ÇANKAYA UNIVERSITY
FACULTY OF ENGINEERING
COMPUTER ENGINEERING DEPARTMENT

Project Report
Version 1

CENG 408
Innovative System Design and Development II

TEAM 8
Transporter Rover

Ahmet Selim Yalçın
201817030
Mustafa Furkan Eroğlu
201817252
Barış Berker Bayram
201811012
Burçak Dolkun
201815020

Advisor: *Serdar ARSLAN*

Table of Contents

List of Figures

List of Tables

Abstract

The Automated Guided Vehicle (AGV), first developed in 1953, is a driverless transport vehicle utilized for horizontal movement. AGVs are powered by batteries and centrally controlled by computer systems. They are frequently employed in industrial settings such as production facilities and warehouses. When multiple AGVs operate in coordination, they constitute an Automated Guided Vehicle System (AGVS).

Our project **RASIM** (Robotic and Automation System in Manufacturing) aims to transport bulky and heavy items within a warehouse, enhancing Industry 4.0 applications and producing such products domestically. To ensure the safe operation of Automated Guided Vehicles (AGVs), this article introduces a visual surveillance system that fully utilizes measurements from forward and downward-facing cameras.

The AGV's forward camera estimates its position and orientation by tracking landmarks in the detected environment using advanced image sequences. This system also enables us to employ the mapping capabilities provided by the Robot Operating System (ROS) software. Additionally, the downward-facing camera is utilized to detect QR codes affixed to the ground. These QR codes help determine the AGV's pose within an absolute reference frame.

By integrating ROS software and image processing techniques, the detected QR codes guide the AGV to the most accurate position within a virtual shipping area we create. This integration ensures precise navigation and enhances the overall efficiency and safety of AGV operations in the warehouse.

1. INTRODUCTION

Automated Guided Vehicles (AGVs) represent a critical advancement in industrial automation, enhancing efficiency and accuracy in material handling processes. AGVs are self-propelled vehicles that transport materials in various settings, such as manufacturing floors, warehouses, and distribution centers. Their key features include autonomy, precision, and the ability to operate without human intervention. In the context of Industry 4.0, AGVs play a vital role in streamlining operations and reducing manual labor.

This project focuses on developing an AGV system designed to operate within a controlled environment featuring three key terminals: a parking station, a product pickup point, and a drop-off point. The AGV is equipped with a line-following sensor to navigate predetermined paths accurately. The system is designed to perform automated material handling tasks efficiently using advanced sensors and control systems.

At the product pickup point, the AGV aligns itself precisely using a platform mechanism to collect the designated item. The AGV then proceeds to the drop-off point, where it deposits the item with similar precision. To ensure smooth and accurate entry and exit from the terminals, the AGV employs image processing techniques facilitated by the Nvidia Jetson Nano. This sophisticated image processing capability allows the AGV to detect and respond to visual cues, ensuring it performs tasks accurately and efficiently.

The parking operation is another critical aspect of the project. The AGV is expected to navigate to the parking station autonomously and perform the parking maneuver without any external guidance. This involves sophisticated path planning and control algorithms to ensure that the vehicle parks in the designated spot correctly.

In summary, this project aims to develop an advanced AGV system capable of performing essential material handling tasks autonomously. By integrating line-following sensors, image processing, and precise control mechanisms, the AGV will enhance efficiency in industrial environments. The successful implementation of this project will demonstrate the potential of AGVs in automating routine tasks, reducing labor costs, and increasing operational efficiency.

Automated guided vehicle (AGV), It provides benefits in carrier loading. It is possible to travel along the road. In warehouses [1], [2]. General scenarios of AGV Includes transportation of raw materials, semi-finished products production lines, storage, and collection operations storage and distribution [3], [4]. In normal AGV path, AGV may deviate gradually. Deviation from the expected path due to cyber-attacks, wireless command blocks, and untimely adjustments are a few of them [5]. This deviation can cause collisions between AGVs and cargoes, the chaos of transportation, the entire production process and even disaster losses of warehouse personnel [6]. Thus, it is necessary to build an AGV surveillance system to estimate the AGV trajectories online and issue an alarm in the presence of path deviations. Generally, the AGV surveillance methods could be classified into vision based, laser based, magnetic based, wire based, color tape-based methods [6]. Among the surveillance methods, computer vision is a promising category because of its impressive advantages in costs, small size, abundant information, and computational efficiency [6]. For the sake of this, the vision-based surveillance system has been widely explored in both the academic and industrial fields. However, the vision system has the limitation of the inherent disadvantages regarding its long-term drift & considerable computational costs, landmark tracking failures in the context of the texture less environments, scale ambiguities [7]. These problems have impaired its potential applicability's in the AGV surveillance scenarios. To solve these problems, this works makes full sue of the downward looking camera to compensate for the forward-looking camera drawbacks. The downward camera captures the QR codes posted on the floor and estimates the absolute poses with reference to the global warehouse frame. Because the downward-looking visual surveillance has the relatively higher accuracy, it could remarkably reduce the accumulated forward-looking camera estimation drifts and make up for the errors of the estimated AGV poses over the adjacent QR code distances. Despite the advantages of the downward-looking and forward-looking visual surveillance, the fusion still has several issues that need to be solved for AGV surveillance. Firstly, the forward-looking and downward-looking cameras are supposed to be rigidly fixed and precisely calibrated before fusion. Yet the traditional calibrations are unable to find the appropriate visible views [8]. In addition, the solution of minimizing the downward and forward-looking visual residuals may get stuck in the local minimum or even cause system failures due to the inappropriate fusion strategy. To address these issues, this paper designs an innovative forward-downward visual fusion strategy. Firstly, the coordinates between the downward-looking camera and forward-looking camera are calibrated using the SE3 model. The transformation between the multi-view references is estimated by aligning the trajectories in a consistent manner. Secondly, the AGV positions and orientations are jointly estimated by the visual-QR measurements. Lastly, the map points and keyframes are jointly optimized in the modified graph optimization framework. The contributions of this paper are summarized as follows.

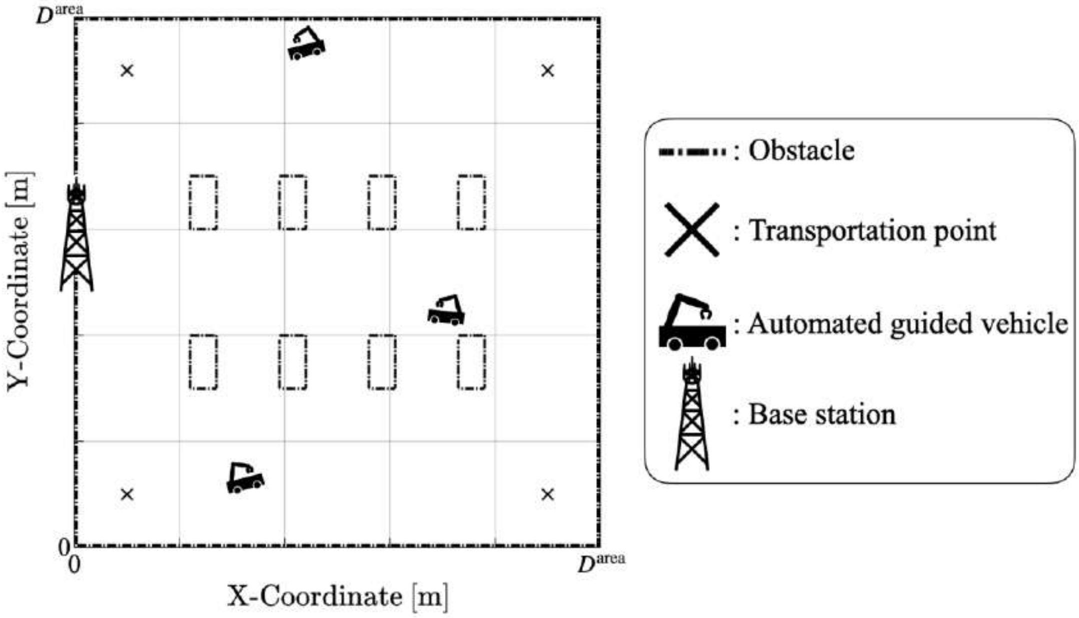


Figure 1. Indoor factory environment.

Automation of storage facility and management has made a new generation of industries that primarily focusses on automated goods transportation in storage facilities, live shipment tracking, automated dispatch, and merchant resource management. A Warehouse management system must satisfy the following:

• To decrease the time of storage retrieval inside very large storage facilities.

• To minimize the storage space to accommodate more goods in smallest possible storage area [9],[10].

• To decrease the time to take a product from placing order by a consumer to delivery of the order to consumer.

• To increase the sales and operations capability of a storage facility [11].

• To observe the current trading environment and to pull insights of the consumer behavior there by increasing sales and generating profits. 11

• To perfectly synchronize IT infrastructure and real time data analytics. To continuously operate the storage facility and handle huge loads of goods and products, shifting to an automated robotic solution increases the chance of faster manipulation and transport of goods, faster dispatching, and faster processing time [12],[13]. To increase the capacity of a warehouse to withhold a huge stock either the area of the warehouse needs to be enormous, or stocks must be piled up on one another. This results in low space for human controlled vehicles inside the warehouse to move the stock to one place to another which sometimes leads to congestion and jams there by clogging up entire orders and halting the dispatch process. This problem can be efficiently solved by automated robotic warehouse transport bots which are controlled by sophisticated algorithms and technologies like data acquisition, software synchronization, job scheduling, real time updates and inter robot communication and positioning systems that periodically adapt to the warehouse operation procedure and can be changed and modified by humans when needed [14].

1.1 Background

The AGV can tow objects behind them in trailers to which they can autonomously attach. The trailers can be used to move raw materials or finished products. The AGV can also store objects on a bed. The objects can be placed on a set of motorized rollers (conveyors) and then pushed off by reversing them. AGVs are employed in nearly every industry, including pulp, paper, metals, newspaper, and general manufacturing. Transporting materials such as food, linen or medicine in hospitals is also done.

An AGV can also be called a laser guided vehicle (LGV). In Germany the technology is also called *Fahrerloses Transport system* (FTS) and in Sweden *förarlösa truckar*. Lower cost versions of AGVs are often called Automated Guided Carts (AGCs) and are usually guided by magnetic tape. The term AMR is sometimes used to differentiate the mobile robots that do not rely in their navigation on extra infrastructure in the environment (like magnetic strips or visual markers) from those that do; the latter are then called AGVs.

AGVs are available in a variety of models and can be used to move products on an assembly line, transport goods throughout a plant or warehouse, and deliver loads.

The first AGV was brought to market in the 1950s, by Barrett Electronics of Northbrook, Illinois, and at the time it was simply a tow truck that followed a wire in the floor instead of a rail. Out of this technology came a new type of AGV, which follows invisible UV markers on the floor instead of being towed by a chain. The first such system was deployed at the Willis Tower (formerly Sears Tower) in Chicago, Illinois to deliver mail throughout its offices.

Over the years the technology has become more sophisticated and today automated vehicles are mainly Laser navigated e.g. LGV (Laser Guided Vehicle). In an automated process, LGVs are programmed to communicate with other robots to ensure a product is moved smoothly through the warehouse, whether it is being stored for future use or sent directly to shipping areas. Today, the AGV plays an important role in the design of new factories and warehouses, safely moving goods to their rightful destination.

This new era continues to this day and AGVSs are more popular than ever. This can for example be seen in the number of publications with the phrase “Automated Guided Vehicle” in its title, abstract or keywords from the first digital publications in 1974, and up to now, see figure 1.2. It is obvious that the trend has been increasing during this period, with a temporary decline in the 1980s.

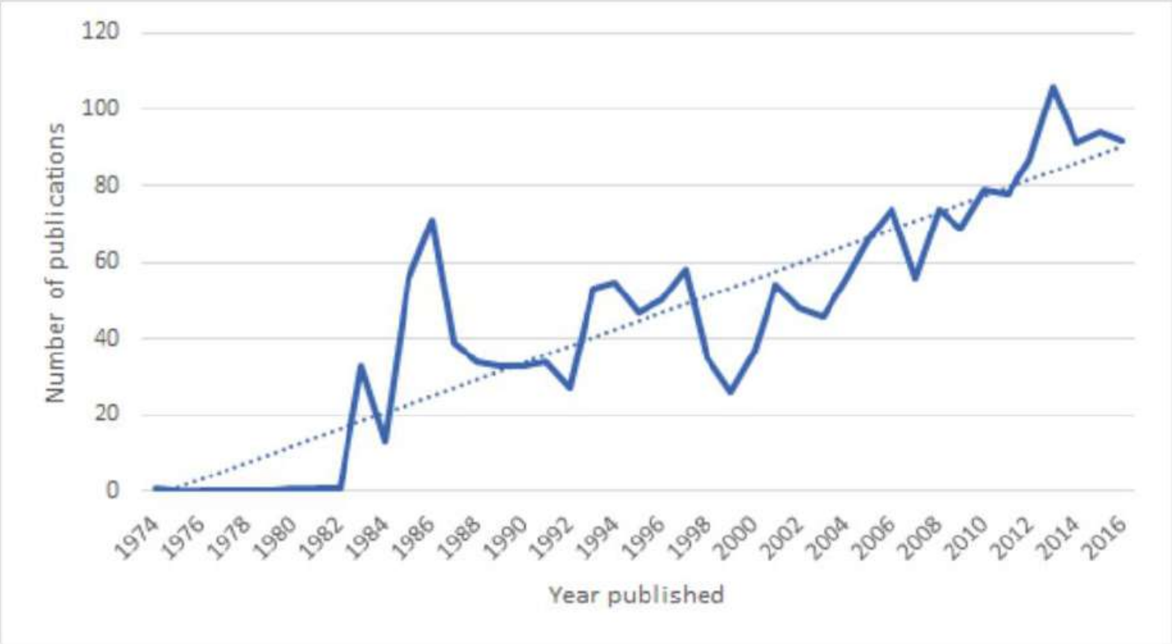


Figure 2 - An illustration of the number of publications with the phrase "Automated Guided Vehicle" in its title, abstract or key words.

In the beginning of the 80’s Scania started to implement AGVSs as well. According to Staffan Garås, Senior Advisor for Global Logistics Development at Scania, there was an automation trend in Sweden and Scania therefore started implementing AGVSs with active inductive guidance. These were however phased out in the middle of the 90s as a new trend reached Scania, lean production. Focus shifted from having a lot of material close to the production line to having the needed amount of material close to the line. This increased the importance of having a flexible material handling system, which the AGVS did not provide. Garås (2017) also adds that there were technical issues with the big and complicated AGVSs that led to too many disturbances in Scania’s production.

Technology has since developed a great deal and modern AGVSs come with many more advantages than the early ones. Lundgren (2017) has seen that AGVSs are a part of the big trend Industry 4.0 and that many of the German automobile manufacturers are implementing this new technology. The technology is also spreading to new market segments such as retail. In 2012 Amazon, the largest Internet-based retailer in the United States acquired the AGVS producer Kiva Systems and deployed 15 000 AGVs across 10 of its warehouses with the aim to reduce delivery lead times and increase customer service levels (D'Andrea, 2012). This new field of application increases the speed of technological development even more.

1.2 Problem Definition

The implementation of Automated Guided Vehicle Systems (AGVSs) in material handling processes requires a structured strategy for effective utilization of this new technology. In previous instances where new technologies were introduced across various production and warehouse sites without unified guidelines, the lack of standardization led to discrepancies in the equipment used, as seen with the installation of conveyor belts. This resulted in different conveyor systems being adopted at various production sites, leading to inefficiencies.

Recognizing the increasing trend and benefits of AGVs—such as reducing non-value adding time in manufacturing due to transportation, enhancing safety, and decreasing energy consumption. There is now a need for a well-defined strategy on how to implement AGVSs effectively. Includes identifying the relevant technological aspects critical to the successful integration and operation of these systems.

1.3 Research Purpose, Question and Objective

The aim of this undergraduate thesis is to map the technology related to Automated Guided Vehicle Systems (AGVS) and develop a guide on how and where to effectively apply this technology.

The research questions are:

- What is the current state of AGVS technology and what innovations are expected in the future?
- Which AGVS technology should be used in material handling processes?
- What are the main factors that determine the suitability of AGVS for the material handling process?

The aim of this undergraduate thesis is to express why AGVS should be preferred and will be presented as a decision matrix that simplifies the process of determining its optimal application. To develop this decision matrix, various technological aspects of AGVSs need to be comprehensively mapped. In addition, it is necessary to analyze the factors affecting the selection of AGVS in material handling processes.

1.4 Focus and Delimitations

The initial idea was to focus the report only on AGVs in the material handling process on the production floor. Specifically, this includes activities from the arrival of goods at the production site to the renewal of the production line, as well as the transportation of finished goods from the production line. This means that pure warehouse operations, other types of automation such as conveyor belts, and in-depth studies of assembly AGVs are excluded from this study.

Another topic that is only briefly discussed in this study is load sensing technology. Although an interesting topic, it is considered an add-on to AGV rather than part of the core AGVS technology.

The focus of this report is to map AGVS technology and create a guide on how and where to use this technology.

1.5 Target Group

The target audience for this report comprises individuals with a basic understanding of logistics and a keen interest in learning more about AGVs. The primary audience includes professionals within the logistics sector and members of the project teams responsible for the implementation of AGVSs.

1.6 Organization of The Project

Table 1 – Gantt Chart of project

PROJECT MANAGEMENT PLAN																		
Job Definition	Task Group Areas of Responsibility	Situation	1. Week 2.10.2023	2. Week 9.10.2023	3. Week 16.10.2023	4. Week 23.10.2023	5. Week 30.10.2023	6.Week 6.11.2023	7. Week 13.11.2023	8.Week 20.11.2023	9. Week 27.11.2023	10.Week 4.12.2023	11.Week 11.12.2023	12. Week 18.12.2023	13. Week 25.12.2023	14.Week 1.01.2024	15.Week 8.01.2024	16.Week 15.01.2024
Team Setup	Ahmet Selim YALÇIN	Completed																
Literature Study	Tüm Ekip	Continues																
Mechanical Design Process	Burçak DOĞAN	Continues																
Manufacturing Documentation	Burçak DOĞAN	Not started																
Mechanical Production Process	Burçak DOĞAN	Not started																
Mechanical Assembly	Burçak DOĞAN	Not started																
Structural Analysis Tests	Burçak DOĞAN	Not started																
Hardware Design Process	Mustafa Furkan EROĞLU	Continues																
Electronic Card String	Mustafa Furkan EROĞLU	Not started																
Cabling Production Process	Mustafa Furkan EROĞLU	Not started																
Testing of Electronic Cards	Mustafa Furkan EROĞLU	Not started																
Software Process	Barış Berker BAYRAM	Continues																
Software Testing	Barış Berker BAYRAM	Not started																
System Integration Process	Ahmet Selim YALÇIN	Not started																
Stock control	Ahmet Selim YALÇIN	Not started																
Supply Process	Ahmet Selim YALÇIN	Not started																
System Testing	Ahmet Selim YALÇIN	Not started																
Driving Test	Ahmet Selim YALÇIN	Not started																
First Report Preparation	Tüm Ekip	Continues																
First Report Presentation	Tüm Ekip	Not started																
Second Report Preparation	Tüm Ekip	Not started																
Second Report Presentation	Tüm Ekip	Not started																

1.7 Structure of System

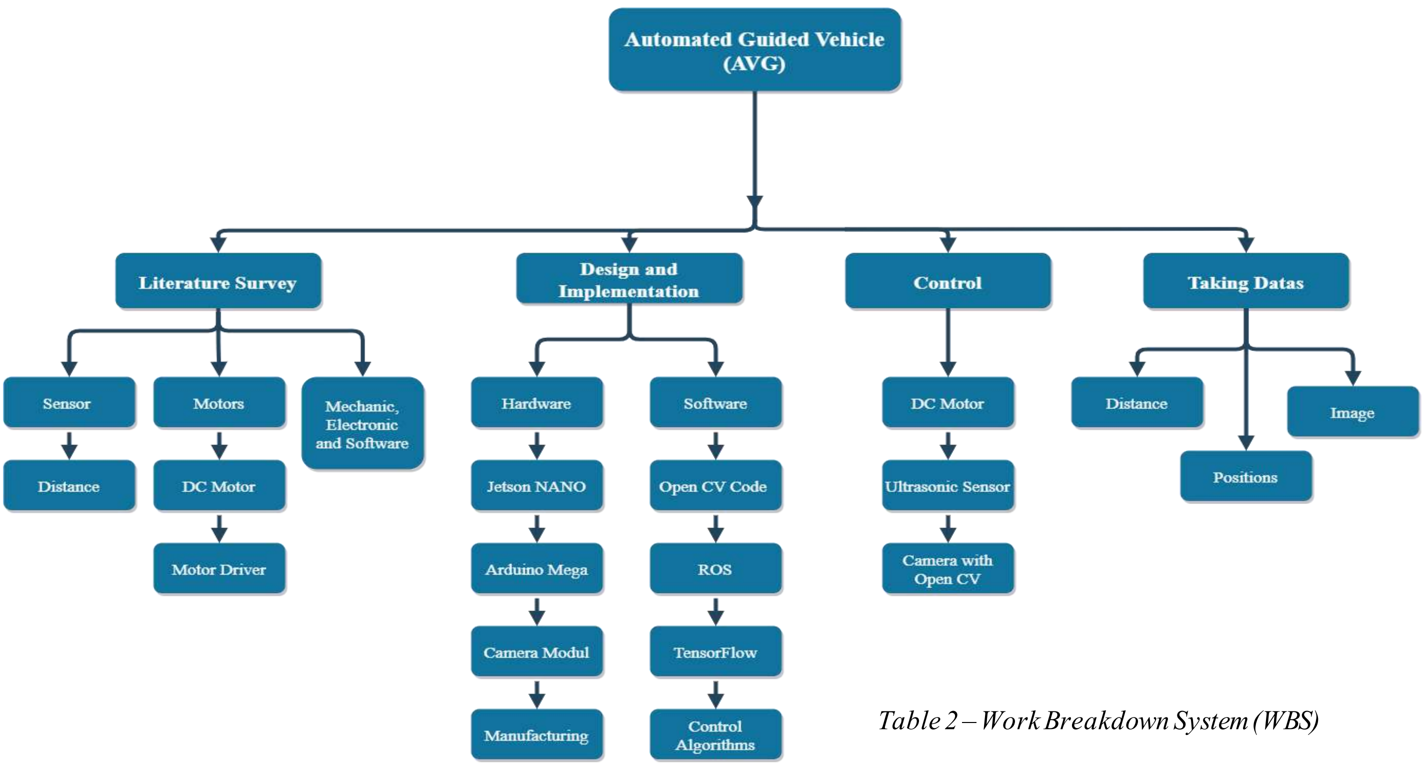


Table 2 – Work Breakdown System (WBS)

2. Related Works

Multi-view vision based mobile robotics navigation has attained great attention in the robotics and industrial communities because of its impressive navigation performance achieved. This section will briefly introduce the related works on forward-looking visual based and downward-looking visual based ground vehicle navigation. The merits and drawbacks of each navigation technology are introduced and discussed. Marius et al. [9] proposed a high-performance autonomous inventory MAV for operation inside warehouses. The MAV navigates along warehouse aisles and detects the placed stock in the shelves alongside its path with a multi-modal sensor setup containing an RFID reader and two high-resolution cameras. Draganjac et al. [10] presented an algorithm for decentralized control of multiple automated guided vehicles performing transportation tasks within industrial and warehousing environments. By running on each vehicle in the system, the algorithm provides vehicles with capabilities for autonomous path planning and motion coordination. Wang et al. [11] proposed a method for extracting deviation of AGV navigation path based on chromaticity threshold segmentation. This method adopts the idea of pre-closed-loop control, realizing AGV's path automatic tracking, key point turning, automatic driving, and recognition of the navigation line laid on the ground through the path tracking controller. However, in factories or workshops with a single scene, there exist problems such as significant changes in lighting and excessive brightness or darkness, which result in issues with setting the chromaticity threshold and consequently lead to the failure of AGV's navigation system and erroneous path planning. Pan et al. [12] proposed a Spatial Convolutional Neural Network (SCANN) to detect lanes by replacing layer-to-layer convolution in traditional feature maps with spatial slice-to-slice convolution, thereby transmitting information between rows and columns of pixels. However, the overall time complexity increases and real-time performance decreases, making it difficult to quickly identify whether there are anomalies in the AGV's driving process. Guan et al. [13] obtained the optimal solution by using the prior knowledge of the reference plane in holography estimation and aligning the gravity vector between two views. They separated the point correspondences into two sets to decouple the camera pose estimation. However, the distinction between far and near points was ambiguous, resulting in sub-optimal estimation results. Saurer et al. [14] proposed a relative pose estimation algorithm that utilizes the known common direction and correlated environment. This method can derive fewer point correspondences for relative motion estimation. However, in practical scenarios, the camera's degrees of freedom are reduced, and aligning the camera requires accurate calibration, which is not suitable for typical scenarios such as indoor or road driving. Tan et al. [15] proposed an efficient lane detection algorithm based on iterative search and Ransac curve fitting, which improves the robustness of the vehicle detect AprilTags2 and obtain visual pose estimation, which is then merged with kinematic pose estimation to obtain the AGV pose. However, the authors used a weighted average method to fuse the estimated poses from AprilTags2 and the poses calculated from the kinematic model, with weights based on empirical values, and did not use an algorithm to calculate optimal weight values for data fusion, resulting in low fusion accuracy. Bergamasco et al. [22] proposed a new benchmark marker system that relies on a robust framework of cyclic codes to provide excellent information redundancy, precise detection, and robustness against various types of noise. However, during the motion of the AGV, not only the baseline pose of the downward-facing camera is required, but also real-time detection of the AGV pose from a first-person perspective is needed. Using visual detection technology to detect the motion status (position, attitude, velocity) of AGV can quickly identify whether the AGV system has anomalies. Existing detection methods are mostly based on the first-person view, which results in long-term error accumulation and does not consider the impact of occlusion and lighting changes in factory environments on detection, leading to low detection accuracy. At the same time, the downward-facing camera has a limited field of view and can only capture a limited range of images, resulting in distortion in capturing image details.

2.1 Explorative, descriptive, explanatory and normative studies

When choosing which form of study that will be carried out the existing amount of information in the area of knowledge can be important. Explorative, or investigatory, studies are often used when there is little knowledge in the field studied and you are attempting to obtain a fundamental understanding. Descriptive studies are used when there is a fundamental understanding of the field studied and the aim is to describe, but not explain, relations. An explanatory study is used when searching for deeper knowledge about a field of study and the goal is to both describe and explain. Finally, normative study is used when there is already knowledge and an understanding of the field of study and the aim is to provide guidance and suggest measures (Björklund and Paulsson, 2014 p.64).

There is a lot of information about some areas of AGVS technology but not all. Areas such as guidance on how to program a certain type of AGV are well covered by literature, but there exists very little information regarding what factors that are important when implementing an AGVS. The purpose of the master thesis is, as mentioned earlier in the report, to map the technology connected to AGVSs and create a guideline for Scania on how and where to use the technology. The mapping of the technology will be done in a primary phase where an explanatory study is appropriate. The field of AGVSs will be described and explained. In the second phase a guideline for Scania will be created. In this part a normative study is more appropriate as guidance will be provided and measures suggested. This will be done based on the collected information in the primary phase which creates a thorough knowledge and understanding of the field.

2.2 Chosen Method

The method chosen for this report is illustrated in Figure 2.1. The mapping of the technology will be based on data collected from two primary sources: existing theory and suppliers. The theoretical framework will be derived from books and scholarly articles related to AGVSs and relevant areas impacting AGVSs. This information will be summarized in different sections within the frame of reference part of the report. Data from suppliers will be gathered through interviews and, in some cases, visits to their production sites. This information will be compiled in the empirical section of the report. Additionally, interviews with reference companies will be included in the empirical part to validate the suppliers' information and identify any discrepancies.

By analyzing and comparing theoretical insights with the empirical data collected, a comprehensive map of AGVS technology will be created. This map will be integrated with information from the theoretical framework and suppliers regarding the implementation of AGVSs. From this, a guideline will be developed to assist in selecting a suitable AGV system and determining the optimal locations for its implementation.

2.3 System Model

Figure 1 shows a material handling system for an indoor factory considered in this paper with a square area of Darea[m] on one side having multiple obstacles (such as walls, shelves, pillars, etc.) The communication system consists of a single BS multiple AGVs, which are indexed by i, where i ∈ I and I = {1, 2, . . . , I}. BS and multiple obstacles are fixed and placed as shown in Fig. 1. The two-dimensional location of BS is expressed by u^{bs} = [x^{bs} [m], y^{bs} [m]] ^{T}and the height of BS antenna is denoted by H^{bs} [m]. The two-dimensional locations of multiple obstacles are represented by a set u^{obs}. Each AGV travels to its allocated destination to transport materials and goods during a simulation time T^{sim} [sec]. The two-dimensional location of AGV i at time t, where t ∈ [0, T^{sim}], is represented by u_i(t) and the antenna height of AGV i is denoted by Hi [m]. This paper assumes AGV i follow the motion equation given as follows [14]:

$$x_i(t) = x_i(0) + \int_0^t v_i(t)\cos\theta_i(t)\,dt \quad (1a)$$

$$y_i(t) = y_i(0) + \int_0^t v_i(t)\sin\theta_i(t)\,dt \quad (1b)$$

$$v_i(t) = v_i(0) + \int_{t_0}^t \dot{v}(t')\,dt' \quad (1c)$$

$$\theta_i(t) = \theta_i(0) + \int_0^t \left(\omega_i(0) + \int_0^{t'} \dot{\omega}(t'')\,dt''\right) dt' \quad (1d)$$

where v_i (t) [m/sec], w_i (t) [rad/sec], θ_i (t) [rad] are the linear and angular velocities and the heading direction of AGV i at time t. Furthermore, the velocity and acceleration are constrained as follows:

$$-V \le v_i(t) \le V \quad (2a)$$

$$-\Omega \le \omega_i(t) \le \Omega \quad (2b)$$

$$-A_v \le \dot{v}_i(t) \le A_v \quad (2c)$$

$$-A_\omega \le \dot{\omega}_i(t) \le A_\omega \quad (2d)$$

where V [m/sec], Ω[rad/sec], A^{v}[m/ sec^{2}], A^{w}[rad/sec^{2}], are maximum velocities and accelerations. This paper assumes that each AGV communicates with BS, i.e., transmits sensing information and images as it travels to its destination for material handling.

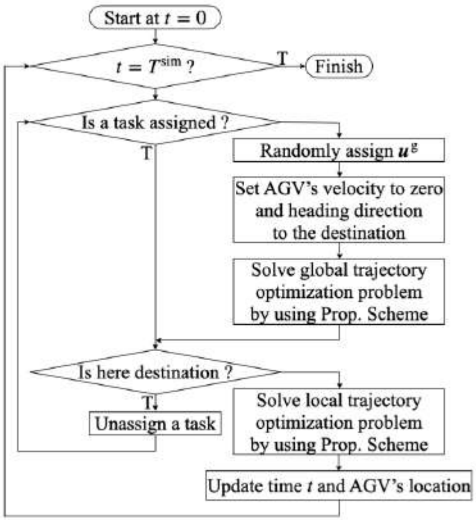


Fig. 3. Task assignment and path planning in material handling.

A. Task assignment

For material handling, a task operation model is illustrated in Fig. 3. The tasks are assigned to AGV i with instructions to arrive at random destination represented by $u_i^g(t)$, where u g $u_i^g(t) \in U^g$ and U^g is denoted as a set of transportation points showed in Fig. 1. Each AGV travels according to its assigned task by local path planning, given a global optimal trajectory.

B. Uplink communication

All AGVs communicate with BS over one available channel with bandwidth *B*[Hz] for transmitting information. To avoid signal interference among AGVs, time division multiple access (TDMA) is applied.

The distance between AGV i and BS at time *t* is denoted by

$$d_i(t) = \sqrt{\| u_i(t) - u^{bs} \|^2 + (H_i - H^{bs})^2}$$

For simplicity, AGV-to-BS channel is dominated by NLoS paths [15], so that the channel gain is described as

$$\gamma_{i,\mathrm{dB}}(t) = -10a \log_{10} d_i(t) - b - 10c \log_{10} f - \xi_{dB}(d_i(t)),$$

Where $a, b, c,$ f [GHz] denotes path loss coefficient, reference offset, carrier frequency factor, and carrier frequency, respectively, and ξ is the shadowing component with spatial correlation [16]. Then, the signal-to-noise ratio (SNR) is given by

$$\Gamma_i(t) = \frac{P_i \gamma_i(t)}{\sigma^2}, \qquad (3)$$

where P_i [W] is the transmit power of AGV i and σ^2 [W] is the additive white Gaussian noise (AWGN) power.

C. Problem formulation

The aim of this paper is to maximize the achievable spectral efficiency in each task operation time by optimizing the trajectory. Thus, the optimization problem for AGV i can be formulated as follows:

$$
\begin{aligned}
\text{(P1):} \quad & \underset{\{\boldsymbol{u}_i(t), \forall t \in (t_0, t_0+\tau_{t_0}]\}}{\text{maximize}} \int_{t_0}^{t_0+\tau_{t_0}} \log_2\left(1+\Gamma_i(t)\right) dt && \text{(4a)}\\
\text{s.t.} \quad & \boldsymbol{u}_i(t_0+\tau_{t_0}) = \boldsymbol{u}_i^{\mathrm{g}}(t_0) && \text{(4b)}\\
& v_i(t_0) = 0, && \text{(4c)}\\
& \omega_i(t_0) = 0, && \text{(4d)}\\
& \theta_i(t_0) = \arg\left(\boldsymbol{u}_i^{\mathrm{g}}(t_0) - \boldsymbol{u}_i(t_0)\right), && \text{(4e)}\\
& \|\boldsymbol{u}_i(t) - \boldsymbol{u}\| \geq D^{\mathrm{hit}}, \forall \boldsymbol{u} \in \mathcal{U}^{\mathrm{obs}} && \text{(4f)}\\
& \|\boldsymbol{u}_i(t) - \boldsymbol{u}_j(t)\| \geq D^{\mathrm{hit}}, \forall j \in \mathcal{I}\backslash i && \text{(4g)}\\
& (1a) - (2d)
\end{aligned}
$$

Where t_0 [sec] and $(t_0 + \tau_{t_0})$[sec] are the start and finish time of each task operation, and D^{hit} [m] is the minimum distance to ensure collision avoidance. Here, (4b) constrains the arrival at the destination within the travel time, (4c) - (4e) indicates that the velocities and heading are initialized when a task is assigned, (1a) - (2d) constraints motion, and (4f) - (4g) constraints collision avoidance. Note that (P1) is NP-hard and difficult to solve due to the non-convex (4a), nonholonomic movement (1a) - (1d), and combinatorial restrictions (4g), which needs to be transformed and relaxation of restrictions.

2.4 Engineering Calculations and Analysis

Numerical techniques and a range of software applications were employed in the study to analyze the designs. (h) is the height at which the scissor lift platform should be in its open configuration. At this height, the angle (α) between the scissor material and higher platform was established. It was anticipated that a 50 kg weight would disperse the burden on the higher level. The weight that the platform must lift is governed by this load as well as the materials utilized to construct the upper platform and scissor profiles. Kinematic analysis was carried out and the rotation of the mobile robot interacting with the platform was schematized.

2.4.1 Kinematic Representation of a Mobile Robot Doing a Point-Turn

This section will give the kinematic model of a mobile robot (RASIM) with two wheels in the middle that are powered by these wheels. The vehicle has driving wheels on both the left and right sides, as well as two ball casters on the front and back.

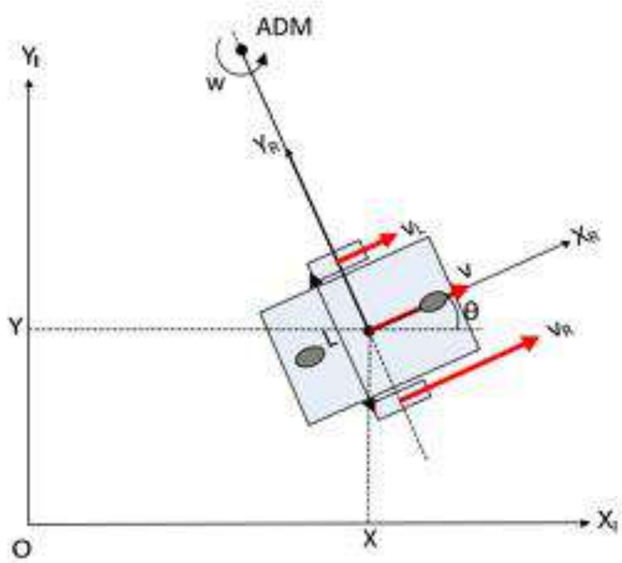


Figure 4 RASIM kinematic model visualization (Arslan, 2011)

The following equation provides the vehicle's sequence vector.

$$Q=[X \quad Y \quad \alpha]^T$$

X and Y in the RASIM alignment indicate the robot's location within the whole set of axes. and the vehicle's direction in the axis set is indicated by θ. The vehicle's orientation in the axis set is indicated by α. The variables and matrices that were previously specified and adjusted to the kind of RASIM in this design may be found in the following equations for the kinematic model of mobile robots.

$$\acute{p}= S(q)\dot{\eta}$$

$$\begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} cos\theta & 0 \\ sin\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ w \end{bmatrix}$$

$$\eta = \begin{bmatrix} v \\ w \end{bmatrix}$$

$$S(q) = \begin{bmatrix} cos\theta & 0 \\ sin\theta & 0 \\ 0 & 1 \end{bmatrix}$$

About the vehicle, the velocity vector in the RASIM's local axis set is lateral. A row matrix is one way to express a matrix. A moving robot turning a point the following equation provides the matrix specified for.

$$A(q) = [-sin\theta \quad cos\theta \quad 0]$$

2.4.2 Point Rotation and Dynamic Modeling of a Mobile Robot

The dynamic model of a point-turning robot will be shown in this part.

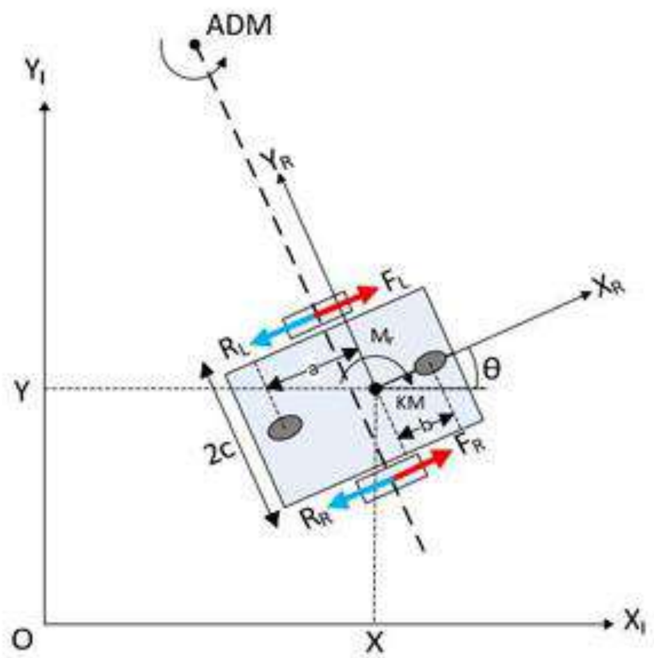


Figure 5 RASIM dynamic model representation with point rotation (Arslan, 2011)

$$\ddot{q} = S\,(q)\dot{\eta} + \dot{S}(q)\eta$$

$$q = \begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{\theta} \end{bmatrix}, \qquad S\,(q) = \begin{bmatrix} cos\theta & 0 \\ sin\theta & 0 \\ 0 & 1 \end{bmatrix}, \qquad \eta = \begin{bmatrix} v \\ w \end{bmatrix}$$

$$S^T MS\dot{\eta} + S^T MS\dot{\eta} + S^T R + S^T \tau_d = S^T B_\tau - S^T A^T \lambda$$

The robot's dynamic model has zero components because of non-holonomic limitations.

$$\bar{M}\dot{\eta} + \bar{M}\eta + \bar{R} + \overline{\tau_d} = \bar{B}\tau$$

Below are the explicit expressions for the matrices and vectors found in the equations.

$$\bar{M} = S^T MS,\ \bar{C} = S^T M\dot{S},\ \bar{R} = S^T R,\ \overline{\tau_d} = S^T \tau_d,\ \bar{B} = S^T B$$

2.4.3 Kinematic Calculation of The Scissor Lift

A visualization of the scissor lift in the open position is shown in Figure.

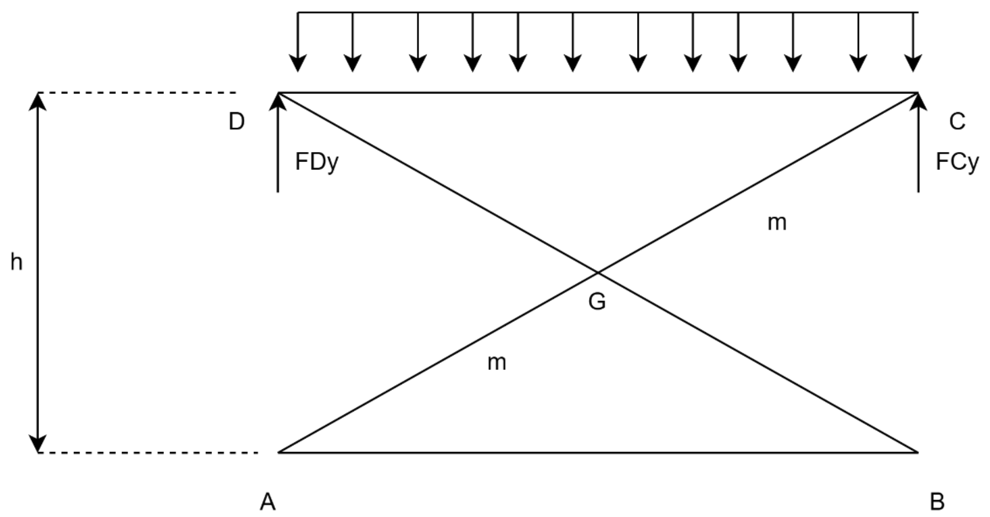


Figure 6 Illustration of the scissor lift system a free drawing diagram

w = $m_{top} \times \frac{g}{L}$

w= $50 \times \frac{9.81}{415}$ = *1.18 N/mm*

W=1.18× 415=490 N

The assumption of moment rotation was made to rotate counterclockwise. This presumption applies to all ensure that moment equations may use it.

$\sum M_D$=0 -W (W_{length}) + $2F_{Cy}(length\ of\ platform) = 0$

$\sum F_Y$=0 $2F_{Dy}$ - W+ $2F_{Cy}$ = 0

The first equation is changed to reflect the force produced by the weight. The scissors at 45°, the platform's length measurements were made.

-490(200) +2(118.07) =0

F_{Cy}= 118.07 N

Likewise, if is F_{Cy} written in the other equation,
2(F_{Dy})-490+2(118.07) =0

F_{Dy}=126.9 N

The forces $F_{Dy}$and F_{Cy} calculated here are used to calculate the load on the profile arm,

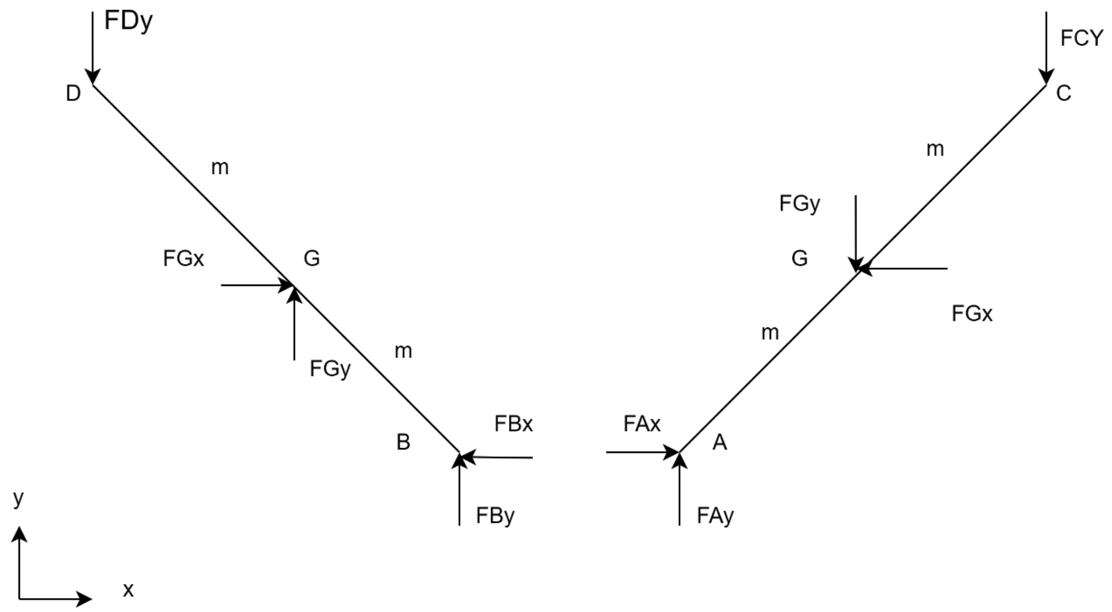


Figure 7 Diagram for lower cross profiles

$\sum M_B$= 0 F_{Dy} ×2×cosα−F_{Gy}× m ×sin(90-α) −F_{Gx}×m×sinα=0

$\sum F_x$=0 $F_{Gx} - F_{Bx} = 0$

$\sum F_y$=0 -F_{Dy}+F_{Gy} + F_{By} = 0

$\sum M_D$= 0 − F_{Cy} ×2× m ×cosα−F_{Gy}× m ×sin(90-α) + F_{Gx}×m×sinα=0

$\sum F_x$=0 $F_{Ax} - F_{Gx} = 0$

$\sum F_y$=0 -$F_{Cy} - F_{Gy} + F_{Ay} = 0$

If F_{Gy} is left alone in the moment equation taken from point B and simplifications are made:

$F_{Gy}=2\times F_{Dy}-F_{Gx}\times tan\alpha = 0$

If F_{Gx} is left alone in the moment equation taken from point D and simplifications are made:

$$F_{Gy}=2\times F_{Dy} - F_{Gx}\times tan\alpha$$

And,

$$F_{Gx} = \frac{2F_{Cy}}{tan\alpha + \frac{F_{Gy}}{tan\alpha}}$$

Thus,

$F_{Gy}=F_{Dy} - F_{Cy}$

The operations were performed assuming that the internal angles of the scissors were at 45°.

$$\text{Cos (45) =Sin (45)} =\sqrt{2}/2 \cong 0.7071$$

$$\sum M_D= 0 \quad - F_{Cy}\times 2\times m \times cos\alpha - F_{Gy}\times m \times sin(90\text{-}\alpha) + F_{Gx}\times m\times sin\alpha=0$$

$$F_{Gy}=2\times F_{Dy} - F_{Gx}\times tan\alpha$$

$$F_{Gx} = \frac{2F_{Cy}}{tan\alpha + \frac{F_{Gy}}{tan\alpha}}$$

$$-118.07\times 2\times 200\times 0.7071+F_{Gy}\times 200\times 0.7071-F_{Gx}\times 200\times 0.7071=0$$

$$-33294.2+141,42F_{Gy}-141.42F_{Gx}=0$$

$$F_{Gy} - F_{Gx}=\frac{33294.2}{141.42}=235.5 \text{ N}$$

$$\sum M_B= 0 \quad F_{Dy}\times 2\times cos\alpha - F_{Gy}\times m \times sin(90\text{-}\alpha) - F_{Gx}\times m\times sin\alpha=0$$

$$126.93\times 2\times 0.7071-F_{Gy}\times 0.7071-F_{Gx}\times 0.7071=0$$

$$179.4-0.7071F_{Gy}-0.7071F_{Gx}=0$$

$$F_{Gy} + F_{Gx}=\frac{179.4}{0.7071}=253.5 \text{ N}$$

Then,

$$F_{Gy} - F_{Gx} = \frac{33294.2}{141.42} = 235.5 \text{ N}$$

$$F_{Gy} + F_{Gx} = \frac{179.4}{0.7071} = 253.5 \text{ N}$$

$$F_{Gy} = 0$$

$$F_{Gx} = 693.0N$$

$$F_{Ax} - F_{Gx} = 0$$

$$F_{Ax} = F_{Gx}$$

$$F_{Ax} = 693.0N$$

$$-F_{Cy} - F_{Gy} = F_{Ay}$$

$$F_{Ay} = 362.57 \text{ N}$$

$$F_{Gx} - F_{Bx} = 0$$

$$-F_{Dy} + F_{Gy} + F_{By} = 0$$

$$F_{Gx} = F_{Bx}$$

$$F_{Bx} = 693.0N$$

$$F_{By} = 346.5N$$

The force component with the biggest size is chosen, and its product is obtained. The maximum force that should be applied to the design is therefore determined.

Angle α	$\boldsymbol{F_{Dy}}$ **(N)**	$\boldsymbol{F_{Cy}}$ **(N)**	$\boldsymbol{F_{Gx}}$ **(N)**	$\boldsymbol{F_{Gy}}$ **(N)**	$\boldsymbol{F_{Ax}}$ **(N)**	$\boldsymbol{F_{Ay}}$ **(N)**
5°	**245.5**	**245.5**	**5611.4**	**-20.8**	**2806.8**	**224.7**
10°	**248.8**	**248.8**	**2823.1**	**0**	**1411.6**	**248.8**
20°	**260.8**	**260.8**	**1432.7**	**0**	**716.5**	**260.8**
30°	**282.8**	**282.8**	**979.7**	**0**	**490.0**	**282.8**
45°	**346.5**	**346.5**	**693.0**	**0**	**346.5**	**346.5**

Table 3 forces associated with the scissor arm's angular locations

The force component with the greatest size is chosen, and its product is obtained. The maximum force that should be applied to the design is therefore determined.

F = $\sqrt{F_X{}^2 + F_y{}^2}$

F_{Ax} = $\sqrt{FA_X{}^2 + FA_y{}^2}$ = 2815.77 N

The greatest force operating on the platform is found to be 2815.77 N once the equation is solved.

2.4.4 Top Plate Analysis

The scissors attain their furthest separation from one another when the platform is closed. analysis of the top table performed while the site was closed.

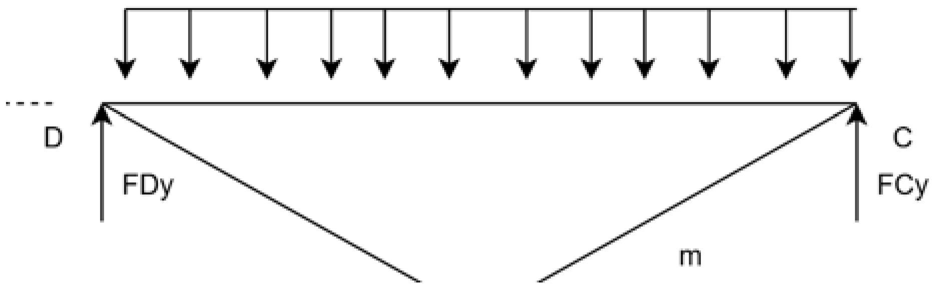


Figure 8 Free body diagram for top plate

$$F_{Dy} = F_{Cy} \approx 350 \text{ N}$$

$$\text{W}=2\times F_{Cy} = 700\ N$$

$$\text{w} = \frac{W}{L} = \frac{700}{200} = 3.5 \text{ N/mm}$$

Me=w $\times \frac{L^2}{8}$ =75348.4Nmm

$\sigma_t = \frac{\text{Me}}{We} \le \frac{\sigma\text{ak}}{S}$ (N/mm^2)

$We = \frac{\text{bh^2}}{6}$ =(480×*10^2)/6 =8000* *mm^3*

The material used was stainless steel of grade 304. 10 mm is the nominal wall thickness (h) used as a benchmark.

$\sigma_t = \frac{75348.4}{8000} \leq \frac{480}{10}$

$9.41 \frac{\mathrm{N}}{mm^2} \leq 48 \frac{\mathrm{N}}{mm^2}$

As seen from the calculations, the top table is made of 304 stainless steel, 50 kg carrying capacity safe for its capacity.

2.4.5 Scissor Profile Size Determination

The truss profiles were ideally made of the same material as the top plate.

Nominal Wall Thickness (mm)	Rm - Tensile Strength (MPa)	ReH - Minimum Yield Strength (MPa)	KV- Impact Energy (J)	A- Min. Elongation (%)
3-250	515 - 720	205	40 - 60	40 - 50

Figure 9 Mechanical properties of 304 Stainless Steel (Anonymous)

When the profile is closed and the load is greatest, a basic torsion analysis is done.

F_{Ax} = 2806.8 N (compression force)

F_{Ay} =224.7 N

Me=224.7 × *205= 46063.5 Nmm*

Profile dimensions were chosen to be 50×10 mm.

$We=\frac{bh^2}{6}$ =4166.6mm^3

A=50 × *10 = 500 mm^2*

$\sigma_e = \frac{46063.5}{4166.6}$=11.05 N/mm^2

$\sigma_t=\frac{F}{A}$= 5.61 N/mm^2

$\sigma_{Total}=\sigma_e+\sigma_t$=16.6 N/mm^2

$\sigma_{factor}=\frac{480}{5}$=96 $N/mm2$

$16.6\frac{\mathrm{N}}{mm^\wedge 2} \le 96 \frac{\mathrm{N}}{mm^\wedge 2}$

Design that uses materials and proportions chosen in accordance with the procedures used can function safely.

2.4.6 Pin Calculation Used for Joining Profiles

The G point is where the strongest link in the design is located. While the platform is closed, this force is at work. As a result, calculations were done using this force.

Pin material from TS2337 standards:

for shear material

σak=295 $N/mm2$,

and for pin material from ST 50-2

σak=480 $N/mm2$.

The connecting pin is subjected to shear stresses based on design specifications and will not break.

$$\tau = \frac{F_{yük}}{\frac{n\pi d^2}{4}} \le \tau_{em} = \frac{\tau_{ak}}{S}$$

N= number of cross-sections forced to cut

τak=0.45×σak

τak=0.45×295=132.75 N/mm2

$$\tau = \frac{2806.8}{981.7} \le \frac{132.75}{5}$$

$$2.85 \frac{\mathrm{N}}{mm^\wedge 2} \le 26.5 \frac{\mathrm{N}}{mm^\wedge 2}$$

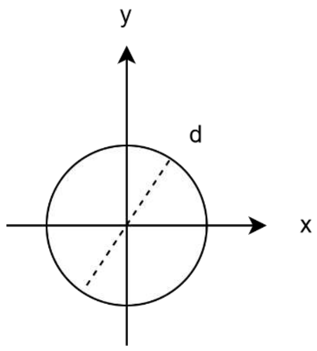


Figure 10 Pin cross-sectional view

Based on calculations, the pin is safe.

2.4.7 Manufacturing Processes

- Cutting and Forming: Metal sheets or profiles are appropriately bent and cut.

- Welding: Screws or welding are used to link parts. Wheels and chassis are put together.
  4
- Assembly is the process of putting wheels, engines, and other mechanical parts together.

After the design of **RASIM** was completed, the workspace of HSC ULV was used to start the production process. At first, the square profiles of the chassis were cut to specified dimensions and welded. Then the wheels were mounted to the motors. The connections with the chassis were made using welding. Afterwards, 4 ball casters were fixed to the bottom cover. The most reasonable products available in the company were used. The bottom cover was fixed with screws. Profiles were cut and welded for the scissor system. The dimensions were made suitable in the Miller lathe machine. After the scissors were finished, the linear actuator motor was mounted on the chassis. The total mass of the vehicle was 15,715 kg (excluding chassis and motors linear actuator). The weight of the hoist is 2.875 kg. The total weight is around 20 kg (including cables and software devices).

2.4.8 Ansys Applications

The design of mobile robots was based on theoretical computations. The final vehicle design takes into consideration the relevant dimensions and specifications. These designs were 3D created using the Catia software and SolidWorks. Strong The ANSYS software was used to examine the parts of the automobile model.

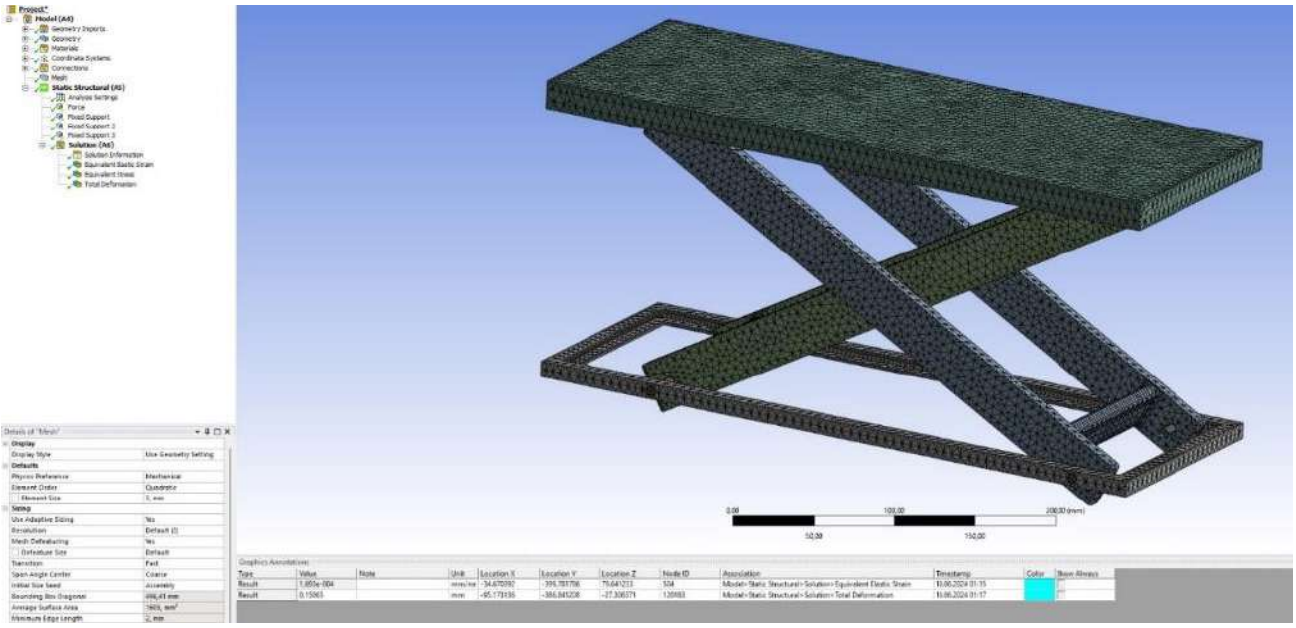


Figure 11 Mesh Structure of Scissor Lift Platform

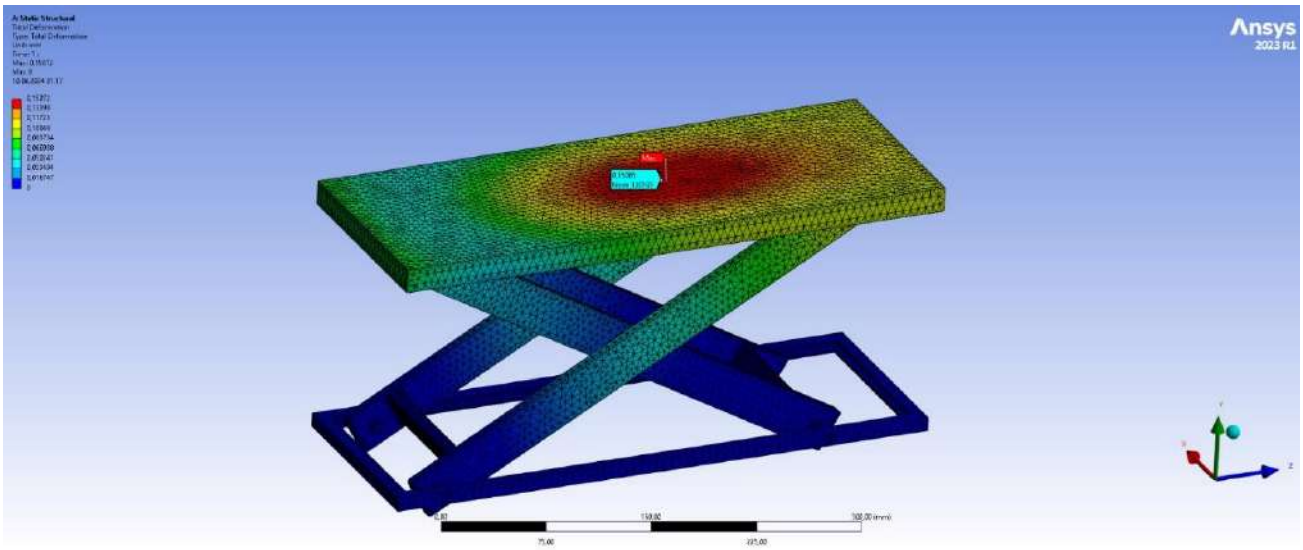


Figure 12 Deflection on the Scissor Platform with the Maximum Force to Be Used

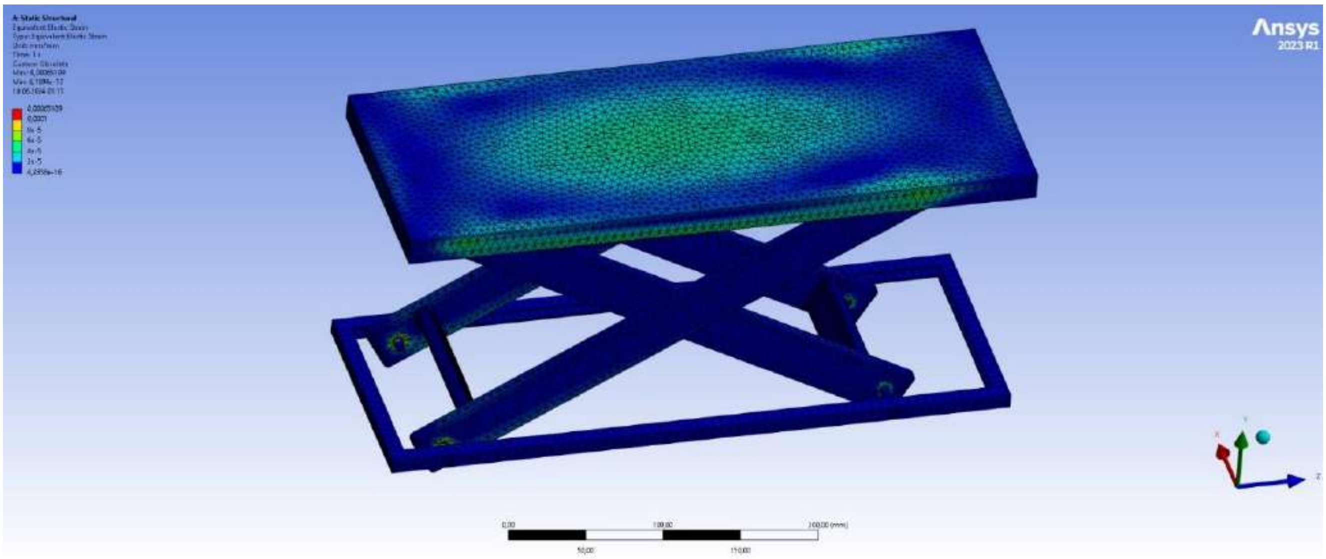


Figure 131 Equivalent elastic strain

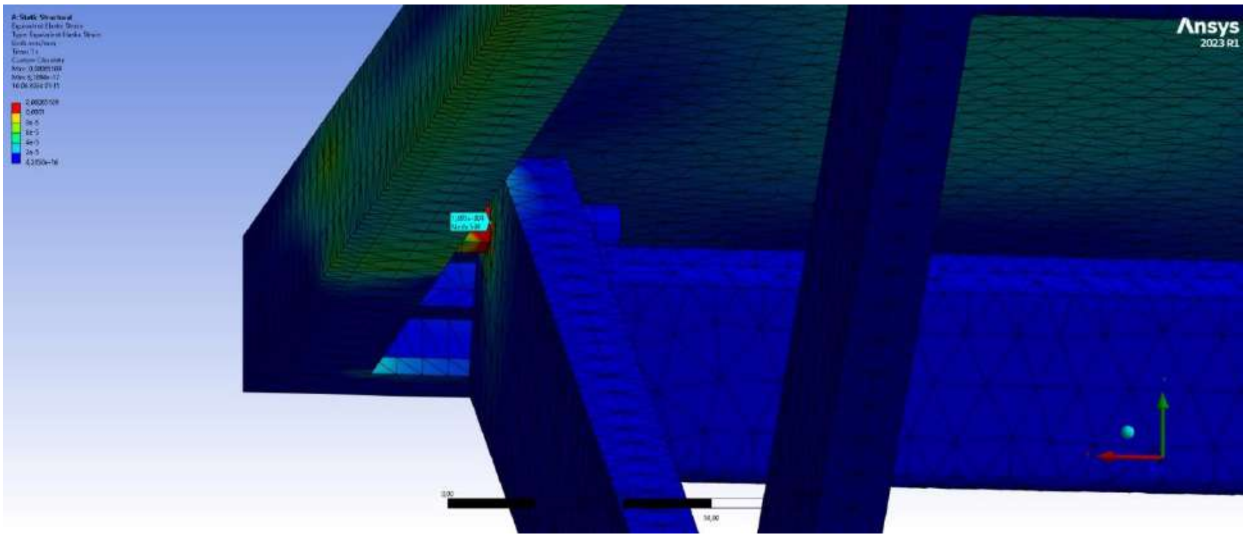


Figure 14 Bomb view of equivalent elastic strain

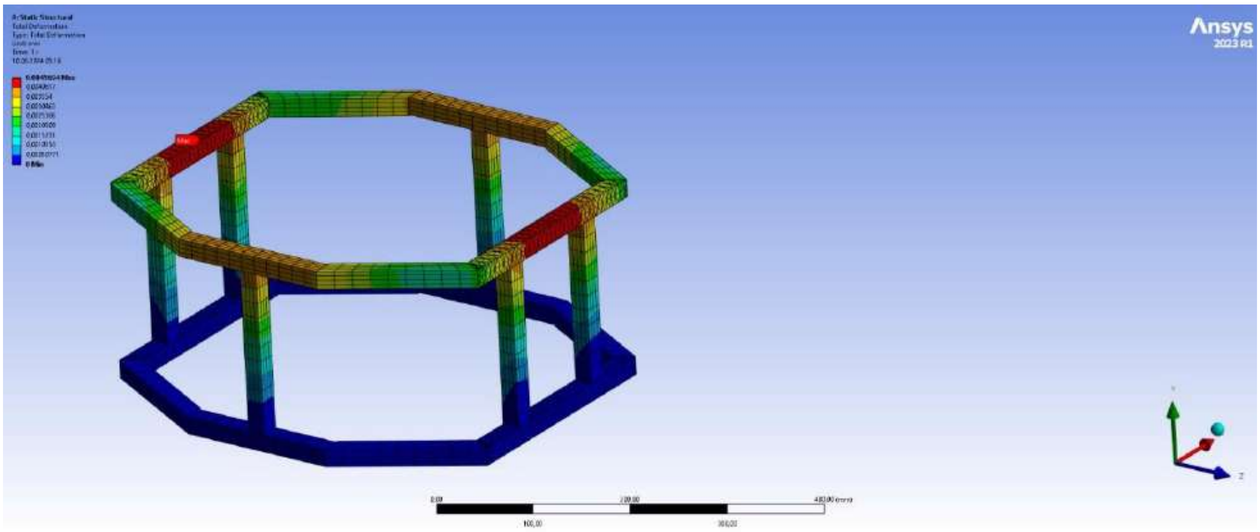


Figure 15 Total deformation of chassis

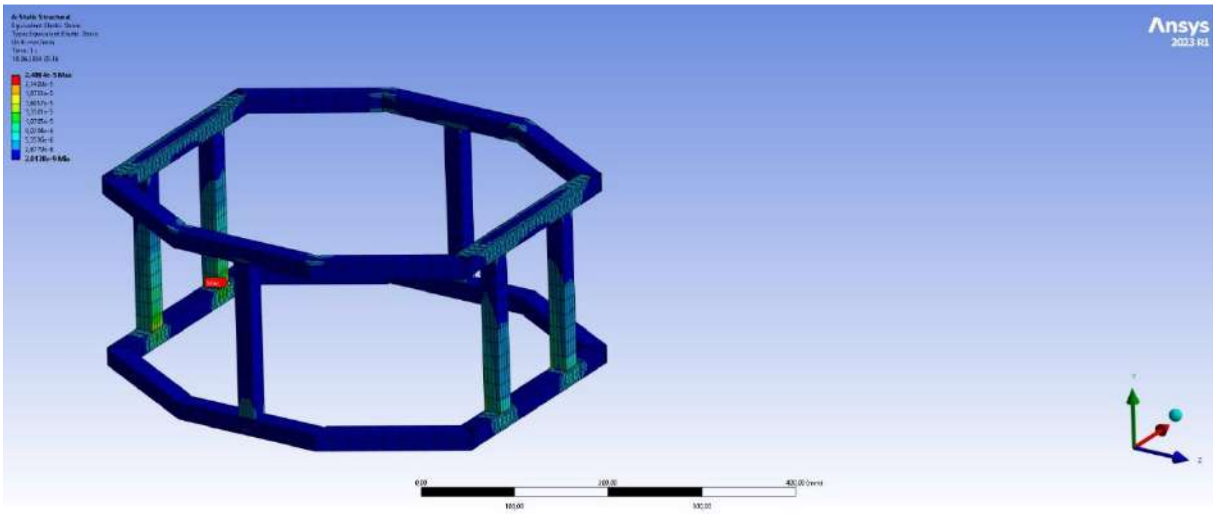


Figure 16 Equivalent elastic strain of the chassis

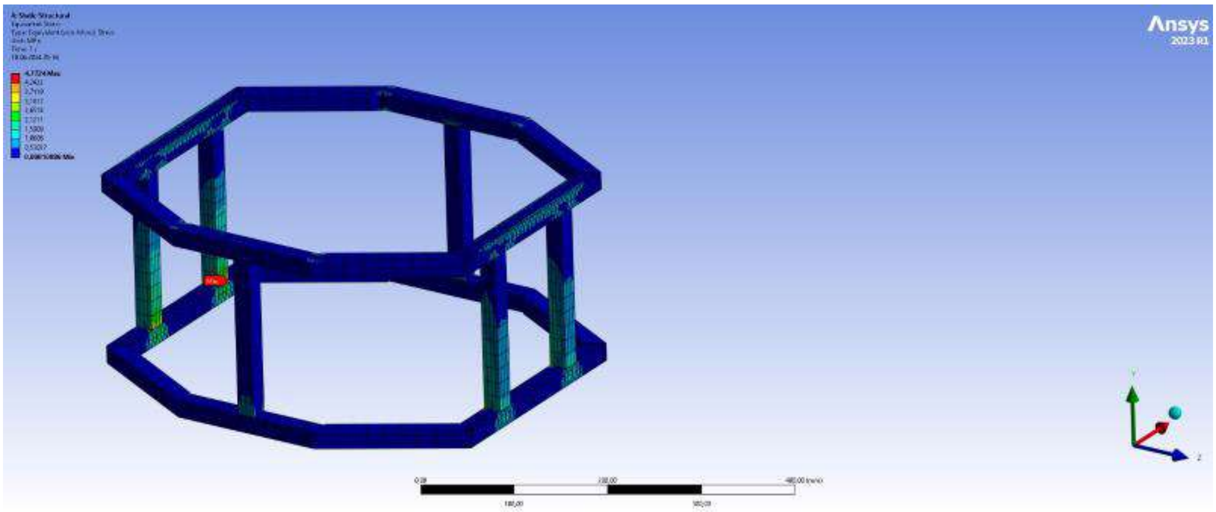


Figure 17 Equivalent (von- Mises) stress of the chassis

2.5 Design of an AGV System

In this section of the report, it is concluded that the dimensions of the design should have a length of 420 mm, a width of 317 mm and a height of 220 mm because of the scanning and examination of the data results during the design of the automatic guided vehicle. The load capacity calculated so far and expected to lift is 5 kg and it is agreed that it should be capable of lifting a load up to 60 mm into the air. It was decided that the vehicle should have two large and three small (drunk wheels).

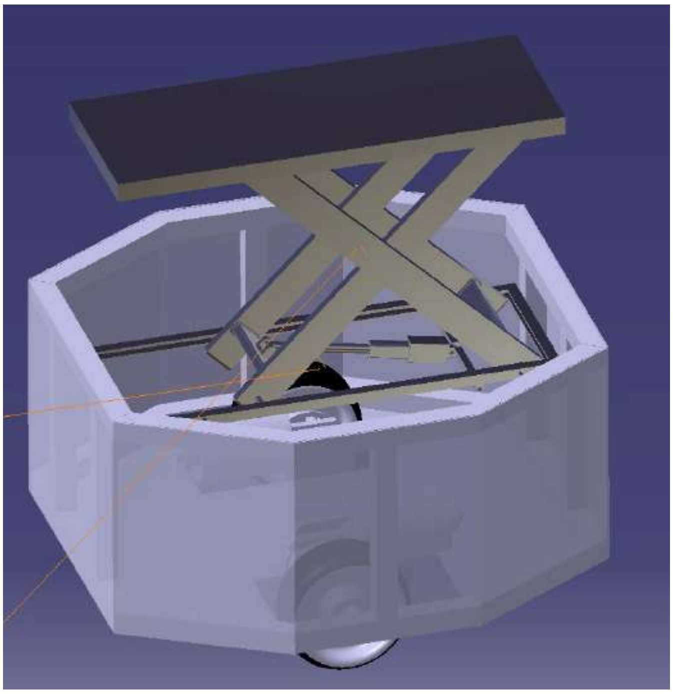


Figure 18 AGV Upper System

2.6 Frame Design

This design approach increases the strength of the system and as a result the system can lift more weight. A static analysis is conducted to validate the strength of the aluminum structure. Specifically, the model of the structure is tested, in different loads, to estimate the limits of the design. Our main concern is to fmd out if the yield limit of the material used is exceeded and plastic deformations are developed. For the estimation of the stresses developed, the ANSYS specialized Finite Element Analysis @EA) program is used. Finite Element Analysis is used more and more in the development of new products, providing the manufacturers with low-cost redesign and optimization times. [21]

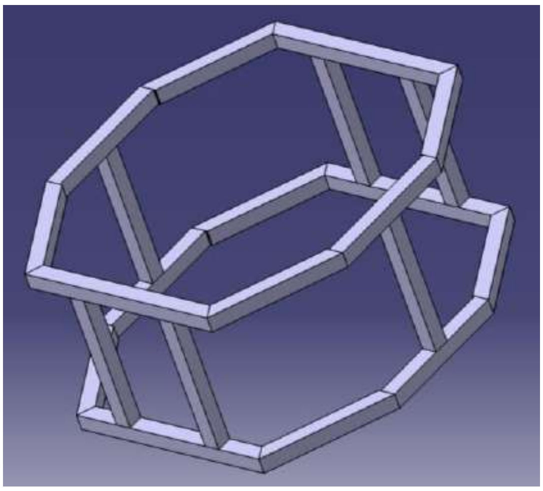


Figure 19 Frame Design

2.7 Motor Selection

Motors The motor of AGV generally is mounted with an encoder to measure the distance of path. The motor type which are used in AGV are named Geared DC motor, brushless DC motor, servo motor. These motors are selected based on their impacts on flexibility and accuracy of AGV movement. Drive mode is divided into single wheel drive, differential drive and omnidirectional drive. Single wheel drive means a driving wheel has the function of walking and steering and two driven wheels are fixed. Differential drive has two drive wheels that use the speed difference to realize rotation. Omnidirectional drive is much more flexible. With two driving and rotatable wheels, both parallel movement and differential drive function are available [22].

2.8 Load Transfer Methodology

Unit load AGV has many kinds of decks (e.g. lift, roller, chain, *etc*) to transport materials. In the consideration of the transfer requirements of automobile assembly production lines, we suggest a material transport control methodology to locate AGV beside the load stand accurately and retrieve and deposit the pallet automatically. [23].

2.9 Load-Transfer Mechanism Design

To execute the retrieving and depositing behaviors required in the long-travel materials transport process, a both sides three-level push-pull load-transfer mechanism is designed. This mechanism contains a lifting module and a translational module, as shown in [23].

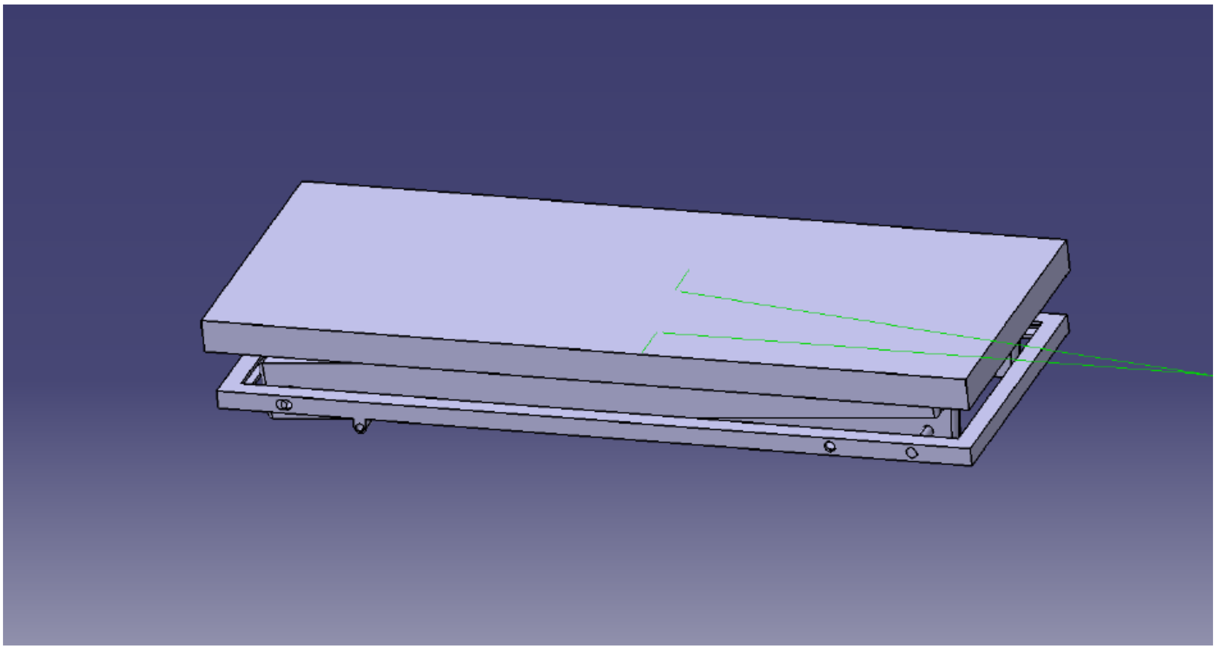


Figure 20 Load-Transfer Mechanism Closed Version

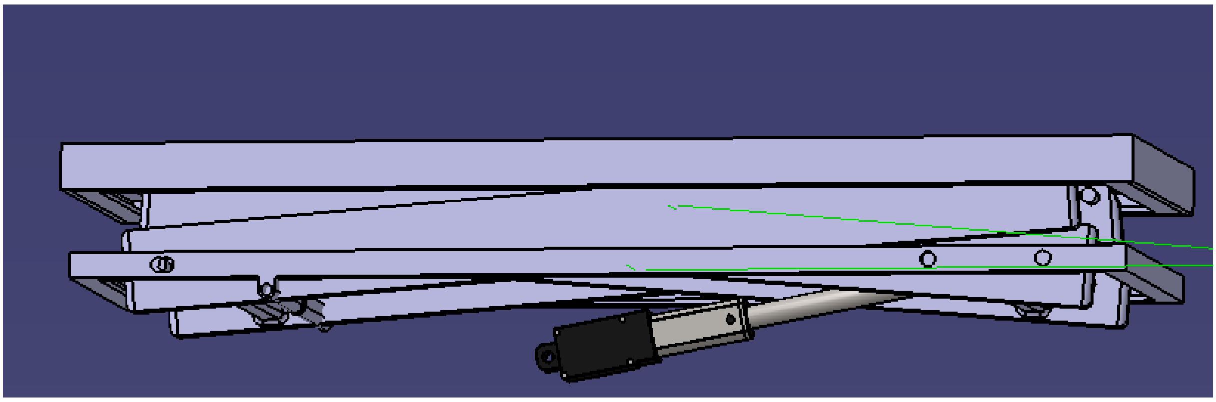


Figure 21 Load-Transfer Mechanism Closed Version Another View

2.10 Manufacturing Process

The manufacturing process of the vehicle involves five distinct components. The initial phase focuses on crafting the outer shell, employing sheet metal plates for the exoskeleton. These plates are meticulously cut to the required size and shaped using a bending method. Subsequently, the shaped pieces are welded together, resulting in the completion of the exoskeleton. Moving on to the second aspect, the vehicle skeleton incorporates aluminum sigma profiles. These profiles are cut to specified dimensions and interconnected using apparatus, forming the essential framework. The third component encompasses the production of in-wheel motors and passive wheels responsible for mobility, designed in accordance with the Digital Twin specifications and fortified with epoxy, hydrophobic spray, and tubing for sealing. The fourth facet pertains to the carrier system, constructed from iron profiles and sheet metal plates. The iron profiles are cut, melted together, and welded, while the sheet metal undergoes cutting, bending, and welding processes before the two sections are merged. Finally, the fifth part involves the creation of the lift system and the conveyor belt system. The lift system incorporates iron profiles and sheet metal plates, fused through welding, while the conveyor belt system integrates iron profiles, pulleys, belts, and bearings, providing efficient movement facilitated by the fusion-welded pulleys and strategically placed bearings. This comprehensive process ensures the meticulous fabrication of each integral component, culminating in the production of a fully functional vehicle. [25] Cutting and Forming: Metal sheets or profiles are appropriately bent and cut.

- **Welding:** Screws or welding are used to link parts. Wheels and chassis are put together.

- **Assembly:** Is the process of putting wheels, engines, and other mechanical parts together.

After the design of RASIM was completed, the workspace of HSC ULV was used to start the production process. At first, the square profiles of the chassis were cut to specified dimensions and welded. Then the wheels were mounted to the motors. The connections with the chassis were made using welding. Afterwards, 4 ball casters were fixed to the bottom cover. The most reasonable products available in the company were used. The bottom cover was fixed with screws. Profiles were cut and welded for the scissor system. The dimensions were made suitable in the Miller lathe machine. After the scissors were finished, the linear actuator motor was mounted on the chassis. The total mass of the vehicle was 15,715 kg (excluding chassis and motors linear actuator). The weight of the hoist is 2.875 kg. The total weight is around 20 kg (including cables and software devices).

2.11 System Configuration

The configuration of the vehicle forces is shown in figure IV. It is seen that the two sprockets in the middle of the AGV are driving wheels, which are actuated separately by two DC motors. So, there are two trajectories of line and arc for this kind of AGV. The center of linking line between two driving wheels is the kinematics origin of AGV. [26] Design and Development of an Automated Guided Vehicle for Educational Purposes, Khosro Bijanrostami Submitted to the Institute of Graduate Studies and Research in partial fulfillment of the requirements for the Degree of Master of Science in Mechanical Engineering Eastern Mediterranean University September 2011 Gazimağusa, North Cyprus,

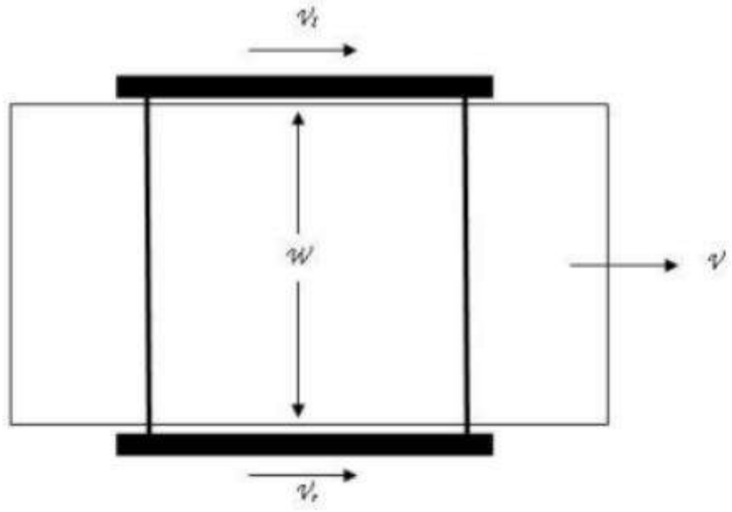


Figure IV: Vehicle Forces

V= linear AGV velocity (mm/s) = angular AGV velocity (rad/s) W = distance between two sprockets = 400mm Vr = velocity of right wheel (mm/s) Vl = velocity of left wheel (mm/s).

3.1 AGV Types

Automated Guided Vehicles (AGVs) come in many forms with various attributes and applications. While many attempts have been made to describe them, there is no official definition. According to Müller (1983), an AGV is a driverless transport system used for the horizontal movement of materials. Ganesharajah, Hall, and Sriskandarajah (1998) define AGVs as battery-powered, driverless vehicles that are centrally computer-controlled and independently addressable, used for moving jobs between workstations on a factory floor. Despite numerous definitions, the common characteristic of an AGV is that it is a vehicle that moves materials horizontally without a human driver. This can occur in production environments, warehouses, hospitals, or other locations where material transportation is necessary.

Given the wide range of applications, AGVs can vary significantly in appearance and attributes. Ullrich (2015, p.132) suggests that the best way to categorize AGVs is by the loads they transport. This chapter describes and illustrates various AGVs to provide an overview of their potential applications. Initially, three types of pallet handling AGVs are described: specially designed forklift AGVs, forklift AGVs as automated serial equipment, and piggyback AGVs. Subsequently, the trailer-handling AGV, the towing vehicle, is discussed, followed by the underride AGV, which handles roller containers. Finally, the assembly AGV, which carries assembly objects, the heavy load AGV, which transports heavy objects, and the diesel AGV, which operates outdoors, are described. Although Ullrich (2015, p.132) also mentions mini-AGVs and people movers, these are not included in this discussion due to their rarity and irrelevance to this project.

3.1.1 The Forklift AGV

3.1.1.1 *Specially Designed Forklift AGV*

Specially designed forklifts can look as in figure 3.1. This vehicle's load unit is pallets or forklift compatible containers. It can be used independently or with other AGVs, then managed by an AGV guidance control system (Ullrich, 2015 p.133).

Figure 3.1 - Specially designed forklift AGVs (AGVE, 2017)

3.1.1.2 *Forklift AGV as Automated Serial Equipment*

A Forklift AGV as automated serial equipment has the same qualities as Specially designed forklift AGVs but with the difference that it has space for a human to sit or stand in it and steer it manually (Ullrich, 2015 p.134), see figure 3.2 for an example. The advantage of a Forklift AGV as automated serial equipment is that it can be made of serially produced vehicles from forklift manufacturers' standard ranges. The serially produced vehicle is enhanced with safety equipment, guidance and navigation components. Some series are more appropriate to automate than others as there needs to be space in the forklift to add all the extra components. The sensors also need to be able to be placed so that the necessary parts of the AGV are covered (Ullrich, 2015 p.134). According to Ullrich (2015, p.134) there are two camps in the business regarding forklift AGVs. Those who prefer Specially designed forklift AGVs and those who see more advantages in Forklift AGVs as automated serial equipment. He argues that advantages with the first category are that they are designed for permanent use and extended service life, have optimal integration of all additionally needed components, i.e. no space problems, and that they account for an automation-compatible energy concept with the possibility of automatic battery change or charging. Forklift AGVs as automated serial equipment on the other hand come with cost advantages through serial manufacture and have proven service and replacement part availability (Ullrich, 2015 p.134-135).

Figure 3.2 - A serial produced forklift AGV (Toyota-forklifts.eu, 2017)

3.1.2 The Piggyback AGV

A piggyback AGV can look as in figure 3.3. These AGVs can carry pallets, boxes or containers. In contrast to the forklift AGVs mentioned in the previous subchapter this AGV cannot lift the load directly from the floor but requires a certain height which must be maintained throughout the loading and unloading areas of the AGV. The advantage of this type of AGV is their lateral load handling. They can drive up to the loading area and directly transfer the load without turning and maneuvering as a forklift would have to. This results in less space needed for loading operations and can be done quickly using the conveyor belts (Ullrich, 2015 p.136).

Figure 3.3 - Piggyback AGV. Source Frog according to Ullrich (2015, p.135)

3.1.3 The Towing Vehicle

Towing vehicles tow several trailers behind it. Just as with forklift AGVs there are two categories when it comes to towing AGVs. Either they can be specially designed towing AGVs without a space for a human, see figure 3.4, or they can be towing AGVs as automated serial equipment, see figure 3.5 (Ullrich, 2015 p.136). These two categories are by Ullrich (2015, p.136) not separated as trailer-towing vehicles according to him are considerably less common than forklift AGVs.

Figure 3.4 - A specially designed towing vehicle. Source dpm according to Ullrich (2015, p.136)

3.1.4 The Underride AGV

An underride AGV goes under a roller cart or material wagon and lifts it slightly, see figure 3.6 (Ullrich, 2015 p.39). Some underride AGVs can read a transponder on the bottom of the container to get instructions on what the specific cart contains and where it is going (Ullrich, 2015 p.72). Ullrich (2015 p.70) points out that this type of AGV has many advantages over for example forklift AGVs. They require less space than many other AGVs as the container itself determines the space required almost entirely. It also has high maneuverability when loading and unloading. This is the standard AGV in environments such as hospitals (Ullrich, 2015 p.132). An underride AGV can also be used as a towing vehicle. In this case the AGV is equipped with a cylinder (Ullrich, 2015 p.46) or a pickup thorn on the upper side of the AGV that hooks into the roller cart, which remains on its wheels while it is being transported (Ullrich, 2015 p.69).

Figure 3.6 - An underride AGV (Swisslog.com, 2017)

3.1.4 The Assembly Line AGV

Assembly line AGVs carry the assembly object on a path where the object is being constructed during transport. Assembly line AGVs are quite different from transport AGVs. Here the assembly object, its size and weight are very important when choosing AGV. The assembly stages are also important to take into consideration. These AGVs usually have simpler navigation systems than other AGVs and move at a slower pace. Personnel safety also works differently as workers are constantly in direct proximity to the AGV (Ullrich, 2015 p.137). An example of an assembly line AGV can be seen in figure 3.7.

Figure 3.7 - An assembly line AGV (Airfloat.com, 2017)

3.2 Electrical and Electronic Design of AGV System

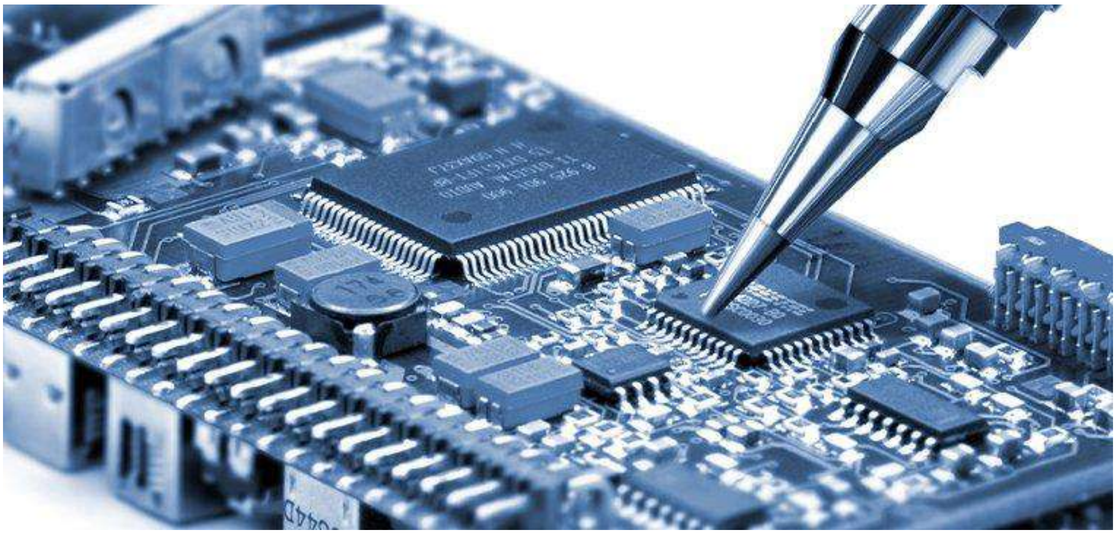


Figure 21 - PCB Cart

Plated through-holes, commonly known as vias, are holes that facilitate connections between the layers of a circuit board, allowing the transmission of electric current. Leaded circuit components mounted on the board are inserted into these holes, serving as pathways for interlayer transmission. In surface mount technology, there are copper or solder-coated pads on which the components rest, providing a flat surface. Some conductive copper regions are interconnected by waterways, which conduct electric current. Thicker waterways are used for higher currents, while thinner ones suffice for lower currents. Printed Circuit Board (PCB) design is typically done using computer programs. Well-known PCB design software includes OrCAD, Proteus (ISIS and ARES), Protel, and PCAD. Considerations in PCB design include trace spacing, via widths, placement of bypass capacitors, positioning of radio frequency-emitting components to minimize interference with integrated circuit elements, isolation using a Faraday cage if needed, and the separation of digital and analog ground planes. Materials like copper-clad laminates, where copper is bonded to materials such as fiberglass, are known as copper-clad FR4. The circuit schematic is transferred onto this material in the form of a thin film. Using a resistive ink or similar substance, conductive paths are drawn on the board, outlining the desired circuitry. The board is then exposed to a specialized acid mixture that selectively removes exposed copper, leaving behind the protected traces. After washing off the remaining ink layer, the circuit board is ready for use.

3.3 Power Electronics Definition

Power electronics is a branch of science that primarily focuses on controlling the energy supplied to a load and transforming different forms of energy. It is a highly attractive and significant discipline within Electrical Engineering. Power electronics requires a fundamental understanding of Mathematics, Circuit Theory, and electronic principles. Power electronics is one of the most crucial branches in the rapidly expanding electronic sector. Initially confined to industrial applications, power electronics circuits and devices have now extended beyond industry boundaries, entering households, offices, and vehicles in today's world.

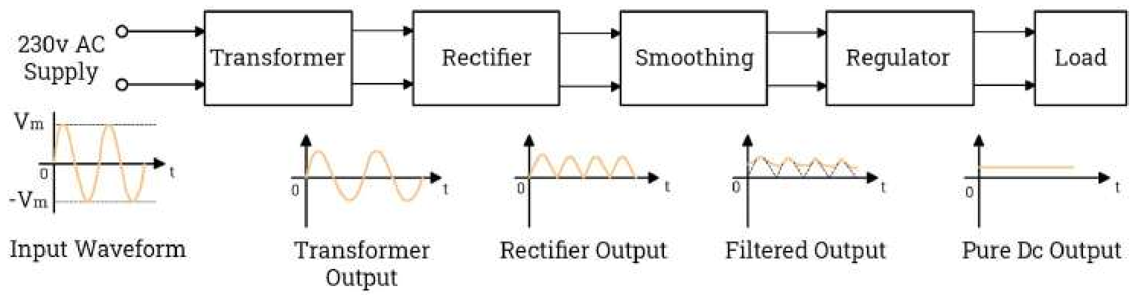


Figure 22 – AC / DC Power Supply Block Diagram

3.4 Power Electronics: Applications and Industrial Usage

3.4.1 Power Electronics System

The fundamental structure of a power electronics system is illustrated in block form. As depicted in the figure, a power system consists of two main sections: the power circuit and the control circuit.

3.4.2 Application Areas of Power Electronics

- **In Industry:**
  - In energy transmission and distribution.
  - For motor control in various applications.
  - Automation of machinery.
  - Heating and cooling processes.
  - In production and assembly industries.
  - Power supply and control systems of various types.
- **In Buildings, Offices, and Homes:**
  - Building automation systems.
  - In heating, cooling, ventilation, and security systems.
  - All office equipment.
  - Entertainment, sports, and gaming devices.
  - Medical devices.
  - Home appliances such as washing machines, dishwashers, refrigerators, air conditioners, vacuum cleaners, etc.
- **In Transportation Vehicles:**
  - Power systems in aircraft and other air vehicles.
  - In railway and metro vehicles and ground systems.
  - Electronic systems in heavy vehicles like buses, trucks, and tractors.
  - Electrical power and control systems in automobiles.
  - In vehicles such as forklifts, mobile cranes, concrete mixers, etc.
- **In Agriculture and Livestock:**
  - Automation and control in greenhouse systems.
  - Open-field irrigation and product control systems.
  - Automation of agricultural machinery.
  - Maintenance and automation in animal farming facilities such as poultry farms and barns.
  - Control of field and land cultivation, planting, and fertilization processes.
  - Applications in seed cultivation and seedling growth studies.

3.4.3 Industrial Applications of Power Electronics

The fundamental static and dynamic industrial applications of power electronics, along with other important industrial application areas, are outlined below.

- **Basic Static Applications:**
  - Uninterruptible Power Supplies (UPS).
  - Switched-Mode Power Supplies (SMPS).
  - Resonant Power Supplies (RMPS).
  - Induction Heating (IH).
  - Electronic Ballasts (EB).
  - High Voltage DC Transmission (HVDC).
  - Static VAR Compensation (SVC).
- **Basic Dynamic Applications:**
  - General DC Motor Control.
  - General AC Motor Control.
  - Control of Squirrel Cage (Short Circuit Rotor) Induction Motors.
  - Control of Wound Rotor (Slip Ring Rotor) Induction Motors.
  - Linear Induction Motor Control.
  - Synchronous Motor Control.
  - Universal Motor Control.
  - Stepper Motor Control.
  - Reluctance Motor Control.
- **Other Significant Applications:**
  - Lighting and Light Control Systems.
  - Heating and Cooling Systems.
  - Soldering and Welding Systems.
  - Melting and Hardening Systems.
  - Screening and Grinding Systems.
  - Elevator and Crane Systems.
  - Escalator and Conveyor Systems.
  - Pump and Compressor Systems.
  - Ventilation and Fan Systems.
  - Alternative Energy Source Systems.

- Battery Charging and Energy Storage Systems.
- Electric Transportation and Electric Vehicle Systems.
- Space and Military Vehicle Systems.
- Earth Excavation and Mining Systems.

Additionally, Power Electronics is extensively integrated into interdisciplinary fields such as Industrial Automation, Mechatronics, and Robotics.

3.5 Lithium-Ion Battery Technology

Research and development activities on secondary (rechargeable) lithium-ion batteries are gaining increasing significance due to the growing demand for portable electric and electronic devices, as well as the rising importance of electric cars in the transportation sector. Lithium-ion batteries, with their high energy density and non-toxic nature, have become preferred choices for mobile phones, laptops, small household appliances, and are favored by environmentally conscious designers and consumers due to their low CO2 emission levels. In the past, lithium-ion batteries, commonly used in portable household appliances, have seen their importance rise with the increasing need for energy storage systems amidst diminishing energy resources. This trend is expected to further escalate with the increasing production of electric cars soon. This article provides a brief introduction to lithium-ion battery technology, discussing its history and the research and development efforts that have been undertaken.

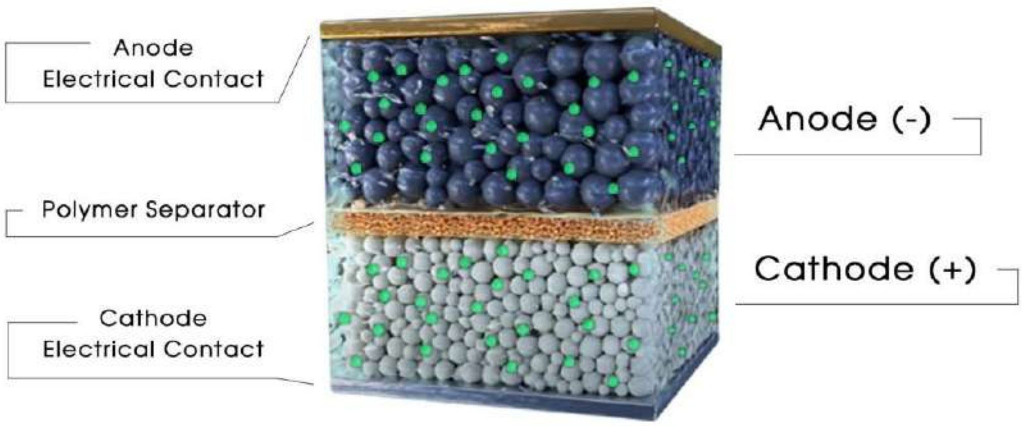


Figure 23 - Lithium-Ion Batteries

3.6 What is a Lithium-Ion Battery?

Rechargeable batteries are commonly known as secondary batteries. They are electrochemical cells that can be recharged and reused after discharge. Below provides a general comparison of lithium-ion batteries with other secondary batteries such as silver-zinc, nickel-zinc, and nickel-hydrogen, highlighting their advantages and disadvantages.

Advantages:

- Maintenance-free due to a sealed cell design.
- Long lifespan.
- Wide operating temperature range.
- Extended shelf life.
- Rapid charging capability.
- High-power discharge capacity.
- High energy efficiency.
- High specific energy and energy density.
- Lack of memory effect.

Disadvantages:

- Cost.
- Degradation at high temperatures.
- Requirement for protective circuits.
- Capacity loss or thermal degradation due
- to overcharging.

3.7 Working Principle of Lithium-Ion Battery:

In rechargeable lithium-ion batteries, cells, responsible for generating and storing energy like in other battery systems, consist primarily of three main components: anode, cathode, and electrolyte. These components play crucial roles in the overall functionality of the battery.

The anode serves as the negative electrode, while the cathode functions as the positive electrode. Positive electrodes typically consist of metal oxides with tunnel or layered structures (Limo). On the other hand, negative electrode materials exhibit layered structures. These structures enable the mutual exchange of Li ions between the positive and negative electrodes during the charge and discharge of the cell/battery. This displacement is defined as a topotactic reaction.

In this reaction, the active materials in the anode and cathode host lithium, which moves as a guest from one electrode to another. Figure XI provides a schematic representation of a lithium-ion cell and the reactions occurring at the negative and positive electrodes.

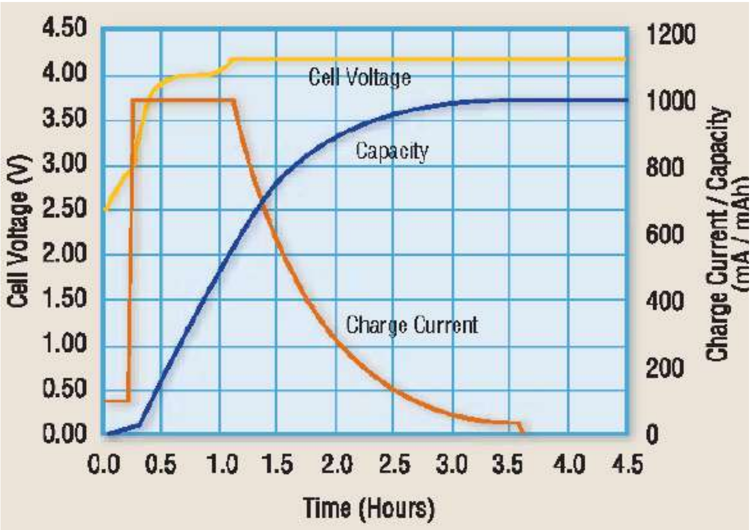


Figure 24 - The charge profile for a Li-Ion battery consists of four stages

3.8 How Lithium-Ion Batteries Are Charged and Discharged?

Firstly, let's define some terms: "Capacity" indicates the maximum amount of energy a battery can store and is given in Ampere-hours (Ah); for example, a 10Ah battery can provide a current of 10 Amperes for 1 hour or 1 Ampere for 10 hours. The "state of charge" reflects how much capacity of the battery has been filled. "Nominal voltage" indicates the voltage a fully charged battery provides under normal conditions. "Terminal voltage" is the voltage the battery provides at a specific moment. A notable characteristic of lithium-ion batteries is that their state of charge is related to the terminal voltage, allowing us to estimate the state of charge by measuring the terminal voltage. Lastly, the manufacturer's specified "current constant" (C) is used as an indicator of the battery's instantaneous power, representing how quickly the battery can be discharged or the maximum instantaneous current it can deliver. For instance, a 1C battery can be discharged from full to empty in 1 hour, while the commonly available 45C batteries on the market can be discharged in approximately 1.5 minutes. A 5000mAh 45C battery can deliver a current of 225 Amperes, which is a significant power that can have serious consequences if not handled carefully.

Charging currents range from 0.5C to 2C, while the capacity of the battery and the current constant are largely independent of each other.

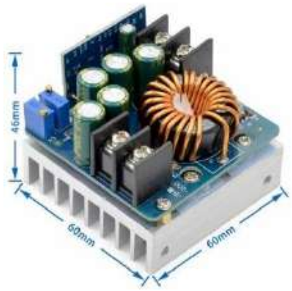
Understanding the state of charge from the terminal voltage makes charging lithium-ion batteries relatively straightforward. To charge the battery, a constant current of 1C is applied until the battery reaches its "maximum terminal voltage." Once this voltage is reached, the voltage is stabilized, and the current is allowed to decrease slowly until the battery is fully charged.

However, there are two crucial points to consider during charging and discharging. Firstly, fully charged and fully discharged terminal voltage values are critical. While charging, the battery is not fully charged at 4.20V terminal voltage, and if it exceeds the maximum terminal voltage allowed for these batteries, which is 4.26V, it can catch fire. Similarly, if it drops below 3V, it can be damaged beyond repair. The second point is a bit more complex: Suppose we need a battery for our laptop with a nominal voltage of 7.4V. We can achieve this by connecting two lithium-ion batteries in series. When charging and discharging these batteries, looking only at the total voltage, we can see that the battery works normally for a while, but as we charge and discharge it, it begins to deteriorate. This is because the capacities of the two batteries connected in series are not equal. As we charge and discharge the battery, the battery with lower capacity reaches its maximum terminal voltage of 4.26V earlier. Since the battery with larger capacity is still less full at that time, its terminal voltage is lower. When the total voltage reaches the maximum value of $2 \times 4.26 = 8.52$V, the battery with smaller capacity exceeds its maximum terminal voltage and gets damaged. Therefore, in applications where a high voltage is created by connecting many lithium-ion batteries in series (such as electric cars), circuits are placed on each battery terminal to monitor the voltage and eliminate excess current if necessary. This may require quite complex systems, and it is one of the challenges in designing electric vehicles. There is such a device in the batteries of your mobile phone and computer, and the battery has more than two terminals to allow the phone to communicate with this device.

The component design employed in this study is essential to the design and functionality of an Automated Guided Vehicle (AGV). AGVs are self-propelled vehicles used for material handling and transportation in industrial environments. The electronic components integrated into the design serve various functions that enable the navigation and autonomous operation of the AGV. These components are critical in ensuring the AGV's efficient and reliable performance in diverse industrial settings.

1. **Battery (DC 12 Volt):** The battery serves as a power source for the AGV, providing the electrical energy needed to drive its various components.

2. **Regulator:** The regulator ensures a stable voltage supply to the AGV's electronic components by maintaining a constant output voltage despite variations in input voltage.

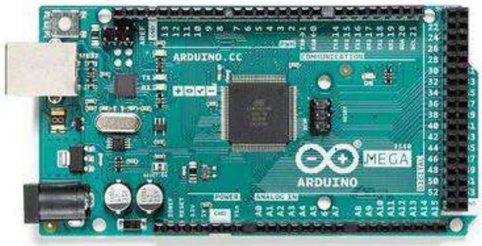


3. **Motor Driver:** The motor drive controls the speed and direction of the AGV motor, translating the input voltage (9-36V DC) to the appropriate output voltage (12V DC) required for motor operation.

4. **DC Motor:** The DC motor is responsible for the mechanical movement of the AGV, converting electrical energy into kinetic energy to propel the vehicle.

5. **Ultrasonic Sensor (HC – SR04):** The ultrasonic sensor measures the distance between the AGV and obstacles in its path using ultrasonic waves, which is crucial for navigation and collision avoidance.

6. **Arduino Mega 2560 R3 Board:** The Arduino Mega 2560 R3 board acts as the central microcontroller, processing input from various sensors and controlling the actuators to enable autonomous operation of the AGV.



7. **Quadruple Line Follower Sensor:** The line follower sensor detects and follows lines on the ground, allowing the AGV to navigate along predefined paths.

8. **Jetson – Nano:** The Jetson Nano is a compact AI computer that provides advanced processing power for machine learning and image processing tasks, enhancing the AGV's autonomous capabilities.

9. **Linear Actuator:** The linear actuator is used for precise linear movements, enabling the AGV to perform specific tasks such as lifting or moving objects.

10. **Fan:** The fan provides necessary cooling to prevent overheating of the electronic components, ensuring optimal performance and longevity.

11. Car Battery (12 Volt): The car battery acts as an additional power source, providing reliable and long-lasting power to support the AGV's operations, especially in demanding environments.

3.9 Cost Analysis

Mechanical Components	Quantity	Unit Price	Total Price	Manufacturer / Seller
Box Profile	8	400	₺ 3.200,00	HSC
Sheet Metal	1	5000	₺ 5.000,00	HSC
Drive Wheel	2	200	₺ 400,00	HSC
Support Wheel	4	100	₺ 400,00	HSC
Wiper Motor	2	2634,34	₺ 5.268,68	HSC
Lead Screw	2	400	₺ 800,00	HSC
Linear Actuator	1	1500	₺ 1.500,00	EBB Mekatronik
Mechanical Labor	1	2000	₺ 2.000,00	HSC
Vehicle Coating	1	6000	₺ 6.000,00	Eren Architecture
Electronic Components	**Quantity**	**Unit Price**	**Total Price**	**Manufacturer / Seller**
Nvidia Jetson Nano Developer Kit	1	13567,2	₺ 13.567,20	Open Zeka
40A Motor Driver	2	380	₺ 760,00	Robocombo
12V 12A Battery	1	2735	₺ 2.735,00	Akü Dünyası
Battery Charger	1	750	₺ 750,00	Akü Dünyası
Stop Button	1	100	₺ 100,00	Robocombo
Logitech Camera	1	2000	₺ 2.000,00	Trendyol
3S Lithium Ion Battery	6	333,33	₺ 1.999,98	Robocombo
BMS	1	200	₺ 200,00	Sedu Aydınlatma ve Elektrik
40W Voltage Step-Down Module	1	702	₺ 702,00	Sedu Aydınlatma ve Elektrik
Cabling Cost	1	2000	₺ 2.000,00	Eroğlu Kablajsan
HC-SR04 Ultrasonic Distance Sensor	4	42,82	₺ 171,28	Robocombo
Arduino Mega 2560 Rev3 (Orijinal)	1	1713,97	₺ 1.713,97	AGM Bilişim
12V DC Fan	1	54,74	₺ 54,74	Robocombo
4-line Follower Sensor Set	1	110	₺ 110,00	Robocombo
Jumper Cable (40 pcs)	3	40	₺ 120,00	Robocombo
Electrical Tapes	5	130	₺ 650,00	Koçtaş
Shrink Tube Set	1	70	₺ 70,00	Robocombo
Perforated Board	2	15	₺ 30,00	Robocombo
Total Price			**₺ 52.302,85**	

Table 4 Cost Analysis

4.1 Software Parts of The AGV System

4.11 Collision Avoidance

The collision avoidance system implemented in the autonomous warehouse robot is a sophisticated integration of sensor technologies and dynamic trajectory adjustments. Utilizing a combination of ultrasonic and infrared sensors, the robot continuously scans its environment to detect both fixed and moving obstacles. These sensors have proven effective in various studies. When an obstacle is detected, the robot's algorithms calculate the optimal path adjustment, allowing it to either maneuver around the obstacle or come to a complete stop. This real-time response ensures the safety of the robot and the goods it transports, minimizing the risk of collisions in the dynamic warehouse setting.

4.12 Line Tracking Management

The robot's navigation is primarily based on a robust line tracking system. Algorithms continuously analyze the feedback from line sensors to ensure the robot stays on the designated path. In cases where the line is interrupted or deviates due to unforeseen circumstances, the robot employs advanced algorithms to autonomously re-establish the correct path. This capability enables the robot to navigate reliably even in situations where the lines may be partially obscured or distorted.

4.13 Dynamic Path Planning

To adapt to unforeseen obstacles or changes in the warehouse layout, the autonomous robot employs dynamic path planning algorithms. These algorithms continuously analyze the surroundings, considering real-time data from sensors. By calculating alternative routes, the robot can efficiently navigate around obstacles and optimize its path to reach its destination in the most time effective manner. This adaptability is crucial for the robot's efficiency in a dynamic and evolving warehouse environment.

4.14 Image Processing for Parking

Upon reaching the delivery or charging point, the robot utilizes advanced image processing techniques for optimal parking. Equipped with cameras, the robot captures images of its surroundings and analyzes them to identify the most suitable parking area. This involves recognizing the available space, avoiding other objects or robots, and ensuring a secure and efficient parking maneuver. This image processing capability enhances the robot’s ability to operate seamlessly within the warehouse infrastructure.

4.15 Object Recognition and Waiting State

For safety during the waiting state between objects or vehicles, the robot incorporates sophisticated object recognition capabilities. These capabilities allow the robot to identify and differentiate between various objects, ensuring that it parks in a secure location. The waiting state is initiated when the robot is positioned between two objects or vehicles, providing a safe and strategic location within the warehouse. This proactive safety measure prevents potential collisions during idle periods.

4.16 Emergency Stop Mechanism

As a fail-safe measure, the autonomous robot is equipped with a rapid emergency stop mechanism. In the event of critical errors, system malfunctions, or unforeseen circumstances jeopardizing safety, the robot can execute an immediate emergency stop. This halts all movements, preventing potential accidents or damage to the robot, goods, or surrounding infrastructure. The emergency stop mechanism aligns with industry standards for safety in autonomous systems.

4.2 Emergency Stop Mechanism

In the context of our autonomous warehouse robot project, where open computing environments play a pivotal role, the inherent susceptibility to diverse security threats necessitates a comprehensive understanding of potential risks. This paper has undertaken a survey of security threats prevalent in such systems, shedding light on the challenges that can impede the seamless operation of our autonomous robot. Additionally, we have delved into an exploration of prominent protection mechanisms devised to counteract these threats. Our investigation reveals a myriad of proposed techniques; however, it is crucial to acknowledge that not all of these solutions are equally practical within the dynamic landscape of autonomous warehouse robotics. Striking a balance between robust security measures and operational feasibility poses a continual challenge in the implementation of protection mechanisms. Despite the strides made in security research, it is evident that there is still work to be done. Remaining security issues must be satisfactorily addressed to ensure the widespread acceptance and realization of the full potential of autonomous warehouse robotics. The dynamic and unpredictable nature of warehouse environments demands adaptive and resilient security solutions. As we progress with our autonomous warehouse robot project, it becomes increasingly apparent that ongoing research is imperative. Through a commitment to innovation and continuous improvement we can fortify our system against emerging threats, ensuring not only the safety and security of the robot but also facilitating its seamless integration into warehouse operations. In this pursuit, our project aligns with the broader goal of advancing the mobile agent paradigm and contributing to the evolution of autonomous systems in open computing environments.

4.3 Software Requirement Specifications (SRS)

4.3.1 Introduction

Automated Guided Vehicle (AGV) technology provides significant advantages in carrier loading, finding widespread applications in warehouses, production lines, and distribution processes. The objective of this project is to develop a reliable and effective AGV surveillance system by addressing challenges encountered in the normal navigation routes of AGVs.

4.3.2 Purpose

The fundamental purpose of this project is to enhance safety in logistics and production processes by mitigating potential risks caused by deviations in the navigation routes of AGVs. Additionally, the developed surveillance system will enable more accurate prediction of AGV movements, facilitating rapid alarm notifications in the presence of abnormal situations. This is crucial for ensuring the safety of warehouse personnel and contributing to the seamless progression of production processes.

4.3.3 Scope

This project focuses on safety and efficiency issues in logistics and production processes where AGVs are utilized. The scope of the project includes the following areas:

- **Safety of AGV Trajectories:** Prevention of collision risks, chaotic situations, and disruptions in production processes arising from deviations in the normal routes of AGVs.

- **Surveillance System Design:** Design and implementation of an AGV surveillance system, incorporating forward and downward-facing cameras.

- **Image Processing and Fusion Strategy:** Calibration of coordinates between forward and downward-looking cameras, development of image processing, and fusion strategies.

4.4 General Description

4.4.1 Glossary

Name	Definition
Actor	Entities utilizing the automatic parking system, including users, environmental sensors, other software and hardware components, Arduino, ROS, etc.
Object Detection	The ability to recognize obstacles and lines in digital images, video feeds, and environmental sensor data.
Computer Vision	The capability to analyze visual and environmental data to comprehend surroundings and recognize objects.
YOLO Models	Deep learning algorithms used for object detection.
C	A general-purpose programming language suitable for fulfilling low-level control and processing requirements in the project.
C++	A general-purpose programming language supporting object-oriented programming, applicable for complex algorithms in the project.
ROS	Robot Operating System; an open-source software framework for developing and managing robotic applications.
Arduino	A popular microcontroller board platform for embedded system projects.

Table 5 – Glossary

4.4.2 Functional Requirements Overview: AVG Project

In the quest to develop a state-of-the-art autonomous warehouse vehicle, functional requirements were meticulously crafted to ensure efficient cargo handling, obstacle avoidance and autonomous parking. The project includes advanced features such as line recognition, cargo collection and transportation, obstacle detection, path planning and precise autonomous parking. Below is an overview of the functional requirements

4.4.2.1 Line Tracking and Cargo Transportation:

- **Line Recognition:** Computer vision techniques should be applied to accurately detect and follow designated lines in the warehouse environment.
- **Cargo Receiving:** Autonomous capabilities will be developed for the vehicle to approach the cargo, recognize it, and safely remove it from a specific pickup point.
- **Cargo Transportation:** Stability and safety during cargo transportation is ensured by using a combination of sensors and robust control algorithms.

4.4.2.2 Unblocking:

- **Obstacle Detection:** We used a combination of computer vision and sensor-based systems to detect obstacles in the vehicle's path.
- **Route Planning:** We applied dynamic path planning algorithms to enable the vehicle to navigate around obstacles and optimize the route to the destination.

- **Emergency stop:** An emergency stop mechanism will be integrated to reduce possible collisions in unforeseen situations.

4.4.2.3 Autonomous Parking:

- **Parking Zone Recognition:** Sensors and computer vision will be used to recognize designated parking zones within the warehouse.
- **Precision Parking:** Precise autonomous parking techniques will be developed to ensure correct positioning of the vehicle in designated parking areas.
- **Automatic Shutdown:** We will implement a system that will enable the vehicle to shut down autonomously after the parking process is successfully completed.

4.4.2.4 System Integration and Communication:

- **Integration with Central System:** We will establish seamless integration with a central system to receive pick-up and drop-off instructions.
- **Communication Protocols:** Implement reliable communication protocols for real-time data exchange between the vehicle and the central system.

4.4.2.5 Safety and security:

- **Emergency Protocols:** Define protocols and mechanisms for handling emergencies, including safe cargo evacuation and systematic shutdown.
- **Security Precautions:** Implement security features to prevent unauthorized access and ensure the integrity and security of cargo during transportation.

4.4.3 Brief Introduction of The Aruco System and The Process of Visual Guidance

ArUco markers have a square shape, featuring a black border, an inner grid was utilised to record a binary-coded number identifier. The marker is named using a dictionary. A set of guidelines for calculating marker identification, performing validation, and implementing error correction are defined in the dictionary. We employ the original ArUco dictionary, in which the marker's unique ID is stored in natural binary code. using bits from the marker's second and fourth columns and uses the remaining bits for parity testing. The first four markers in this vocabulary are shown in Fig 8

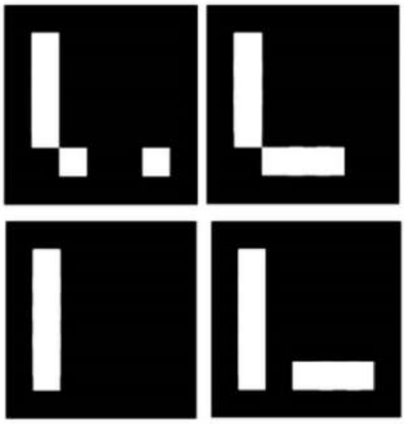


Figure 25 – Markers with ArUco isd 0,1,2,3

The marker is verified using a signature matrix. This matrix's rows each encode a potential pair of two bits. An ArUco marker has only one valid rotation since it is only valid if each of its rows equals one of the signature matrix's rows. The matrix of signatures utilized in this dictionary is shown in Table .4.

Value	Data				
0	1	0	0	0	0
1	1	0	1	1	1
2	0	1	0	0	1
3	0	1	1	1	0

Table 6 - Signature matrix for validating ArUco markers

The marker 1023 in Fig 9. is symmetric horizontally, demonstrating that it is not sufficient to ensure that there is only one potential rotation for each marker. This may be confirmed by evaluating the signature matrix.

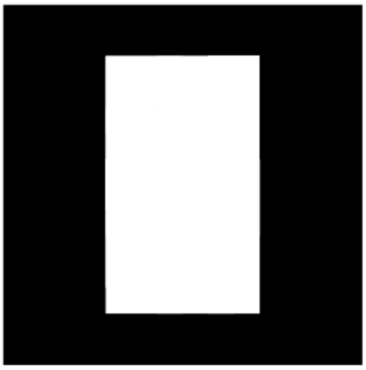


Figure 26 – ArUco Marker 1023

4.4.4 General Constraints and Assumptions: AVG Project

When defining the framework for an autonomous warehouse vehicle project, we consider various constraints and make certain assumptions that guide the development process. These constraints and assumptions serve as fundamental parameters, providing clarity and context for the scope and implementation of the project.

4.4.4.1 Constraints:

- **Hardware Realities:** This project operates within the limits of available hardware resources, including sensors, actuators, and computing capabilities.
- **Budget Realities:** Financial constraints shape projects and require adherence to a given budget throughout development and deployment.
- **Compliance:** Vehicle design and functionality must comply with existing safety and regulatory standards for autonomous systems in warehouse environments.
- **Temporal Pressures:** A predefined timeline dictates the project's pace, necessitating efficient development, testing, and deployment phases.
- **Operational Context:** Tailored for warehouse use, the autonomous vehicle's operation is constrained by the distinctive characteristics and layout of such environments.

4.4.4.2 Assumptions:

- **Stable Illumination:** The project assumes a consistent and sufficient level of lighting within the warehouse for the effective operation of computer vision systems.
- **Obstacle Variety:** Both static and dynamic obstacles are assumed to exist within the warehouse, and the vehicle is designed to adapt to diverse scenarios.
- **Clearly Defined Markings:** Efficient line following presupposes the presence of distinct and discernible markings on the warehouse floor.
- **Charging Infrastructure:** The assumption is that a reliable charging infrastructure is in place within the warehouse to meet the autonomous vehicle's periodic charging needs.
- **Reliable Connectivity:** Stable and uninterrupted network connectivity is assumed to enable seamless real time communication between the autonomous vehicle and the central control system.
- **Navigable Pathways:** The project assumes the availability of unobstructed and navigable pathways within the warehouse for smooth cargo pickup, transportation, and delivery.
- **Localized Operations:** Designed for localized operations, the autonomous vehicle is limited to the defined warehouse space and doesn't consider external navigation.
- **Predictable Cargo Handling:** The project assumes a structured cargo handling process, with predefined pickup and drop-off points to facilitate efficient and automated operations.
- **Emergency Response:** The assumption is that established emergency protocols are in place, ensuring that the autonomous vehicle operates within safety parameters in unforeseen events.

4.4.5 Specific Requirements: AVG Project

The precise demands for the self-governing warehouse vehicle initiative encompass an exhaustive array of functionalities, attributes, and capacities designed to facilitate effective management of goods, seamless maneuvering around obstacles, and self-directed parking within the confines of the warehouse environment.

4.4.5.1 Route Following and Cargo Management

- **Path Recognition:** The automated vehicle is expected to employ sophisticated computer vision algorithms to accurately identify and adhere to designated pathways within the warehouse, ensuring precise navigation.
- **Cargo Retrieval:** A sophisticated mechanism is to be devised to empower the vehicle to independently approach predefined pickup locations, recognize cargo, and securely elevate it onto the vehicle.
- **Cargo Conveyance:** The system will integrate robust cargo handling mechanisms, leveraging sensors and control algorithms to ensure the steadfast transportation of cargo during the vehicle's traversal within the warehouse.

4.4.5.2 Obstacle Evading

- **Obstacle Identification:** A comprehensive suite of sensors, including lidar and ultrasonic sensors, is mandated to be deployed for the detection of both stationary and dynamic obstacles obstructing the vehicle's route.
- **Pathway Planning:** Dynamic algorithms for path planning will be implemented to enable the vehicle to adeptly navigate around obstacles, optimizing routes for the seamless transportation of cargo.
- **Emergency Halting:** An integrated emergency halt system will be put in place, activated by the proximity of obstacles, ensuring immediate cessation to prevent collisions and uphold warehouse safety.

4.4.5.3 Self-Directed Parking

- **Recognition of Parking Zones:** The vehicle is slated to utilize a blend of sensors and computer vision for the identification and localization of designated parking areas within the warehouse.
- **Precision Parking:** Cutting-edge algorithms will be formulated to facilitate meticulous self-directed parking, enabling the vehicle to position itself with precision within specified parking zones.
- **Automated Power Down:** Upon the successful completion of parking, an autonomous shutdown procedure is to be instituted, optimizing energy consumption and ensuring operational efficiency.
-

4.4.5.4 Integration with Central System and Communication

- **Incorporation with the Central System:** The autonomous vehicle is required to seamlessly integrate with a central control system, facilitating efficient communication and coordination of tasks related to pick up, transportation, and delivery.
- **Real-time Data Exchange:** Implementing reliable and low-latency communication protocols is imperative to facilitate real-time data exchange between the autonomous vehicle and the central control system.

4.4.6 Safety and Security Measures

- **Emergency Procedures:** A comprehensive set of emergency procedures will be articulated and implemented to guarantee the safety of both the vehicle and the warehouse environment in unforeseen or emergency scenarios.
- **Controlled Access:** Her system is mandated to integrate robust access control measures, preventing unauthorized access to the vehicle's control systems and cargo, thereby ensuring the security of operations.

4.4.7 User Interface Design

- **Remote Surveillance:** A user-friendly interface is to be devised for remote surveillance, empowering users to monitor the vehicle's status, oversee cargo, and receive real-time updates on operational parameters.
- **Manual Override Option:** The system is to incorporate an alternative for manual override, granting operators the capacity to assume control in situations necessitating manual intervention, thus ensuring adaptability and flexibility.

4.5 Use Case Scenarios for The AVG Project

4.5.1 Cargo Pickup and Delivery

Objective:

- The vehicle picks up cargo from a designated pickup point and securely delivers it to a specified drop-off point.

Scenarios:

- The central control system initiates a cargo pickup request, specifying the pickup and delivery points.
- The vehicle autonomously navigates to the pickup point, adapting its speed to environmental variables such as traffic.
- Utilizing various sensors, the vehicle accurately identifies and picks up the cargo securely.
- During transit, the vehicle dynamically adjusts its route based on real-time environmental conditions.
- Upon reaching the delivery point, the vehicle safely unloads the cargo and completes the task.

4.5.2 Line Following

Objective:

- The vehicle accurately navigates by following designated lines within the warehouse.

Scenarios:

- The vehicle starts its operation and initializes the line-following algorithm.
- Using line-following sensors and cameras, the vehicle continuously tracks and adjusts its position relative to designated lines.
- In cases of poor line quality, the vehicle employs image processing algorithms to enhance line detection and maintain precise navigation.
- The vehicle responds to sudden line deviations or changes, ensuring continuous and accurate line following.

4.5.3 Obstacle Avoidance

Objective:

- The vehicle navigates safely within the warehouse, avoiding both static and dynamic obstacles.

Scenarios:

- The vehicle's sensors constantly scan the surroundings for obstacles, including both static structures and moving entities.
- Upon detecting an obstacle, the vehicle evaluates its position and motion to create a dynamic obstacle map.
- Using path planning algorithms, the vehicle devises an alternative route to circumvent obstacles efficiently.
- In critical situations where collision risk is high, the emergency stop system activates, bringing the vehicle to an immediate halt.

4.5.4 Autonomous Parking

Objective:

- The vehicle autonomously parks in designated areas within the warehouse.

Scenarios:

- The vehicle identifies available parking zones using a combination of sensors and image processing techniques.
- Utilizing precise maneuvering algorithms, the vehicle navigates to the designated parking area.
- During the parking process, the vehicle continuously monitors its surroundings, adjusting its position for optimal parking.

- Upon successful parking, the system updates the vehicle's status, optimizing energy usage, and automatically shutting down.

4.5.5 System Integration and Communication

Objective:

- The vehicle seamlessly communicates and integrates with the central control system.

Scenarios:

- The central control system issues real-time commands to the vehicle, such as initiating tasks or updating routes.
- The vehicle acknowledges received commands and provides constant updates on its status, location, and environmental conditions.
- In the event of communication disruption, the vehicle activates failsafe protocols, ensuring safe operation or stopping when necessary.

4.5.6 Safety and Security

Objective:

- The vehicle is prepared for any security threat and operates safely.

Scenarios:

- The vehicle is equipped with a comprehensive set of security protocols to handle emergencies or unforeseen circumstances.
- If a system error or security vulnerability is detected, the vehicle enters a safe mode, ceasing autonomous operations.
- Emergency buttons or manual control options are available for operators to swiftly intervene in critical situations.
- Continuous surveillance of the vehicle's surroundings ensures prompt threat assessment and avoidance.

4.6 Conclusion

To The development of the Autonomous Guided Vehicle Project has been guided by a comprehensive set of software requirements, encapsulating the diverse functionalities necessary for seamless cargo handling, obstacle navigation, and autonomous parking within a warehouse environment. The meticulous consideration of these requirements serves as the foundation for creating a robust and intelligent autonomous system.

4.6.1 Achievements

Throughout the specification process, the team has achieved a clear understanding of the project's scope, objectives, and critical functionalities. The definition of user roles, identification of system actors, and detailed requirements for object detection, computer vision, and motion planning laid the groundwork for the successful implementation of the autonomous vehicle.

4.6.2 Object Detection and Computer Vision

The integration of Object Detection, leveraging advanced computer vision techniques and YOLO models, empowers the vehicle to accurately identify objects, navigate designated paths using line recognition, and interact with its environment dynamically. The fusion of forward and downward-looking cameras, coupled with innovative QR code recognition through the downward camera, enhances the precision of pose estimation and ensures reliable navigation.

4.6.3 Motion Planning and Navigation

The autonomous vehicle's ability to navigate dynamically within the warehouse, avoiding obstacles and following marked paths, is a testament to the well-defined requirements in the realm of motion planning. The implementation of obstacle avoidance, path planning, and emergency stop protocols contributes to a robust navigation system that prioritizes safety and efficiency.

4.6.4 Autonomous Parking

The project's focus on autonomous parking is underscored by the development of precise parking algorithms, recognition of parking zones, and an automated shutdown procedure upon successful parking. These features not only optimize energy consumption but also enhance the vehicle's overall operational efficiency within the warehouse environment.

4.6.5 Future Considerations

While the current set of software requirements provides a solid foundation, continuous refinement and iteration will be essential to address emerging challenges and technological advancements. Future considerations may include the incorporation of additional sensors, optimization of image processing algorithms, and enhancements to the user interface for more intuitive control and monitoring.

4.6.6 Conclusion Statement

In conclusion, the Software Requirement Specification for the Autonomous Guided Vehicle Project serves as a comprehensive guide for the development team. The delineation of user characteristics, functional requirements, and system constraints ensures a systematic and well-informed approach to building an autonomous vehicle that excels in cargo handling, obstacle navigation, and parking within a warehouse setting.

5 Software Design Description (SDD)

5.1 Purpose

This document is created to elaborate on the software design of the Autonomous Warehouse Vehicle Project and define the software structures, algorithms, and components that will be used in the development stage of the project. The primary objective of our project is to optimize warehouse management processes and make material movement within the warehouse more efficient and reliable

5.2 Project Description

The Autonomous Warehouse Vehicle Project aims to automate cargo transportation and placement processes in warehouse environments. The starting point of this project is to reduce manual handling operations in the warehouse, optimize warehouse management, and strategically redirect human resources to more strategic tasks. The vehicle should be able to follow specified lines, detect obstacles, and park safely in a designated storage area.

5.3 Objectives

The main objectives of our project are as follows:

- **Line Following:** The vehicle should be able to follow a specific line drawn on the floor, allowing it to move along designated routes within the warehouse.
- **Obstacle Avoidance:** The vehicle should be capable of detecting obstacles in its surroundings using sensors and navigate around them safely or choose an alternative route.
- **Autonomous Parking:** The vehicle should be able to park safely in a designated storage area. This process should be accomplished through precise sensors and algorithms.
- **Safety and Efficiency:** The software should adhere to safety standards while executing the vehicle's movements and tasks. Additionally, it should operate efficiently to maximize material movement within the warehouse.

5.4 Definitions

Name	Definition
Autonomous Warehouse Vehicle	A vehicle that automates cargo transportation and placement tasks within a warehouse.
Line Tracking	The process of detecting and following lines drawn on the floor to enable the vehicle to move along specific routes within the warehouse.
Obstacle Avoidance	The capability of the vehicle to detect obstacles within the warehouse using sensors and avoid them safely.
Autonomous Parking	The ability of the vehicle to park safely in a designated storage area. This process is accomplished through precise sensors and algorithms.
Computer Vision	A subfield of Artificial Intelligence focused on recognizing and understanding objects using visual inputs.
YOLO Models	YOLO (You Only Look Once) Models are a family of object detection algorithms used in real-time computer vision applications.
Sensors	Hardware components on the vehicle used to collect environmental data for line tracking, obstacle avoidance, and autonomous parking processes.
Control Algorithms	Software algorithms that guide the vehicle's movements and coordinate tasks.
Database	An organized collection of structured data storing user settings, camera configurations, and statistical data for local processing in Insightio.
UI (User Interface)	The graphical or text-based interface facilitating interaction between the user and the application.
C and C++	General-purpose programming languages used for various applications, ranging from low-level operations to high-level applications.
Arduino	An electronic platform and programming language used for embedded systems and prototyping.
Jetson Nano	An embedded system and artificial intelligence computing platform produced by NVIDIA. Designed for high-performance computing and deep learning applications.

Table 7 – Project definitions

5.5 System Overview

5.5.1 Project Objectives

The Autonomous Guided Vehicle Project encompasses an automation system designed to perform cargo transportation and placement tasks within a warehouse. The primary objectives are as follows:

- Achieving autonomous pickup and delivery of cargo.
- Integrating line tracking, obstacle avoidance, and autonomous parking capabilities.
- Enhancing material movement within the warehouse to optimize operational efficiency.

5.5.2 System Architecture

The autonomous guided vehicle includes a set of integrated systems and components. The key components are as follows:

- **Line Tracking Module:** Detects lines drawn on the floor to enable the vehicle to move along specific routes.
- **Obstacle Detection Module:** Utilizes sensors to detect obstacles around the vehicle and develops strategies to safely avoid these obstacles.
- **Autonomous Parking Module:** Employs algorithms and precision sensors to park the vehicle safely in a designated storage area.
- **Control Algorithms:** Software algorithms coordinating various tasks and directing the vehicle's movements.

5.5.3 Software and Programming Languages

The project will utilize C and C++ programming languages, offering features suitable for general purpose programming and embedded systems. Additionally, the Arduino and Jetson Nano platforms will be integrated into the project.

5.5.4 External Libraries and Frameworks

External libraries such as OpenCV for line tracking and object detection and frameworks like YOLO (You Only Look Once) will be used in the project.

5.5.5 System Integration

The integration of the autonomous warehouse vehicle system involves the interaction of various components and modules. Line tracking, obstacle avoidance, and autonomous parking modules are integrated by control algorithms, and an interface is provided.

Algorithms

```
//load engine
int RPWM = 2; // right PWM
int LPWM = 3; // left PWM
int R_EN = 4; // right Enable
int L_EN = 5; // left Enable

// sağ motor
int RPWM1 = 6; // Sağ PWM
int LPWM1 = 7; // Sol PWM
int R_EN1 = 8; // Sağ Enable
int L_EN1 = 9; // Sol Enable

// sol motor
int RPWM2 = 10; // Sağ PWM-
int LPWM2 = 11; // Sol PWM
int R_EN2 = 12; // Sağ Enable
int L_EN2 = 13; // Sol Enable

// renk sensörleri
const int sensorRight = A0; // Sağ sensör
const int sensorRightMid = A1; // Sağ ortadaki sensör
const int sensorLeftMid = A2; // Sol ortadaki sensör
const int sensorLeft = A3; // Sol sensör

// ön sensör
const int ontrigPin = 22;
const int onechoPin = 23;

// arka sensör
const int arkatrigPin = 28;
const int arkaechoPin = 29;
```

```

const int righttrigPin = 32;
const int rightechoPin = 33;
const int lefttrigPin = 34;
const int leftechoPin = 35;

// Sensör eşiği
const int threshold = 450;

// Engel mesafesi (cm cinsinden)
const int stopDistanceMin = 20;
const int stopDistanceMax = 30;
```

Figure 27 – Arduino ADK Code

At the beginning of the code, the pins required for the motors (2 DC motors and 1 Linear Actuator) and sensors used in the vehicle are defined. For the control of three motors, right and left PWM pins and enable pins are assigned. (RPWM, LPWM, R_EN, L_EN pins are the right and left PWM pins and enable pins for the load motor. RPWM1, LPWM1, R_EN1, L_EN1 pins are right and left PWM pins and enable pins for the right engine. RPWM2, LPWM2, R_EN2, L_EN2 pins are right and left PWM pins and enable pins for the left motor.) The load engine pins you see in the code are the pins of the motor driver attached to the linear actuator. The pins labeled as the right motor belong to the right motor driver, and those labeled as the left motor belong to the left motor driver. These three motor drivers are BTS7960B motor drivers. Additionally, one IR color sensor and four ultrasonic sensors are defined. The IR sensor has four sensors that read black and white outputs. These sensors are connected to the Arduino pins A0, A1, A2, and A3. The trig and echo pins of the ultrasonic sensors are also given in the code. (ontrigPin, onechoPin pins trig and echo pins for the front ultrasonic sensor; arkatrigPin, arkaechoPin pins trig and echo pins for the rear ultrasonic sensor; righttrigPin, right echo Pin pins trig and echo pins for the right ultrasonic sensor; lefttrigPin, leftechoPy pins trigger and echo pins for the left ultrasonic sensor.) The sensor threshold and obstacle detection distance constants are defined to determine under which conditions the sensors will be activated. In addition, there are two fixed values that you see in the code, one of which is the threshold constant, the detection threshold of color sensors, and the other is the stopDistanceMin, stopDistanceMax constants, distance limits for obstacle detection.

```
void setup() {
  Serial.begin(9600); // Seri iletişimi başlat
  // Pinlerin çıkış olarak ayarlanması
  pinMode(RPWM, OUTPUT);
  pinMode(LPWM, OUTPUT);
  pinMode(R_EN, OUTPUT);
  pinMode(L_EN, OUTPUT);
  pinMode(RPWM1, OUTPUT);
  pinMode(LPWM1, OUTPUT);
  pinMode(R_EN1, OUTPUT);
  pinMode(L_EN1, OUTPUT);
  pinMode(RPWM2, OUTPUT);
  pinMode(LPWM2, OUTPUT);
  pinMode(R_EN2, OUTPUT);
  pinMode(L_EN2, OUTPUT);
  // Enable pinlerini HIGH yaparak motor sürücüyü aktif hale getirme
  digitalWrite(R_EN, HIGH);
  digitalWrite(L_EN, HIGH);
  digitalWrite(R_EN1, HIGH);
  digitalWrite(L_EN1, HIGH);
  digitalWrite(R_EN2, HIGH);
  digitalWrite(L_EN2, HIGH);
  // renk sensörlerini aktif etme
  pinMode(sensorRight, INPUT);
  pinMode(sensorRightMid, INPUT);
  pinMode(sensorLeftMid, INPUT);
  pinMode(sensorLeft, INPUT);

```

```
74    // Ultrasonik sensör pinleri ayarlanır
75    pinMode(ontrigPin, OUTPUT);
76    pinMode(onechoPin, INPUT);
77    pinMode(arkatrigPin, OUTPUT);
78    pinMode(arkaechoPin, INPUT);
79    pinMode(righttrigPin, OUTPUT);
80    pinMode(rightechoPin, INPUT);
81    pinMode(lefttrigPin, OUTPUT);
82    pinMode(leftechoPin, INPUT);
83  }
```

Figure 28 – Arduino Setup Function

The setup function is executed once the program is started and sets the input/output modes of various pins. The output pins of our motor drives between lines 49 - 60 are set as the output pin on the Arduino. We are making the enable pins high to activate the motor drives between lines 61 - 67. We activate the color sensors between lines 69 and 72. This sensor is the sensors of the black line Decking. between 75 - 82 pins, we Decode the ultrasonic sensors. Here, the trig pins are set as the output pin and the echo pins are set as the input pin.

```
void loop() {
  if (Serial.available()) {
    String command = Serial.readStringUntil('\n');
    command.trim();

    if (command == "forward") {
      next();
    } else if (command == "stop") {
      dur();
    } else if (command == "right") {
      sag();
    } else if (command == "left") {
      sol();
    }
  }

  // Renk sensörlerinin değerlerini oku
  int rightValue = analogRead(sensorRight);
  int rightMidValue = analogRead(sensorRightMid);
  int leftMidValue = analogRead(sensorLeftMid);
  int leftValue = analogRead(sensorLeft);

```

```
  // Renk sensörlerinin değerlerini seri monitöre yazdır
  Serial.print("Right: ");
  Serial.print(rightValue);
  Serial.print(" | RightMid: ");
  Serial.print(rightMidValue);
  Serial.print(" | LeftMid: ");
  Serial.print(leftMidValue);
  Serial.print(" | Left: ");
  Serial.println(leftValue);

  if ( rightMidValue == 1023 && leftMidValue == 1023 || rightValue == 0 &&  leftValue== 0 ){
    next();
  }else if (rightMidValue == 1023 && leftMidValue == 0 || rightValue == 1023 &&  leftValue== 0){
    sol();
  }else if (rightMidValue == 0 && leftMidValue == 1023 || rightValue == 0 &&  leftValue== 1023){
    sag();
  }else if (rightMidValue == 0 && leftMidValue == 0 || rightValue == 0 &&  leftValue== 0){
    dur();
  }

  delay(500);
}

```

Figure 29 – Arduino Loop Function

The loop function begins by checking if there is any data available on the serial port. If data is available, it reads the incoming string until a newline character is encountered, then trims any leading or trailing whitespace from the string. Depending on the received command, the function calls different functions to control the movement of the motors. If the command is "forward", the next ()

function is called to move the robot forward. If the command is "stop", the dur() function is called to stop the robot. If the command is "right", the sag () function is called to turn the robot to the right, and if the command is "left", the sol() function is called to turn the robot to the left. Following this, the function reads the values from four color sensors: sensorRight, sensorRightMid, sensorLeftMid, and sensorLeft. These values are then printed to the serial monitor for debugging or monitoring purposes. The function then evaluates the values from these sensors to determine the robot's movement based on the color detection. If both the right middle and left middle sensors read 1023 or both the right and left sensors read 0, indicating the detection of a black line, the robot moves forward by calling the next () function. If the right middle sensor reads 1023 and the left middle sensor reads 0 or the right sensor reads 1023 and the left sensor reads 0, the robot turns left by calling the sol () function. Conversely, if the right middle sensor reads 0 and the left middle sensor reads 1023 or the right sensor reads 0 and the left sensor reads 1023, the robot turns right by calling the sag () function. Lastly, if both the right middle and left middle sensors read 0 or both the right and left sensors read 0, the robot stops by calling the dur () function. A delay of 500 milliseconds is included at the end of the loop to provide a pause between each iteration.

```
int measureDistance(int trigPin, int echoPin) {
  // Ultrasonik sensör ile mesafe ölçümü
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

  long duration = pulseIn(echoPin, HIGH);
  int distance = duration * 0.034 / 2; // Mesafeyi cm cinsine çevir

  return distance;
}
```

Figure 30 – Arduino MeasureDistance Function

The measureDistance function measures the distance using an ultrasonic sensor, which involves sending out a sound wave and then measuring how long it takes for the echo to come back. Here’s how it works: First, the function makes sure the trigger pin (trigPin) is set to LOW for a brief moment (2 microseconds) to clear any previous signals. Then, it sets the trigger pin to HIGH for 10 microseconds, which sends out a sound wave from the ultrasonic sensor. After that, the trigger pin is set back to LOW. Next, the function uses pulseIn to measure the time (in microseconds) that the echo pin (echoPin) stays HIGH, which is the time taken for the sound wave to travel to an object and back. This duration is stored in the duration variable. Finally, the function calculates the distance to the object by using the formula duration * 0.034 / 2. This formula converts the time (duration) into distance in centimeters. The factor 0.034 is derived from the speed of sound in air (approximately 340 meters per second), and dividing by 2 accounts for the round trip of the sound wave (to the object and back). The calculated distance is then returned by the function.

```
void yuka() {
  // İleri hareket için RPWM'i PWM sinyali ile LPWM'i LOW yapma
  analogWrite(RPWM, 255); // Tam hız
  digitalWrite(LPWM, LOW);
}

void asag() {
  // Geri hareket için LPWM'i PWM sinyali ile RPWM'i LOW yapma
  analogWrite(LPWM, 255); // Tam hız
  digitalWrite(RPWM, LOW);
}
```

Figure 31 – Arduino Linear Actuator Function

This code controls a linear actuator to move up and down, operating the lift that the vehicle uses when picking up a load. The yuka function moves the actuator upwards; it does this by sending a PWM signal with a value of 255 to the RPWM pin, which represents full speed, and simultaneously setting the LPWM pin to LOW, causing the actuator to move forward. On the other hand, the asag function moves the actuator downwards; it sends a PWM signal with a value of 255 to the LPWM pin and sets the RPWM pin to LOW, causing the actuator to move backward. These two functions use PWM signals and digital pins to control the direction and speed of the actuator, ensuring the lift operates as needed.

```
void next() {
  // Ön sensör mesafesini ölç
  int distance = measureDistance(ontrigPin, onechoPin);
  if (distance >= stopDistanceMin && distance <= stopDistanceMax) {
    dur();
    Serial.println("Engel algılandı, duruyor");
  } else {
    // İleri hareket için LPWM1 ve RPWM2'yi PWM sinyali ile sür ve RPWM1 ve LPWM2'yi LOW yap
    analogWrite(LPWM1, 255); // Maksimum hız
    analogWrite(RPWM2, 255); // Maksimum hız
    digitalWrite(RPWM1, LOW);
    digitalWrite(LPWM2, LOW);
    Serial.println("İleri gidiyor");
  }
}

void geri() {
  // Arka sensör mesafesini ölç
  int distance = measureDistance(arkatrigPin, arkaechoPin);
  if (distance >= stopDistanceMin && distance <= stopDistanceMax) {
    dur();
    Serial.println("Arkada engel algılandı, duruyor");
  } else {
    // Geri hareket için LPWM'i PWM sinyali ile RPWM'i LOW yapma
    analogWrite(LPWM, 255); // Tam hız
    digitalWrite(RPWM, LOW);
    Serial.println("Geri gidiyor");
  }
}
```

Figure 32 – Arduino Motors Controller Function

The next function controls the forward movement of the vehicle by first measuring the distance using the front ultrasonic sensor via the measureDistance function with ontrigPin and onechoPin. If the measured distance falls within the defined range (stopDistanceMin to stopDistanceMax), indicating an obstacle, the dur() function is called to stop the vehicle, and a message "Obstacle detected, stopping" is printed to the serial monitor. If no obstacle is detected, the function drives the vehicle forward by setting LPWM1 and RPWM2 to maximum speed (255) and setting RPWM1 and LPWM2 to LOW, ensuring the vehicle moves forward, with the message "Moving forward" printed to the serial monitor. Similarly, the geri function manages the backward movement by measuring the distance with the rear ultrasonic sensor using arkatrigPin and arkaechoPin. If an obstacle is detected within the specified range, it stops the vehicle and prints "Obstacle detected behind, stopping." Otherwise, it drives the vehicle backward by setting LPWM to maximum speed (255) and RPWM to LOW, printing "Moving backward" to the serial monitor.

```
void sag() {
  // Sağ mesafe sensörünü kontrol et
  int distance = measureDistance(righttrigPin, rightechoPin);
  if (distance >= stopDistanceMin && distance <= stopDistanceMax) {
    dur();
    Serial.println("Sağda engel algılandı, duruyor");
  } else {
    // Sağ motoru tam hızda ileri sür
    analogWrite(RPWM1, 128);
    digitalWrite(LPWM1, LOW);
    // Sol motoru geri sür
    analogWrite(RPWM2, 128);
    digitalWrite(LPWM2, HIGH);
    Serial.println("Sağa dönüyor");
  }
}

void sol() {
  // Sol mesafe sensörünü kontrol et
  int distance = measureDistance(lefttrigPin, leftechoPin);
  if (distance >= stopDistanceMin && distance <= stopDistanceMax) {
    dur();
    Serial.println("Solda engel algılandı, duruyor");
  } else {
    // Sol motoru tam hızda ileri sür
    analogWrite(LPWM1, 128);
    digitalWrite(RPWM1, LOW);
    // Sağ motoru geri sür
    analogWrite(RPWM2, 128);
    digitalWrite(LPWM2, HIGH);
    Serial.println("Sola dönüyor");
  }
}
```

Figure 33 – Arduino Motors Controller Function

The sag function makes the vehicle turn right by first measuring the distance using the right ultrasonic sensor with the measureDistance function, using righttrigPin and rightechoPin. If the distance is within the defined range (stopDistanceMin to stopDistanceMax), indicating an obstacle, the function calls dur() to stop the vehicle and prints "Obstacle detected on the right, stopping" to the serial monitor. If no obstacle is detected, the function proceeds to control the motors to turn the vehicle right. It sets the right motor (RPWM1) to half speed (128) and LPWM1 to LOW to move the right motor forward, while setting the left motor (RPWM2) to half speed (128) and LPWM2 to HIGH to move the left motor backward, thus executing a right turn, and prints "Turning right" to the serial monitor. Similarly, the sol function controls the left turn by measuring the distance with the left ultrasonic sensor using lefttrigPin and leftechoPin. If an obstacle is detected on the left within the specified range, it stops the vehicle and prints "Obstacle detected on the left, stopping." If no obstacle is detected, it sets the left motor (LPWM1) to half speed (128) and RPWM1 to LOW to move the left motor forward, and the right motor (RPWM2) to half speed (128) and LPWM2 to HIGH to move the right motor backward, making the vehicle turn left, and prints "Turning left" to the serial monitor.

4.1 Detailed Test Cases

TC_ID	**Sensor01**
Purpose	Validate the autonomous vehicle's ability to follow commands and avoid obstacles using sensor inputs.
Estimated Time Needed	10 Minutes
Dependency	All sensors and motor drivers should be functional and properly calibrated.
Setup	Place the vehicle in an area with predefined paths and obstacles. Ensure all sensors and motors are connected and powered.
Procedure	**[A01]** Initialize the vehicle system and sensors. **[A02]** Send the "forward" command via serial input. **[A03]** Place an obstacle within 20-30 cm in front of the vehicle. **[A04]** Verify the vehicle stops or avoids the obstacle. **[A05] Send the "right" command via serial input.** **[A06] Verify the vehicle turns right.** **[A07] Send the "left" command via serial input.** **[A08] Verify the vehicle turns left.** **[A09] Send the "stop" command via serial input.** **[A10] Verify the vehicle stops.**
Verification	**[V01]** Observe that the vehicle follows the commands accurately and avoids obstacles as per the logic defined in the code.
Cleanup	Power down the vehicle and disconnect all serial inputs.

TC_ID	
Purpose	Validate the functionality of the three motor drivers for forward, reverse, and stop commands.
Estimated Time Needed	30 Minutes
Dependency	**All motor drivers should be connected and functional.**
Setup	Ensure the motor drivers are connected to their respective motors and the system is powered.
Procedure	**[A01]** Initialize the system and ensure all motor drivers are powered on. **[A02]** Send the "forward" command to the first motor driver. **[A03]** Verify the motor connected to the first driver moves forward. **[A04]** Send the "reverse" command to the first motor driver. **[A05]** Verify the motor connected to the first driver moves in reverse. **[A06]** Send the "stop" command to the first motor driver. **[A07]** Verify the motor connected to the first driver stops. **[A08]** Repeat steps [A02] to [A07] for the second motor driver. **[A09]** Repeat steps [A02] to [A07] for the third motor driver.
Verification	**[V01]** Observe that all motors respond correctly to the forward, reverse, and stop commands as expected.
Cleanup	Turn off the system and ensure all connections are safely disconnected.

Individual Test Results

TC ID	Priority	Run By	Result	Explanation
Pc.01	High	Barış Bayram	Pass	Initialization and line following logic works correctly.
Pc.02	High	Barış Bayram	Fail	Sensor data processing issue, to be fixed in next update.
Pc.03	Medium	Barış Bayram	Pass	Load handling command execution is successful.
Pc.04	High	Barış Bayram	Pass	Obstacle detection and avoidance is functional.
Pc.05	High	Barış Bayram	Pass	Vehicle stops accurately at designated points.
Pc.06	Medium	Barış Bayram	Pass	Command response time is within acceptable limits.
Pc.07	Medium	Barış Bayram	Pass	System recovers gracefully from errors.

Ec.01	High	M.Furkan Eroğlu	Pass	Power distribution to all motors is stable.
Ec.02	Medium	M.Furkan Eroğlu	Fail	Voltage drop observed during load handling, needs investigation.
Ec.03	High	M.Furkan Eroğlu	Pass	Sensors correctly powered and functioning.
Ec.04	High	M.Furkan Eroğlu	Pass	Battery levels are monitored correctly.
Ec.05	Medium	M.Furkan Eroğlu	Pass	Overcurrent protection works as intended.
Ec.06	High	M.Furkan Eroğlu	Pass	All connections are secure and stable.
Ec.07	High	M.Furkan Eroğlu	Pass	Backup power system operates correctly.

References

References:

[1] H. Hu, X. Jia, K. Liu, and B. Sun, “Self-adaptive traffic control model with behavior trees and reinforcement learning for agv in industry 4.0,” IEEE Transactions on Industrial Informatics, vol. 17, no. 12, pp. 7968– 7979, 2021.

[2] H. Mart´ınez-Barbera and D. Herrero-P ´ erez, “Autonomous navigation of ´ an automated guided vehicle in industrial environments,” Robotics and Computer-Integrated Manufacturing, vol. 26, no. 4, pp. 296–311, 2010.

[3] X. Li, J. Wan, H.-N. Dai, M. Imran, M. Xia, and A. Celesti, “A hybrid computing solution and resource scheduling strategy for edge computing in smart manufacturing,” IEEE Transactions on Industrial Informatics, vol. 15, no. 7, pp. 4225–4234, 2019.

[4] A. Gao, R. R. Murphy, W. Chen, G. Dagnino, P. Fischer, M. G. Gutierrez, D. Kundrat, B. J. Nelson, N. Shamsudhin, H. Su et al., “Progress in robotics for combating infectious diseases,” Science Robotics, vol. 6, no. 52, p. eabf1462, 2021.

[5] J. Shang, J. Zhang, and C. Li, “Trajectory tracking control of agv based on time-varying state feedback,” EURASIP Journal on Wireless Communications and Networking, no. 1, pp. 1–12, 2021.

[6] M. De Ryck, M. Versteyhe, and F. Debrouwere, “Automated guided vehicle systems, state-of-theart control algorithms and techniques,” Journal of Manufacturing Systems, vol. 54, pp. 152–173, 2020.

[7] M. Labbe and F. Michaud, “Rtab-map as an open-source lidar and visual ´ simultaneous localization and mapping library for large-scale and longterm online operation,” Journal of Field Robotics, vol. 36, no. 2, pp. 416–446, 2019.

[8] K. Eckenhoff, P. Geneva, J. Bloecker, and G. Huang, “Multi-camera visual-inertial navigation with online intrinsic and extrinsic calibration,” in 2019 International Conference on Robotics and Automation (ICRA). IEEE, 2019, pp. 3158–3164.

[9] Mathew Gadd, Paul Newman. “A Framework for Infrastructure-Free Warehouse Navigation”. IEEE International Conference on Robotics and Automation, 2015.

[10] Ganesan P, G.Sajiv , Megalan Leo.l. “Warehouse Management System using Microprocessor Based Mobile Robotic Approach”. Third International Conference on Science Technology Engineering & Management,2017.

[11] Kaveh Azadeh, René de Koster, and Debjit Roy. “Robotized and Automated Warehouse Systems Review and Recent Developments”, 2017.

[12] Mr. Aniket Avinash Naik, Mrs. Meghana R Khare. Study of “WebSocket Protocol for Real-Time Data Transfer”.International Reasearch Journal of Engineering and Technology,2020.

[13] Liu Qigang, Xiangyang Sun. “Research of Web Real-Time Communication Based on Web Socket”.International Journal of Communications,Networks and System Sciences,January 2012.

[14] Abdullah Saleh Alqahtani, Robert Goodwin. “E-commerce Smartphone Application”. International Journal of Advanced Computer Science and Applications, Vol. 3, No.8,2012.

[15] I. Draganjac, D. Miklic, Z. Kova ´ ciˇ c, G. Vasiljevi ´ c, and S. Bogdan, ´ “Decentralized control of multiagv systems in autonomous warehousing applications,” IEEE Transactions on Automation Science and Engineering, vol. 13, no. 4, pp. 1433–1447, 2016.

[16] X. Pan, J. Shi, P. Luo, X. Wang, and X. Tang, “Spatial as deep: Spatial cnn for traffic scene understanding,” in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 32, no. 1, 2018.

[17] B. Guan, P. Vasseur, C. Demonceaux, and F. Fraundorfer, “Visual odometry using a homography formulation with decoupled rotation and translation estimation using minimal solutions,” in 2018 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2018, pp. 2320–2327.

[18] C. Zhao, B. Fan, J. Hu, Q. Pan, and Z. Xu, “Homography-based camera pose estimation with known gravity direction for uav navigation,” Science China Information Sciences, vol. 64, no. 1, pp. 1– 13, 2021.

[19] D. Fox, W. Burgard, and S. Thrun, “The Dynamic Window Approach to Collision Avoidance,” IEEE Robotics Automation Mag., Mar. 1997.

[20] Design and Development of an Automated Guided Vehicle, Georgios Kaloutsakis the Johns Hopkins University Department of Mechanical Engineering Baltimore, MD 21218, U3.A.

[21] V. Adams and A. Askenazi, Building Better Producis with Finite Element Analysis, Onword Press, U.S.A., 1999.

[22] “AGV Automated Guided Vehicle types of Battery”. [Online]. Available: 44 https://www.agvnetwork.com/agv-types-ofbattery. [Accessed: 01-Nov.-2019].

[23] Design and Control of Material Transport System for Automated Guided Vehicle, Wu Xing, Lou Peihuang, Cai Qixiang, Zhou Chidong, Shen ke, Jin chen College of Mechanical and Electrical Engineering Nanjing University of Aeronautics and Astronautics Nanjing, P. R. China,2012.

[24] Design and Control of Material Transport System for Automated Guided Vehicle, Wu Xing, Lou Peihuang, Cai Qixiang, Zhou Chidong, Shen ke, Jin chen College of Mechanical and Electrical Engineering Nanjing University of Aeronautics and Astronautics Nanjing, P. R. China,2012.

[25]https://cdn.t3kys.com/media/upload/userFormUpload/brsVjKcxuF4eWPb0xGMyaXzDa1KKAsmu.pdf.

[26] Design and Development of an Automated Guided Vehicle for Educational Purposes, Khosro Bijanrostami Submitted to the Institute of Graduate Studies and Research in partial fulfillment of the requirements for the Degree of Master of Science in Mechanical Engineering Eastern Mediterranean University September 2011 Gazimağusa, North Cyprus

[27] Yoshio, M., Brodd, R.J. and Kozawa, A., 2009. Lithium-Ion Batteries:Science and Technologies, Springer Science and Business Media, Newyork, USA, pp. 1-7.

[28] Hackney, S.A. and Kumar, R.V., 2010. High Energy Density Lithium Batteries, Wiley-VCH Verlag GmbH, Weinheim, pp. 70-73.

[29] Linden, D. and Reddy, T.B., 2002. Handbook of Batteries, Third Eddition, McGraw-Hill, Bölüm 22 ve 35.

[30] Ozawa, K., 2009. Lithium-Ion Rechargeable Batteries, Wiley-VCH Verlag GmbH, Weinheim, pp. 5-8.

[31] Nazri, G.A. and Pistoia, G., 2009. Lithium Batteries: Science and Technology, Springer Science and Business Media, Newyork, pp. 8-11.

[32] Uysal, M., Karslıoğlu, R., Guler M.O., Alp, A. and Akbulut, H., 2009. Rod and wire like morphologies of thin oxide developed with plasma oxidation after electro deposition, Materials Letters, 63, 422-424.

[33] Walter, A., Schalkwijk, V. and Scrosati, B., 2002. Advances in Lithium-Ion Batteries, Kluwer Academic/ Plenum Publishers, New York, pp. 30-42.

[34] Besenhard, J.O., 2008. Handbook of Battery Materials, John Wiley & Sons, pp. 50-72.
[35] Dahlin, G.R. and Strom, K.E., 2010. Lithium Batteries:Research, Technology and Applications, Nova Science Pub Incorporated, pp. 32-51

[36]http://search.usa.gov/search?utf8=&afuspto.gov&query=lithium+ion+battery&go=Go,

[37] https://www.sciencedirect.com/science/article/abs/pii/S1369847818301414,

[38]https://www.sciencedirect.com/science/article/abs/pii/S0968090X18302134

[39] M. Gerdts, S. Karrenberg, B. Mueller-Bessler, G. Stock, ''Generating locally optimal trajectories for an automatically driven car'' Optimization and Engineering (2009)

[40] J. Alonso et al. Cartography for cooperative manoeuvres: autopia's new cartography system for cooperative manoeuvres among autonomous vehicles Journal of Navigation (2011)
[41] https://www.ijcai.org/Proceedings/77-2/Papers/002.pd