Autonomous Warehouse Robot

Taylor Harvey, Diandra Reyes, Peter Watson, and Alexander Wozny

***Abstract*—Automation is a useful feature that enhances any mechanical design. The development and implementation of an autonomous robot catered towards performing tasks associated with warehouses is a relevant topic in modern day technology. Key goals of these types of projects include but are not limited to achieving navigational, tracking, and independent execution of the robot while users are only required to send commands. In this paper the motivation, analysis, and overall process of validating several components of an autonomous robot are explained. This is to offer insight on the design thinking necessary to develop a system capable of attaining autonomy. Testing played a large role during this design process for it highlighted discrepancies while demonstrating benefits of incorporating autonomous behavior to a system. Levels of validations were used to measure success starting with simpler tasks such as achieving holonomic steering, localization and material manipulation to more complex tasks such as properly detecting and identifying materials and integrating several components to one system. The report outlines in detail primary experiences of a design team building an autonomous material handling robot contributing to the overall experimentation of combining innovative techniques with standard practices.**

I. Introduction

Automated warehouses are a growing industry popular for using advanced technology, such as robots, in storage facilities. Robots help improve storing and fetching of packages while also lowering employee costs. These robots can change how a warehouse functions by replacing human workers and can work in dark spaces. A company can lower costs of utilities and employment by including robots in their warehouse. Pursuing this project will not only increase profits but will reduce safety hazards that can affect human workers still working in the warehouse.

This project develops an autonomous mobile robot that can work in dark warehouses. The robot is independent enough to track, receive, and store packages on its own. However, the robot will need human aid if something goes wrong in the warehouse. To track packages around the warehouse, packages will have a quick response (QR) code, which are like barcodes except they store more data. These codes are also helpful for creating a map of the warehouse for the robot. Black lines placed around the warehouse shape the robot's path and lead it to each package spot. The robot features a forklift device to lift and hold the packages. For testing, the robot will work in a scaled down warehouse including black lines, QR codes, and packages on small pallets.

This warehouse robot is a holonomic robot, meaning that it is able to translate in any direction along the ground, as well as rotate. It has four universal omni-directional wheels to accomplish this motion.

Because there are different functions the robot must perform, this project seeks to accomplish several tasks. The project can be broken into 3 main functions: locomotion, navigation, and manipulation. The robot must be able to physically move and traverse a flat ground environment. The individual wheels need to be accurately controlled so the robot can move in a specified direction at a controlled speed. The project must also be able to properly and accurately navigate around a prototype dark mock warehouse to a specified package location and move back to the starting point. For this to be done, the robot will have line sensing capabilities and be able to follow closely to the line, notice different line configurations to help localize itself, and use this localization to reach the specified position. Additionally, the robot must be able to manipulate, or lift, a package off a certain shelf height and move to a desired location with the package.

II. Methodology

To control the robot's forward, lateral, and angular motion, each independent wheel needs to be controlled by a motor. The angular velocities of each wheel needs to be determined to achieve the desired motion. To reach a desired robot velocity, inverse kinematics equations are used. These equations are shown below.

$$\begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \\ \omega_4 \end{bmatrix} = \begin{bmatrix} sin(45^\circ) & cos(45^\circ) & R \\ sin(45^\circ) & cos(45^\circ) & R \\ sin(45^\circ) & cos(45^\circ) & R \\ sin(45^\circ) & cos(45^\circ) & R \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ \dot{\theta} \end{bmatrix} \tag{1}$$

The inputs in this equation are v_x, v_y, and $\dot{\theta}$, or desired forward velocity, lateral velocity, and angular velocity. The positive v_x direction is forward; the positive v_y direction is to the left; and the positive $\dot{\theta}$ direction is counterclockwise. The angle in the sines and cosines of the equation are at 45° because the wheels are placed at this angle relative to the possible motions of the robot. The 45° angle allows for the robot to get the maximum forward and lateral velocity from the wheels. In the equation, R is based on the physical dimensions of the robot; it is the Euclidean distance from the center of the robot to the center of each wheel, which for our robot is the same for each wheel since it has equal length and width. The output of the equation is ω_i, or the angular velocity of each wheel, with i representing 1, 2, 3, or 4. A schematic is shown below to better visualize the different components involved in controlling the robot's motion

To measure the position and speed of the wheels, encoders are used. Because there are 4 independently controlled wheels, each needs 2 accessible interrupts to be used from the microcontroller. Because of this, the Arduino Due was selected because of its large number of possible usable interrupts as

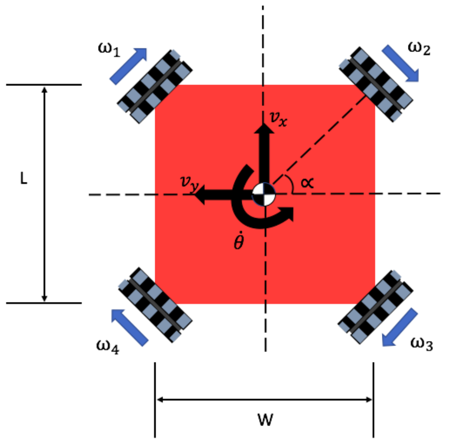

Fig. 1. Robot Motion Schematic

well as its faster processing speed. The wheels are controlled on the Due by using the error in the position and speed of the wheels from the encoder values and using them in a PD controller. The values are converted to an appropriate duty cycle so the wheels output the correct speed.

The lifting mechanism implemented for manipulating the packages was of a forklift-type design. It is made up of a pulley system where a mast is rigidly attached to the chassis and the forks move up because they are attached to a carriage that loops around the mast and is attached to a large motor. The motor has a pulley around it, and a polymer chord is connected to the pulley and to the carriage, allowing the motor to lift the carriage. Using the encoders on the large lifting motor, the position of the forks can be determined and the upward and downward speed of the forks can thus be controlled.

A. *Integration*

Integration is an important part of a project since all of the subsystems are compiled into one large working system. It can be more tedious and difficult than expected and should not be overlooked. Different components might affect the other, causing a possible need for change in a design. A knowledge of how the components work together is needed to avoid complications between subsystems as well as to properly integrate the components so they work synchronously.

For the robot to navigate around the warehouse, the controls for the wheels and sensors must be integrated to work together. The infrared sensor values are used to sense the line and the error in the robot's position. There are four sensors placed in the middle of all the sides of the robot. There are 2 for each direction of travel to minimize the error in the heading angle of the robot. The error is used to calculate the desired angular velocity of the robot, as well as the velocity perpendicular to the motion. For example, if the robot were moving forward, the error would be used to calculate the desired angular and lateral velocity ($\dot{\theta}$ and v_y) to get the robot to move back towards the line. These values are fed into the inverse kinematic equations to determine the desired wheel velocities. The PD controller of the wheels controls the motors so they spin at the desired rate. This produces motion of the robot, allowing it to move to the desired points in the warehouse.

For the lifting mechanism, the mast must be secured firmly to the chassis by a custom aluminum bracket. Forks are firmly secured to the carriage, which is placed around the mast so that it can slide freely up and down. One end of the polymer chord is tied to a notch in the carriage, and the other end is secured to a pulley attached to the lifting motor. The lifting motor is prevented from rotating by the 3D printed casing it is screwed into, which is screwed into the wood base underneath. The motor for lifting must be controlled on the Arduino Due. The function for the lifting mechanism was integrated with the code involving the line sensing so that it could be called whenever the robot reached a shelf. For integrating the lifting mechanism with the drive train, the weight of the package on the forks near the front of the robot needed to be taken into consideration. Having a heavy weight at the front of the robot could cause it to either tip forward or for there to be more friction in the front wheels than the back. This could cause the robot to have trouble steering since the wheels on the front of the AGV would be having more of an effect on its motion. A counterweight was used to help balance the chassis so even friction is applied to all the wheels. Additionally, the controllers for the wheels work to correct any error caused from this in the wheels' desired velocities.

III. Testing and Validation

A. *Setup for Validation*

To control the robot's motion and path, black lines were set up around the mock warehouse to map out where the robot could move to. The lines were laid down in a grid, which simplified the problem, but also mimicked the overall layout of a warehouse. Due to space and resource limitations, the warehouse configuration was made to be a 3 by 1 grid with 3 shelves.

Since the robot will be retrieving packages for an assembly line, the first shelf in the warehouse near the start position represents the assembly line where it will bring packages from inside the warehouse. The second shelf in the middle of the warehouse is a first level shelf, which is used to show the robot can pick up a package from the first level of a shelf. In the back of the warehouse, there is a taller shelf used to represent a second level of shelving. Having the AGV retrieve a package from here will validate the robot can pick up a package from a second level of shelves.

Different levels of validation were used to determine the success of the project. Lower level validation involved easier tests for the robot and its components to accomplish. Higher levels of validation involved increasingly difficult tasks and included an increasingly fully integrated robot. For testing, the robot can be divided into 2 subsystems. The first is the lifting aspect, and the second is the locomotion and drive train system of the robot. These subsystems will be combined into one full system in the end when the whole robot is tested.

B. Lifting

The first level of validation involved testing components that made up the systems of the robot. For the lifting, the forks were first tested with the carriage to insure they were able to lift the maximum load without breaking, or deflecting to the point the package would fall off. Later the mast and the pulley with the full load was tested without being attached to the robot chassis. After it was confirmed that the mast could handle the load and the carriage could easily slide up and down the shaft, the whole system was attached to the robot chassis and the carriage was connected to the motor with the rope. The motor was tested with different voltage sources to get a good feel for the power required to lift. Lastly the whole lifting mechanism was tested to ensure the robot could lift a full load of 25 lbs the height of 1.5 ft without the package tipping.

C. Movement

Once the wheel motors could be controlled using the Arduino Due, the drive train could be tested. To initially test the drive train and wheel controls, several tests were performed.

The first set of tests were very basic and involved programming the robot to move forward and backwards, left and right, and spin clockwise and counterclockwise. Once it was established the motors would move in the proper directions and at equal magnitudes just from watching the robot, the second set of tests were performed. The second set of tests were simple and involved controlling the robot to go a set distance at a set speed. This involved observing whether the robot would stop at the desired distance, which was 1 meter, and whether it reached that distance in a certain amount of time. The robot was tested to move at 0.2 meters per second. A meter stick was placed down next to the path of the robot, starting at the very front of the robot. To pass this test, the robot needed to stop at the edge of the meter stick and reach that location in 5 seconds. This test was timed using a stopwatch. From the first few tests, the robot was close, but it did not reach the exact distance and time constraints. Some hand tuning was performed in the code to get the robot to stop after traveling the specified distance at the proper speed. Additional tests were performed to validate whether the wheel controls were working and that when the wheels and motors were supporting the base level weight (just the weight of the robot), that the robot was able to move straight. A black line was taped to the ground and the robot was lined up so the front was perpendicular with the line. The robot ran through several trials of moving along the line and it moved straight relative to the line. Note that this test was for the individual motor controls and was testing the PD control and not the actual line following capabilities of the robot.

Further tests were conducted on the wheel motor controls by placing weight distributed in different areas of the chassis to determine if the robot could still move relatively straight with uneven loads.

D. Package Identification

To ensure the correct packages are being delivered to the user, tests were done on the PI camera to see if it accurately identified packages. Initially, the camera is tested on its own while it is connected to the raspberry pi and a monitor. The monitor displays the image the raspberry pi is capturing while also displaying the commands along with any errors that may occur when the camera is functioning. The camera is first tested in a well lit environment while a QR code is placed in front of it. To test how far the camera is able to capture the QR code the team placed the QR code three feet away from the camera as a starting point. When observed that the camera was unable to identify the QR code increments of one inch were made until the monitor showed that the camera properly read the QR code. It was at two feet that the camera was able to identify the QR code. Once the maximum distance was determined, the next tests done were at different lightings to see if the camera can capture the QR code. The first level was having all the lights on so the room was well lit and bright; the camera easily captured the image. The next level was turning off most of lights and using the camera in a dim room; the camera also captured the QR code with ease. For the first two tests, the maximum distance between the camera and the QR code did not need to be manipulated. For the final test, the camera was placed in a dark environment along with the QR code, two feet apart. The camera was unable to capture the QR so the distance between camera and QR code decreased exponentially. It was observed in environments that are extremely dark the camera could not detect an image without assistance from a light source. To account for this an LED light was attached to the PI camera. Once again the camera is placed at the maximum distance of two feet and moved closer inch by inch until the camera is able to capture an image; the maximum distance in the dark was observed to be about 12 inches for the camera to be able to detect the QR code. For the sake of this project the maximum distance used for the remainder of testing was twelve inches to avoid any failures regarding the camera.

E. Package Pick Up

Once identified, the system was tested to ensure the lifting mechanism accurately positioned the forks on the vertical direction. Two heights were tested to resemble two separate shelves. The first test was seeing if the robot would recognize two ticks placed along the path that led to the package. The first tick notified the robot to stop and lift the forks to a height that can allow the forks to move between the pallet and shelf. Once the height is set the robot will then move to the second tick and the forks will be parallel to the pallet. When the robot is done moving forward, the lifting mechanism will begin to lift the pallet along with the package and move backwards to the main line path once item completes the lift. The lift then descends the package to the initial height until it is ready to deliver the package to the drop off location. To ensure the functions were being passed properly, the lifting cycle was ran along one section or aisle of the model warehouse and only with one path leading to a package.

F. Full System Validation

The last level of validation involved validating all electrical and mechanical components of the system working together. After all components were assembled and the mock warehouse was set up properly, the robot's ability to move to a proper package location, move to the package, lift up the package, and return to the final destination was tested. For validation, the robot either completes or fails the task.

The first test involves training the robot so that it knows the locations of the packages around the warehouse. The robot is sent to move around the different intersections and different paths to each package and scans each QR code using its Raspberry Pi camera. From here, the robot knows the location and contents of each package.

Next, all the robot's capabilities will be tested. The robot will perform 2 runs around the warehouse. The first will involve picking up a package from the first level shelf (lower level shelf). For the second run, the robot will move to the back of the warehouse to lift up a package from the second level shelf. This shows the robot is able to lift packages from different heights or levels.

IV. Results

At the final stage of validations, the robot passed several tests and achieved many of the goals set out for this project. For the drive train, the robot successfully moved in every linear, lateral, diagonal, and angular direction on command. The speed of the robot was also able to be controlled, as well as the actual robot position to be determined. The holonomic nature of the system occurred with less than five percent error by utilizing a PD controller for the motors that controlled the wheels along with the encoders to receive our variables for the controller. The line sensing of the system on the left, front, back, and right side of the robot's body functioned properly and was capable of performing the desired navigation. The robot was capable of sensing and following a line towards linear and lateral directions along with changing directions efficiently utilizing a PI controller. For the lifting mechanism, the Maxon motor used to lift the desired packages effectively lifted the goal load of 25 pounds by using a PD controller, as well as carefully picking up and placing packages at desired locations of the model warehouse. The integrity of the system from the base frame to the mast for the lifting mechanism maintained itself throughout all testing and demonstrations. Regarding the software aspect of the system, the network communication between user interface, Raspberry PI, and Arduino Due effectively executed commands from start to finish. The PI camera competently located and scanned QR codes for package identification and inventory tracking both in a well-lit environment and in the dark.

V. Conclusion

The omni-directional wheels were interesting to use since they are an unconventional drive train. There are different advantages and disadvantages of omni-directional wheels. The omni-wheels ended up working better than expected. While it was believed they would have much more slip and undesired movement in the system, especially when picking up packages. One advantage, in the case of our project, was the omni-wheels increased maneuverability of the system. The steering the robot used was not as difficult to implement into the warehouse as compared to a different type of steering. Making sure the robot stayed on the line while turning and not slipping did not need to be accounted for. With the omni-directional wheels, the robot would just change the direction it was moving in without having to spin. Another advantage is an increase in torque and being able to move with heavier loads. Because there are more motors moving the drive train, a larger load can be placed on the drive train because more wheels contribute to the movement. One disadvantage of the omni-directional wheels is they take up more power since they each draw current to move. This causes the robot to run out of battery faster. Additionally, determining the cause of a maintenance problem is more difficult because more components are used in the system, so it is harder to determine which wheel is not working if the robot is not moving properly. The wheels also begin to squeak after a lot of use, so they will have to be lubricated after heavy use. The wheels are also more complicated to attach because the mounts have to be at 45°.

Although the aluminum chassis worked for our application, a different material chassis should have been chosen or designed. Although aluminum is lightweight, which was advantageous, it also bent relatively easily during the beginning of our testing process. A stronger, less bendable material would have been preferred.

For the lifting mechanism, there were also more viable options than the pulley design that was chosen. There are a lot of additional concerns and potential problems that could go wrong with the mechanism since there are multiple components. One problem that had to be dealt with was the stretching of the polymer chord that lifted up the carriage and attached to the motor. Whenever a large weight was placed on the chord, it would stretch. This displacement was not measured by the encoders since the motor shaft itself did not rotate. Although it was small, this problem had to be considered and could potentially cause problems if more precision is required.

References

Harvey, T., Watson, P., Reyes, D., amp; Wozny, A. (2021, April 12). Operation Manual: Team 506 [Word].