementary units in an artificial neural network [46].

Weights and Biases

Weights and biases are the learnable parameters of a machine learning model.

$$Output = Weight * Input + Bias \tag{2.3}$$

Activation Functions

Activation functions are required in deep neural networks to introduce non-linearity into the model [47]. The three most common activation functions are shown in Figure 2.11

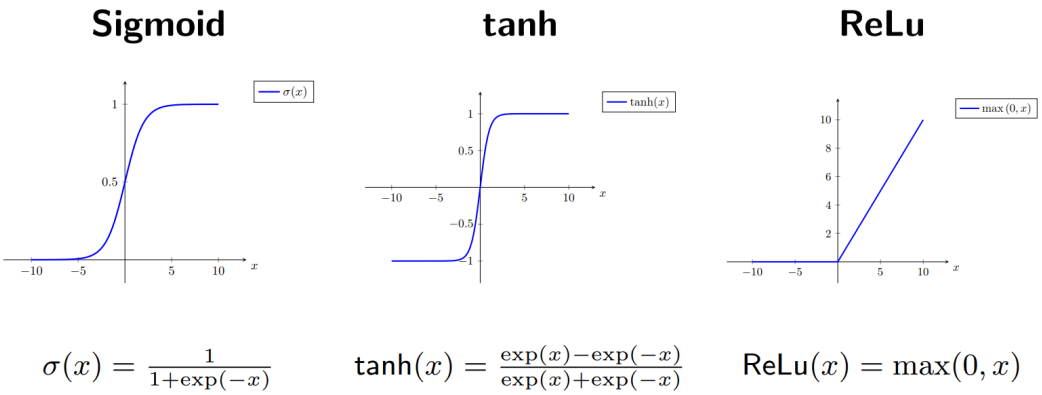

Figure 2.11: Activation functions [48]

Parameter Initialisation

There are two main methods for initialising the learnable parameters for the neural network:

1. Xavier Initialisation

   Xavier initialisation, generally used in combination with the *tanh* activation function, initialises the weights in the network with a small Gaussian value with zero mean and a specific variance σ based on fan_{in} (number of incoming neurons) and fan_{out} (number of outgoing neurons) [49]:

$$\sigma = \sqrt{\frac{1}{fan_{in} + fan_{out}}} \tag{2.4}$$

2. He Initialisation

   He initialisation, generally used with the ReLU activation function, is similar to Xavier initialisation but with the factor multiplied by two [50]:

$$\sigma = \sqrt{\frac{2}{fan_{in}}} \tag{2.5}$$

Mini-Batch Gradient Descent

Mini-batch gradient descent is an algorithm that splits the training datasets into small batches which are then used to calculate model error and update model parameters [51].

Stochastic Gradient Descent Optimiser

The purpose of an optimiser is to update the weights and biases of the neural network in order to minimise its loss (L). A vanilla Stochastic Gradient Descent (SGD) algorithm updates the weights (w) by subtracting the current weight by a factor (i.e. η, the learning rate) of its gradient, as seen in Equation: 2.6 [48]:

$$w_{new} = w - \eta \frac{\partial L}{\partial w} \tag{2.6}$$

Variations of this equation are commonly known as SGD optimisers. There are three main variations of how they differ [48]:

1. Adapt Gradient $\frac{\partial L}{\partial w}$
2. Adapt Learning Rate η
3. Adapt both Gradient $\frac{\partial L}{\partial w}$ and Learning Rate η

Some of the most popular variations of SGD optimisers can be seen below:

Table 2.1: Summary of SGD optimisers

Optimiser	Year	Learning Rate	Gradient
Momentum	1964		✓
AdaGrad	2011	✓	
RMSprop	2012	✓	
Adadelta	2012	✓	
Nesterov	2013		✓
Adam	2014	✓	✓
AdaMax	2015	✓	✓
Nadam	2015	✓	✓
Cyclical Learning Rate	2015	✓	
AMSGrad	2018	✓	

Batch Normalisation

Batch normalisation is a method of normalising the inputs at each layer which could reduce internal co-variate shift. According to the original paper that introduced this process, batch normalisation allows the model to have higher learning rates, be less sensitive to parameter initialisation, and gain a faster convergence rate [52].

Loss Functions

The loss function is a measure of the performance of a neural network model with respect to its given training sample and the expected output. Two common loss functions are discussed:

1. Mean Squared Error

   The Mean Squared Error (MSE) shows how close a regression line is to a set of points. The equation is defined in 2.7, where n is the number of input neurons, Y_i are the observed values, and $\hat{Y}_i$ are the predicted values.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2 \tag{2.7}$$

2. Cross-Entropy Loss

   Cross-entropy loss measures the performance of the classification model whose output is a probability between 0 and 1. It is defined in Equation 2.8, where (t_i, p_i) = *(truth label, softmax probability for the* i^{th} *class).*

$$L_{CE} = -\sum_{i=1}^{n} t_i \log(p_i) \tag{2.8}$$

Cost Function

A cost function is a loss function with the addition of a regularisation term. A regularisation term is added when computing the performance of a neural network model to prevent overfitting the network model during training [53]. It will increase the bias but decrease the variance of the model. An example of a cost function, cross-entropy loss with L2 regularisation, is defined in Equation 2.9.

$$J_{ridge}(\mathcal{D}, w, b) = \frac{1}{2}\sum_{(x,y)\in\mathcal{D}} L_{CE}(y, w^T x + b) + \lambda\|w\|^2 \tag{2.9}$$

Neural Network Comparison

ANNs can be either shallow or deep [54]. Networks are called shallow when they have only one hidden layer, and deep when the number of hidden layers is greater than one. Some of the most common neural network architectures are listed below

[55]. To make the distinction, the Deep Neural Network (DNN) is differentiated from the ANN.

- Artificial Neural Network (ANN)

  A shallow neural network model that only has one hidden layer.

- Deep Neural Network (DNN)

  A deep neural network model, also known as a feedforward neural network, that has more than one hidden layer.

- Convolutional Neural Network (CNN)

  The name CNN comes from the convolutional layers within the architecture. Unlike ANNs and DNNs, CNNs do not flatten the image for training, but instead capture the spatial and temporal dependencies in an image through the application of a sliding kernel/filter. This kind of neural network is the most popular in computer vision.

- Recurrent Neural Network (RNN)

  The name RNN comes from the recurrent layers in the architecture. Contrary to the models described above, RNN is time dependent. These networks are designed to process sequential data and is most popular within natural language processing.

2.3.4 Convolutional Neural Network

The process of a convolutional neural network can be separated into two parts - feature extraction and classification. This section presents some basic information regarding CNN components.

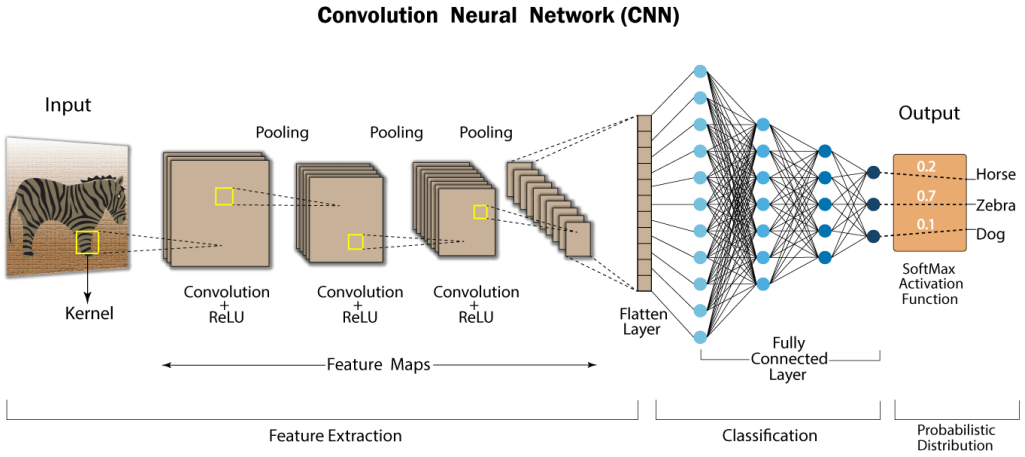

Figure 2.12: CNN structure [56]

Filters/Kernels

Filters or Kernels are used to extract features such as edges from the images. They are matrices that slide over the input data, perform element-wise multiplication within the sub-regions of the input data, and generate the output as matrices, which is called a "Feature Map". An example is shown in Figure 2.13.

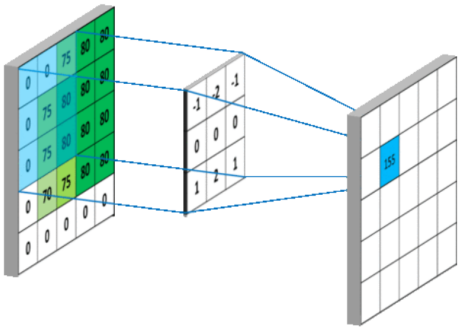

Figure 2.13: Filter and Feature Map [57]

Zero Padding

Zero padding is a kind of same-padding technique to preserve the original input size for the output of the convolutional layer. This technique adds a border of pixels all with zero value around the edges of the input images as shown in Figure 2.14.

0	0	0	0	0	0	0
0	1	1	1	0	0	0
0	0	1	1	1	0	0
0	0	0	1	1	1	0
0	0	0	1	1	0	0
0	0	1	1	0	0	0
0	0	0	0	0	0	0

Figure 2.14: Zero padding [58]

Pooling

There are two main types of pooling - max pooling and average pooling. Max pooling takes the maximum value from a specific region in the feature map, while average pooling takes the average as shown in Figure 2.15.

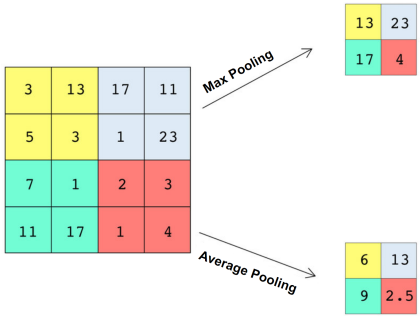

Figure 2.15: Max Pooling and Average Pooling

2.3.5 Data Augmentation

Neural networks typically require large, diversified datasets in order to efficiently train the network and accurately predict its classes. With an increase in complexity of the network (i.e. a larger number of hidden layers and therefore a larger number of hidden nodes), there is also an increase in the number of trainable parameters. These parameters are how the network maps the inputs (for example, an image of a dog) to the relevant outputs (i.e. the label ”dog”). With an increase in parameters, more data is needed to train them. However, oftentimes the amount of data is limited, which can result in an unregularised and therefore inaccurate network. To solve this problem, data augmentation can be introduced with which new data is synthesised by altering the data that is already available. This not only increases the dataset, but also diversifies it [59]. Some of the techniques for data augmentation of images include [60]:

- Geometric transformations
  - Scaling
  - Cropping
  - Flipping
  - Rotation
  - Translation
  - Gaussian Noise
- Photometric (colour) transformations
  - Altering brightness
  - Altering contrast
  - Altering saturation
  - Altering hue
  - Changing to grayscale

Some of these techniques are demonstrated in Figure 2.16.

Figure 2.16: Examples of image transformations used in data augmentation [59]

Utilising data augmentation techniques such as these has been proven to increase the accuracy of neural network classifications [61].

Chapter 3

Concept Design

In this chapter, the generated concepts of the prototype are discussed. The concepts are divided into two major systems - mechanical structure and software. The functional design will also be presented.

3.1 Functional Design

It is important to take the desired functionalities from the technical requirements into account when designing a concept. Some of the technical requirements have a direct influence on the hardware that is chosen, such as TR1c and TR4a, but other requirements like TR1a and TR5a describe functionalities of the prototype that can be achieved in many different ways and with the use of different hardware. It was carefully chosen, from the theory covered in the state of the art analysis, that a prototype wheel loader of a scale of roughly 1:10, including a chassis, lifting mechanism, propulsion system, steering mechanism and perception system, was to be developed. The technical and stakeholder requirements provide a list of functionalities for the prototype, including:

- An image recognition and image localisation system
- A navigation and localisation system
- A vehicle and actuator control system
- An energy storage
- A power management system

For the two first functionalities, it was decided to use the Nvidia Jetson AGX Xavier, due to the high demands on computational power that these functionalities require in order to function with minor latency, and the limitations of the available space on the prototype. Although vehicle and actuation control can be handled by the same processor, it was decided to develop those functionalities using separate hardware.

This benefits the prototype's modularity due to the introduced possibility to replace the hardware that is implemented for the autonomous system without affecting the vehicle's mechanical capabilities. The choice also benefits parallel development since the mechanical structure, the mechatronic systems, and the ROS systems can be tested individually during development, without the need of other systems' presence.

To justify how the systems would interconnect and communicate with each other, meet their power demands, and how the functionalities should be implemented, a functional architecture was developed early on and used as a guide for concept generation and hardware planning. The functional architecture can be seen in Figure A.1 in Appendix A, and illustrates how the systems are implemented. The systems are assigned to four different groups to describe the subsystems that can be developed in parallel and tested individually. Some of the hardware mentioned in the functional architecture was chosen through the help of a comparison process that is explained in the following section.

3.2 Comparison Matrix

The comparison matrix is a tool to visualise similarities and differences between simple products as well as complex and abstract concepts. The matrix helps to organise and classify the elements in which products are compared. The features and characteristics of each element are evaluated according to a set of criteria, enabling easy recognition of their advantages and disadvantages in order to bolster the decision making process.

To use this method, the following steps are necessary. Firstly, criteria are chosen for evaluation, then each concept receives scores in a specified scale for every criterion. In this project, the concepts are initially scored from 1 to 5 for each criterion, where 5 is the best. Thereafter, the criteria are compared to each other and ranked according to importance to weight their influence on the final result. The final step is to multiply the concepts' initial scores for every category by the weighted criteria to receive the final evaluated values for each concept.

3.3 Mechanical Structure

In this section, the design choices that have been made regarding the mechanical structure will be evaluated and explained. The concept design for the vehicle will be presented.

3.3.1 Chassis

The prototype will operate indoors with ideal environmental conditions, so the construction is not obligated to withstand weather conditions such as rain or sub-

zero temperatures. Therefore, the construction of the chassis will be focused on mechanical properties to withstand the loads that will act on it as well as ensuring proper dimensions to fit all components. Different concepts were taken into account and evaluated using the comparison matrix method. Three designs were considered:

- Customised stiff chassis
- Zeux concept chassis (scissor frame)
- Construction based on RC rally car's chassis available on the market

The designs were evaluated using these factors:

- Weight of the finished chassis assembly
- Reliability in the presence of unfavorable factors
- Complexity of the construction
- Cost, i.e. the expense of purchasing the necessary components and materials
- Accessibility of the assembly parts in the market

The result of this process is visible in Appendix B in Figure B.1 where, in the pentagonal graph, the value of each concept is presented. Figure B.1 shows that the best candidate is the stiff chassis concept, however this option lacks some core functionalities, such as the scissor feature for better versatility. Therefore, the Zeux frame is instead selected to better fulfill the scope of the project, since this will not induce any major influence on the cost and reliability. The final scores of the three concepts, with the full score calculations available in Table B.1, Appendix B, were:

Table 3.1: Final score of chassis evaluation

Type of chassis	Score
Customised stiff	11.5
Scissor frame concept	9.125
RC rally car's	6.375

The dimensions of the chassis frame are currently set to be 138 mm in width, 395 mm in length, and 65 mm in height. The main parts of the body are designed to be manufactured from steel sheet metal and connected with each other using conventional methods, such as screws or blind rivets. The design is presented in Figure 3.1.

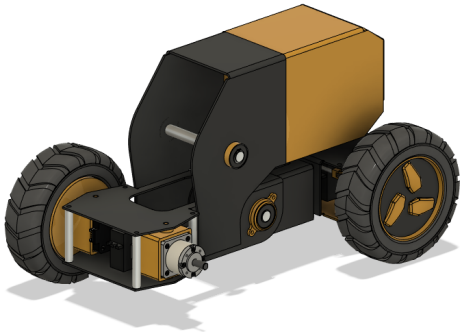

Figure 3.1: The chassis design

Scissor Frame Mechanism

The scissor frame mechanism is realised by joining two parts of the chassis to allow them to pivot. The front part is connected to the lifting mechanism and the main body. The rear part of the scissor mechanism is connected to the front of the chassis with a linear actuator responsible for lifting the construction and pivoting. This is demonstrated in Figure 3.2.

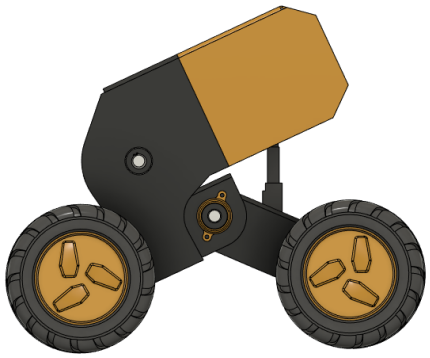

Figure 3.2: The design of the chassis with scissor frame

The scissor frame mechanism allows the vehicle to add functionalities such as:

- Adjusting the height of the bucket by moving the rear axle
- Adjusting the angle of the bucket to the ground
- Adjusting the height of the vehicle to navigate on rougher terrain

The scissor frame is a custom design for the purpose of this project, inspired by the Zeux concept.

3.3.2 Drive Train

In this subsection drive train components will be evaluated and with explanations of the functionalities that were considered. The solutions will be presented.

Wheels

The wheels will be designed to have enough elasticity to maintain constant contact with the ground. Therefore, no additional suspension will be needed. The choice of wheel is narrowed down to these three types:

- Air-filled rubber tires
- Airless rubber tires
- Custom hollow tires

Airless rubber tires were chosen due to their availability on the market, cost, and reliability.

Propulsion System

For the propulsion system, the options taken into consideration are two- or four-wheel drive (WD) and number of motors used. The systems were evaluated using these factors:

- Cost, i.e. the expense of purchasing the necessary components and materials
- Accessibility of the assembly parts in the market
- Weight/Dimension ratio, influence of the system on vehicle weight
- Controlability of the vehicle with the propulsion system
- Reliability in the presence of unfavorable factors

The evaluated results are visible in Table 3.2 and Figure B.2, with the score calculations available in Table B.2, Appendix B. These results show that the best solution is 2WD with two motors, one for each front wheel. This solution provides a substantial amount of power while maintaining a good grip to the ground due to the motors' placement on the front axle, where most of the vehicle's weight is focused. Therefore, a two motor solution is chosen due to the simplicity that it provides to the powertrain. It removes the necessity of using differential mechanisms, cardan joints, and other axles that would deliver momentum to the wheels. The differential steering is solved through adjusted control of the motor speed that takes into account the differences in wheel rotary speed when taking turns.

Table 3.2: Final score of propulsion system evaluation

Type of propulsion system	Number of motors	Score
2 WD	1	8.625
2 WD	2	10.125
4 WD	1	6.125
4 WD	2	7.125
4 WD	4	9.375

Propulsion Motors

Propulsion motors were selected to further develop the vehicle model. The potential candidates were brushless DC motors, stepper motors, and DC motors. An evaluation graph, shown in Figure B.3, Appendix B, was created to illustrate the decision process with the following criteria which are based on TR3a and SR6:

- Cost, i.e. the expense of purchasing the motor and corresponding parts
- Accessibility of the motor's mechanisms
- Power to weight ratio
- Complexity of implementation and control
- Reliability of the motor's behaviour

The candidates had a very similar performance in cost, accessibility, and power to weight ratio. The stepper motor is preferable in controlability whereas the brushless DC motor is preferable in reliability. The final, evaluated scores are presented in Table 3.3 and the calculations to determine these scores are presented in Appendix B, Table B.3.

Table 3.3: Final score of propulsion motor evaluation

Type of Motor	Score
Brushless DC motor	9.25
Stepper motor	9.625
DC motor	7.875

Conclusively, the stepper motors scored highest in the evaluation graph. The overall grading of stepper motors and brushless DC motors were approximately identical. The decision was made by assessing the significance of reliability and complexity. Simpler motors would simplify the embedded system control. On the other hand, reliable motors would provide a stable vehicle performance. The stepper motors could expedite the electronic design and control programming, whilst being a reliable solution. At the same time, the excess stability provided by the brushless DC motor

is redundant for this application. As a result of the above analysis, stepper motors were chosen as the propulsion motors.

Steering

Three main steering mechanisms were featured for matrix comparison: Double Ackermann, articulated steering, and mecanum wheels. Mecanum wheels were considered for the many functionalities that they provide without requiring mounting of an actual steering mechanism. The main flaw with this solution is that if there is waste or other obstacles on the ground, that could interfere with the mechanics of the wheels, hindering movement. Therefore, mecanum wheels were ruled out.

Articulated steering can provide the turning angle necessary to achieve prototype manoeuvrability and is highly popular in wheel loader constructions. However, since the bucket is mounted to the chassis, it would move along with the chassis while turning which makes the vehicle itself difficult to steer in cramped spaces. Meanwhile, Ackermann steering limits the movement of the bucket which is preferable in the context of confined spaces.

The designs were evaluated using these factors:

- Cost, i.e. the expense of purchasing the necessary components and materials
- Accessibility of the assembly parts in the market
- Weight/Dimension ratio, influence of the system on vehicle weight
- Controlability of the vehicle with the steering solution
- Reliability in the presence of unfavorable factors

The weighting diagram of the presented systems with the above criteria is located in Appendix B in Figure B.4. The evaluated scores of the steering solutions are shown in Table 3.4, and how they were calculated in Table B.4, Appendix B.

Table 3.4: Final score of steering mechanisms evaluation

Type of steering mechanisms	Score
Mecanum wheels	8.75
Ackermann steering	9.5
Articulated steering	8

To decrease the turning radius even further using the Ackermann solution, the mechanism can be applied to both front and rear axles. With one steering servomechanism for each axle and a system of rods which connect both wheels in one axle, the wheels are turned. The steering system is presented in Figure 3.3.

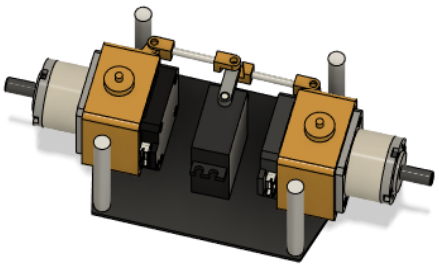

Figure 3.3: Design of the Ackermann steering mechanism

The minimum turning radius is designed to reach a value of 450 mm, as seen in Figure 3.4.

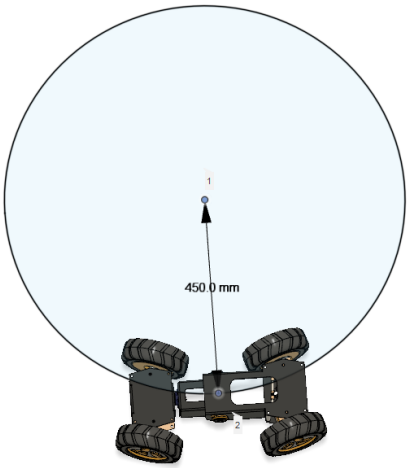

Figure 3.4: Turning radius of the double Ackermann solution, 450 mm

3.3.3 Lifting Mechanism

The potential lifting mechanisms and tool attachments of the prototype model were presented in Sections 2.1.2 through 2.1.3. The two lifting systems considered were kinematic linkages and sliding systems, which were examined with regards to the following criteria:

- Cost, i.e. expense of purchasing components and materials
- Accessibility of the assembly parts in the market
- Flexibility of the lifting mechanisms and motion speed
- Complexity of controlling the lifting mechanisms
- Reliability, i.e. accuracy and precision of lifting mechanisms

The final scores of the lifting mechanisms are tabulated in Table 3.5, and how this was calculated is shown in Table B.5, Appendix B.

Table 3.5: Final score of lifting mechanisms evaluation

Type of lifting mechanisms	Score
Sliding system	8.25
Kinematic linkage	11.75

The kinematic linkage mechanism has a higher score than the sliding system. This result was due to the lack of agility of the sliding system since a tilting cylinder would be needed to perform the loading and unloading motions. Even though the sliding system had a more straightforward control principle, this advantage was not significant. Correspondingly, the kinematic linkages were preferable over the sliding system and was therefore chosen as the final concept for continued work.

Tool Attachment

After deciding the lifting system's mechanical design, an appropriate tool attachment could be selected. A bucket would be the appropriate tool for the prototype model. This decision was made after discussions with the stakeholders, where the importance of collecting plastic bale fragments was emphasised. Among the three possible solutions, only the bucket has the ability to preserve the fragments. Tynes and bale grabbers are not able to achieve this purpose. Consequently, the bucket will be used in the project prototype.

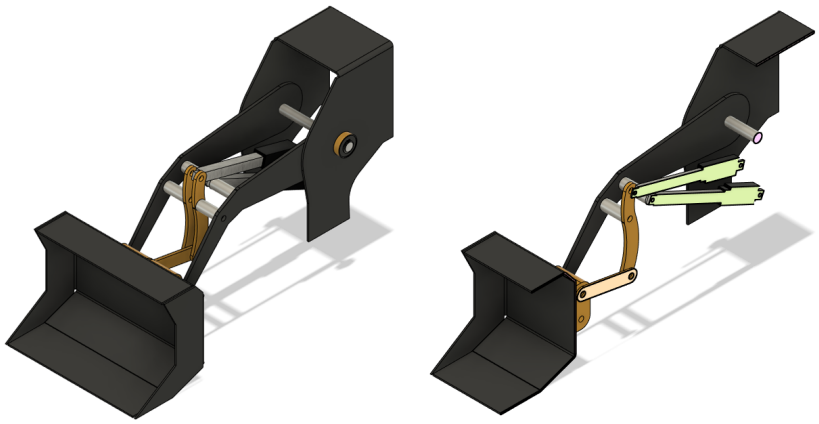

Figure 3.5: Design of the lifting mechanism

A CAD drawing of the current lifting mechanism design is shown in Figure 3.5. A customised bucket will be driven by Z-bar linkages. The Z-bar linkages will be composed of pins and customised metallic bars. Sidewise planer linkages are connected by cylindrical bars and pins. Two linear actuators are attached in the

mid-plane of the mechanical structure to drive the lifting and unloading motion, as described in Section 2.1.2. The bucket will be manufactured by sheet metal and linkage components have a simple geometry which will enable them to be easily machine-manufactured.

3.4 Software

In this section, the design choices that have been made with regards to the prototype's software implementation (navigation and image recognition) will be motivated and explained. This is in accordance with SR1 and SR2.

3.4.1 Navigation

Since the prototype is required to localise and orient itself, as stated in TR5a, high demands are put on its navigation system. In order for the prototype to perform the task, it has to build its own map of the environment in which it operates. Moreover, the prototype also has to localise the loading and unloading points for delivery of the plastic bales. To perform these tasks, a SLAM algorithm will be used with the operating system, ROS.

Mapping

In order to create its own map, the prototype will receive input from a camera and, using a SLAM algorithm, create a map of its surroundings, see Figure 3.6. For mapping, a LiDAR is often used because it provides a 360° input of the surroundings. However, a camera with stereo vision, such as Stereolabs' ZED 2 camera, is considered sufficient in this project, since the mapping will be constrained to a small environment.

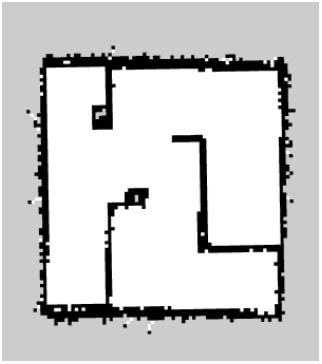

Figure 3.6: Map generated from SLAM based algorithm [62]

Localisation

The second crucial part of navigation will be localisation. The robot has to know where to pick up the plastic bales and where to drop them off, as well as its own position. The information about the robot's movement and position will come from a built-in IMU in the camera.

3.4.2 Image Recognition

For image recognition, detecting the plastic bales and differentiating their edges belongs to the segmentation category. The network will be developed in the programming language Python with the deep learning library PyTorch. To handle the processing power required to run image recognition with a CNN in real-time, a powerful GPU, such as the NVIDIA Xavier AGX 32GB, will suffice. Further information for the CNN solution is discussed below.

Convolutional Neural Network

In this project, the Convolutional Neural Network architecture is chosen. There are three main reasons why a CNN is preferred over a feed-forward neural network for image recognition:

1. Using Spatial Structure

   A fully-connected feed-forward neural network learns by flattening out an image into vectors, which at the same time diminishes the spatial structure of an image. This leads to a low accuracy of prediction, and little to zero accuracy in cases of complex images with pixel dependencies. The usage of a sliding window in CNN however, can capture the spatial and temporal dependencies in an image since each neuron only connects to its receptive field.

2. Parameter Sharing and Local Connectivity

   The number of parameters in a neural network grows rapidly with an increase in the number of hidden layers. In a fully-connected feed-forward neural network, it would require a very high number of neurons even in a shallow architecture. This architecture is generally impractical for larger inputs such as high resolution images. Comparing to that, the architecture of a CNN successfully tackles this issue through parameter sharing and local connectivity.

   - Parameter Sharing

     The vectors of weights and biases are called filters and represent particular features of the input. Parameter sharing is the sharing of the same filters between many neurons. This reduces the memory required during training since these filters are used across all receptive fields that share the filter, as opposed to each receptive field having its own weights and biases vector in a fully-connected feed-forward neural network.

   - Local Connectivity

     Local connectivity is the concept of each convolutional neuron connecting and processing data only for its receptive field (subset of the input image), unlike a fully-connected feed-forward neural network where all the neurons are fully connected.

3. Reduce Model Overfitting

   With less parameters needed in a CNN, the possibility of overfitting the model is therefore reduced.

For the design of this project's convolutional neural network architecture, it was decided to implement He initialisation to avoid vanishing and/or exploding gradient problems, ReLU as the activation function, zero padding technique when performing convolution, cross-entropy loss with L2 regularisation as the cost function, and Adaptive Moment Estimation (Adam) as the optimiser. It was also decided to utilise batch normalisation to accelerate the training process and to use mini-batch gradient descent during training.

Choice of Activation Function

There are three reasons why a rectified linear unit (ReLU) is chosen for the activation function.

1. Vanishing Gradients Problem

   As opposed to the sigmoid and tanh activation functions, ReLU does not saturate large positive numbers which may otherwise cause the vanishing gradient problem when a network increases in depth.

2. Computation Efficiency

   The ReLU activation function is relatively cheap to compute when compared to sigmoid and tanh.

3. Faster Convergence

   In practice, the training of a ReLU network converges much faster than the sigmoid/tanh activation functions.

Dataset

Because no repository currently exists of labelled plastic bale images, the dataset for this project will have to be created independently. This will have to be done by collecting images online. Due to the time-consuming nature of this data collection, data augmentation, with geometric and photometric transformations, will be utilised to both expand and diversify the dataset.

Chapter 4

Future Work

In this chapter, the time plan, budget, and team organisation is presented.

4.1 Time Plan

The project is divided into four different phases, where each phase consists of multiple tasks for each sub team. The primary goal of the first half of the project is to have all materials and components ordered and received by the end of Phase 2. In the beginning of Phase 3, each sub team will start working on their own designated tasks. All developments are to be completed by the end of Phase 3. Two weeks at the end of Phase 3 are to be reserved for system testing, validation, verification, and improvements. Phase 4 lasts for one month and will consist of preparing the final presentation and report. An overview of the project plan is shown in Table 4.1 and an in-depth project plan is presented in a GANTT chart in Appendix B.

Table 4.1: Project phases overview

Phase	Time	Description
Phase 1	31 Mar - 31 May	Spring report, CAD model, Purchase Order, Concept design
Phase 2	1 Jun - 22 Aug	Ordering materials and components
Phase 3	23 Aug - 16 Nov	Mechanical, Mechatronics, and Software system construction
Phase 4	17 Nov - Final	Final report, presentation slides

4.2 Budget

As stated in the stakeholder requirements, found in Section 1.3, the given budget of 50 000 SEK should not be exceeded. In order to ensure that the budget is not exceeded, a cost analysis, found in Appendix D, was made. In the cost analysis, all parts and materials are listed including the quantity of each component. The estimated price for each segment is based on retailers' price found online. The price for each segment was summed up to give a total cost estimation.

4.3 Organisation

The project team consists of stakeholders from Volvo CE, one team coach, and nine team members. The team is separated into three subteams, namely the Mechanical, Mechatronics, and Software teams, as shown in Figure 4.1. The Mechanical team will be in charge of the manufacturing of components and overall vehicle construction. The Mechatronics team will be responsible for the electrical harness, the control of motors and actuators, and the vehicle control system. The Software team will manage the development of the neural network, mapping, and localisation algorithms.

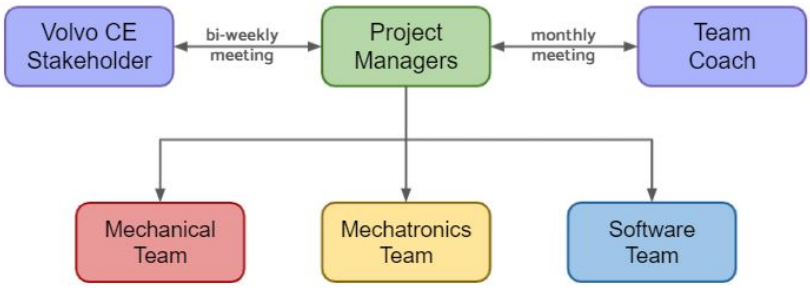

Figure 4.1: Team structure overview

4.3.1 Communication and Project Management

A bi-weekly meeting with the stakeholders and a monthly meeting with the team coach are held via the meeting platform Zoom. The communication and project management platforms used within the team are listed in Table 4.2.

Table 4.2: Communication and project management overview

Purpose	Chosen Platform
Project management platform	Projectplace
Documentation storage	Google Shared Drive
Meeting scheduling platform	Google Calendar
Communication platform	Slack, Messenger
Virtual meeting platform	Zoom

Bibliography

[1] *Plastic in Sweden: Facts and Practical Advice*. 2019. URL: http://naturvardsverket.se/Documents/publ-filer/978-91-620-8854-5.pdf?pid=26005.

[2] *What is Waste Incineration?* URL: https://www.conserve-energy-future.com/advantages-and-disadvantages-incineration.php. (Accessed: 19.05.2021).

[3] *Volvo CE and LEGO Technic Introduce Zeux*. URL: https://www.volvoce.com/global/en/this-is-volvo-ce/what-we-believe-in/innovation/zeux/. (Accessed: 19.05.2021).

[4] Shravan H. Gawande, A. A. Muley, Rahul N. Yerrawar. 'Optimization of Torsional Stiffness for Heavy Commercial Vehicle Chassis Frame'. In: (2018). DOI: 10.1007/s42154-018-0044-6.

[5] Anh-Tuan Dang, Dinh-Ngoc Nguyen, Dang-Hao Nguyen. 'A Study of Scissor Lifts Using Parameter Design'. In: (2020).

[6] *Lego Zeux concept*. URL: https://www.lego.com/pl-pl/kids/sets/technic/volvo-concept-wheel-loader-zeux-88a0d74a84374016a6b37363d0c41cc4. (Accessed: 10.05.2021).

[7] Sameer Prabhu, Jeffrey Wendlandt. 'Multi-Domain Modeling and Simulation of an Electro-Hydraulic Implement System'. In: (2006). DOI: 10.4271/2006-01-3490. URL: https://www.researchgate.net/publication/268378859_Multi-Domain_Modeling_and_Simulation_of_an_Electro-Hydraulic_Implement_System.

[8] Torcan life equipment. *What is forklift? Working mechanism where is it used?* 2020. URL: https://torcanlift.com/what-is-forklift-working-mechanism-where-it-is-used/. (Accessed: 19.05.2021).

[9] Prodig. *Bale Grab*. 2021. URL: http://www.prodigattachments.com/en/products/agricultural/6-bale-grab. (Accessed: 19.05.2021).

[10] *Grading Bucket*. URL: https://www.volvoce.com/europe/en/attachments/wheel-loader-attachments/loader-buckets/grading-lwl/. (Accessed: 19.05.2021).

[11] *Forklift Forks*. URL: https://www.liftruck.co.uk/shop/forklift-forks/forklift-forks.html. (Accessed: 19.05.2021).

[12] *1803 mm (71 In) Bale grab.* URL: https://www.warrencat.com/new-equipment/attachments/bale-grabs/1803-mm-71-in-bale-grab/. (Accessed: 19.05.2021).

[13] Manfred Harrer, Peter Pfeffer. *Steering Handbook*. Springer International Publishing AG Switzerland. ISBN: ISBN 978-3-319-05448-3. DOI: 10.1007/978-3-319-05449-0.

[14] W. A. Wolfe. 'Analytical Design of an Ackermann Steering Linkage'. In: (1959).

[15] Quan Qiua, Zhengqiang Fanb, Zhijun Menga, Qing Zhangb, Yue Conga, Bin Lia, Ning Wangc, Chunjiang Zhaod. 'Extended Ackerman Steering Principle for the coordinated movement control of a four wheel drive agricultural mobile robot'. In: (2018).

[16] Wikipedia. *Ackermann Steering Geometry.* URL: https://en.wikipedia.org/wiki/Ackermann_steering_geometry. (Accessed: 19.05.2021).

[17] Stephen L. Dickerson, Brett D. Lapin. 'Control of an Omni-Directional Robotic Vehicle with Mecanum Wheels'. In: (1991).

[18] Patent Yogi. *Reinventing wheels - Mecanum wheels that can move a vehicle in any direction.* URL: https://patentyogi.com/american-inventor/reinventing-wheels-mecanum-wheels-that-can-move-a-vehicle-in-any-direction/. (Accessed: 19.05.2021).

[19] Matthias Felden, Patrick Bütterling, Peter Jeck, Lutz Eckstein, Kay Hameyer. 'Electric vehicle drive trains: From the specification sheet to the drive-train concept'. In: (2010). DOI: 10.1109/EPEPEMC.2010.5606531.

[20] Claudio Annicchiarico , Mirko Rinchi , Stefano Pellari , Renzo Capitani. 'Design of a Semi Active Differential to Improve the Vehicle Dynamics'. In: (2014).

[21] David J Purdy, Dave Simner. 'Brief investigation into the effect on suspension motions of high unsprung mass'. In: *Journal of Battlefield Technology* (2004). ISSN: 1440-5113.

[22] Professor Dr.-Ing. habil. Hartmut Janocha (Ed.) *Actuators, Basics and Applications.* Universtat des Saarlandes. DOI: 10.1007/978-3-662-05587-8.

[23] ROS. *About ROS.* URL: https://www.ros.org/about-ros/. (Accessed: 19.05.2021).

[24] ROS. *Packages.* URL: https://wiki.ros.org/Packages. (Accessed: 19.05.2021).

[25] NVIDIA. *What Is Simultaneous Localization and Mapping?* URL: https://blogs.nvidia.com/blog/2019/07/25/what-is-simultaneous-localization-and-mapping-nvidia-jetson-isaac-sdk/. (Accessed: 19.05.2021).

[26] Veli Ilci Andand, Charles Toth. 'High Definition 3D Map Creation Using GNSS/IMU/LiDAR sensor Integration to Support Autonomous Vehicle Navigation'. In: (2019). DOI: 10.3390/s20030899.

[27] Xiaofeng Ren, Dieter Fox, Kurt Konolige. 'Change Their Perception'. In: (2013). DOI: 10.1109/MRA.2013.2253409.

[28] A. Kolb, R. Koch, R. Larsen. 'Time-of-Flight Cameras in Computer Graphics'. In: (2010). DOI: https://doi.org/10.1111/j.1467-8659.2009.01583.x.

[29] Manaf A. Mahammed, Amera I. Melhum, Faris A. Kochery. 'Object Distance Measurement by Stereo VISION'. In: *International Journal of Science and Applied Information Technology (IJSAIT)* 2.2 (2013). ISSN: 2278-3083.

[30] You li, Javier Ibanez-Guzman. 'Lidar for Autonomous Driving: The Principles, Challenges, and Trends for Automotive Lidar and Perception Systems'. In: (2020). DOI: 10.1109/MSP.2020.2973615.

[31] M. Kutila, P. Pyykönen, et al. 'Automotive LiDAR performance verification in fog and rain'. In: (2018). DOI: 10.1109/ITSC.2018.8569624.

[32] Norhafizan Ahmad, Raja Ariffin Raja Ghazilla, et al. 'Reviews on Various Inertial Measurement Unit (IMU) Sensor Application'. In: *International Journal of Signal Processing Systems* 1.2 (2013). DOI: 10.12720/ijsps.1.2.256-262.

[33] Daehee Won, Jongsun Ahn, et al. 'Performance Improvement of Inertial Navigation System by Using Magnetometer with vehicle Dynamic Constraints'. In: (2015). DOI: 10.1155/2015/435062.

[34] Mohammad O. A. Aqel, Mohammmad H. Marhaban, et al. 'Review of visual odometry: types, approaches, challenges, and applications'. In: (2016). DOI: 10.1186/s40064-016-3573-7.

[35] Ji Zhang, Sanjiv Singh. 'LOAM: Lidar Odometry and Mapping in Real-time'. In: (2014). DOI: 10.15607/RSS.2014.X.007.

[36] E.W Dijkstra. 'A note on two problems in connexion with graphs'. In: (1959). DOI: 10.1007/BF01386390.

[37] Atta Ur Rehman et al. 'Emergency Evacuation Guidance System for Underground Miners'. In: (Feb. 2019).

[38] Blake Lin. *A* Pathfinding Algorithm in 2D Grid*. URL: https://medium.com/akatsuki-taiwan-technology/a-pathfinding-algorithm-in-2d-grid-11f5a5354cc2. (Accessed: 19.05.2021).

[39] M.Z Azmi, T Ito. 'Artificial Potential Field with Discrete Map Transformation for Feasible Indoor Path Planning'. In: (2020). DOI: 10.3390/app10248987.

[40] AvMaria M. P. Petrou, Costas Petrou. *Image Processing: The Fundamentals*. John Wiley Sons Ltd. ISBN: 978-0-470-74586-4.

[41] Xavier P., Burgos-Artizzu, et al. 'Real-time image processing for crop/weed discrimination in maize fields'. In: 75 (2011).

[42] John Canny. 'A Computational Approach to Edge Detection'. In: Pami-8.6 (1986).

[43] MathWorks. *Edge Detection*. URL: https://se.mathworks.com/discovery/edge-detection.html. (Accessed: 19.05.2021).

[44] Pulkit Sharma. *Image Classification vs. Object Detection vs. Image Segmentation*. 2019. URL: https://medium.com/analytics-vidhya/image-classification-vs-object-detection-vs-image-segmentation-f36db85fe81. (Accessed: 19.05.2021).

[45] DeepLizard. *Machine Learning Deep Learning Fundamentals*. URL: https://deeplizard.com/learn/video/FK77zZxaBoI. (Accessed: 20.05.2021).

[46] Rami A. Alzahrani. 'Neuromorphic Circuits With Neural Modulation Enhancing the Information Content of Neural Signaling'. In: (2020).

[47] Chigozie Enyinna Nwankpa. 'Activation Functions: Comparison of Trends in Practice and Research for Deep Learning'. In: (2018).

[48] Josephine. *KTH DD2424 course slides*. URL: https://www.kth.se/student/kurser/kurs/DD2424?l=en. (Accessed: 20.05.2021).

[49] Xavier Glorot, Yoshua Bengio. 'Understanding the difficulty of training deep feedforward neural networks'. In: (2010).

[50] He. 'Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification'. In: (2015).

[51] Sebastian Ruder. 'An overview of gradient descent optimization algorithms' In: (2020).

[52] Sergey Ioffe, Christian Szegedy. 'Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift'. In: (2015).

[53] Jan Kukačka. 'Regularization for Deep Learning: A Taxonomy'. In: (2017).

[54] Saad Albawi. 'Understanding of a Convolutional Neural Network'. In: (2017). DOI: 10.1109/ICEngTechnol.2017.8308186.

[55] Aravind Pai. *CNN vs. RNN vs. ANN – Analyzing 3 Types of Neural Networks in Deep Learning*. 2020. URL: https://www.analyticsvidhya.com/blog/2020/02/cnn-vs-rnn-vs-mlp-analyzing-3-types-of-neural-networks-in-deep-learning/. (Accessed: 19.05.2021).

[56] Patent Yogi. *Convolutional Neural Network — Deep Learning*. URL: https://developersbreach.com/convolution-neural-network-deep-learning/. (Accessed: 20.05.2021).

[57] MLNotebook. *Machine Learning Deep Learning Fundamentals*. URL: https://mlnotebook.github.io/post/CNN1/. (Accessed: 20.05.2021).

[58] 호롤리. *호다닥 공부해보는, CNN (Convolutional Neural Networks)*. URL: https://gruuuuu.github.io/machine-learning/cnn-doc/. (Accessed: 20.05.2021).

[59] Great Learning Team, Arun K. *Understanding Data Augmentation — What is Data Augmentation and how it works?* 2020. URL: https://www.mygreatlearning.com/blog/understanding-data-augmentation/. (Accessed: 19.05.2021).

[60] Shorten C., Khoshgoftaar T.M. 'A survey on Image Data Augmentation for Deep Learning'. In: 6.60 (2019). DOI: https://doi.org/10.1186/s40537-019-0197-0.

[61] J. Wang, L. Perez. 'The Effectiveness of Data Augmentation in Image Classification using Deep Learning'. In: (2017). URL: https://arxiv.org/pdf/1712.04621.pdf.

[62] *Turtlebot3: Slam.* URL: https://emanual.robotis.com/docs/en/platform/turtlebot3/slam/. (Accessed: 21.05.2021).

Appendix A

Functional Architecture

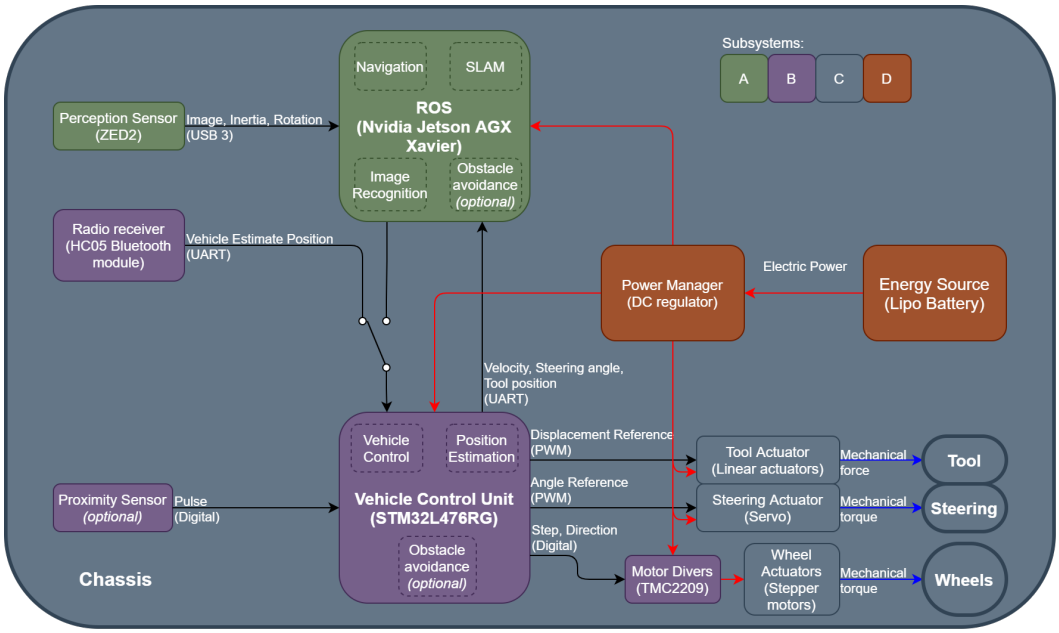

Figure A.1: Functional architecture of the prototype, with the functions divided into four different categories for subsystem identification

Appendix B

Evaluation of the design concepts

Table B.1: Weighted evaluation of the chassis concepts

		Zeux chassis		Stiff custom chassis		RC car chassis	
Criteria	Criteria Weight	Original Score	Weighted Score	Original Score	Weighted Score	Original Score	Weighted Score
Weight	25%	3	0,75	4	1	5	1,25
Reliability	87.5%	4	3.5	5	4.375	3	2.625
Complexity	75%	4	3	5	3.75	2	1.5
Cost	50%	3	1.5	4	2	1	0.5
Accessibility	12.5%	3	0.375	3	0.375	4	0.5
Weighted score for each concept			**9.125**		**11.5**		**6.375**

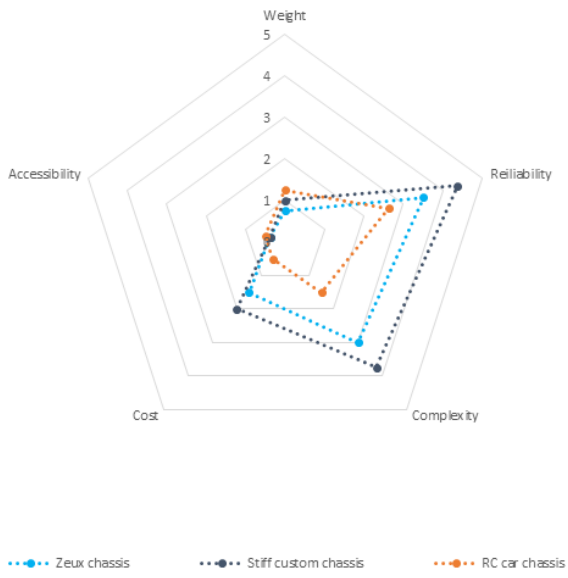

Figure B.1: Chassis concept evaluation diagram

Table B.2: Weighted evaluation of the motor drive type concepts

		2WD,1M		2WD,2M		4WD,1M		4WD,2M		4WD,4M	
Criteria	Criteria Weight	Original Score	Weighted Score	Original Score	Weighted Score	Original Score	Weighted Score	Original Score	Weighted Score	Original Score	Weighted Score
Cost	37.5%	4	1.5	3	1.125	3	1.125	2	0.75	2	0.75
Accessibility	0%	3	0	5	0	1	0	2	0	4	0
Weight/Dimension	37.5%	5	1,875	3	1,125	4	1,5	3	1,125	2	0,75
Controlability	87.5%	3	2,625	5	4,375	2	1,75	3	2,625	4	3,5
Reliability	87.5%	3	2,625	4	3,5	2	1,75	3	2,625	5	4,375
Weighted score for each concept			**8.625**		**10.125**		**6.125**		**7.125**		**9.375**

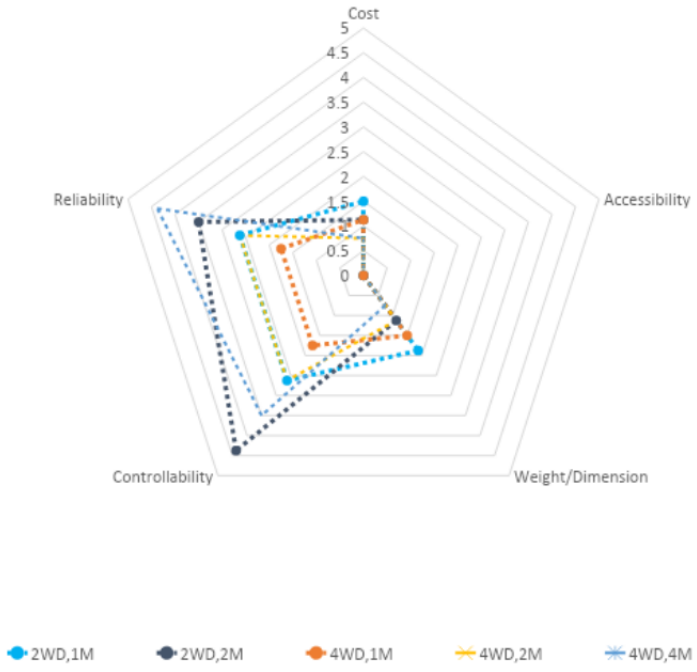

Figure B.2: Motor drive type concept evaluation diagram

Table B.3: Weighted evaluation of the propulsion motor types

		Brushless DC motor		Stepper motor		DC motor	
Criteria	Criteria Weight	Original Score	Weighted Score	Original Score	Weighted Score	Original Score	Weighted Score
Cost	37.5%	2	0.75	4	1.5	3	1.125
Accessibility	12.5%	2	0.25	5	0.625	4	0.5
Power/Weight	25%	5	1.25	2	0.5	4	1
Complexity	87.5%	3	2.625	5	4.375	3	2.625
Reliability	87,5%	5	4.375	3	2.625	3	2.625
Weighted score for each concept			**9.25**		**9.625**		**7.875**

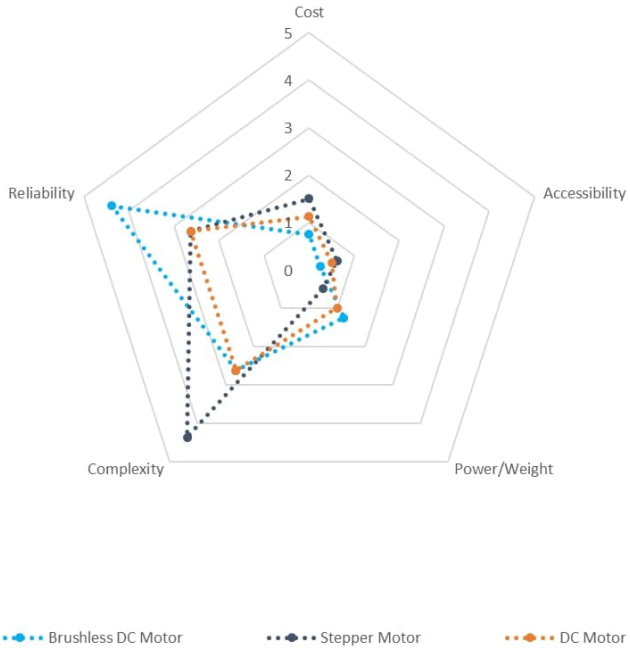

Figure B.3: Weighted propulsion motor evaluation diagram

Table B.4: Weighted evaluation of the steering concepts

		Mecanum wheels		Ackermann steering		Articulated steering	
Criteria	Criteria Weight	Original Score	Weighted Score	Original Score	Weighted Score	Original Score	Weighted Score
Cost	25%	2	0.5	4	1	4	1
Accessibility	50%	5	2.5	3	1.5	3	1.5
Weight/Dimension	0%	3	0	3	0	3	0
Controlability	75%	5	3.75	4	3	2	1.5
Reliability	100%	2	2	4	4	4	4
Weighted score for each concept			**8.75**		**9.5**		**8**

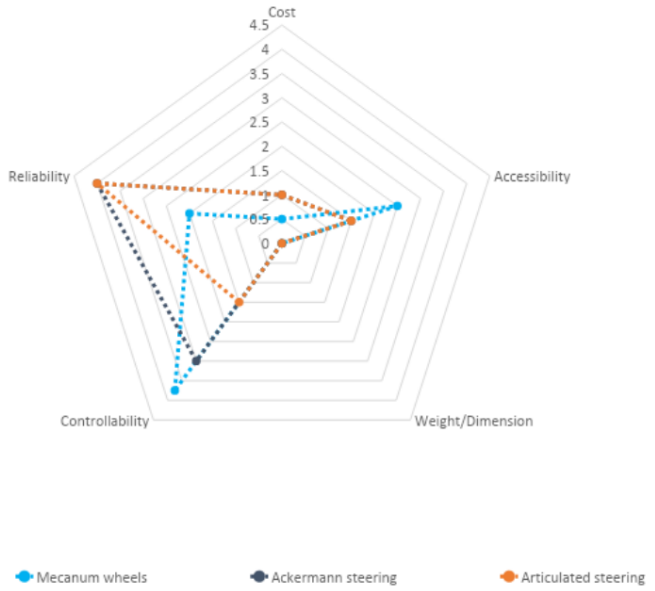

Figure B.4: Steering concept evaluation diagram

Table B.5: Weighted evaluation of the lifting mechanism concepts

		Sliding system		Kinematic linkage	
Criteria	Criteria Weight	Original Score	Weighted Score	Original Score	Weighted Score
Cost	37.5%	3	1.125	4	1.5
Accessibility	12.5%	5	0.625	4	0.5
Flexibility	87.5%	3	2.625	5	4.375
Complexity	25%	5	1.25	4	1
Reliability	87.5%	3	2.625	5	4.375
Weighted score for each concept			**8.25**		**11.75**

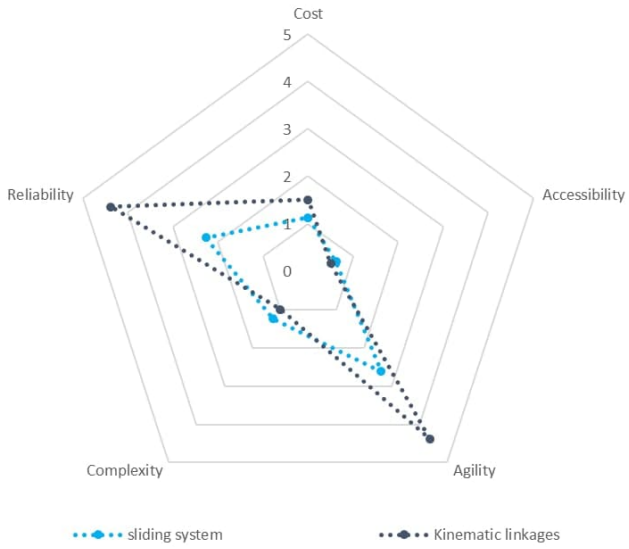

Figure B.5: Weighted lifting system evaluation diagram

Appendix C

GANTT chart

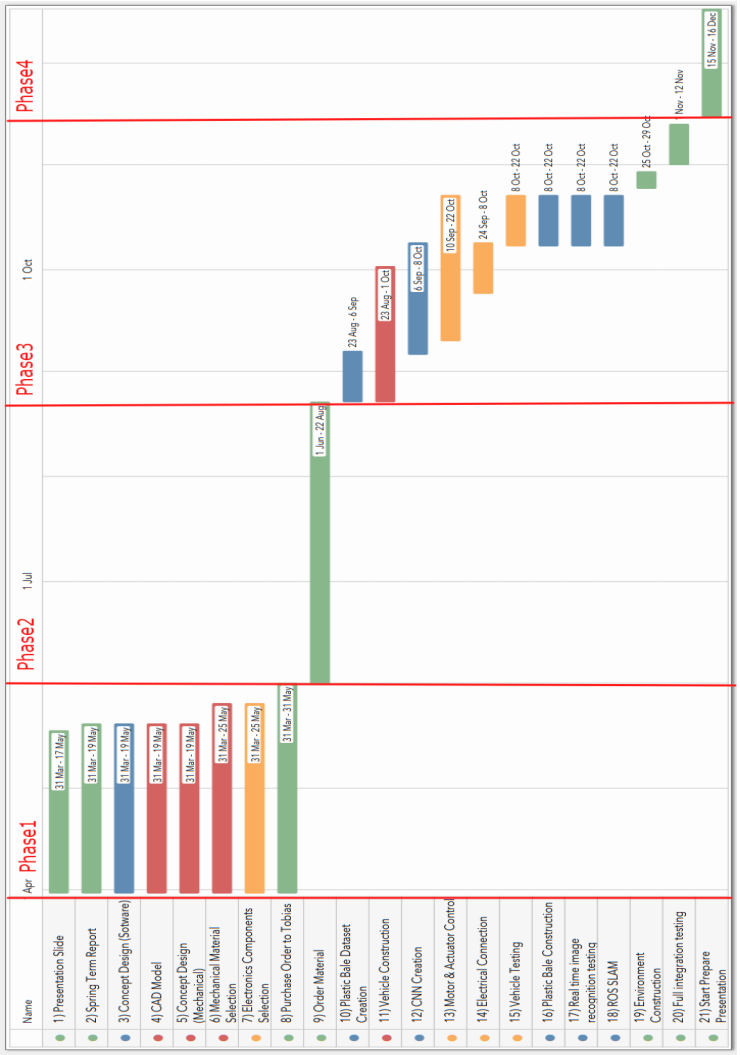

Figure C.1: GANTT chart

Appendix D

Budget

Part No.	Part name	Subgroup	Notes	Unit	Quantity	Price/unit (SEK)	Price total (SEK)	Link	Article No.	Store
1	NUCLEO-L476RG	Mechatronics	Microcontroller		2	124,25	248,5	link	497-15881-ND	Digikey
2	Nvidia Jetson AGX Xavier	Software	GPU		1	5828,74	5828,74	link	945-82972-0045-	Arrow
3	ZIPPY Compact 8000mAh 5	Mechatronics	Battery		2	763,04	1526,08	link	9067000489-0	Hobbyking
4	Zed2	Software	Camera, IMU		1	3740	3740	link		Stereolabs
5	L16-R Miniature Linear Serv	Mechatronics	3 or 4 Qty		4	581,01	2324,04	link	L16-R	Actuonix
6	Kitchen scale	Mechatronics	Load cell		1	99	99	link	1006334	Netonenet
7	Luxorparts Delbar kopplings	Mechatronics	Jump Wires, male-female		1	89,9	89,9	link	87900	Kjell
8	Luxorparts Delbar kopplings	Mechatronics	Jump Wires, female-female		1	89,9	89,9	link	87906	Kjell
9	Breadboard Jumper Wire Kl	Mechatronics	Breadboard kit		1	117	117	link	301-15-119	Elfa
10	Nema 17 Stepper Bipolar L4	Mechatronics	Stepper motor		2	216,51	433,02	link	17HS13-0404S-(	Stepperonline
11	Oisolerade DC/DC-omvandl	Mechatronics	DC regulator, 6A		1	132,27	132,27	link	894-APXW012A	Mouser
12	TMC2209 STEPPER MOTO	Mechatronics	Stepper driver TMC2209		1	277,22	277,22	link		
13	Sheet metal (steel)	Mechanics	Check if KTH has	2,0 x 1000 x 500	2	255	510	link	503105	Montano
14	Digital Servo 20kg / 0.16sec	Mechatronics			2	203,51	407,02	link	068000026-0	Hobbyking
15	Nylon XT90 Connectors Mal	Mechatronics	XT90 connectors (male)		1	29,98	29,98	link	015000267-0	Hobbyking
16	F7010 - Fuse 10A 32V Red,	Mechatronics	Fuse, 10A		25	0,8982	22,455	link	301-71-841	Elfa
17	H7810 - Fuse Holder MiniO	Mechatronics	Fuse holder, MiniOTO		3	15,6	46,8	link	110-29-328	Elfa

Budget:	50000	SEK
Parts cost:	15921,925	SEK
Shipping cost:	0	SEK
Total Cost:	15921,925	SEK
Remaining Budget:	34078,075	SEK

Figure D.1: The components and costs in the budget calculation