Propelled and steered converter dolly for improved shunting of semi-trailers at goods terminals

Master's thesis in Systems, Control and Mechatronics
Master's thesis in Automotive Engineering

NEEL KACHHAWAH
CLIVE RAHUL MISQUITH

Department of Mechanics and Maritime Sciences
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2022
www.chalmers.se

Master's thesis 2021:34

Propelled and steered converter dolly for improved shunting of semi-trailers at goods terminals

NEEL KACHHAWAH
CLIVE RAHUL MISQUITH

Department of Mechanics and Maritime Sciences
Division of Vehicle Engineering and Autonomous Systems
Chalmers University of Technology
Gothenburg, Sweden 2022

Propelled and steered converter dolly for improved shunting of semi-trailers at goods terminals

NEEL KACHHAWAH, CLIVE RAHUL MISQUITH

© NEEL KACHHAWAH, CLIVE RAHUL MISQUITH, 2022.

Supervisor: Fredrik von Corswant, REVERE Lab; Toheed Ghandriz, Vehicle Dynamics; Maliheh Sadeghi Kati, Electrical Engineering
Examiner: Bengt Jacobson, Department Mechanics and Maritime Sciences, Division of Vehicle Engineering and Autonomous Systems

Master's Thesis 2022:34
Department Mechanics and Maritime Sciences
Division of Vehicle Engineering and Autonomous Systems
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: Scaled-down(1:14) model of Volvo truck (used as surrogate for i-Dolly) and semi-trailer combination.

Typeset in LaTeX
Gothenburg, Sweden 2022

Propelled and Steered Converter Dolly for Improved Shunting of Semi-Trailers on Goods Terminals

NEEL KACHHAWAH
CLIVE RAHUL MISQUITH
Department of Mechanics and Maritime Sciences
Division of Vehicle Engineering and Autonomous Systems
Chalmers University of Technology

Abstract

Distribution of semi-trailers from goods distribution terminals to customer loading docks are carried out by human drivers. Goods are transported to these distribution centers longer and heavier vehicles (LHV). These truck-trailer combinations need to be reconfigured in order to reroute traffic to customer end points. This requires a lot of driver skill and hence increases time and cost. These inland distribution centers or dry ports present an opportunity to improve shunting operation efficiencies through automation of a self-steered and self-propelled intelligent dolly (i-Dolly). This thesis explores the dominant methods of localization, path planning and path following used in autonomous systems. MATLAB Simulations for certain reversing maneuver of a single iDolly unit were performed using differential equations that described the reversing motion. The simulations were used to correlate results found during the practical tests on the scaled down model. For the physical tests, an over-head camera was used to implement a local positioning system with the use of ArUco markers. A scaled-down truck, used as surrogate for an i-dolly, was augmented with a cubic spline path planning algorithm and a path following stanley controller. The resulting system showed to have a maximum cross track error of 0.3 cm from the planned course when the coupling maneuver was successful. It was observed that the system was at times unsuccessful in the coupling maneuver, during which the cross track errors reached values of 1.5 cm. The recommended expansion on the work done would be to investigate higher precision hardware for localization and tracking at a small scale as well as setting up tests on more maneuvers. These could be further applied to full-scale model.

Keywords: i-Dolly, autonomous systems, ArUco markers, cubic spline, Stanley controller, longer and heavier vehicles (LHV), semi-trailers, transportation

Acknowledgements

We would to thank Fredrik von Corswant of REVERE Lab, Chalmers for giving us this opportunity and for also being a mentor and friend throughout the thesis. His constant optimism and support have encouraged us to always be positive.
We thank our academic supervisors Toheed Ghandriz and Maliheh Sadeghi Kati for their continuous guidance, feedback and support. Their knowledge and feedback have been invaluable to us especially during these times of distant contact and online meetings.
We would also like to thank Bengt J H Jacobson, our examiner at Chalmers University of Technology for his thorough feedback and guidance at every stage of the thesis.
A very special vote of thanks goes to Diego Jauregui. He has been a pillar of support to us during this thesis. His support with the tools and techniques needed to achieve our goal were indispensable.

We thank our family and friends for always being there to support us during good times as well as the tough times.

Neel Kachhawah & Clive Rahul Misquith
Gothenburg, June 2021

Contents

1
Introduction

1.1 Motivation for this research

Transportation in Europe, by longer and heavier vehicles (LHV) has been increasingly considered for their ability to transport a high volume of goods at relatively lower CO_2 emissions and costs [1]. These modes of transport, however, are beneficial at large distances. The efficiency is good due to the large volume of goods and maneuverability of the truck is fairly simpler when moving in the forward direction. The lesser area available for maneuvering in distribution center (or dry port) as well as customer loading docks prompts the shunting of individual trailers and containers within these areas.

To understand the purpose of this thesis project, it is beneficial to look into the current working of non-autonomous transportation in and around logistics terminals, specifically in Gothenburg. What happens typically is that containers or goods are transported on semi-trailers by LHVs, at times an A-double combination, from ports to a local in-land distribution center (or dry port). From these dry ports, the semi-trailers are redistributed individually to the end customer for unloading.

Currently, these semi-trailers have to be individually shunted to the last docking station by human-driven trucks for distances of up to 10 km, sometimes partly on public roads. For this purpose, the semi-trailers need to be reconfigured to the individual trucks. This requires a lot of effort from the driver. It also prevents the driver from engaging in other important tasks.

In the future, the semi-trailer units are aimed to be transported by an automated and electric transport solution called an intelligent-dolly (i-Dolly), with functions that include automatic coupling, shunting and parking of semi-trailers. Such a solution would provide benefits such as reduced, or rather completely removed, local fuel exhaust gases, improved time efficiency and reduced operating costs since no driver is needed for such repetitive maneuvers.

The purpose of this project is to develop algorithms for the envisioned i-Dolly. These solutions were developed onto a 1:14 down-scaled dolly-trailer combination model and tested in a scaled environment with reasonable assumptions of vehicle speed, environmental constraints and resource availability. The solutions were tested and the results were successful within reasonable margins of error. The errors were investigated and feasible corrections were suggested.

1.2 Research Questions

The following questions are investigated:

- How can coupling and/or shunting operations at terminals and local distribution of semi-trailers be performed efficiently by using remotely controlled dollies?
- What methods of localization, path planning and control can be used to achieve these operations in a scaled-down environment?

1.3 Objectives

The envisioned solution can be divided into primary and secondary objectives.

1.3.1 Primary Objectives

- Develop path-following controller for the coupling of the i-Dolly to the semi-trailer.
- Develop algorithms for autonomous maneuvering within the scaled-down port as well as reversing of the i-Dolly and semi-trailer to the loading dock of the goods terminal.

1.3.2 Secondary Objectives

- Use kinematic vehicle model for testing and designing purposes.
- Visualize and create a scaled-down model of the dry port and goods terminal for testing purposes.
- Develop algorithms for localization and tracking of the i-Dolly.

1.4 Limitations

- Use of a single-track vehicle model with collapsed axles which assumes Ackermann geometry and hence neglect factors such as tire scrub.
- Localization and tracking with the use of ArUco markers has been used to mimic real-world GPS positioning systems. No development on real-world application of ArUco will be done in this thesis. No comparison of this will be done to GPS usage in real-world application.
- The tires on the scaled-down truck do not exhibit the real-world tire due to less traction. These are directly related to the inaccuracy of axle loads on the truck.

2
Theory

This chapter of the report introduces various research areas under localization, path planning, and path-following controllers. It explores the working of different techniques as well as why the may or may not be suitable for use in this thesis project. Many techniques were investigated. Those presented below are the few prominent ones that were looked into extensively. The following logic flow figure 2.1 explains how the information would flow in order to control a vehicle autonomously:

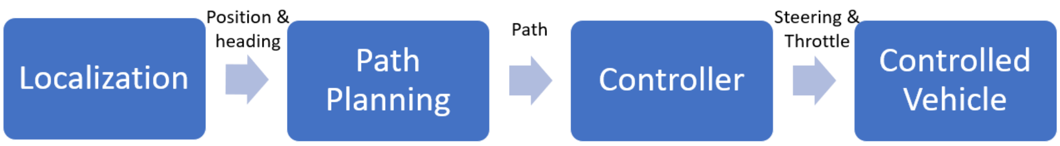

Figure 2.1: Logic Flow for controlling a vehicle

2.1 Localization

The positioning system for the project was chosen after studying different technologies such as Global Positioning Systems (GPS), LiDAR Simulataneous Localization and Mapping (SLAM) and ArUco Marker detection using Camera Vision. The advantages and disadvantages are discussed in the following sections.

2.1.1 Global Positioning System

The localization of autonomous vehicles is typically achieved using Global Positioning System, along with a fusion of Inertial Measurement Unit data and Odometer data. GPS provides global coordinates of the device but causes errors due to physical obstructions. They are usually fused with IMU and odometer data to provide a better estimate of the pose of the vehicle. However, GPS commonly provide horizontal accuracy of 3 metres [15]. For the purpose of this project, which is scaled down 1/14 times, this method of positioning would not prove feasible as the size of the scaled down model is in the order of centimetres.

2.1.2 LiDAR SLAM

Another possible method for localization is the use of 2D LiDAR Simulataneous Localization and Mapping (SLAM). This approach would require a 2D LiDAR to be mounted on the scaled down truck model. The LiDAR reading could be used to build a virtual map of the surrounding. Based on loop closure and point cloud mapping, the developed algorithm would allow determining the position of the object. However, using this method would require the development of an accurate algorithm which would consume a lot of time and effort. This is not being used here to reduce the complexity and due to the fact that real-time application of this approach would be out of the scope of this thesis.

2.1.3 ArUco Markers

As an alternative, a ceiling-mounted, downward-facing camera along with ArUco markers can be used as seen in [19]. The ArUco markers would be affixed upon the i-Dolly, the trailer, the loading bay. A camera would read these markers and tracking would be done using the OpenCV library on Python. These markers allow the tracking of X and Y coordinates in a 2D plane as well as the orientation with respect to the virtual x-axis. The benefit of using this method is the reduced complexity and that it mimics the real-life combination of GPS-IMU-odometer fused data. Additionally, ArUco markers may be used to assist the coupling and reverse maneuvers.

2.2 Path Planning

Path-planning algorithms are crucial to understanding how an autonomous vehicle selects what path to take from the start position to a goal state. Various path planning algorithms have been developed for indoor motion. For the purpose of this project, it is important to take a look at some of the most prominent path planning algorithms and what their advantages would be in the given use case.

2.2.1 A* Algorithm

The A* algorithm was first introduced by Hart, Nilsson & Raphael in 1968 [7]. It is a grid-based path generation algorithm, which evaluates the optimal path based on a cost function

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

where g(n) is the cost of reaching a particular node on the grid map and h(n) is the cost of reaching the goal state from that same node. The heuristic cost h(n) is used to direct the search algorithm towards the goal state node in the grid. Obstacles can be pre-defined into the algorithm and nodes corresponding to these obstacles are avoided by the path planner. The path is generated such that obstacles are avoided while keeping an acceptable optimal path to the goal.
Although this algorithm more or less guarantees a solution, the computational time required to create the path is not very feasible even in low-speed maneuvers.

2.2.2 Rapidly-exploring Random Trees (RRT) Algorithm

A quicker algorithm than the A* is the Rapidly-exploring Random Trees algorithm or simply RRT. The main advantage of this method is the ability to explore large

unknown spaces in order to generate paths to the goal state. As seen in [8], the algorithm works by first generating a random point in the work space, joining the closest branch node to that point, and iterating this till it reaches the goal state.
This is also one of the main disadvantages of this algorithm. Since, the generation of a point in the work space is stochastic in nature, the computational time required to return an acceptable path is unknown. There is also no guarantee that a generated path would be optimal. The path generated in each iteration would not remain constant. Instead, due to its stochastic nature, every path generated would be different even if the goal and initial states remain unchanged.
Furthermore, similar to the A* algorithm, the generated way-points from these algorithms do no provide a smooth path that is needed for a path-following controller.

2.2.3 Cubic Spline

Spline is a special type of piecewise polynomial. Piecewise is a function which is defined by multiple sub-functions for different intervals. Cubic spline is used to create a smooth path between the given waypoints. This is done by joining the n points by n-1 cubic polynomials. One of the few advantages of using a cubic spline (or spline) interpolation is it helps reduce the complexities which are present if a higher degree polynomial is used. Overfitting is one of the problems faced when using a higher degree polynomial as seen in[20]. Higher order polynomials usually get an oscillatory behaviour which are not very realistic paths for a vehicle to follow.

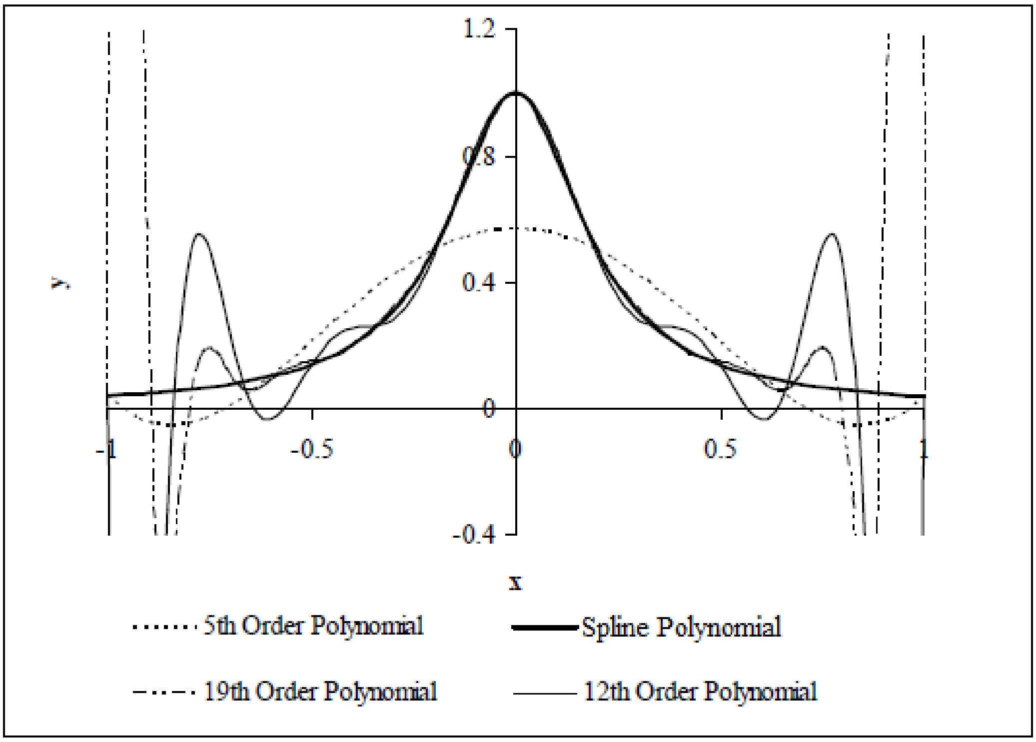

Figure 2.2: Spline vs Higher order Polynomials [20]

While on the other hand, using linear equations to join the given n points does not provide a smooth curve. A spline curve also provides with the heading angles which are smooth varying values for each segment and hence it helps in giving more stable and less fluctuating control inputs.

2.3 Controller

2.3.1 Pure Pursuit Controller

A path tracking controller like the pure pursuit controller influences the motion of the vehicle based on a reference path and the geometrical kinematics of the vehicle model. It tracks a 'look-ahead' point on the reference path. The steering angle input is based on the look-ahead distance and geometrical parameters of the vehicle.

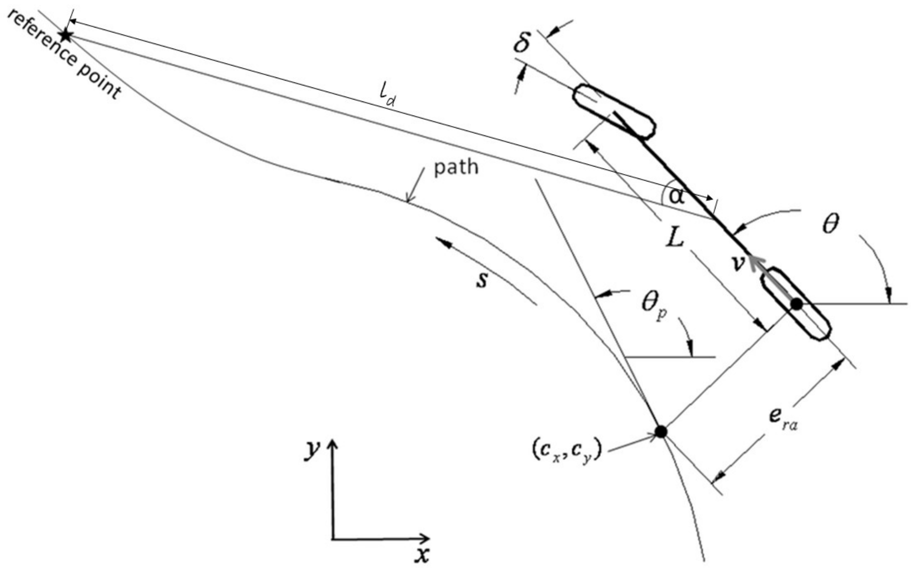

Figure 2.3: Vehicle Kinematic Model and look-ahead distance [10]

A reference point along the length of the vehicle is used as the point from which the look-ahead distance l_d is calculated to the reference trajectory. The equation for calculating the required road wheel angle is given as

$$\delta = \arctan\left(\frac{2L\sin\alpha}{l_d}\right) \tag{2.2}$$

Hence, the steering angle request is proportional to the deviation of the vehicle from the reference trajectory. However, since the controller does not account for the cross-track error given by $e_r a$, it tends to cause the vehicle to oscillate around the reference trajectory before stabilizing. This would be especially prominent along paths of high curvature and at higher speeds which can cause high degree of off-shoots.

2.3.2 Stanley Controller

A Stanley controller was developed by Stanford University for their racing team in the DARPA Grand Challenge 2005. Stanley controller is similar to a pure pursuit controller but it tracks the front axle instead of the rear axle. The benefits of using a Stanley controller are that it can be applied to nonlinear system models and it is less computationally expensive compared to a non-linear MPC controller. This controller considers a single-track kinematic model of the vehicle. The vehicle speed is controlled by a proportional integral controller and is uses minimum computing resources. The steering control law considers the heading error, the crosstrack error and the vehicle velocity. The typical speeds for getting an accurate result from this controller are lesser compared to pure pursuit or MPC controller. This is extremely suitable for this project as all the maneuvers involved in this work have relatively low speeds.

2.4 MATLAB Simulations

In order to theoretically analyze the performance of the Stanley controller's path following ability, a MATLAB simulation was set up. This MATLAB simulation set up consists of a vehicle model, a path file consisting of path coordinates and corresponding yaw values created beforehand, and a Stanley controller for path following the path. The aim of the simulations is to tune the stanley controller and correlate its performance and path following ability to the physical setup. The simulations could then be used for future study and tuning of the controller for different maneuver types.

2.4.1 Vehicle Model

The model used for the simulation was developed as a research project at Chalmers University of Technology in collaboration with engineers at Volvo Autonomous Solutions and Volvo GTT [5].
The equations describing the motion of the vehicle are of a single-track model [6]. The equations for tyre dynamics assume a linear tyre model with no lateral load transfer. The simulation model also includes effects of combined slip, road gradient and air resistance. A single unit vehicle with two axles - steered first axle and driven second axle is simulated.

2.4.2 Path Coordinates and Orientation

The path is generated by using a few input way-points which are fed to a way-point generator. It also consists of yaw values at each point which is defined by X and Y coordinates which are stored as a '.mat' file. This acts as the input to our controller.

3

Method

This chapter of the report begins with an outline of the simulations and hardware components that form the i-Dolly and trailer combination, the network communication setup, and the over-head camera setup. It then reviews the communication software used between the various components. Further, it explains the implementation of the selected localization technique used in the project. It also describes the strategy used by the path planning and control algorithms to achieve the maneuvers that have been envisioned.

3.1 Simulations

The set up utilizes the vehicle model and equations of motion to simulate the motion of the truck along the predefined trajectory. By integrating the Ordinary Differential Equations, we arrive at the vehicle states such as vehicle position (x, y), vehicle longitudinal and lateral speeds (v_x, v_y), yaw angle and yaw angle rate (ψ, $\dot{\psi}$) and wheel angle (δ).

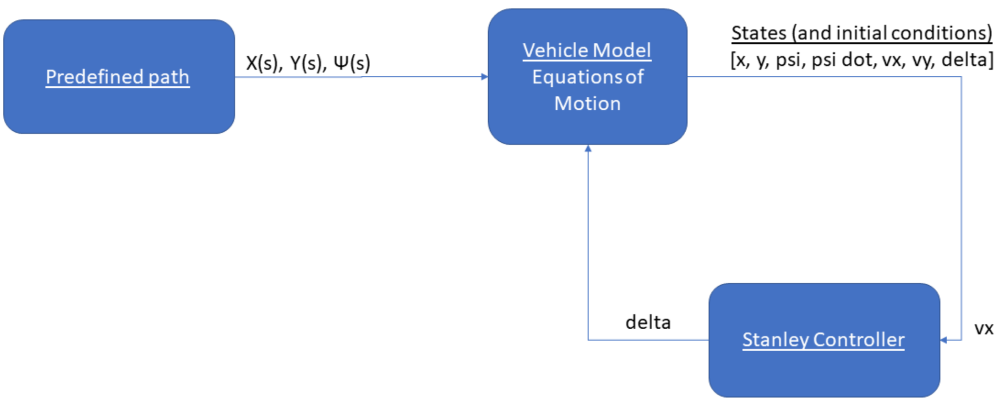

Figure 3.1: Block diagram of the simulation setup

3.2 Hardware Setup

3.2.1 i-Dolly and Trailer

The i-Dolly and trailer combination is represented by a 1:14 scaled-down model of a Volvo truck and trailer. The i-Dolly consists of two motors for steering and propulsion. It also houses the Electronic Control Unit (ECU) which provide steering and propulsion signals to the i-Dolly. Both rear axles are have equal loading. These axles have propulsion capability. The front axle is only steerable. The Truck/i-Dolly dimensions are as shown in figure 3.2.

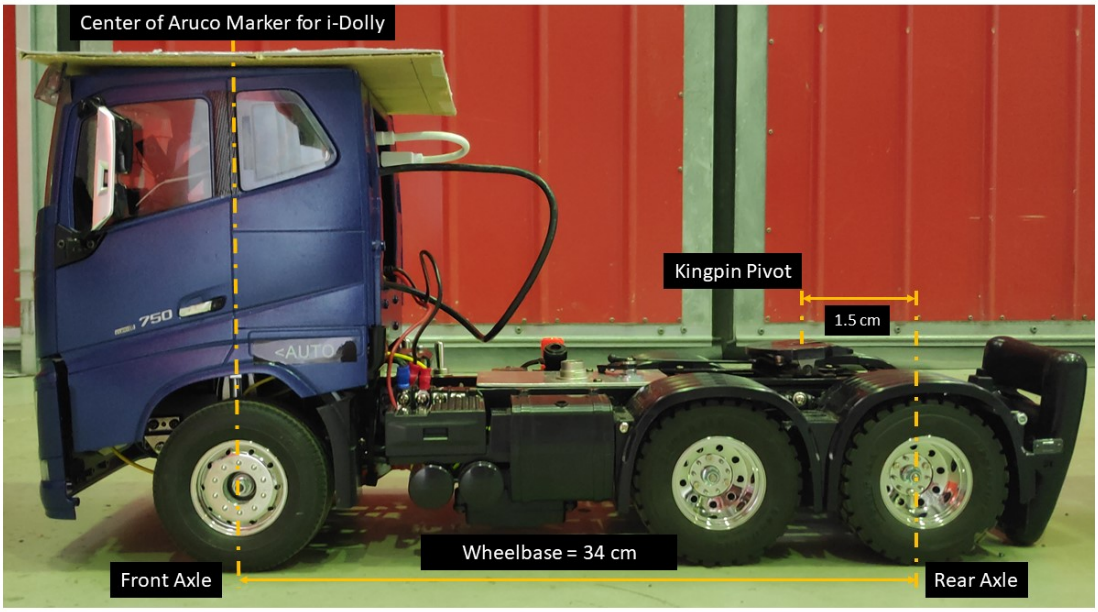

Figure 3.2: Scaled-down 1:14 model of Volvo truck used as i-Dolly

In order to provide the ECU with control signals from the controller, there exists a BeagleBone Board (BBB) which uses Pulse Width Modulated (PWM) signals sent through its General Purpose Input Output (GPIO) pins. The BBB receives control input via a Raspberry Pi (RPI). The RPI, although used as an intermediary, is required as a modular device to have options for using on-board cameras or LiDAR in future work. The trailer shown in figure 3.3 represents trailers that would be parked at dry ports and need to be transported to customer loading docks a few kilometers from the dry port. The rear axles of the trailer are free-rolling. They are neither steerable nor are they driven. All three axles are load bearing. The i-Dolly would be required to maneuver to the location of the trailer and reverse into the kingpin.

Figure 3.3: Scaled-down 1:14 model of Volvo Trailer used for coupling operation to i-Dolly

3.2.2 Network Diagram

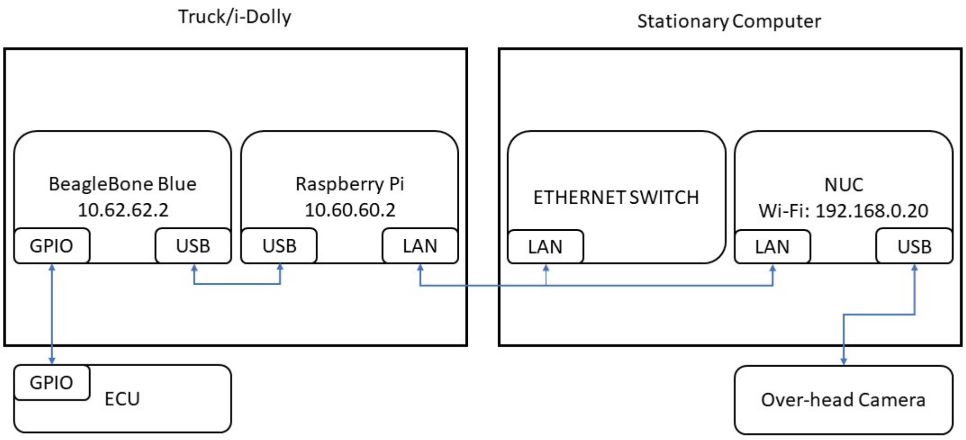

Figure 3.4: Network Diagram

The hardware setup includes an over-head camera, an Intel NUC mini PC, an Ethernet switch, a Raspberry pi board, a BeagleBone Blue board, and a scaled-down model of a Volvo truck which was used to mimic a self-steered and self-propelled intelligent dolly (i-Dolly). Figure 3.6 shows the physical setup of the hardware. The LAN connection between i-Dolly and the NUC computer is through a 3m Ethernet cable to allow for maneuvers in a large radius.

3.2.3 Overhead Camera

The camera is mounted at a height of 3.4m from the ground. The frame of view of the computer is a 3x3.5m area below within which the i-Dolly would be able to carry out maneuvers.

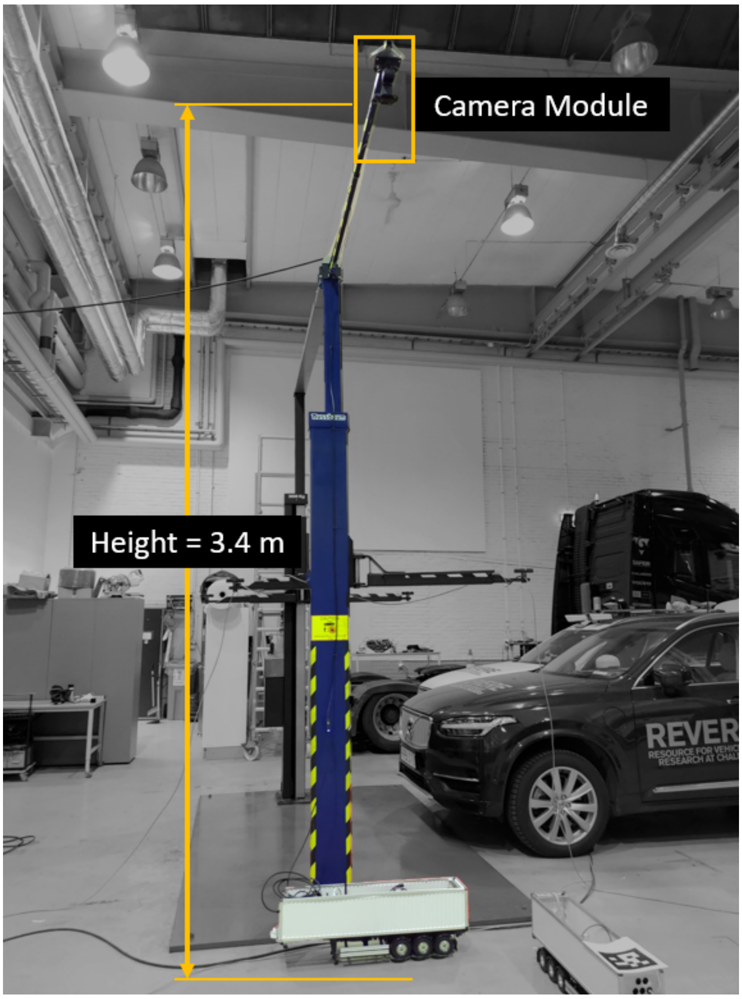

Figure 3.5: Camera mounted at a height above the ground

3.3 Software Configuration

3.3.1 Logic Diagram

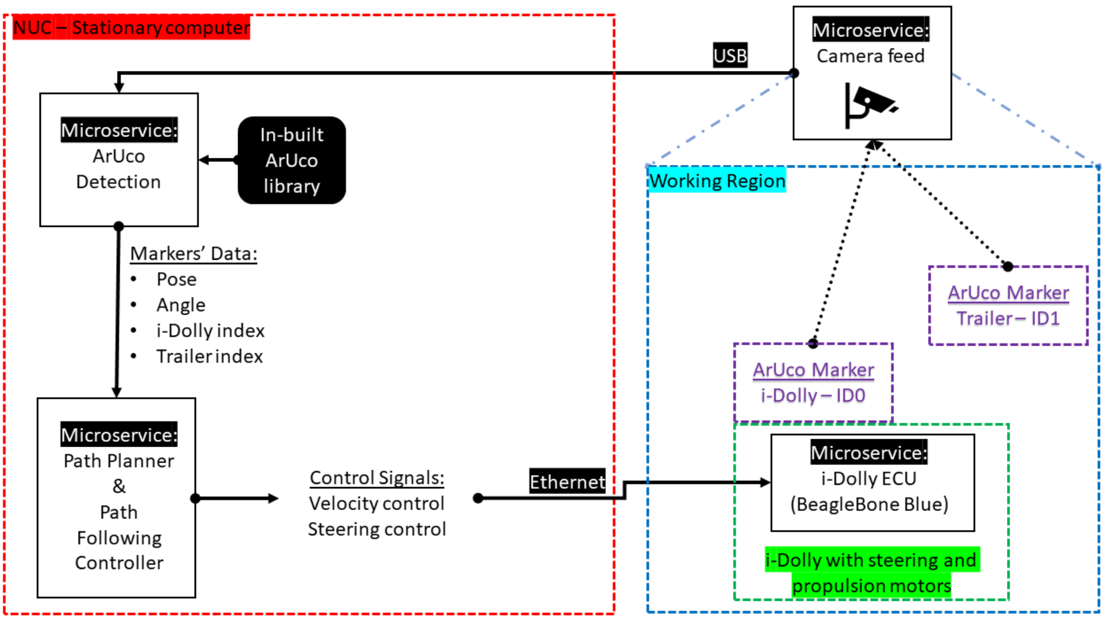

Figure 3.6: Microservice logic flow in openDLV environment

3.3.2 Microservices and openDLV

For the purpose of establishing communication between the various devices, computers and actuators for controlling the i-Dolly, containerized software microservices was utilized[11].

3.4 Localization - ArUco markers & camera vision

The tracking of the scaled vehicle using position and pose would be important in order to implement a path planning algorithm, as well as a path tracking algorithm. For a full-scale truck, a sufficiently accurate GPS system would be adequate to provide us with this data. Due to the scale of this project however, it would become necessary that one use a system that works with an acceptable level of accuracy.

ArUco markers are an open-source library of fiducial markers [4] that can be used with camera vision tools such as OpenCV. By using OpenCV, detection of these markers, and their coordinates, is possible. These coordinates may then be converted from pixel coordinates and distances to metric distances by using a suitable calibration. Furthermore, the corner coordinates of these markers, with simple coordinate geometry, make it possible to estimate the orientation of the markers with respect to a virtual x-axis.

An overhead camera placed at a height of 3.7m is used to detect markers placed within the field of view below. ArUco markers are placed on the scaled vehicle model below.

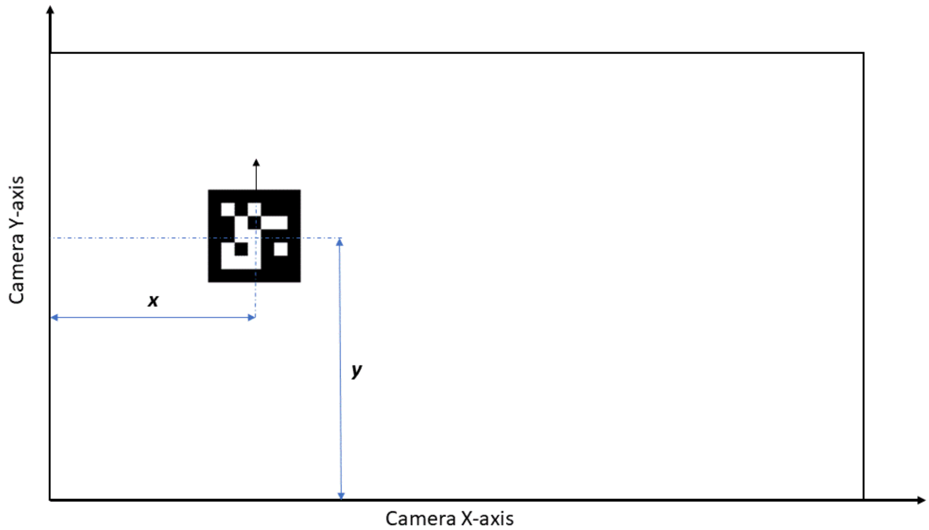

Figure 3.7: X and Y coordinates using ArUco Marker detection

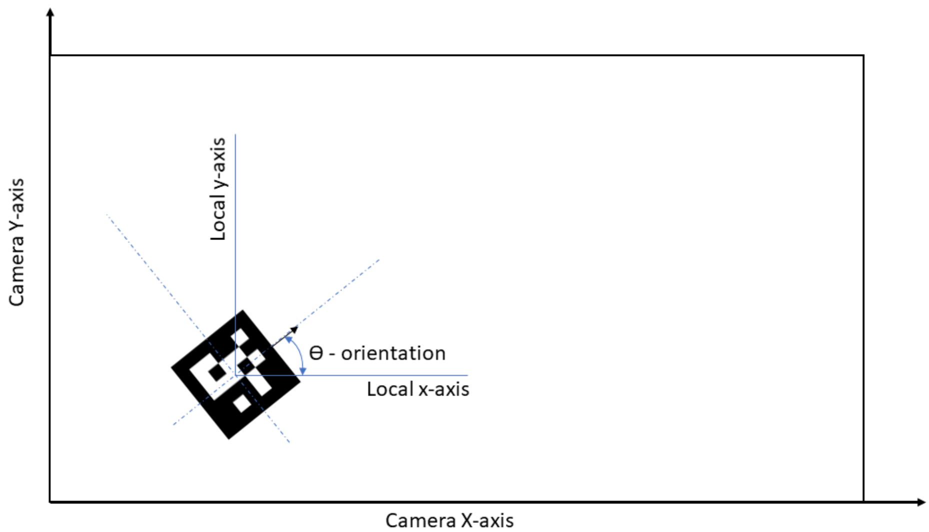

Figure 3.8: Orientation using ArUco Marker detection

3.4.1 Calibration

A calibration of the ArUco marker algorithm is needed to convert the pixel coordinates obtained from the camera frame to X and Y coordinates in terms of centimetres.
In order to achieve this, two ArUco markers were placed at a fixed horizontal distance from one another. The pixel distance between the two markers was obtained from the camera frame. Thus, the conversion factor "pixel2cm" to convert 'n' pixel units to centimeters were calculated as follows:

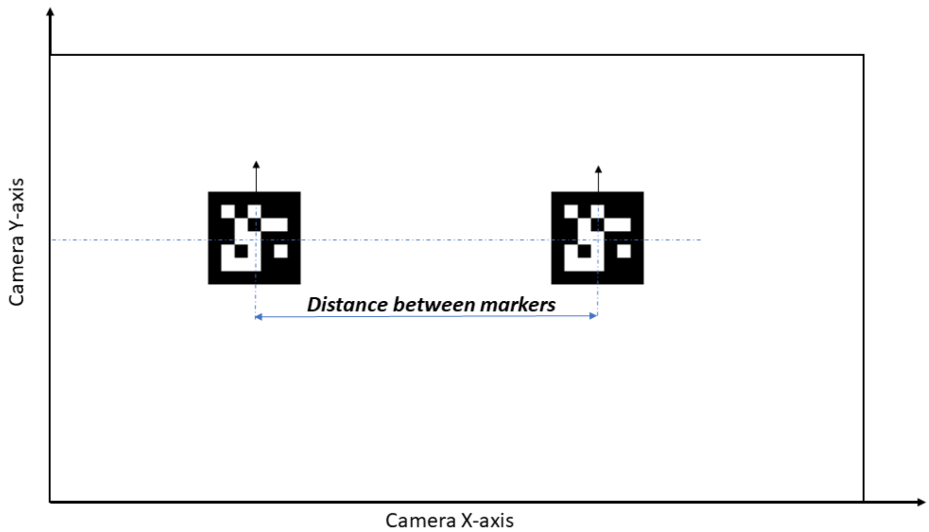

Figure 3.9: Determining the pixel to distance conversion factor by using a fixed horizontal distance

$$pixel2cm = \frac{\text{Distance between markers [cm]}}{\text{Distance between markers [pixels]}}$$

For the purpose of determining the orientation of the markers, the simple trigonometric tangent function was applied to the marker's corner coordinates.

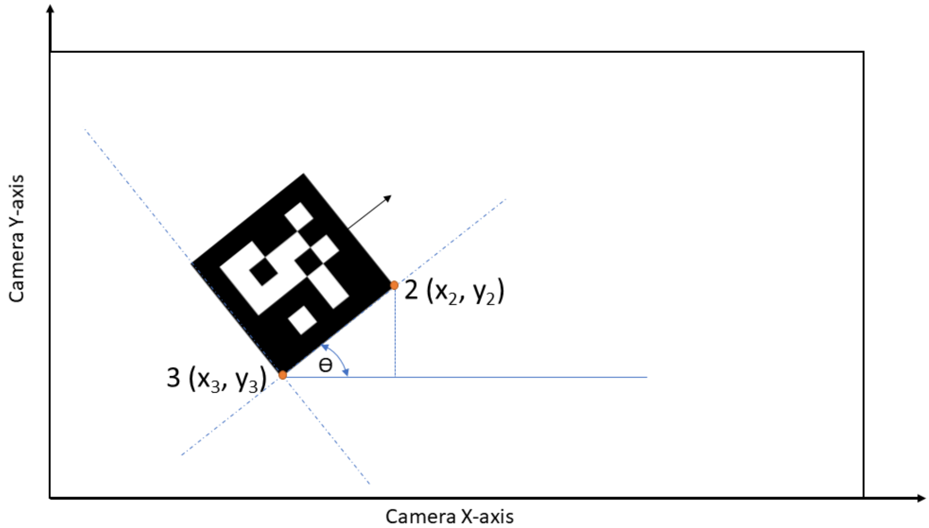

Figure 3.10: Determining the ArUco marker orientation using corner co-ordinates

$$\theta = \arctan(\frac{|y_2 - y_3|}{|x_2 - x_3|})$$

3.5 Path Planning

The path for a vehicle should be smooth for various reasons such as: keeping the movement of vehicle realistic, reduce wear and tear of moving parts, reduce energy consumption, etc. For this purpose, the cubic spline interpolation technique was utilized as it forms a smooth curve between n points which lie in the interval [a, b] where a is the starting point and b is the final point[13].

$$a = x_0 < x_1 < \cdots < x_n = b \tag{3.1}$$

Spline creates a cubic polynomial between every 2 points in the interval which is of the form:

$$f_i(x) = a_i + b_i\,(x - x_i) + c_i\,(x - x_i)^2 + d_i\,(x - x_i)^3 \tag{3.2}$$

Here, the first and second derivatives of the function are continuous in the interval [a, b]. By making sure that the first and second derivatives are continuous, a curve can be generated that is continuous and passes through all the control points. The second derivative matches the curvature at the control points, while the first derivative matches the slope and the function itself is continuous which means that it passes through all the control points as seen in 3.11.

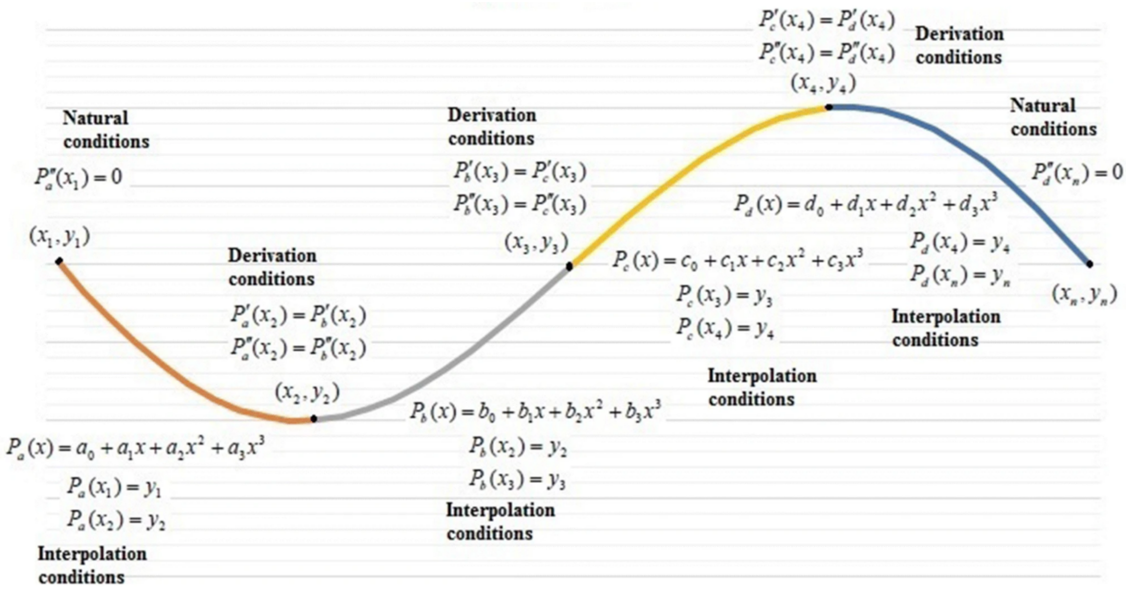

Figure 3.11: Cubic Spline [12]

In this project, the spline is used to interpolate the starting point, the target points and a few more way-points in between. These in-between way-points are called control points and are calculated in order to keep the ending and beginning of the path relatively straight. This is done for a reverse coupling maneuver as it is important to keep the orientation of the semi-trailer and i-Dolly aligned and in a straight line. This helps in keeping the kingpin exactly above the fifth wheel.

3.6 Selection of controller

As mentioned above 2.3, for the purpose of demonstration a Stanley controller is chosen over pure pursuit controller. This is done because a pure pursuit controller only tracks the heading error while a Stanley controller tracks heading and cross-track error (explained below 3.7).

3.7 Stanley Controller

A Stanley controller is comprised of parts:

- The first one is the kinematic model, which neglects the vehicle inertia and is effective for low speed driving. These constrains make the controller globally stable.
- The second part is the dynamic model which is more complicated and more accurate as it includes the inertia, steering servo actuation and tyre slip. This model is also closer to the realistic dynamics of a vehicle. The dynamic model deals with the non linear part of the vehicle model such as tyre slip effects,

sidelip of tyres, gradient and steering servo motor. For the purpose of low speed maneuver, the dynamic modelling will be neglected.

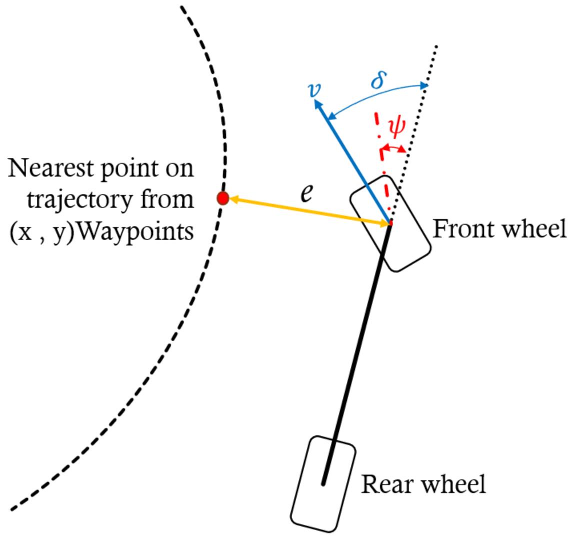

Figure 3.12: Single Track Kinematic Model for Stanley Controller

The kinematic model considers a single track model of the vehicle as shown in 3.12. As seen in the figure, the cross-track error at time t is $e(t)$ - calculated from the front axle (or the tracking point) to the nearest point on the trajectory from the cubic spline path planner. There is also heading error which is tracked by: $(\psi(t) - \delta(t))$. Here $\delta(t)$ is the road wheel angle with respect to the vehicle and $\psi(t)$ is the heading angle of the vehicle with respect to the nearest point on the trajectory. The derivative of the cross-track error is

$$\dot{e}(t) = v(t)\sin(\psi(t) - \delta(t)) \tag{3.3}$$

where the wheel angle is limited to $|\delta(t)| < \delta_{\max}$ and $v(t)$ is the vehicle speed. The yaw rate is given by:

$$\dot{\psi}(t) = r(t) = -\frac{v(t)\sin(\delta(t))}{L} \tag{3.4}$$

where L is the wheelbase of the vehicle. From the above equations, the road wheel steering control $\delta(t)$ is defined as:

$$\delta(t) = \begin{cases} \psi_{ctrl}(t) + \arctan\frac{k*e(t)}{k_{soft}*v_x(t)} & \text{if } \left|\psi_{ctrl}(t) + \arctan\frac{k*e(t)}{k_{soft}*v_x(t)}\right| < \delta_{\max} \\ \delta_{\max} & \text{if } \psi_{ctrl}(t) + \arctan\frac{k*e(t)}{k_{soft}*v_x(t)} \geq \delta_{\max} \\ -\delta_{\max} & \text{if } \psi_{ctrl}(t) + \arctan\frac{k*e(t)}{k_{soft}*v_x(t)} \leq -\delta_{\max} \end{cases} \tag{3.5}$$

where k and k_{soft} are gain factors used to tune the steering controller. For a reversing maneuver, the error calculation will remain the same but the rear wheels will be used for steering.

3.8 Maneuvers

In a dry port and goods terminal, there are at least three basic maneuvers which can be used to describe the movement of an i-Dolly:

- Reversing of the i-Dolly for coupling the fifth wheel.
- Shunting of i-Dolly with semi-trailer.
- Reversing of i-Dolly with semi-trailer into a loading terminal.

These maneuvers will be explained more below.

3.8.1 Assumptions

It is necessary make the following assumptions so as to achieve successful coupling of the semi-trailer with the i-Dolly:

- The start position and orientation of the i-Dolly and semi-trailer are pre-defined for coupling.
- The orientation of the i-Dolly and semi-trailer in their start positions are pre-defined.
- The tyre model is assumed to replicate a real life truck.
- For low speed manoeuvres the single track model of the vehicle is accurate enough.
- There are no obstacles in the vehicle path.

3.8.2 Reversing of the i-Dolly for coupling the fifth wheel

Based on the known starting positions of the i-Dolly, a path planning algorithm is created. This path planning algorithm is based on geometry which is visualized as shown in figure 3.13.

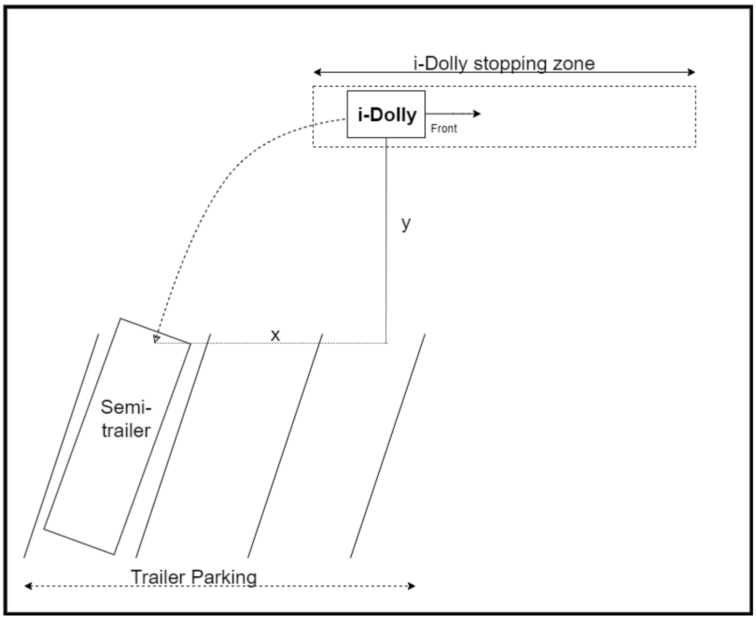

Figure 3.13: Geometrical representation of path for coupling maneuver

The radius of turn is constant as it is calculated using the start positions which are always fixed. The road wheel angle is calculated based on this radius and the vehicle model which is given as an input to the i-Dolly. To make the i-Dolly follow the path, it is proposed to utilize a Stanley controller for keeping minimum possible complexity while retaining performance. The i-Dolly is made to stop very near the trailer and then the i-Dolly reversing begins. This reversing can be done with the help of ArUco marker on the front of the trailer and a reversing camera on the i-Dolly.

3.8.3 Shunting of i-Dolly with semi-trailer

Shunting will involve the combination of i-Dolly with semi-trailer to be maneuvered from the trailer parking (or location where semi-trailer was coupled) to the loading dock. This maneuver will involve movement of the combination on a pre-defined and obstacle free path (as there is no obstacle detection in this thesis).

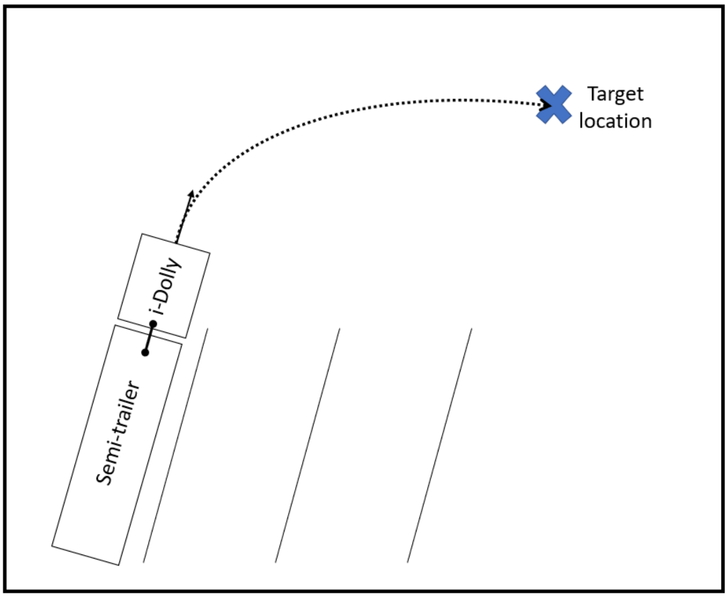

Figure 3.14: Geometrical representation of path for shunting

3.8.4 Reversing of i-Dolly with semi-trailer into a loading terminal

The concept is similar to reversing the i-Dolly to couple the trailer but with an augmented vehicle model. The i-Dolly and semi-trailer will stop in a predefined stopping zone and then it will reverse. There will be ArUco markers on the loading dock to give a position of the loading bay and then there will be ArUco in front of the loading bay which will give help in reversing accurately.

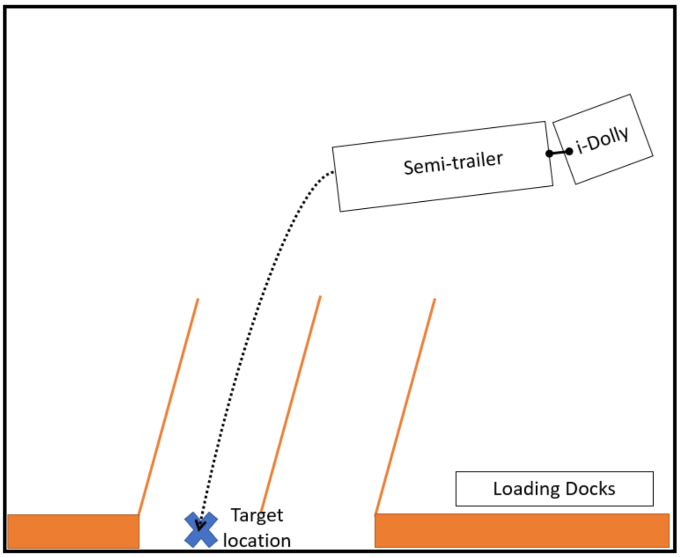

Figure 3.15: Geometrical representation of path for reversing into loading terminal maneuver

4

Results and discussion

This chapter of the report summarises the outcomes while testing the controller which is done on a 1/14 scaled model. It discusses the outcomes in each task - localization, path planning and path following. Following that, there is a summary of testing the same controller on a non-linear simulation model of truck unit. It also explains the errors obtained how they affect the final objectives.

4.1 Localization

The configuration of camera height 3.4m with a marker edge size of 17cm proved to have the least error as shown in table 4.1.

Table 4.1: Variation in measurement error with a change in camera position and Aruco marker size

Height above ground (m)	Marker size (cm)	Pixel length (cm/pixel)	Error (cm)
3.7	15	0.244	1.65
3.7	17	0.245	1.35
3.4	17	0.228	0.95

In order to obtain coordinates of the i-Dolly and semi-trailer with the best possible accuracy, parameters such as camera height from the ground and Aruco marker size were varied and the error in reading a fixed distance was measured and tabulated. The basic requirement was to have the camera at a height which provides enough ground working space for the manoeuvres. The height of camera affects the marker reading precision and the available working space as follows:

- With the increase in height, there was more ground space for manoeuvres but it resulted in an increase in marker detection error and image edge distortion.
- With the decrease in height, there is a great loss of ground space which is a concern as there is a 1/14 scaled model of semi-trailer which is included in the demonstration.

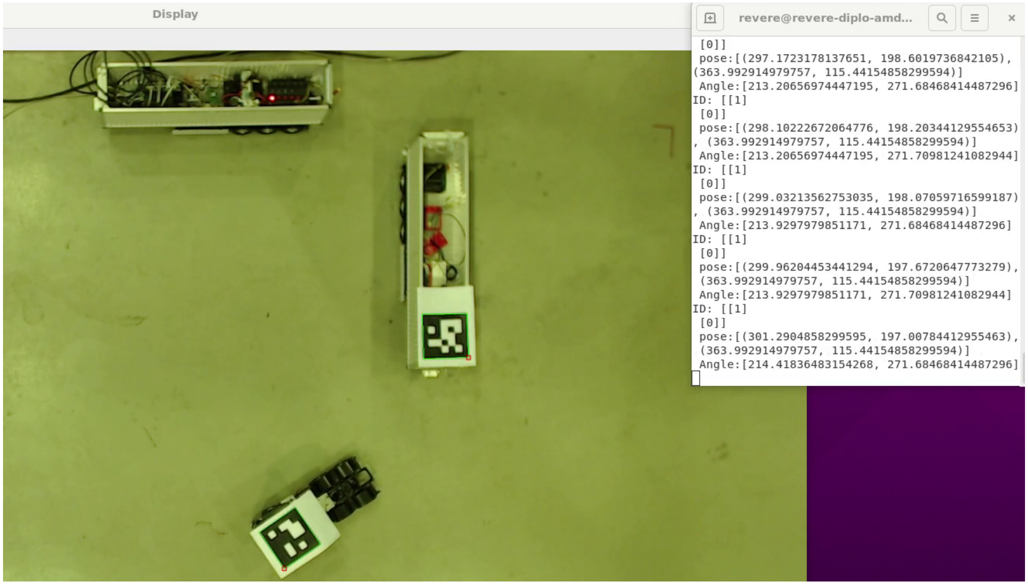

Figure 4.1: Detection of Aruco markers along with position and orientation

One of the error which was unaccounted in the design of the localization was the vibrations in the camera mounting. Even small vibrations in the mounting mechanism resulted in an error in the marker position reading due to the height of camera, which eventually affected the final results.

4.2 Path planner

Trajectories were generated using positions of the i-dolly and trailer as inputs (as seen in figure 4.2). The starting point of the path was the front axle position of the i-dolly and the end point was the kingpin on the trailer. For the purpose of having credible results, the path was generated for three different starting positions with respect to the trailer: left side (figure 4.2), right side (figure 4.3) and straight ahead (figure 4.4) of the trailer which is shown in the figures below.

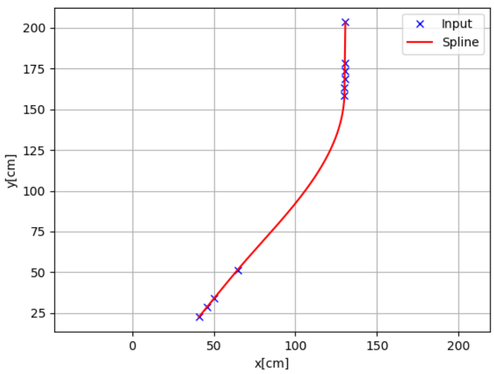

Figure 4.3: Path generated when i-dolly starts on the right hand side ahead of the trailer

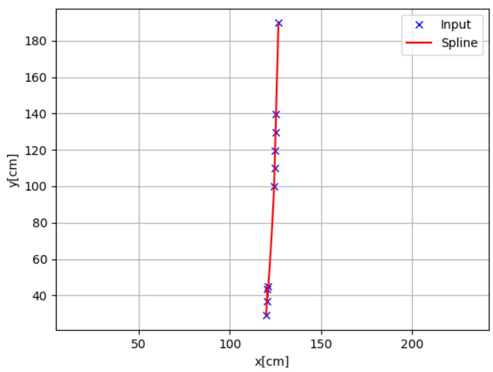

Figure 4.4: Path generated when i-dolly starts ahead of the trailer

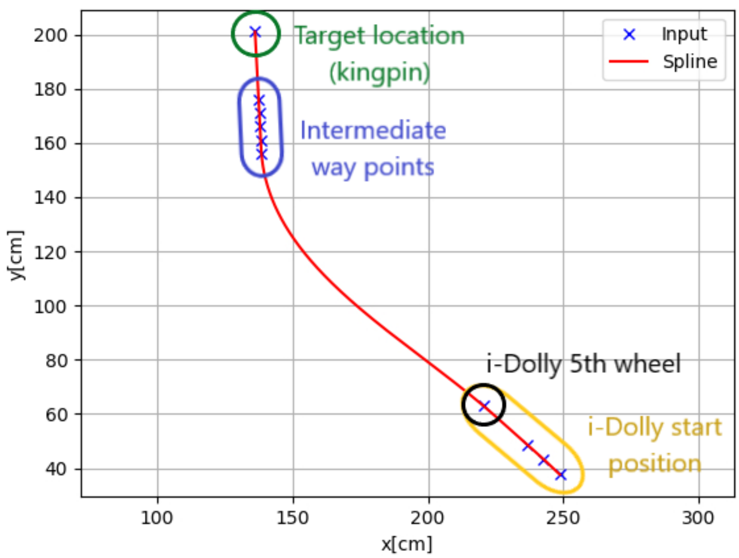

Figure 4.2: Path generated when i-dolly starts on the left hand side ahead of the trailer

As seen in the above figures, multiple way points in front of the trailer (intermediate points) are also used as inputs for path planning. This is to ensure that the i-dolly reverses so as to maintain an orientation parallel to the trailer. A buffer distance was maintained between the kingpin and the last few way points to make sure the last part of reversing was in a straight line (as it happens in real life). This also helped in making sure that the controller had enough distance in the end to correct its orientation and couple. After many trials, an appropriate buffer value was chosen as there were downsides to choosing a high and low buffer value which are discussed later. By changing the buffer values, the intermediate way points also changed which affected the path as follows:

- An increase in the buffer value meant that the intermediate way points were further from the king pin position. While this was helpful in making sure that the last part of reversing was in a straight line and there was quite a lot of distance to get on the correct path, it had a big disadvantage. Due to a large gap between the intermediate way points and the target, the path generated had a very high curvature which the i-Dolly was unable to follow due to physical limitations on the maximum road wheel angle angle.

- If the buffer value was reduced, the intermediate way points came closer to the target position. Due to this, the path generated had relatively less curvature but there was not enough distance for the i-Dolly to correct its orientation at the end. This meant the i-Dolly would try to couple with the trailer at an angle which is not an ideal situation.

4.3 Test with Stanley controller

As explained earlier in section 3.7, the Stanley controller considers the rear driven wheels as front axle for reversing. There are various factors which affect the performance of the Stanley controller which are stated below and explained later:

- Tracking point
- road wheel angle input
- Vehicle velocity
- Heading and cross-track error

4.3.1 Tracking Point

The tracking point 3.7 was chosen to be 42.5cm behind the ArUco marker which was placed exactly above the front axle. This corresponds to a distance of 8.5 cm behind the driven axle. The controller stops when the tracking point is closest to the target location.

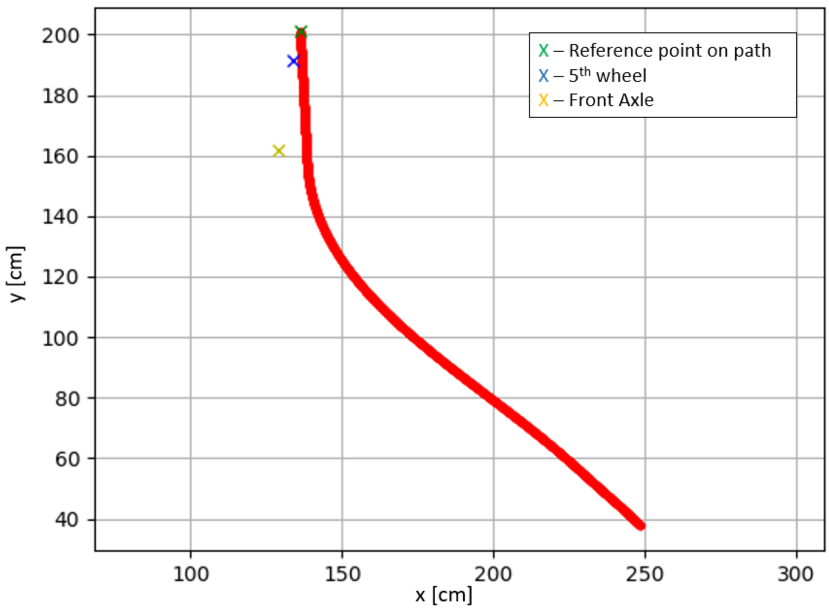

Figure 4.5: Tracking points after the controller reached the goal position

The cross-track error and heading error are based on the tracking point and hence it varies the controller performance significantly. The variation of performance can be explained as follows:

- If the tracking point is chosen to be much further behind the driven axle, the vehicle goes off course and does not reach the required goal. This happens because the controller gives the correction signals based on the error of the tracking point and the planned path much earlier than needed which makes the vehicle deviate from the path.
- If the tracking point is chosen too close to the fifth wheel, the controller fails to complete the required maneuver of coupling. This happens because control

signals for correction are received much later than needed and which gives very less distance for the controller to correct the vehicle path.

4.3.2 Wheel angle input

Due to the physical limitations of the 1/14 scaled model, the i-Dolly had a maximum of 0.56 radians (or 32.6 degrees) available wheel input on each side.

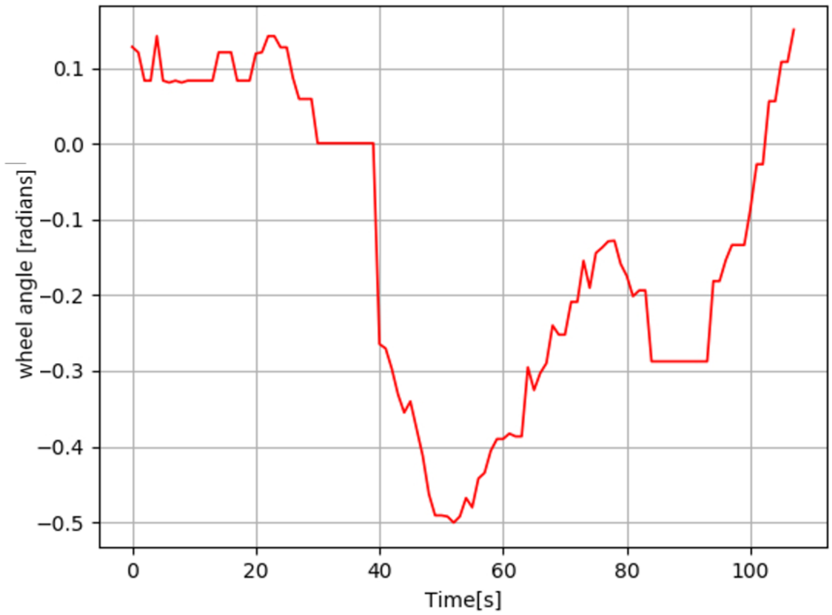

Figure 4.6: Road wheel angle input

There were some inbuilt errors in the hardware such as wheel wobble and steering servo motor error which sometimes resulted in failure in coupling. The error by which coupling maneuver failed due to this was always below 2 cm.

4.3.3 Vehicle velocity

The vehicle velocity that was chosen for the maneuver was 0.4 km/h. This was primarily due to the communication mechanism in openDLV software. If the speed was kept higher than 0.4 km/h, the camera software was not able to track the vehicle in real time, instead it gave the localization values late (i.e delayed in time). In order to track the i-Dolly and semi-trailer in real time, it was important to match the frequency at which the camera was giving output and the movement of the vehicle. For this purpose, it was decided to keep velocity below 0.4 km/h.

4.3.4 Heading and cross-track error

The heading and cross-track error have shows similar characteristics as seen from the figures 4.8 and 4.9 below. Similar characteristic is seen in the road wheel angle input as well. These corresponds to the deviation of the vehicle from the generated path. As the vehicle goes off course, the error values increase and hence the steering input increases to correct it. The gain values for cross-track error chosen here are $k = 0.15$ and $k_{soft} = 8$.

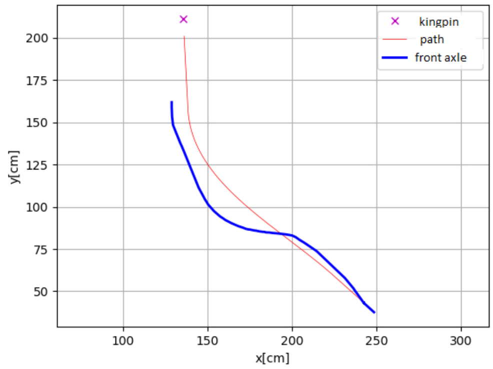

Figure 4.7: Front axle trajectory with respect to path

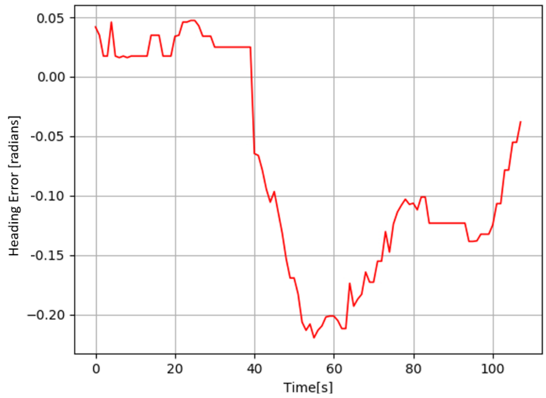

Figure 4.8: Heading error

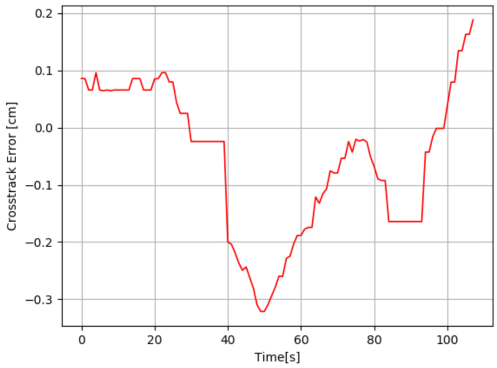

Figure 4.9: cross-track error

The heading and cross-track error depends on the tracking point as explained in section 3.7. The cross-track error has its own gain values k and k_{soft} which affects controller performance. Changing both the gain values at the same time in a similar manner does not yield good results because of the way they are used. Hence for the purpose of testing, k_{soft} was kept constant and the 'k' value varied, to observe the following:

- With the increase in k value, the controller overshoots by a lot from the path in both directions.
- With a very low k value, the controller takes an unusually long time to correct the cross-track error.

4.4 Simulation

The simulation was repeated for different trajectory in all 4 quadrants to verify the controller performance. This was used as a proof to confirm the controller performance irrespective of the vehicle coordinates and orientation. The speed of the vehicle while reversing was a constant 8kmph. The simulation considers the truck to be life-sized with tuned tyre parameters that fit experimental data. The tracking was very similar to what was observed in the physical model.

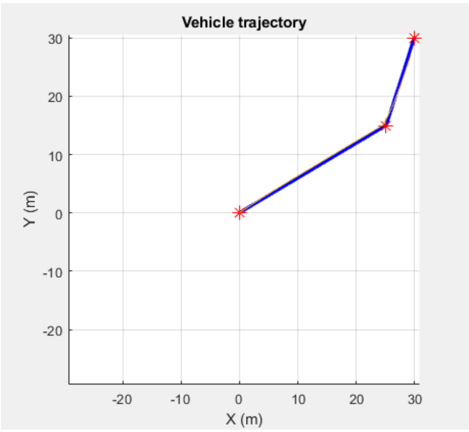

Figure 4.10: Reversing path in quadrant 1

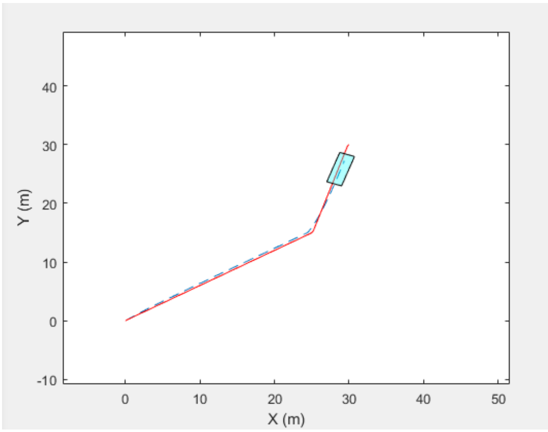

Figure 4.11: Reversing path tracking in quadrant 1

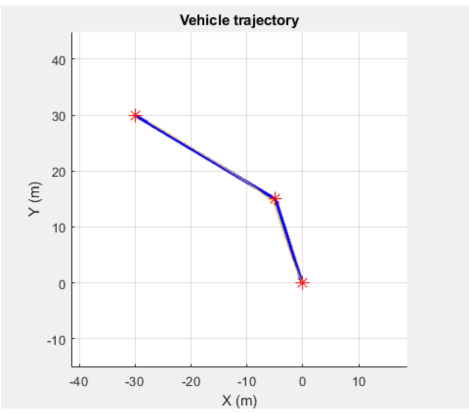

Figure 4.12: Reversing path in quadrant 2

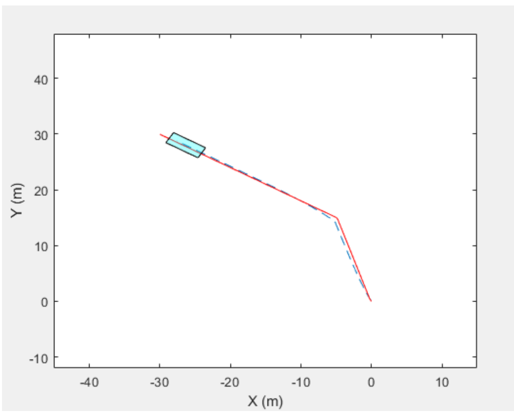

Figure 4.13: Reversing path tracking in quadrant 2

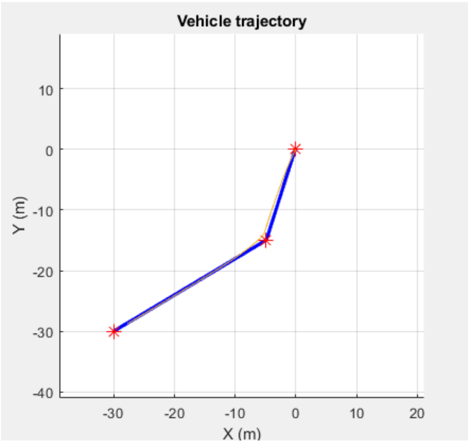

Figure 4.14: Reversing path in quadrant 3

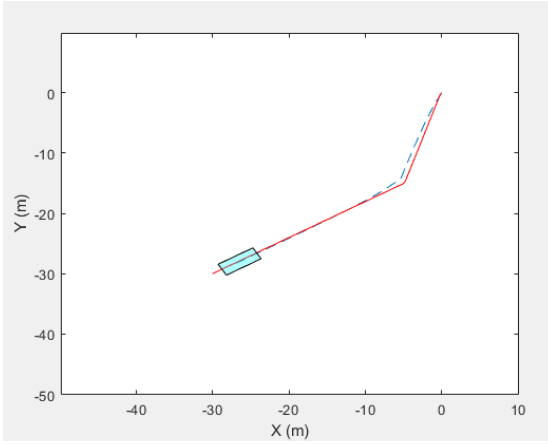

Figure 4.15: Reversing path tracking in quadrant 3

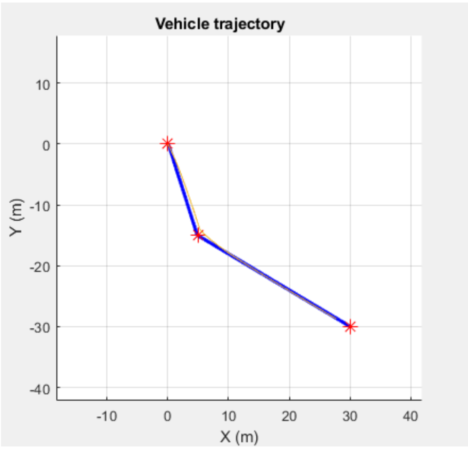

Figure 4.16: Reversing path in quadrant 4

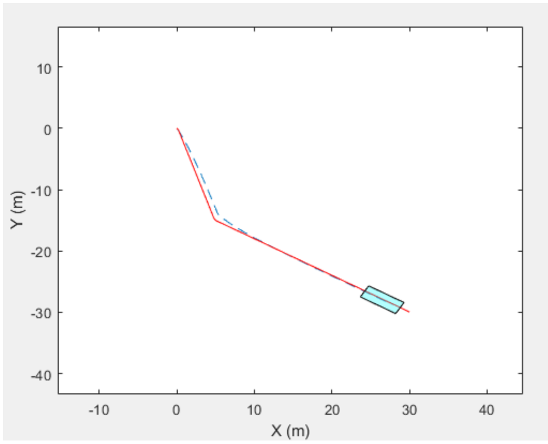

Figure 4.17: Reversing path tracking in quadrant 4

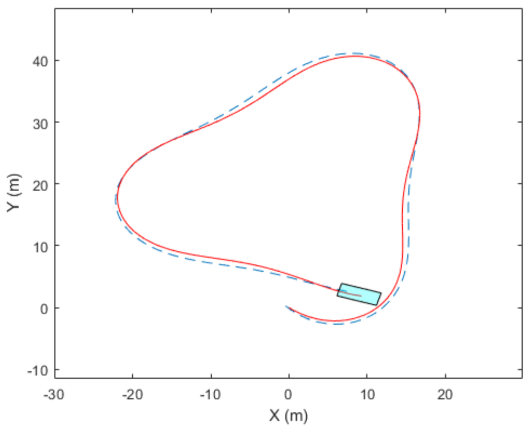

Figure 4.18: Reversing in all 4 quadrants

5

Conclusion

The purpose of this thesis was to investigate a solution to automate the different maneuvers at dry ports in order to reduce human involvement in repetitive tasks. This was demonstrated on a scaled-down model using an over head camera and ArUco markers for localization, a cubic path planner for path generation and Stanley controller for path following. A simulation model was used to verify the findings from the scaled-down model.

The proposed localization method was able to provide the position of the i-Dolly and semi-trailer with a maximum error of 0.95 cm. For a model of this size, it was found that this level of accuracy was not enough for obtaining consistent results. These errors could also be traced back to error in the camera perception. The marker parameters allowed us to extract the positions and orientations of the start and end points. These parameters along with intermediate way points allowed us to generate a path using cubic spline interpolation. This method produced smooth curves which would be realistic for a vehicle to follow. The results from this method were consistent for different start and end positions.

The Stanley controller was a simple yet effective controller used for path following. The reversing maneuver was performed with an accuracy of 0.3 cm. However there were instances when the maneuver was not successful. In this case, the error was 1.5 cm. This occurred due to a culmination of various factors such as: wheel wobble present in the model, steering servo motor error, reduced traction on wheels due to minute dust particles and inaccuracy from ArUco localization.

The thesis was structured around performing the following three maneuvers:

- Reversing the i-Dolly for coupling.
- Shunting of i-Dolly with semi-trailer.
- Reversing of i-Dolly with semi-trailer into a loading terminal.

Due to hardware and software failures and the limited time for the thesis work, only one maneuver which is reversing the i-Dolly for coupling was demonstrated. The next step in order to successfully solve the problem statement would be to perform the remaining two maneuvers based on the methods and results presented in this report. In order to obtain a more accurate controller, simulations were be performed to further tune the controller based on the path curvature required. Furthermore, to increase the consistency of obtaining the desired results, high accuracy hardware could be used.

Bibliography

[1] Larsson, L. (2020, July). Longer and heavier vehicles: The ultimate guide. Volvo Trucks knowledge hub. Retrieved from https://knowledgehub.volvotrucks.com/sustainable-and-profitable-business/longer-and-heavier-vehicles-the-ultimate-guide

[2] Magnusson, N., Odenman, T. (2012). Improving absolute position estimates of an automotive vehicle using GPS in sensor fusion.

[3] KUSUMAKAR, R., KURAL, K., TOMAR, A. S., PYMAN, B. (2017). Autonomous Parking for Articulated Vehicles. HAN University of Applied Science, Netherlands.

[4] Garrido-Jurado, S., Muñoz-Salinas, R., Madrid-Cuevas, F. J., Marín-Jiménez, M. J. (2014). Automatic generation and detection of highly reliable fiducial markers under occlusion. Pattern Recognition, 47(6), 2280-2292.

[5] Ghandriz, T., Jacobson, B., Nilsson, P., Laine, L., & Fröjd, N. (2020). Computationally Efficient Nonlinear One-and Two-Track Models for Multitrailer Road Vehicles. IEEE Access, 8, 203854-203875.

[6] T. Ghandriz. Multitrailer Vehicle Simulation: Generation and IntegrationOf Differential Equations. Accessed: Nov. 6, 2020, doi: 10.5281/zenodo.4248263.

[7] Hart, P. E., Nilsson, N. J., Raphael, B. (1968). A formal basis for the heuristic determination of minimum cost paths. IEEE transactions on Systems Science and Cybernetics, 4(2), 100-107.

[8] LaValle, S. M. (1998). Rapidly-exploring random trees: A new tool for path planning.

[9] Ang, K. H., Chong, G., Li, Y. (2005). PID control system analysis, design, and technology. IEEE transactions on control systems technology, 13(4), 559-576.

[10] Funk, N., Alatur, N., Deuber, R., Gonon, F., Messikommer, N., Nubert, J., ... Siegwart, R. (2017). Autonomous electric race car design. arXiv preprint arXiv:1711.00548.

[11] Berger, C., Nguyen, B., Benderius, O. (2017, April). Containerized development and microservices for self-driving vehicles: Experiences best practices. In 2017 IEEE International Conference on Software Architecture Workshops (ICSAW) (pp. 7-12). IEEE.

[12] Magalhaes, P. A. A., Junior, P. A. A. M., Magalhaes, C. A., Magalhaes, A. L. M. A. (2021). New formulas of numerical quadrature using spline interpolation. Archives of Computational Methods in Engineering, 28(2), 553-576.

[13] Lian, J., Yu, W., Xiao, K., Liu, W. (2020). Cubic Spline Interpolation-Based Robot Path Planning Using a Chaotic Adaptive Particle Swarm Optimization Algorithm. Mathematical Problems in Engineering, 2020.

[14] Jacobson, B. (2016). Vehicle dynamics compendium for course MMF062; edition 2016. Chalmers University of Technology.
[15] Stenborg, E. (2017). Localization for Autonomous Vehicles (Doctoral dissertation, Chalmers Tekniska Hogskola (Sweden)).
[16] KUSUMAKAR, R., KURAL, K., TOMAR, A. S., & PYMAN, B. (2017). Autonomous Parking for Articulated Vehicles. HAN University of Applied Science, Netherlands.
[17] Stranden, J. E. (2019). Autonomous driving of a small-scale electric truck model with dynamic wireless charging (Master's thesis, NTNU).
[18] Matsushita, K., & Murakami, T. (2006, November). Backward motion control for articulated vehicles with double trailers considering driver's input. In IECON 2006-32nd Annual Conference on IEEE Industrial Electronics (pp. 3052-3057). IEEE.
[19] Larrazabal, A. J., Cena, C. G., & Martínez, C. E. (2019). Video-oculography eye tracking towards clinical applications: A review. Computers in biology and medicine, 108, 57-66.
[20] Kaw, A., Keteltas, M. (2009). Spline Method of Interpolation. General Engineering.

A

Appendix 1

A.1 Maneuvers starting from right side of the trailer

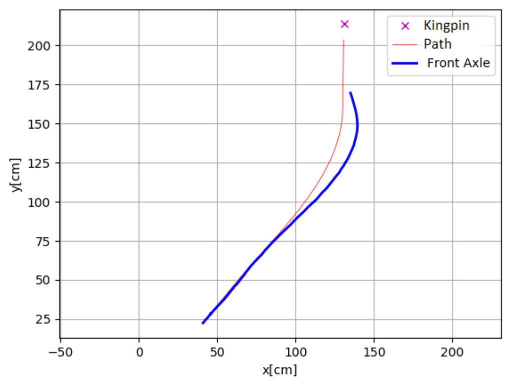

Figure A.1: Front axle trajectory with respect to path

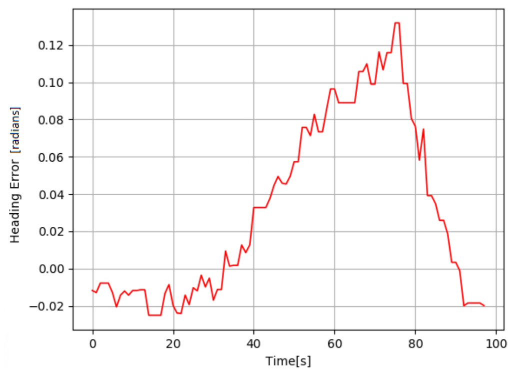

Figure A.2: Heading error

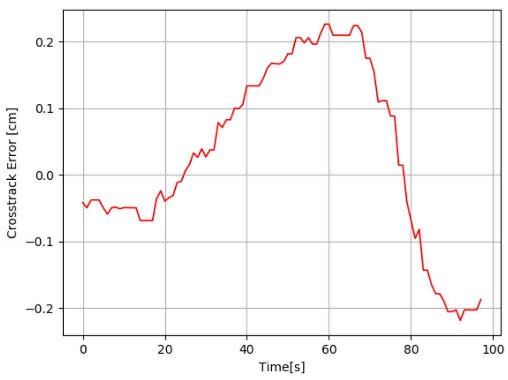

Figure A.3: cross-track error

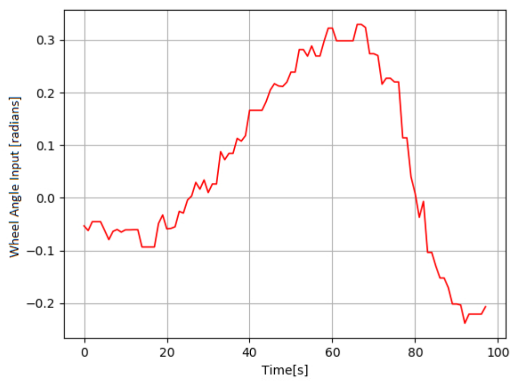

Figure A.4: Road wheel angle input

A.2 Maneuvers starting from front of the trailer

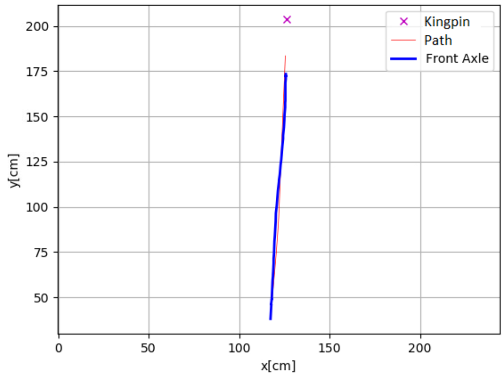

Figure A.5: Front axle trajectory with respect to path

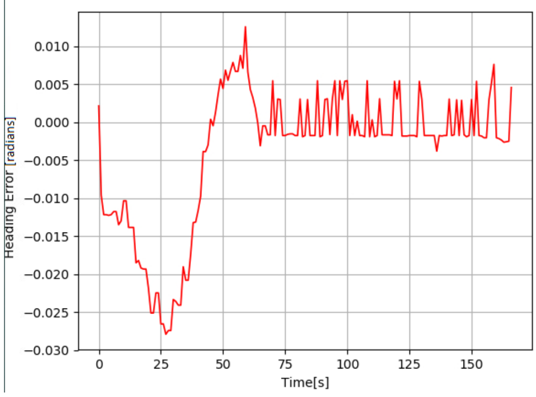

Figure A.6: Heading error

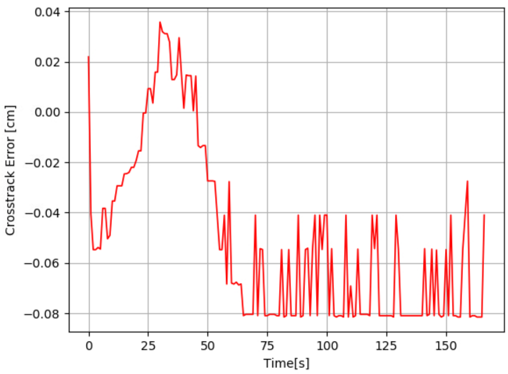

Figure A.7: cross-track error

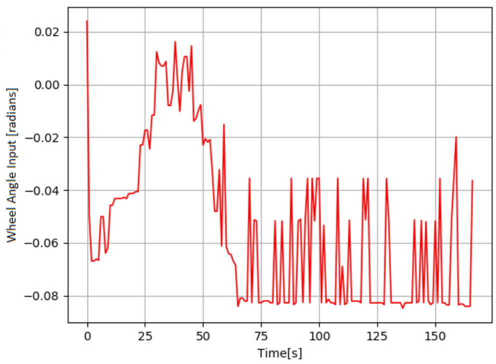

Figure A.8: Road wheel angle input

DEPARTMENT OF SOME SUBJECT OR TECHNOLOGY
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden
www.chalmers.se

CHALMERS
UNIVERSITY OF TECHNOLOGY