
Practical Pallet Engagement with an Autonomous Forklift

Lachlan Barnes

A thesis submitted in fulfilment of the requirements for the degree of Master of
Engineering in Computer Systems, the University of Auckland, April 2022.

Abstract

This thesis outlines the design, implementation and evaluation of an autonomous pallet engagement system. The primary difference between the proposed work and others is a focus on pallet engagement while in warehouse racking. Despite this operation being fundamental for many warehousing workflows, previous studies are confined to pallet engagement in open-space. Autonomous pallet engagement within racking is a difficult problem, with tight tolerances that require precise localization, motion planning and control.

Four major steps were undertaken to create a proof-of-concept autonomous pallet engagement system. First, software was designed and implemented using the robotic operating system (ROS). This software was then tested, optimized and iterated in a simulation. Next, the developed system was integrated with a forklift which had been retrofitted for autonomous operation. Finally, test infrastructure was built to help quantify the system's performance. The Bayesian optimization algorithm was used to tune particular parameters relating to engagement in simulation. This algorithm excels at optimizing complex, black-box functions. As the optimization algorithm demands a large amount of repeated simulation, it was executed using the high-performance computing clusters provided by the New Zealand eScience Infrastructure (NeSI).

To rigorously evaluate the developed proof-of-concept system, the industry-standard ASTM F3499 ("Standard Test Method for Confirming the Docking Performance of A-UGVs") was used for evaluation. This standard required 29 successive consecutive trials in order to confirm docking repeatability within 90 % confidence, based on a 95 % confidence in the test infrastructure. The proof-of-concept design met this requirement in simulation and real-world testing using perfect knowledge of the pallet pose and starting from a fixed location. To further evaluate the system in simulation, it was then tested using a virtual PDS (which would report the pose of the pallet when certain conditions were met). Using this virtual PDS, and when started at a random location relative to the pallet's pose, the system was still able to complete 29 successive consecutive trials.

Evaluation of the developed proof-of-concept showed that ROS was suitable to use for an automated pallet engagement system and that simulation could be used to optimize and tune parameters relating to engagement. A relevant industry standard was considered and confirmed the reliability of the system. Furthermore, the proof-of-concept demonstrates successful pallet engagement within an aisle of warehouse racking, which had not been previously demonstrated in academic research.

Acknowledgements

I would like to express appreciation to my supervisor Professor Bruce MacDonald, who supported and guided me through this project. I would also like to extend my thanks to all those at Crown who enabled this project to happen and who freely offered support and technical advice. I would like to thank Sian Phillips in particular, who was my mentor during the project and a constant source of invaluable advice.

I wish to acknowledge the use of New Zealand eScience Infrastructure (NeSI) high performance computing facilities, and consulting support services as part of this research. New Zealand's national facilities are provided by NeSI and funded jointly by NeSI's collaborator institutions and through the Ministry of Business, Innovation & Employment's Research Infrastructure programme. URL <https://www.nesi.org.nz>.

Contents

List of Figures	vi
List of Tables	viii
1 Introduction	1
1.1 Context	2
1.1.1 Motivation	2
1.1.2 Objective	3
1.1.3 Structure	3
2 Literature Review	4
2.1 Warehouse Operations	5
2.2 Autonomous Guided Vehicles	5
2.3 Motion Planning	6
2.4 Automated Pallet Engagement	7
2.5 Test Methodology and Infrastructure	9
2.6 ROS	9
2.7 Pallet Detection	10
2.7.1 Vision-based systems	10
2.7.2 Rangefinder-based systems	11
2.7.3 Industry Application	13
2.8 Summary	13
3 Requirements	15
3.1 Engagement Task Requirements	16
3.2 Limitation 1: APES with Localization	18
3.3 Limitation 2: Simulation	19
3.4 Limitation 3: Lack of Performance Results	20
3.5 Consolidated Requirements	21

4 Design	22
4.1 Analysis	23
4.1.1 Requirement 1: Simulation	23
4.1.2 Requirement 2: Obstacle Avoidance	24
4.1.3 Requirement 3: Engagement Performance	24
4.2 Design	27
4.2.1 Hardware Overview	27
4.2.2 Additional Hardware	29
4.2.3 ROS System	30
4.2.4 Simulation	38
4.2.5 Test Infrastructure	39
4.3 Implementation	40
4.3.1 PID Controller Tuning	41
4.3.2 Path Planning	42
4.3.3 Path Following	43
4.3.4 Simulation	45
4.3.5 Test Infrastructure	48
5 Evaluation	53
5.1 Experimental Design	54
5.1.1 Simulation Testing	54
5.1.2 Real-world Testing	58
5.2 Results	60
5.3 Vive Validation	60
5.4 Simulation Testing	61
5.4.1 Real-world Testing	63
5.4.2 Simulation Versus Real-World Results	65
5.5 Summary	68
6 Conclusions	69
6.1 Conclusions	70
6.2 Existing Work	71

6.3 Future Work	72
References	73

List of Figures

1	General depiction of warehouse workflow. Order picking is generally the most labor intensive task. Diagram recreated with permission (Copyright © 1998–2017) from John J. Bartholdi, III and Steven T. Hackman [5].	5
2	Simple ROS graph, reproduced from [27], CC BY 3.0.	10
3	Coordinate system of the pallet used for engagement and the forklift. The difference between these coordinate systems as measured from the pallet, is the engagement error. This engagement error is denoted as X_e , Y_e , Z_e , and κ_e	23
4	Components of a Crown Narrow-Aisle Reach Truck. The model shown is the Crown RM6000, which was used for this project after being retrofitted for automation.	29
5	Hardware layout interfacing the ROS system with the RM6000 forklift.	31
6	Overview of the ROS software.	32
7	Actionlib client-server interaction. Diagram based on the likeness of [74], CC BY 3.0	33
8	High level system overview describing how Google Cartographer performs SLAM. Reproduced with permission from [75], © 2016 IEEE.	33
9	High level system overview of ROS Navigation. Reproduced with permission http://wiki.ros.org/move_base , CC BY 3.0	34
10	Overview of PID controllers.	35
11	Top level state machine for pallet engagement. The figure is a screenshot of the graphical interface provided by the Flexbe software [86].	36
12	Pallet Pose Tracker Interface.	38
13	Interface between APES, the Gazebo simulation and reality.	39
14	Visualization from Rviz [90] and Gazebo [87] during a simulated pallet engagement.	39
15	PID design process from “Modern Control Systems, Global Edition, 13th Edition”, reproduced with permission from [99], © 2017)	41
16	Illustration of the impact of altering the spline parameters on the resulting curve interpolation. Reproduced with permission from [102], © 2007 IEEE	42
17	Path planning algorithm.	44
18	Containerization workflow, between local and remote compilation. The optional link to the “Build image” process within the Sylabs Cloud container represents the option of remotely building the image.	47

19	Overview of the procedure for calculate the transform between the HTC Vive coordinate system, and the grid coordinate system.	48
20	Vive mounting and calibration hardware.	49
21	Overview of the procedure for calibrating the proposed test infrastructure.	50
22	Overview of the procedure used to tune selected parameters of the APES.	52
23	Pallet engagement process.	56
24	The PDS is shown in orange.	58
25	Illustration of different error values reported from the Vive.	59
26	Real-world experimental setup.	60
27	Impact on altering the starting location of the APES on the path following characteristics.	66
28	Difference in engagement error reported by the Vive and Gazebo.	67

List of Tables

1	Comparison of common geometric interpolation for path planning. The table is based on a similar table presented by Molter and Fottner [20] © 2011 IEEE.	8
2	Requirements for PID controllers.	17
3	Comparison of different sensing technologies available for SLAM, based on work by [69].	26
4	Vive bias and performance requirements.	27
5	Vive bias and performance requirement creation based on the engagement task performance margins.	27
6	Comparison of systems which could be used for the test infrastructure.	28
7	RM6000 Specifications. ^a Load capacity is subject to derating at height. ^b Average lower speed is 1.3 m/s. ^c Due to a different fork-type being used on the forklift, and wires being run to the fork-carriage, hoist was limited to 264 mm to 2520 mm.	30
8	Characteristics of Pepperl+Fuchs R2000 laser scanner.	31
9	Description of all the base states used to create the state machine and a list of all the implemented states. Some of the state instances are not shown in the overview diagram as they were nested within concurrency containers.	37
10	Comparison of state-of-the-art HTC Vive evaluation studies, based on the work by Ikbal et al., [98]. RMSE refers to the root mean squared error.	40
11	Deterministic evaluation of the developed APES in simulation.	54
12	Vive evaluation results. ^a There was no set measurement error requirement for yaw using ASTM F3499.	60
13	Engagement performance summary for engagement test one in simulation. The X_e , and Z_e are the adjusted errors.	61
14	Engagement performance summary for engagement experiment two in simulation. ^a All the unsuccessful trials were a result of contact with a pallet.	62
15	Statistical analysis of tuned APES parameters versus heuristically tuned parameters. Statistically significant results (if the HDI > 95%) are highlighted. CT error refers to the cross track error and HD error refers to the heading error. ^a These were not the arithmetic means of the data (as reported previously), but the pooled empirical mean calculated during the Bayesian t-test.	63
16	Comparison of the parameters selected manually (i.e., “guess-and-check”) and through the Bayesian optimization method.	63

17	Comparison of PID performance set out in the requirements versus, the achieved PID performance.	64
18	Engagement performance of real-world testing from a constant starting pose. ^a All the trials that exceeded the prescribed TPM's were a result of a κ_e of greater magnitude than 7.5°.	64
19	Engagement performance of real-world testing from a moving starting location. ^a The single failure was caused by contact with the pallet. ^b One of the trial that exceeded the TPM's exceeded the prescribed κ_e , and the other exceeded both the κ_e , and Z_e	65

Acronyms

- AGV** Autonomous Guided Vehicle. 2, 5
- APES** Autonomous Pallet Engagement System. vi–viii, 3, 16–21, 23–27, 34, 36, 38, 39, 43, 45–49, 51, 52, 54–57, 59–66, 68
- CS** Coordinate System. 8, 26, 37, 55, 57
- FF** Forks First. 29, 45, 54, 56, 61–66, 68
- FOV** Field of View. 31, 38, 57
- NeSI** New Zealand eScience Infrastructure. 46, 47, 57, 72
- PDS** Pallet Detection System. vii, 19, 25, 36, 38, 49, 51, 57, 58, 62, 63, 68
- PID** Proportional Integral Derivative. viii, 17, 18, 34, 35, 41, 42, 63
- PRM** Probabilistic Road Map. 6, 7
- PUF** Power Unit First. 29, 54, 56, 61–66, 68
- RMSE** Root Mean Squared Error. viii, 40, 60
- ROS** Robotic Operating System. vi, 5, 30, 31, 33, 34, 36–38, 43, 47, 51, 54–57
- RRT** Rapidly-exploring Random Tree. 6
- SLAM** Simultaneous Localization and Mapping. vi, viii, 26, 31, 33
- SSH** Secure Shell Protocol. 47
- TOF** Time of Flight. 12, 57
- TPM** Task Performance Margin. ix, 17, 61, 62, 64, 65, 68
- VPDS** Virtual Pallet Detection System. 37, 57, 61
- WMS** Warehouse Management System. 2, 18, 31

1

Introduction

1.1 Context

The overall objective of the project¹ was to investigate methods of improving pallet engagement with an autonomous forklift. After identifying state of the art methods, an area selected for focus was pallet engagement within racking. This was because it is a common warehousing workflow, with limited literature addressing proposing possible autonomous solutions. The scope of the project involved the design, simulation, and implementation of a system capable of autonomous pallet engagement, as well as a system capable of evaluating the performance of the developed pallet engagement system. The forklift used during the project was a Crown RM6000 Narrow Aisle Reach Truck retrofitted for automation. Testing of the developed system was done at a Crown research and development facility in Auckland, New Zealand.

1.1.1 Motivation

Autonomously guided vehicles (AGV's) have existed since the the 1950's [1], with the initial iterations' having tracking wires mounted in the ground or ceiling. Although such designs could provide a reliable method of navigation, they were inflexible. Newer designs of autonomous forklifts use a variety of sensing technologies including computer vision and laser range-finders (2- and 3-D). The current market size of autonomous forklifts is estimated to be between \$3.52 Bn USD (2020) [2] and \$2.00 Bn USD (2021) [3] and is predicted to increase to \$5.69 Bn USD (2027) [2] and \$4.11 Bn USD (2029) [3]. A large driver of this market size is a labour shortage [2]. It has been previously predicted that between 2018-2028 that 52.7 % of manufacturing jobs will remain vacant [4]. The reasons cited for this deficit are a "shifting skill set due to the introduction of new advanced technology and automation," followed by "negative perception of students/their parents toward the manufacturing industry" [4]. Autonomous forklifts provide several advantages over conventional driver operated forklifts [2], [3]:

- Greater efficiency
- Less product damage
- Steady working rate
- Increased safety
- Decreased labour cost

This project's motivation is to contribute to the development of autonomous forklifts by proposing a solution to engagement within racking (a common warehousing workflow). By designing a solution that is capable of engaging with pallets without perfect knowledge of the pallets pose (i.e., where the actual pose of the pallet may differ from an expected pose), it is hypothesised that the developed system may additionally be used alongside driver operated forklifts. The developed system will also use localization to enable its use with a warehouse management system (WMS).

¹This project was as part of a Callaghan Innovation Fellowship Grant between Crown Equipment Corporation (Crown), and the University of Auckland.

1.1.2 Objective

The overall project objective is to create a proof-of-concept design of an autonomous pallet engagement system (APES), which is capable of working in a practical environment. Particular emphasis is placed on the ability to navigate and engage with pallets while in racking. The project will strive to achieve four primary goals: “design”, “simulate”, “integrate”, and, “test”.

1. *Design*: Iteratively design an autonomous pallet engagement system in software.
2. *Simulate*: Use a simulated environment to debug, optimize, and test the designed autonomous pallet engagement system.
3. *Integrate*: Integrate the developed pallet engagement system with the real-world functionality of an autonomous lift truck.
4. *Test*: Through repeated testing, demonstrate the degree of reliability, precision and accuracy of the designed pallet engagement system.

1.1.3 Structure

The material is organized in the following structure.

- **Chapter 1 - Introduction:** This first section dissects the motivations and objectives of the project.
- **Chapter 2 - Literature Review:** An examination of the state of the art in automated pallet engagement. Selected literature is examined from the fields of robotics simulation, robotic control, and the current pallet engagement solutions.
- **Chapter 3 - Requirements:** A set of limitations are established from the current literature in automated pallet engagement. These limitations are used to generate a set of requirements for the project.
- **Chapter 4 - Design:** The requirements in the previous section are used to specify how to implement the solution. Additional detail is provided about implementation of the proposed design especially where the specifics are not obvious.
- **Chapter 5 - Evaluation:** The evaluation methods and experimental design are described and the results presented.
- **Chapter 6 - Conclusions:** This section summarizes the project’s conclusions and suggests future avenues to extend the research.

2

Literature Review

The literature review will begin by giving background on warehouse operations and AGV's, before covering the state-of-the art in autonomous pallet engagement, and testing methodologies. An overview of the operating principals of the robotic operating system (ROS) will be provided, followed by a summary of the different options for pallet detection.

2.1 Warehouse Operations

Understanding warehouse operations is pivotal in identifying bottlenecks and areas that can be improved through automation. Most warehousing workflows can be divided into four distinct categories; receiving, put-away, order-picking, and packing and shipping (see Figure 1[5]). Receiving is when goods arrive, are unloaded and placed in a staging area in anticipation of being put away. Often received goods arrive in larger units, such as pallets. Overall, approximately 10 % of typical distribution centres' costs will be consumed by this stage. Next, pallets are put-away in a storage area; a common type of storage is aisles of racking. When products are put away, they often have to be recorded so that the goods' position is digitized to guide order-pickers. This stage accounts for approximately 15 % of the typical warehouse operating expense [6]. A large contributor to this cost is the distance which has to be travelled by goods being stored. After a customer's order is received, several operations occur: the order must be verified (e.g., to ensure sufficient stock), a pick list must be created, accompanying documentation created (e.g., shipping documentation), and the order-picking scheduled. Generally, these operations are coordinated by a WMS. WMSs can leverage information from barcoding and radiofrequency tags to offer opportunities to improve warehouse operations. These opportunities include: real-time control of the warehouse operation, easy communication with other parts of the supply chain, and higher levels of automation [7]. Normally 55 % of the operational costs are consumed by this stage (including order picking) [6]. Packing and shipping form the final stage of the normal warehouse workflow and typically involve the consolidation of a customer's order and packing it into large units for shipping (e.g., cases, pallets).

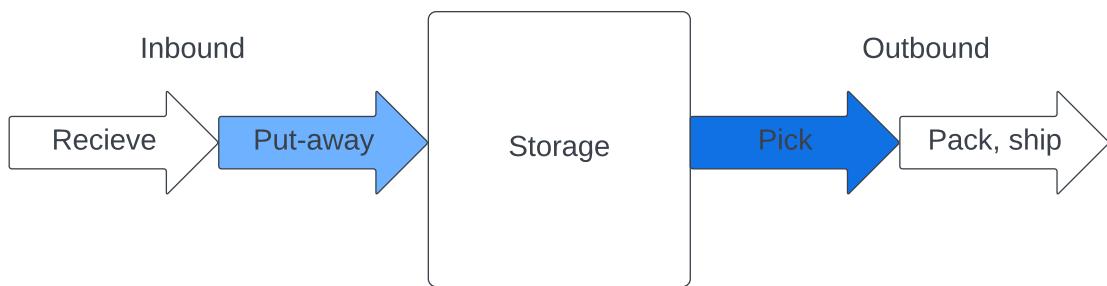


Figure 1: General depiction of warehouse workflow. Order picking is generally the most labor intensive task. Diagram recreated with permission (Copyright © 1998–2017) from John J. Bartholdi, III and Steven T. Hackman [5].

2.2 Autonomous Guided Vehicles

As mentioned in Bostelman and Hong's 2016 review, [8], autonomous guided vehicles (AGVs) have existed since the 1950s, and since inception, there has been a focus on docking maneuvers as they are needed for a significant amount of warehousing workflows. Bostelman and Hong define docking as the arrival and stopping at a position re-

lative to another object. Docking can include positioning the vehicle or the equipment onboard the vehicle relative to another object, for example, a tray station, trailer, or pallet [8]. Early AGVs used electric wires buried underground or painted lines for navigation. This type of vehicle navigation was a low-complexity solution. However, any changes in the vehicle's paths (e.g., due to warehouse layout) would be costly due to the need to make physical changes to the warehouse. Newer designs have built upon limitations with a common approach of using 2D laser range finders for localization. Sometimes this localization process is assisted with reflective beacons, which serve as optical landmarks. Otherwise, the natural features of the environments can be used for localization.

2.3 Motion Planning

The following section covers the different motion planning algorithms available. Path planning is the problem of finding a path from a starting point to an ending location within a configuration space (see below) subject to constraints (e.g., the robot's kinematic and dynamic properties, obstacles).

First introduced by Lozano-Perez in the context of obstacle avoidance, the configuration space refers to the set of all possible states or configurations of a robot [9]. The free configuration space is a subset of this space, denoting all configurations which do not result in a collision. Three main types of path planning methodology will be discussed: variational, graph-search and sampling-based methods. Variational methods optimize a trajectory function using a cost function that considers kinematics constraints and nearby obstacles. The merit of variational methods is their fast convergence to a locally optimal solution. However, these methods cannot find a globally optimal solution inherently, and the quality of their solutions rely heavily on the initial guess. Graph-search methods (also referred to as lattice methods) first discretize the free configuration space – representing it as a graph – before searching for the lowest-cost path. Sampling-based methods input steering commands to explore the workspace and generate a graph of reachable points (also known as a reachability graph). Once the end goal is within the graph, a suitable path can be traced using the graph.

A popular gradient-based variation method was developed by Zucker et al. and can be used to optimize trajectories (e.g., produce short trajectories with small acceleration and jerk) even when initialized with unfeasible ones [10]. It is a popular choice for joint trajectory optimization, where the smoothness of motion is essential, however, it has not seen much use in motion planning. This may be due to the computational and implementation complexity of such methods, as well as their inability to find the globally optimal solution inherently. Graph search methods do not have this inherent limitation, however, they are limited to optimizing across a discretized version of the free configuration space. Therefore, they may fail to find a feasible solution or the most optimal solution. The configuration space can be discretized by sampling it while checking for collisions, to generate a graph of allowable paths – also known as a roadmap. A popular implementation is Probabilistic Roadmaps (PRM) [11] which uses random sampling to generate roadmaps. As PRM uses straight lines to join samples, this may result in paths that are kinematically unfeasible or suboptimal for particular vehicles. After constructing the roadmap, a graph search such as Dijkstra's, or A*, can be used to find the shortest path.

Rapidly-exploring Random Trees (RRT) [12] is a sampling-based method that can find a

feasible solution in high-dimensional configuration space quickly and is probabilistically complete. Probabilistic completeness is an important metric when evaluating sampling-based metrics and refers to a motion planner's ability to converge on the optimal solution with increasing computational time [13]. Asymptotic optimality denotes a method's ability to converge to the lowest-cost path with increasing computational time [13]. There have been two major improvements to the original PRM algorithm, RRT* and RRTX. The former achieves asymptotic optimality [14], whereas the latter is designed for dynamic environments [15]. In both methods, the control inputs are randomly sampled while checking for collisions to construct a reachability graph originating from the starting location and extending outwards.

2.4 Automated Pallet Engagement

The previous section described general techniques for path planning and following. This section will cover the navigation approaches currently implemented in autonomous pallet engagement systems in academic research.

The state-of-the-art navigation current used in autonomous pallet engagement can generally be split into two categories: approaches that use control algorithms to navigate, and approaches with a separate path planning the following stage. Karaman and colleagues [16] proposed an algorithm to detect the pose of pallets from 3D laser range-finder scans and to navigate a forklift around open space in loading and unloading a truck bed. They propose a closed-loop control strategy to drive the forklift when engaging with the pallet. If δ is the steering command to the vehicle, and e_y and e_θ are the lateral and angular errors to the goal point, the control strategy can be defined as:

$$\delta = K_y \tan^{-1}(e_y) + K_\theta e_\theta$$

The problem with this closed-loop control algorithm is that it produces a change in steering angle linearly proportionally to the lateral and angular error to the goal point. Although it may be possible to tune the gains (K_y, K_θ) so that they are appropriate for a particular vehicle and engagement workflow. The gains would likely have to be re-tuned if a vehicle with different dynamic/kinematic properties was used, or a different type of engagement workflow was needed, otherwise the control algorithm may produce an overly aggressive, or passive response.

The approach by Pages et al. [17] uses a simple strategy for path planning. The path to the pallet is broken into two segments: a straight path that is parallel to the middle of the pallet, and a straight path from the vehicle to the intermediate point. In this approach, the steering wheel angle is set based on the planned trajectory. This approach is simple, however, the authors have overlooked some key considerations. When the vehicle reaches the intermediate point, there is an expectation that the wheel angle is able to change instantly in order for the trajectory to be correctly followed. Either the approach calls for the vehicles to stop at the intermediate point, or this is a point where path following error is introduced. The approach also has no closed-loop feedback. This is problematic because, for example, if the commanded steering angle was causing the vehicle to not meet the desired trajectory, no additional action would be taken. The authors do not specify any control strategy for adjusting the vehicle's speed (e.g., reducing its speed as it approaches the endpoint), so it's likely a constant speed was used. The strategy of forming the navigation task as an optimization task (variational path planning) was used by Balivo et al. [18]. Their control strategy can be defined as the minimization of the weighted sum of two

cost functions with the intention of forming the shortest and fastest path while conforming to the dynamic behaviour of the AGV.

Seelinger and colleagues [19] use both a fifth-order polynomial planner and a third-order polynomial planner when engaging with the pallets. The fifth-order polynomial was utilized when approaching, and the third-order was used when closer (to avoid sharp steering changes). The steering angle is calculated directly based on the planned trajectory such that there is an inherent assumption that the vehicle is exactly tracking the planned path. Similar to Pages et al., [17] no information is given about the speed used for the drive wheel. To help address this assumption, the current error from the planned trajectory (specifically the lateral and angular error from the vehicles CS) is calculated. If either of the following conditions is met the path is replanned, the total distance travelled since replanned exceeds 20 inch, the Euclidean distance of the X, and Y error from the path exceeded 3 inch, or, the error in orientation from the path exceeds 3°.

A common approach when creating paths to engage with pallets is to use geometric interpolation to construct paths [19]–[22]. One advantage of this is greater control over the kinematic properties of the path created. A path that has a continuous arc but not a continuous curvature can be referred to as having G¹ continuity. For a non-holonomic robot (where the controllable degrees of freedom is less than the total degrees of freedom), such as the forklift used in the project, a G¹ path means the forklift must stop at non-continuous curvature points to change its steering angle, otherwise, it will deviate from the path. Therefore, a minimum requirement for non-holonomic robots is generally G² paths, which means all curves will have a continuous change in curvature. A comparison of common geometric methods are shown in Table 1 .

	Continuity	Steering limit	Flexibility
Euler spiral	max. G ²	Yes	Low
Femat's spiral	max. G ²	Yes	Low
Bézier curve	Depends on number of control points.	No	High
$\eta^n \rightarrow G^n$		No	High

Table 1: Comparison of common geometric interpolation for path planning. The table is based on a similar table presented by Molter and Fottner [20] © 2011 IEEE.

Another advantage of using geometric models is they allow path planning without discretization of the robot's configuration space. Discretization occurs as it allows for the rapid searching of the robot's configuration space for possible routes, commonly by repeating a shape such as an arc. However, the disadvantage of discretization is that it may introduce a form of error as the goal point may not exist exactly on the discretized map. The major disadvantage of geometric models for path planning is they do not inherently allow for obstacle avoidance. Hentschel et al., [22], offers an interesting method of creating a path for pallet engagement from waypoints by concatenating lines and polar spline segments. This results in a path with constant curvature. Their method allows for limited obstacle avoidance by varying the lateral deviation of the path from the way-points. This method is based on work by Thrun et al., [23].

In contrast to previous decentralized approaches, Behrje et al, [21], propose an autonomous pallet engagement system that uses localization (using adaptive Monte Carlo localization with data from a ToF camera). When the APES was provided with the pallets pose

and started at different positions it was able to pickup the pallet four times successfully (the authors do not give an indication of the system's reliability). The authors state that after the second trial the pallet is slightly offset and rotated (i.e., in comparison to the pose provided to the APES), however, they do not state by how much. They then test starting at the same pose but with a different starting orientation and their system is able to pickup the pallet three out of four times. With the exception of Behrje et al. [21] the other methods proposed use decentralized navigation, i.e., they use the relative pose provided by a pallet detection sensor (PDS) for navigation. However, this approach is likely impractical for normal warehousing use. Most warehousing workflows with an APES would use a WMS to command the APES to a particular location, e.g., to either to put away or retrieve a pallet.

2.5 Test Methodology and Infrastructure

When considering the evaluation procedure for the APES, two primary areas of literature were examined: relevant industry standards and existing methodology in academia. The first standard considered was ASTM (formerly the American Society for Testing and Materials) F3499 [24], labelled "Standard Test Method for Confirming the Docking Performance of A-UGVs". The standard proposed a procedure for confirming a vehicle's ability to performing a docking maneuver within pre-set reliability margins. There was another standard related to robotic navigation which was produced by ISO (International Organization for Standardization), 18646-2 called "Robotics - Performance criteria and related test methods for service robots Part 2: Navigation" [25]. However, this standard focused more on obstacle avoidance rather than precise docking.

Several authors who have developed autonomous pallet engagement systems attempt to quantify the performance of their system through various metrics. Seelinger et al., [19], give the success rate of engagement which is 98/100 when detecting pallets using fiducial markers and 11/13 when using the pallet's natural features. Similarly, Karaman et al., [16] provides the engagement success rate as 35/38 (when detecting pallets using 3D laser ranger finders). In addition, the authors provide the final lateral alignment error and angular error (i.e., error along the Y axis, and κ error, see Figure 3) based on the forklift's estimated position (based on a Dublin's vehicle model [26], and the desired path).

2.6 ROS

ROS (robotic operating system) is a framework for robotic software development which provides structured communication between machines as well as hardware abstraction, low-level device control and common algorithms such as path planning and localization. Although ROS is not a real-time system (i.e., systems designed to work to specific time constraints), the newest version of ROS (ROS2), is real-time compatible.

According to the original authors of ROS [27], it was developed to be:

- Peer-to-peer
- Multi-lingual
- Tool-based
- Thin

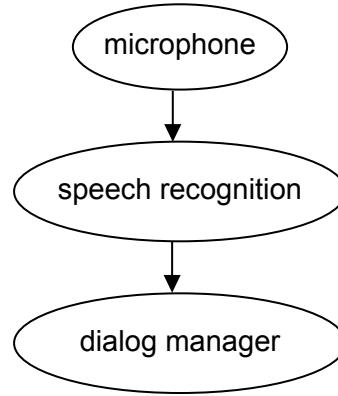


Figure 2: Simple ROS graph, reproduced from [27], CC BY 3.0.

- Free and Open-Source

The underlying concepts of ROS are nodes, messages, topics and services. Nodes are processing units which perform computation. Nodes interact with one another through messages, which are strictly typed data structures based off data primitives (integer, Boolean etc.) which can be nested arbitrarily deep. Nodes communicate by publishing messages on topics, and subscribing to topics. A simple pipeline, where data is directly passed from one node to another is shown in Figure 2. The publish subscribe model is useful for asynchronous communication, however, for synchronous communication services are used. A node is able to advertise a service with a particular name, which allows other nodes to send a request and receive a response from the node advertising the service.

ROS offers several attractive features for robotic development. Its modular structure helps improve debugging, especially as the decentralized topography allows for specific sub modules to be tested in isolation. In addition, ROS offers tools for data recordings and playback, developed subsystems, real-time monitoring and integration with a simulated environment. ROS has some limitations; it does not offer real-time operation, requires network connectivity and is computationally intensive.

2.7 Pallet Detection

Pallet localisation (i.e., finding relative pose of the pallet), typically involves three steps: capturing the scene, finding features indicative of a pallet and using the detected features to find the relative pose of the pallet. Most sensing systems for pallet localization fall into two broad categories: a) those which use vision systems and calculate distances using extrinsic camera calibration, and b) those which use rangefinder-based sensors (e.g., laser range finders and time-of-flight cameras) where distances are sensed.

2.7.1 Vision-based systems

Using vision systems to localize pallets has been well established. Garibotto et al. introduced the idea in 1996 by using a single calibrated camera to identify a pallet's central stringers [28]. Based on the pallet's relative position and orientation, their system would actuate the tines of a retro-fitted forklift once it was a short distance from a pallet. Pages

et al. showed the possible accuracy of using a single calibrated camera and segmenting the image based on colour and geometric characteristics [17]. However, the method used by Pages et al. made several unrealistic assumptions. First, for their method to work they had to adapt their testing environment such that its walls were white and the floor green. Second, to extrinsically calibrate their camera, they used the Coplanar Tsai method [29] which is used to determine the camera's external (pose relative to world coordinate system) and internal parameters (principal point, focal length and distortion coefficients). These parameters can be used to calculate the pose of objects from images taken from the calibrated camera. However, this calibration method imposes a constraint that the pallets could only be accurately localized if placed on the ground. This is because the length of the pallet contacting the ground is used to estimate the relative pose of the pallet. Cui et al, [30] and Byun et al [31], both utilize the Coplanar Tsai [29] calibration method and use various dimensions of the pallet to calibrate the camera used for sensing. This method of estimating the pallets pose (i.e., using Coplanar Tsai calibration [29]), limits the general applicability of the proposed of pallet sensing as it generally uses one or more known length of the pallet, which may differ based on the type of pallet used.

One method of simplifying the pallet detection problem is by using fiducial markers. Seelinger et al. used this approach with two calibrated cameras and three fiducial markers attached to the pallet [32]. However, adding fiducial markers to pallets is impractical as it increases the cost – both initial and ongoing – of such an implementation solution. Mohammad et al. proposed a solution using fiducial markers and therefore has the same inherent drawbacks [33]. Seelinger et al. developed a technique to localize pallets by detecting their edges and fitting a model of the pallet to determine their location (i.e., using “natural features”). However, their natural features identification method lacked robustness as it required a modified test environment for it to work.

Nygards et al. proposed using the structured light technique where a visible laser and a camera would localize a pallet [34]. However, this method is susceptible to the drawbacks of both sensors. First, the method has an inbuilt assumption that the pallet's vertical location is known (i.e., for the laser line to intersect with it). Second, the performance of the video camera is highly dependent on illumination and lighting. Wang et al. used a similar method but with a field-programmable gate array (FPGA) and a digital signal processor (DSP) [35]. Characteristic of the structured light technique, there was a significant increase in localization error with distance [34], [35] for both techniques.

A more complex methodology was proposed by Cucchiara et al., where a hierarchy of features were extracted from an image and used for classification by a decision tree [36]. Despite the complexity of their method, the provided results have a high false detection rate of between 4-14%. As the consequences for a falsely detected pallet may be severe, this performance would be wholly unsuitable for practical implementation.

2.7.2 Rangefinder-based systems

Rangefinder based solutions (i.e., laser scanners, LIDAR, and time-of-flight cameras) have also been proposed for pallet detection. Leckling et al. were the first to attempt pallet localization with LIDAR [37]. They were capable of localizing pallets by placing two reflectors on the corners of the pallets and matching scans against a model of the pallet.

However, like the fiducial marker based methods, it is undesirable as it places a requirement for custom made pallets. A similar model-fitting approach which used both a laser scanner and a camera, was introduced by Baglivo et al. [38]. Building on these methods, He et al. proposed a method that would detect a pallet's corners by post-processing laser scans to find continuous segments (i.e. adjacent points with similar values within a threshold) [39]. Like that proposed by Leckling et al. and Baglivo et al., this method is intrinsically dependent on the pallet model (i.e., its dimensions) and requires stable feature extraction for accurate pallet localization.

Mohamed et al. took a novel approach, using a Faster R-CNN (i.e. a machine learning algorithm for object detection) to localize pallets [40]. To reduce the likelihood of false positives, they tracked potential pallets using a Kalman filter and only confirmed pallets once they had exceeded a given confidence. As acknowledged in the authors' paper, this method failed to present a solution for the detection of multiple pallets. Another novel approach was by Baglivo et al. who used a combination of a laser scanner and camera for pallet localization [18]. However, the inherent complexity of two sensors for pallet detection makes it less desirable for practical applications. Laser-based range finding was used alongside a deterministic linear programming for pallet and truck-bed localization [16]. A user could direct the forklift to pick up or drop off a pallet using a tablet and circling the region on the tablet. However, the main limitations of this approach are the high cost of the three laser rangefinders used for pallet and truck-bed localization and the loss of pallet detection accuracy at specific poses.

Time-of-flight cameras (ToF) work by emitting a modulated light source and measuring the phase-shift returned light to infer a scene's distance measurements. They have recently had success in automotive safety systems [41], gesture recognition [42] and 3-D mapping [43]. When ToF cameras were compared to stereo cameras, ToF cameras were found to be more accurate – albeit with lower image resolution [44]. The first use of a ToF camera to detect pallets was by Kleinert et al., who integrated the ToF cameras into a forklifts' tines and developed a method to reduce noise in ToF images [45]. They subsequently developed a driver assistance feature that would indicate what steering and throttle inputs a driver should use to engage with a pallet [46]. Xiao et al. showed the capability of ToF cameras when used for pallet localization [47]. Their methodology was threefold. Firstly, ToF range images were segmented into planes. Next the planes were filtered based on heuristic rules. Finally, templates of pallets were slid across the planes, and the similarity computed to determine pallet matches. They used the well-researched and widely available ToF sensor (i.e. a Microsoft Kinect 2.0 camera) for pallet localization and ground-placed fiducial markers for navigation prior to pallet engagement. Using ground placed fiducials in a commercial implementation would be difficult due to the installation time and cost and the maintenance associated with replacing worn markers. Molter et al. also used a Microsoft Kinect 2.0 to provide ToF measurements and segmented the range images into planes [48]. In a similar method to Xiao et al., they used a model-fitting approach to localize the pallet. However, they further processed the laser scans to find the centroids of the wooden blocks.

Haanpaa et al. proposed the use of two commercial ToF cameras IFM O3D303 and 3D201, for engaging with custom containers [49], [50]. The authors specify engagement in a confined space (i.e. in an intermodal container) as a system requirement, however, they gave no indication of how or if they fulfilled this requirement.

2.7.3 Industry Application

Similarly there is a mix of vision systems and rangefinder-based approaches for pallet localization and engagement in filed patents. Crown Equipment Corporation (Crown) is the assignee of ten patents that describe identifying the corners of a pallet from a gray-scale image, determining the height of the pallet's central stringer and hole openings, and tracking potential pallets ([51]–[60]). All patents were filed on the same date with Seelinger M. (first author of [32]) as one of the inventors. However, none of the patents mentioned above appears to make use of the method Seelinger M. proposed in [19]. Toyota Industries Corporation (Toyota) also has a patent that details a slightly different strategy where a subset of the image is extracted framing one of the pallet holes at the pallet's front surface. Based on this extracted image, the pallet's relative position and orientation is calculated [61].

However, in recent years, most patents relating to pallet localization and engagement have used ToF cameras. John Bean Technologies Corp (JBT) is the assignee of a patent, which appears to use several similar techniques to the methodology described by Xiao et al. [47], [62]. Interestingly both were published/filed in the same year (i.e. 2017). In both methods, the RANSAC technique for plane extraction is used, and in the case of the JBT patent, a single plane is selected based on a pallet distance cue. This is similar to the heuristic rules used by Xiao et al. to filter planes. Next, a template-sliding algorithm using a pallet model is applied to the plane to determine the pallet's exact location. The JBT patent use cross-correlation to measure the similarity between the pallet template and the binarized plane. Xiao et al., in contrast, use a custom similarity metric.

Boston Dynamics Inc (Boston Dynamics) is the owner of a patent describing a control system that uses a ToF camera to detect features from a pallet and repositions the tines to engage the pallet [63]. Although their proposed design is limited to picking pallets up from the ground, it has some obstacle avoidance by populating occupancy grids from a ToF sensor and adjusting the forklift's motion based on an exclusion zone around the forklift which dynamically resizes when the forklift approaches the pallet. A patent assigned to The Raymond Corporation (Raymond Corporation) describes a sensor suite for pallet engagement comprising: two ToF cameras, two fork-tip sensors, a load weight sensor, and a load seated sensor [64]. The ToF cameras are used for pallet and racking identification. The fork-tip sensors check for objects in the tine's sensor's proximity. The load weight sensor determines when the tines are lowered or raised sufficiently, and the load seated sensor checks for when the tines are sufficiently inserted into the pallet.

2.8 Summary

The process of putting away and retrieving goods during order picking is critical for most warehousing workflows, encompassing 70 % of the typical warehouse operating expense [6], [65]. Therefore, autonomous pallet engagement systems that are capable of storing and retrieving pallets from within aisles of racking is a worthwhile pursuit. However, despite this motivation, no publicly available academic literature, found to date, completed this task; most systems chosen assume that the forklift was in open space (i.e., there are no nearby obstacles), with only Hentschel et al., offering a limited solution for avoiding obstacles [22].

When analysing the normal warehousing workflow, it is apparent that interfacing with a WMS is important for an APES. This is because the WMS would receive the delivery information and customer orders, and could therefore, direct autonomous systems. Despite this, most state-of-the art autonomous pallet engagement systems do not use localization which would enable the WMS to know their position, and therefore direct them within a warehouse.

A limited number of studies provide some performance results, such as the engagement success rate and the final cross track error and angular error. However, to confirm the reliability of an autonomous pallet engagement system, to a level acceptable by relevant industry standards, the performance of the system needs to be more rigorously tested.

The research to date primarily has been focused on a single methodology with minimal iteration. This is likely due to the time and cost associated with comparing multiple approaches. Therefore, it indicates that alternative methods of iterating a design, for example with simulation should be considered.

Several pallet engagement sensors were examined, and although the project's scope does not involve designing a pallet engagement sensor they are a critical component of an autonomous pallet engagement system. Both vision-based and rangefinder-based systems have their respective limitations. Vision-based systems are characterized by low cost however, they are highly dependent on illumination conditions, and their method of extrinsic calibration may impose constraints on their systems. Fiducial markers have been proposed to address the need for extrinsic calibration yet the installation and upkeep of such markers make them unsuitable for practical applications. ToF cameras show promise and are being used in some commercial solutions. However, the publicly available methods which have used with ToF camera, provide little information about the performance of it when used as part of a APES.

The limitations of the state-of-the art can be summarized as:

- Lack of methods that use localization alongside automated pallet engagement.
- Need to simulate an automated pallet engagement system.
- Lack of performance results about pallet engagement performance and reliability.

3

Requirements

First, the requirements of the system will be described and quantified. Next, the limitation of the state of the art literature will be used to create additional requirements for the project. These limitations are the:

- Lack of methods that use localization alongside automated pallet engagement.
- Need to simulate an automated pallet engagement system.
- Lack of performance results about pallet engagement performance and reliability.

These limitations are expanded upon below, and used alongside the objectives and constraints of this project to build a set of requirements for this project, which are described in Section 3.5.

3.1 Engagement Task Requirements

It is crucial that an APES can demonstrate repeatable, controlled actions to be suitable for industrial operation. The standard by ASTM (formerly American Society for Testing and Materials) on the test method for confirming the docking performance of A-UGVs (autonomous - unmanned ground vehicles) [24] was used to derive the requirements of this project. The scope of the test standard included the docking performance of up to three axes and, therefore, was applicable to derive the requirements for the pallet engagement task. The requirements derived from the standard are used to indicate the proposed APES suitability for industrial use but are not a comprehensive description of the standard.

The standard requires a start and endpoint to be described. The APES begins at the start point during a trial, drives to the endpoint (while avoiding obstacles), and engages with a pallet that forms the endpoint. This procedure repeats a specified number of times (a minimum of 29). However, no failure is allowed. The standard specifies a minimum of 29 trials to ensure statistical significance with a 90% probability of success of a trial and 95% confidence in the test apparatus. Failure conditions for a trial are:

- *Task Performance Margins*: The APES should be capable of engaging with the pallet without exceeding the task performance margins (maximum translational and rotational error allowed).
- *Movement beyond the Defined Area*: The APES should be capable of repeated engagements without moving past its defined testing/operational area.
- *Depleted Battery*: If the power source of the A-UGV fails during a trial, this constitutes a failure. However, the power source may be replenished or replaced in between trials.
- *Human Intervention*: During a test, any configuration, alteration, maintenance or repair to the A-UGV is considered a failure. Human intervention is acceptable between repetitions as long as the configuration or test settings are not changed.
- *E-Stop Operation*: If the E-stop is used at any point during a test, this is a failure condition. This is important as E-stop usage could prevent movement outside the defined area, slow the vehicle's approach to the endpoint or modify the vehicle's dynamics.

The task performance margins (TPM) define the maximum translational (X_e , Y_e , Z_r) and rotational (κ_e) error which still constitute a successful engagement (see Figure 3). These margins were calculated based on the geometry of the pallet used for testing. This project exclusively used New Zealand Pallets manufactured by Loscam (Loscam, Auckland, New Zealand). The most restrictive dimension for engagement is the inner height of the pallet pocket, which measures 106 mm, compared to the height of the tine of 45 mm. This therefore, allows for a maximum of ± 30.5 mm of error before collision with the pallet. This calculation was used to create the 30.5 mm task performance requirement along the Z axis. The other task performance margins were calculated likewise using the geometry of the pallet and forklift tines, with the exception of yaws (κ) TPM. This task performance value was calculated directly from the pallet and tines geometry, so a reasonable value of 7.5° was selected.

Because the ASTM standard is generalised to docking and does not focus on engagement in a narrow aisle, additional requirements should be considered. The maximum height, and minimum aisle width a forklift can operate, will control the amount of storage space that can be utilised. Aisle widths are commonly 3.7-4.6 m for counterbalance lift trucks and 2.1-2.7 m for reach and double reach trucks [5]. When APES and human forklift drivers operate within the same space (i.e., interacting with the same stock), a key consideration is that the APES has to be capable with imperfect knowledge of the pallet's pose, as driver-operated vehicles may not completely precisely place pallets in their designated poses.

Engagement with a pallet requires properly tuned controllers to ensure that they are sufficiently responsive while ensuring the resulting overshoot is within the allowed limits. The maximum overshoot for each actuator was calculated based on approximately one third of the TPM for that actuator, in order to avoid exceeding the TPM. For example the TPM for the Z axis was 30.5 mm therefore, the hoist actuator was designed to have no more than 10 mm of overshoot. The PID controller's performance (i.e., the overshoot and settling time) is determined based on a step input, therefore, the range of values comprising this step input are specified in Table 2. This is because the range of the test input will inform the PID performance results. For example a greater range of step input will generally result in a larger settling time irrespective of the controller's performance. The range of the test inputs was chosen based on the minimum and maximum limits for each actuator (see Table 7 for the actuator limits).

	Test input range	Task performance margin	Maximum overshoot	Maximum rise time	Maximum settling time
Hoist	1500 mm	30.5 mm	10 mm	2s	3s
Reach	375 mm	120 mm	30 mm	2s	3s
Sideshift	75 mm	101.5 mm	30 mm	2s	3s
Tilt	3°	n/a	1.0°	2s	3s

Table 2: Requirements for PID controllers.

Requirements

The following requirements have been found by analysing the challenges involved in autonomously engaging with a pallet.

- *Task Performance Margins:* The APES should be capable of engaging with the pallet with the following maximum error margins; X<120 mm, Y<101.5 mm, Z<30.5 mm, k<7.5°.
- *Reliability and Robustness:* The APES must complete 29 repeated engagements based on the failure conditions of standard ASTM F3499 [24].
- *Aisle Width:* The APES must be capable of operating within aisles as narrow as 3.2 m.
- *Goal Variability:* The APES must be capable of following a dynamic goal (i.e., the goal may be updated based on new sensor input, such as from a pallet detection sensor).
- *PID Controllers:* The PID controllers should meet or exceed the performance requirements specified in Table 2.

3.2 Limitation 1: APES with Localization

Creating an automatic pallet engagement system that uses localization data has four key advantages over the current approaches proposed in the literature. First, modern warehouses commonly use racking arranged into aisles to maximise product density. This is because normally it is more economical to expand a warehouse upward rather than outwards, so a common engagement workflow is engaging with pallets in aisles where a tight turn is required before engagement. However, all publicly available academic approaches to pallet engagement have confined their scope to engagement in open-space (i.e., engaging pallets without considering nearby obstacles). An APES which used localization data, may identify obstacles such as adjacent racking, and plan engagement paths accordingly and therefore enable new engagement workflows such as:

- Pallet engagement within racking arranged in aisles.
- Pallet engagement within shipping containers.
- Pallet engagement from conveyor belts.

Another key benefit of using localization with an APES is that it could allow for integration with a WMS. A WMS has information about the current warehouse inventory and orders and can be used to direct autonomous guided vehicles such as an APES. However, for a WMS to coordinate one or more APESs, it must have localization information. This localization information will allow a WMS to know where in the warehouse the vehicle is and therefore, which product the APES should engage with next. However, as space is a commodity in a warehouse, the aisles in which forklifts operate are carefully optimized for: a specific type of forklift and using only the maximum needed storage space. Therefore, the localization system must be highly accurate to enable it to navigate within the narrow aisles. Sensors that could be used for localization include:

- Monocular cameras using fiducial markers.
- Monocular cameras using natural features.
- 2D and 3D laser range finders using natural features.

- Time-of-flight cameras using natural features.

Requirements

By analysing limitation 1, the following requirements can be defined.

- *Plan*: Paths must be planned to avoid obstacles, whilst considering the dimensions of the forklift.
- *Predict*: Approaching collisions with obstacles must be predicted, and appropriate action taken.
- *Repeat*: The forklift must be able to repeatably and reliably engage with a target while avoiding obstacles.

3.3 Limitation 2: Simulation

Simulation will add value to the creation of a pallet engagement system through several mechanisms. First, current research on pallet engagement has tended to focus on presenting a single prototype, as opposed to showing an iterative design. A key reason for this may be due to the time associated with iterating testing and refining solutions that involve a complex system (e.g., an APES). Some examples of parts of the APES which may be tested and iteratively refined in simulation include:

- Pallet detection sensor.
- Pallet detection algorithm.
- Path planning or following algorithm.
- Novel application of pallet engagement.

Another benefit of simulation is proving the reliability of a prototype. In real-world testing it can take a significant amount of time to demonstrate the reliability through repetitive testing (e.g., using the requirements of the standard ASTM F3499 [24]). If the prototype is expected to perform with experimental variables (e.g., different pallet positions and heights), the time needed to sufficiently prove its reliability increases further. Simulation allows for a system to be rapidly tested a large number of times while consistently varying system parameters, to optimize metrics. Metrics to optimize could include the pallet engagement error (e.g., by varying the PDS configuration), or minimizing the precision requirements of the laser range finder used for localization while still fulfilling certain localization requirements.

The third benefit of simulation is through acceleration of the development process. If a particular simulation scenario results in failure, then data relating to this failure can be logged, with a fix applied and the scenario retested. This should result in less time spent trying to diagnose software issues using the actual hardware which may take longer and have limited availability. This is only possible if the simulation is deterministic, (i.e., when the same output is produced for the same input), then system failures can easily be reproduced by repeating the input which caused the failure. Deterministic simulation also helps track performance changes to the system, as any observed performance changes can be correlated to changes to the system as opposed to random variables in the simulation. The simulation should also produce results which are able to be compared with

results collected from real-world experimentation. This will give insight to any deficits in the simulation, which will allow for improvements to it.

However, there are downsides associated with using simulated testing. Testing in simulation is not the same as testing in the real-world so it should be expected that there will be differences between the two. These differences may arise because of assumptions made within the simulation. For example the physics simulation may not adequately represent the real-world physics. Additionally, assumptions made about the system's kinematic or dynamic properties may result in divergent outcomes when compared with real-world results. Simulations can be validated [66], [67], however, the time required to validate a simulation, particularly to a high level of confidence is significant [66]. This therefore, generally still requires the system to be validated with real-world result, which may differ from the results achieved in simulation.

Requirements

By analysing limitation 2, the following requirements for the systems are needed to provide a suitable simulation environment. Although simulation, is not *strictly*, needed to implement an APES, the limitations of the literature show that the addition of it would be beneficial.

- *Deterministic*: The simulation should be deterministic, to enable repeatability of simulated engagements.
- *Repeatably*: The simulation should be capable of running repeated trials (i.e., where the simulation is started, stopped, and reset), where experimental parameters should be able to be changed in-between trials.
- *Validation*: The results from the simulation should be comparable to real-world test results.

3.4 Limitation 3: Lack of Performance Results

A limitation of the published APES literature is they have few results about the performance of their systems. Some papers give no indication of the achieved engagement success rate [17], [50], others gave the engagements success rate, however, there was limited performance results beyond that. Some examples of performance results which could be calculated and presented include:

- Engagement success rate and reliability.
- Path following error and variance, e.g., cross-track and heading error.
- Engagement error and variance across multiple trials.
- Engagement duration, i.e., the time taken to complete the engagement.
- Maximum pallet displacement and orientation from expected position.
- Localization error.
- Fork control PID performance.

By expanding the results provided there should be two distinct advantages. Firstly, by better measuring the performance of the system, it should allow for a more transparent comparison to other methods. For example, if the path following metrics are provided, then they can be compared with alternative path following algorithms in literature. In addition, there are several relevant industry standards for pallet engagement such as ASTM F3499 [24]. If the relevant performance results are calculated, the system can be then be compared against such standards which will give insight into whether the system is fit for purpose.

By allowing for more transparency around the performance of the pallet engagement system, it will allow for the proposed system to be better designed if adopted by industry. For example, the maximum pallet displacement and orientation the APES can handle will influence the precision needed when placing the pallets, either by manual or automated drivers. More transparency into the performance of the APES will allow for its strengths and limitations to be analyzed, and therefore, to be improved upon.

Requirements

The following requirements for the system have been created by addressing the needs of limitations outlined in Section 3.4.

- *Reliability*: The reliability of the system must be quantified, and clearly communicated.
- *Engagement Performance*: The engagement performance of the proposed system must be quantified and given. This engagement performance must be representative of the system's real-world suitability.
- *Engagement Variance*: The variance of the engagement performance, should be given. The variance should be calculated from a sufficient number of trials such that it's representative of normal operation.

3.5 Consolidated Requirements

The following requirements have been formed by consolidating the requirements of the previous sections.

- *Simulation*: To help develop the APES, there should be a deterministic simulation which is capable of modifying experimental variables across repeated experiments. The simulation results should be comparable with real-world test results.
- *Obstacle Avoidance*: The APES must be capable of generating obstacle-free paths and engaging with the pallet while in aisles as narrow as 3.2 m.
- *Engagement Performance*: The APES should be capable of engaging with the pallet with the following maximum error margins: X<120 mm, Y<101.5 mm, Z<30.5 mm, k<7.5°.
- *Reliability and Robustness*: The APES must complete 29 repeated engagement based on the failure conditions of standard ASTM F3499 [24].

4

Design

4.1 Analysis

This section builds on the requirements presented in section 3.5 detailing what is needed to meet them. The requirements break the project into three parts:

- Designing a simulation to design, test and refine an APES solution
- Implementing the proposed solution on real hardware
- Ensuring the proposed solution meets the project requirements.

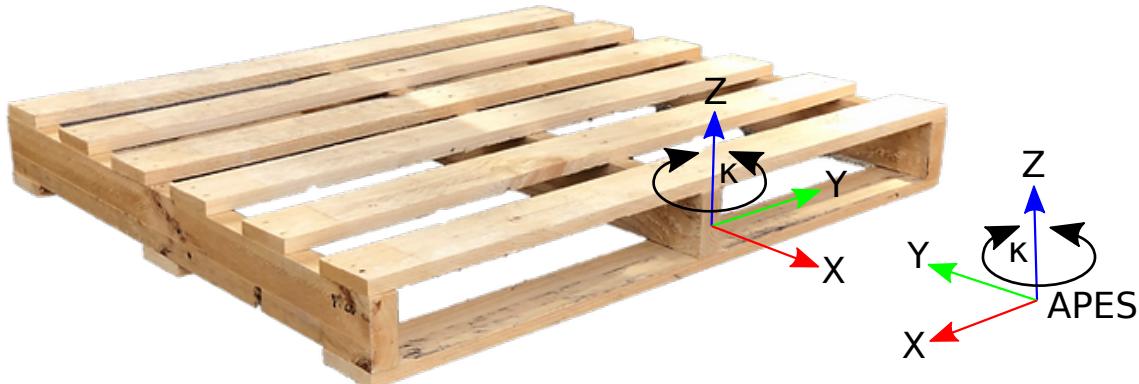


Figure 3: Coordinate system of the pallet used for engagement and the forklift. The difference between these coordinate systems as measured from the pallet, is the engagement error. This engagement error is denoted as X_e , Y_e , Z_e , and κ_e .

4.1.1 Requirement 1: Simulation

“To help develop the APES, there should be a deterministic simulation which is capable of modifying experimental variables across repeated experiments. The simulation results should be comparable with real-world test results.” (See Section 3.5).

Deterministic simulation:

A deterministic simulation means that uncertainty is eliminated from the simulation; for a set of inputs, the outcomes will be the same. One of these inputs generally will be a random number (referred to as the random seed), which will be used as a starting point for a random number generator. The APES should also be deterministic, so if errors occur within a simulation, the simulation can be easily reproduced.

Modifiable Experimental Parameters:

One issue identified in Section 3.3, was that previous work often had minimal design iteration before arriving at a final solution. Therefore, the simulation must allow for changing of experimental parameters, to facilitate design iterations. Examples of these experimental parameters include:

- Characteristics of the pallet detection sensor.
- Characteristics of the sensor used for localization.

- The path planning and following algorithm used.
- The difference between anticipate and actual pallet pose.

Some of these experimental parameters may be discrete (i.e., the path-following algorithm), whereas others continuous (e.g., the pallet detection sensor mounting angle). Multiple parameters must be able to be changed simultaneously

Comparability to Real-World Results:

The results from the simulation must be capable of being compared to real-world results. This means that all the results collected during real-world experiments (e.g., engagement performance), must also be collected during simulation experiments. The results collected during simulation should be done in a similar process to that of real-world testing to ensure the data is valid for comparisons. This may include:

- Starting the forklift at the same position.
- Having the same starting position variability for the forklift.
- Having similar kinematic and dynamic characteristics of the forklift across simulated and real-world testing.

4.1.2 Requirement 2: Obstacle Avoidance

"The APES must be capable of generating obstacle-free paths and engaging with the pallet while in aisles as narrow as 3.2 m." (See Section 3.5).

When generating obstacle-free paths, it is important to consider the scope of the APES engagement algorithm. The APES engagement algorithm is only designed to be responsible for controlling the motions of the vehicle when approaching the pallet, whereas another path planning and following algorithm is intended to oversee general navigation. The motivation for this distinction is due to a difference in functions. During general navigation, the priority normally is to create a globally optimal path with minimum interaction with other obstacles (other forklifts, pedestrians, etc.), however, there are no strict error tolerances for the goal-point. Whereas, during pallet engagement, obstacles must be avoided however, more emphasis should be placed on reducing error with the goal point.

4.1.3 Requirement 3: Engagement Performance

"The APES should be capable of engaging with the pallet with the following maximum error margins: X<120 mm, Y<101.5 mm, Z<30.5 mm, k<7.5°." (See Section 3.5).

The engagement error refers to the translational and rotational difference between the centre-point of the middle stringer of the pallet and the tracking point. The tracking point is used to control alignment of the tines with the pallet. Because the forklift's tines had a non-constant cross-section (i.e., thinner at the tip and wider at the base), it was easier for the tines to attempt to engage higher in the pallet pocket, as opposed to its true center. Therefore, the tracking point was offset 20 mm down from the top surface of the tines along the Z-axis. Likewise, to ensure the tines did not contact the pallet, the tracking point was offset 350 mm from the base of the tines. Only the rotational error around the z-axis was considered (denoted κ), as the other rotational errors were assumed to be

minimal. This is a reasonable assumption as a forklift generally has no other rotational degrees of freedom, and warehouse floors and racking are usually level to a practical tolerance.

Several factors will likely influence the maximum engagement performance achievable, including:

1. The APES lateral alignment with the pallet (i.e., the alignment across the y axis) after the APES has turned on the spot and is about to drive forward to engage with the pallet.
2. The path planning and following algorithm.
3. The characteristics of the pallet detection system (PDS).
4. The localization accuracy of the APES.

APES Lateral Alignment:

Before the APES engages with a pallet it has to complete an approximately 90° turn to face the pallet. Arriving at the correct position to begin the turn is important as after the turn is complete, the APES has limited ability to compensate for lateral error due to the vehicle's non-holonomic constraints, and the limited space within the racking. To calculate the appropriate position to turn the APES several options could be explored: the APES kinematic model could be determined, experimental data on turning characteristics could be collected, or the turn point could be iteratively changed until the desired behavior is observed.

Path Planning and Following Algorithm:

The selected path planning and following algorithm will influence the APES engagement performance. Algorithms that work well together and are suited to the domain of pallet engagement should enable precise engagement and fast adjustment to error. A common path planning technique is to discretize the configuration space, however, this is generally best suited to navigation involving reactive obstacle avoidance as opposed to precise maneuvering. This is because the discretization process will inherently add error to the goal point (i.e., as the goal-point is discretized), and as path planning techniques that operate on the discretized graph often result in paths ill-suited for non-holonomic vehicles.

Pallet Detection System Characteristics:

An important characteristic of the pallet detection sensor is how soon it can provide an updated pose of the pallet (i.e., as opposed to the assumed pose). If the APES does not receive an updated pallet pose from the pallet detection sensor before it starts turning to face the pallet, its ability to adapt to deviations between the updated pallet pose and the assumed pose will be limited. The accuracy of the PDS will ultimately determine the maximum possible engagement performance. The PDS response time may also influence the engagement performance as a low-response time system may be able to employ the techniques of visual servoing [68]. In this case, the error reported from the PDS would feed directly into the APES control systems. However, suppose the system has a high-response time (e.g., several seconds). In that case, it may be necessary for the pallet's pose to be transformed into another frame, such as the global localization frame, and for this frame to be used by the control system, with the additional frame being used to provide feedback as opposed to the PDS. If the pose is transformed into an additional frame (such as the global localization frame), this will add another source of error.

Name	Description
Acoustic	Primarily used for underwater systems which are unsuitable for laser or visual sensors. Its low spatial resolution and range limits possible usage.
2D Laser Range Finder	A common choice for indoor navigation due to its precise distance measuring and excellent spatial resolution. The cost of 2D laser range finders varies significantly, with more expensive units generally having a greater angular resolution, sensing accuracy and sensing speed. Localization performance may deteriorate when used in environments with limited features (e.g., a long corridor).
3D Laser Range Finder	A common choice for indoor navigation when sensing in 3D space is required. As with 2D laser range finders, it offers precise distance measuring and excellent spatial resolution. This type of sensor is generally more expensive compared with the other sensor choices.
Visual	Visual SLAM can be implemented using either monocular or stereovision. Both approaches produce feature-rich data, which extends the range of environments they can work in, however, they may be affected by lighting conditions.
ToF Camera	Provides both visual and 3D depth information simultaneously. As depth information is directly calculated, it allows for generally more precise localization, than with purely a visual sensor. Certain lighting conditions can significantly impact the performance of the sensor.

Table 3: Comparison of different sensing technologies available for SLAM, based on work by [69].

APES Localization Performance:

If pallet engagement does not direct sensing (e.g., a pallet detection sensor), but instead relies on prior information and real-time localisation, then the localization accuracy will have a significant influence on the engagement performance. There are several factors that may impact the localization accuracy: the sensor used for localization (see Table 3), the algorithm used for localization, and the quality of the map used for localization.

Test Infrastructure:

It is necessary for a test infrastructure to be designed and built to measure the APES's pallet engagement performance to evaluate if it meets the prescribed requirements (see Section 3.4). The test infrastructure should be used to measure the relative X, Y, Z and κ error between the pallet and the APES (i.e., from the pallet's CS).

There are several factors to consider when designing the test infrastructure:

- Reliability
- Robustness
- Accuracy

The standard ASTM F3499 [24] is designed to describe a testing methodology which can demonstrate and confirm a vehicle's operational reliability. However, to demonstrate this reliability, it has to be confirmed that the results collected are a valid representation of the

	X	Y	Z	κ
Task performance margin	120 mm	101.5 mm	30.5 mm	7.5°
Max test infrastructure precision and bias	2.5 mm	2.5 mm	2.5 mm	0.375°

Table 4: Vive bias and performance requirements.

system, and not unduly influenced by error. A maximum test infrastructure measurement error of 5 % of the TPM's, was allowed by ASTM F3499, and used for this project. A summary of the resulting maximum measurement error for each TPM is summarized in Table 5. Additionally, the project's requirement of 29 consecutive successful trials is based on a binomial distribution, which calculates the necessary number of successful trials to achieve the necessary confidence in the system, and test infrastructure. The requirement is based on a 95 % confidence in the test infrastructure and a 90 % probability of success threshold. The 5 % confidence in the test infrastructure and the task performance margins (see Section 3.5), can therefore be used to calculate the maximum allowed precision and bias for each axis of measurement. These specific requirements are summarized in Table 5.

	X	Y	Z	κ
Task performance margin	50 mm	50 mm	50 mm	7.5°
Max test infrastructure precision and bias	2.5 mm	2.5 mm	2.5 mm	0.375°

Table 5: Vive bias and performance requirement creation based on the engagement task performance margins.

The reliability and robustness of the test infrastructure depends heavily on the type of technology used to record the engagement error. The ASTM F3499 standard presents two possible options [24] for recording engagement error: grid paper with laser pointers and fiducial markers. Several other options were considered and summarized in Table 6.

4.2 Design

In this section the design used to implement the APES, previously specified, will be expanded upon. This process was iterative, based on experimentation, and was adapted as issues were identified. This section will start by giving a description of the hardware used for the project, before showing the overall software structure of the project, and giving more detail on each constituent component. Finally there will be a description of the test infrastructure.

4.2.1 Hardware Overview

This section starts by giving an overview of the type of forklift used for this project, how communication with the forklift was established, the additional hardware added to the forklift, and finally, the software design used to allow to for automatic pallet engagement.

Options from ASTM F3499						
Technology	Cost	Setup & Test Time	Robustness	Accuracy	Notes	
Grid paper	Low	High	High	Low	An extremely low cost option with an easily determinable measurement error (based on grid spacing). However it can become difficult with more than two degrees of freedom.	
Fiducial markers	Low	Low	Low	Medium	This option uses monocular cameras to calculate the pose of fiducial markers. This method can be accurate [70] when tracking markers on a planar surface, however, when tracking a system with more than two degrees of freedom significant error from the parallax effect can be introduced.	
Alternative options						
Virtual reality tracking	Medium	Low	Medium	Medium	A lower cost option than using multiple cameras with passive markers, while still preserving the capability of accurately tracking objects within three degrees of freedom.	
Multiple cameras with passive markers	High	Low	High	High	This is generally the most accurate [71], [72], however, expensive option. There are several commercial systems with varying costs.	

Table 6: Comparison of systems which could be used for the test infrastructure.

The research platform used was a Crown RM6000 Narrow-Aisle Reach Truck, which had been retrofitted for automation. The RM6000 is an electronically actuated lift-truck (referred to as a forklift) that has been designed to operate in narrow aisles and is equipped with a reach mechanism that allows the tines to extend to engage with a pallet. Generally Reach trucks are also equipped with “outriggers”, which extend forward under the forks and are used to stabilize the vehicle (see Figure 4). The motion of the forklift comes from the drive/steer wheel, which is controlled by adjusting the wheel speed and angle. The combination of this command (i.e., any combination of wheel speed and angle), is referred to as a “traction” command. The forklift has four actuators to move the tines: hoist, reach, side-shift and tilt. The side-shift mechanism moves the forks laterally and can be useful when the forklift is laterally offset from the pallet. The tilt mechanism can tilt the forks and is generally used to tilt the forks upward when travelling. The limits of these actuators are

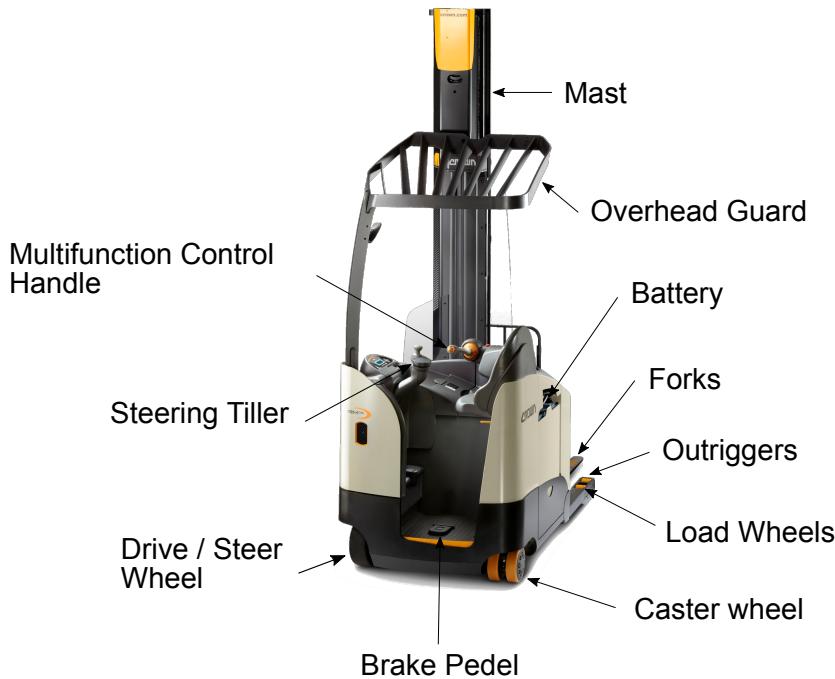


Figure 4: Components of a Crown Narrow-Aisle Reach Truck. The model shown is the Crown RM6000, which was used for this project after being retrofitted for automation.

summarized in Table 7.

The forklift normally travels in the direction of its battery (referred to as power unit first or PUF). However, when interacting with pallets, the forklift often travels in the direction of its forks (referred to as forks first or FF). This is in contrast to counterbalance forklifts which generally travel forks forward.

Communication with the forklift was done using the CAN protocol via a Canbus USB adapter. The vehicle could be operated in either “manual” or “auto” mode, with the transition triggered by a switch on the forklift. These two modes govern whether the forklift is controlled in the standard manner (e.g., the steering tiller, multi-function control and brake pedal), or autonomously. When in auto mode, the forklift can be stopped immediately using one of the four emergency stop buttons (E-stop) placed on the forklift or using the wireless E-stop (see Figure 5).

4.2.2 Additional Hardware

Three major additional pieces of hardware were used alongside the RM6000. These were an R2000 laser scanner (Pepperl and Fuchs, Mannheim, Germany) an onboard and an offboard laptop (Dell, Round Rock, Texas, United States). The R2000 is a 2-D LIDAR based laser scanner that was used for localization and mapping and was selected for its high scan rate (10-50Hz) and low angle resolution of 0.014° [73]. The onboard laptop was used to communicate with the forklift via CAN messages and with the laser scanner over Ethernet. The offboard laptop was wirelessly connected to the onboard laptop, where it could monitor and trigger automation processes of the forklift.

General Information	
Load Capacity (kg)	1450 ^a
Power (V)	36
Dimensions	
Wheelbase (mm)	1648
Width (mm)	1346
Performance	
Travel Speed (Power Unit First)(m/s)	3.2
Travel Speed (Forks First)(m/s)	2.6
Hoist Lift Speed Unlanded (m/s)	0.81
Hoist Lower Unlanded (m/s)	0.56-1.2 ^b
Actuator Limits	
Hoist (mm)	0-7620 ^c
Reach (mm)	0-592
Sideshift (mm)	-51-51
Tilt (°)	3-4

Table 7: RM6000 Specifications.^a Load capacity is subject to derating at height. ^b Average lower speed is 1.3 m/s. ^c Due to a different fork-type being used on the forklift, and wires being run to the fork-carriage, hoist was limited to 264 mm to 2520 mm.

4.2.3 ROS System

The proposed pallet engagement system is written in ROS based on a mixture of Python and C++ (see Figure 6). This system fulfils the following key tasks introduced previously:

- Mapping and localization
- Path planning and following
- Open and closed-loop control
- Finite-state automaton
- Pallet pose tracking

A standardized interface for tasks was implemented to improve the maintainability and extendability of the code-base. This functionality was achieved with the actionlib package provided by default by ROS [74]. In ROS based systems it is common for one node to ask another to perform a task (e.g., to move a robot to a goal point). ROS services can provide this functionality. However, if the task takes a significant amount of time to execute it is useful to be able to cancel the task, get feedback on its progress or supersede it with another. The actionlib package provides this functionality, breaking the engagement task into fundamental actions. The engagement task can be distilled into two fundamental actions: moving the forklift and closed-loop control. More detail is provided for both actions in the following sections. The action client is required to provide the goal and notify the server if it wishes to cancel an active goal (see Figure 7)

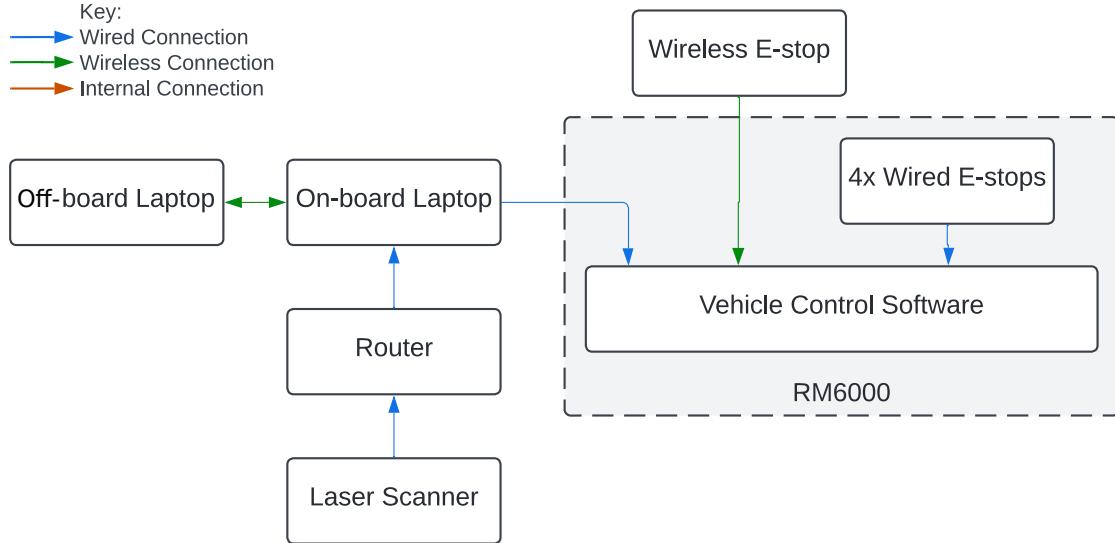


Figure 5: Hardware layout interfacing the ROS system with the RM6000 forklift.

Mapping and Localization

Google Cartographer was used to map the forklift’s environment and to localize it within its environment [75]. Google Cartographer is a type of SLAM (simultaneous localization and mapping) algorithm designed for portable laser range-finders (both 2- and 3-D). Localization was needed, as the designed solution should work alongside a warehouse management software (WMS), which would provide the forklift with a specific slot location for engagement. When Google Cartographer was compared to other SLAM algorithms (specifically Gmapping, tinySLAM, and Hector SLAM), using ground truth data, the former was generally found to outperform the latter [76], [77]. In real-world testing, the R2000 laser scanner was used to provide scan data to Google Cartographer, however, during simulated testing, this laser scanner was replaced with a simulated laser scanner. This simulated laser scanner was given the same parameters as the R2000, which are summarized in Table 8.

Scan rate	50Hz
Samples per scan	1680
Angular FOV	–180° to 180°
Angular resolution	0.014°
Range	0.1 m to 30 m
Range resolution	0.001 m
Typical inaccuracy	±25 mm

Table 8: Characteristics of Pepperl+Fuchs R2000 laser scanner.

A high-level description of how Cartographer performs SLAM is shown in Figure 8. There are two main components: the local SLAM, and the global SLAM. The local SLAM is responsible for forming submaps of the environment, whereas global SLAM attempts to find loop closure constraints between the submaps to reduce drift over time. Cartographer can also be used to provide odometry. Odometry is calculated using the non-loop-closed pose from the local SLAM result and can often supersede the accuracy of conventional wheel odometers due to the accuracy of most laser scanners. Once Cartographer has

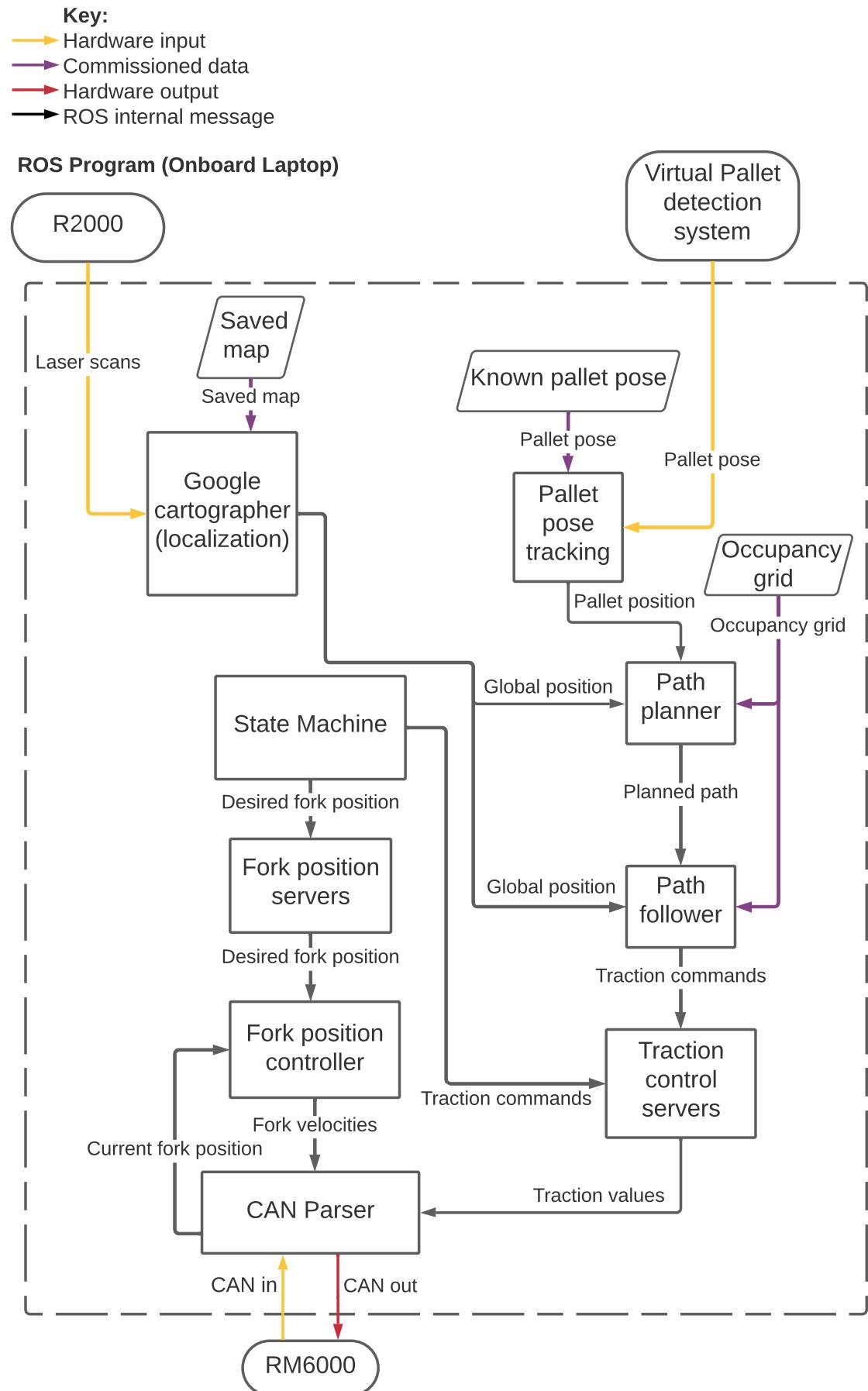


Figure 6: Overview of the ROS software.

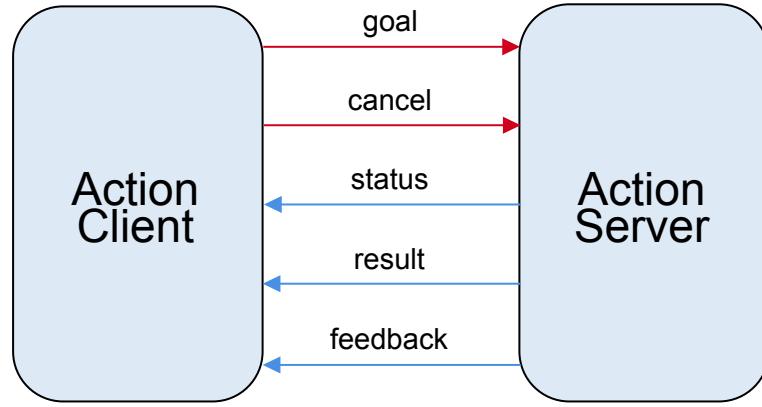


Figure 7: Actionlib client-server interaction. Diagram based on the likeness of [74], CC BY 3.0

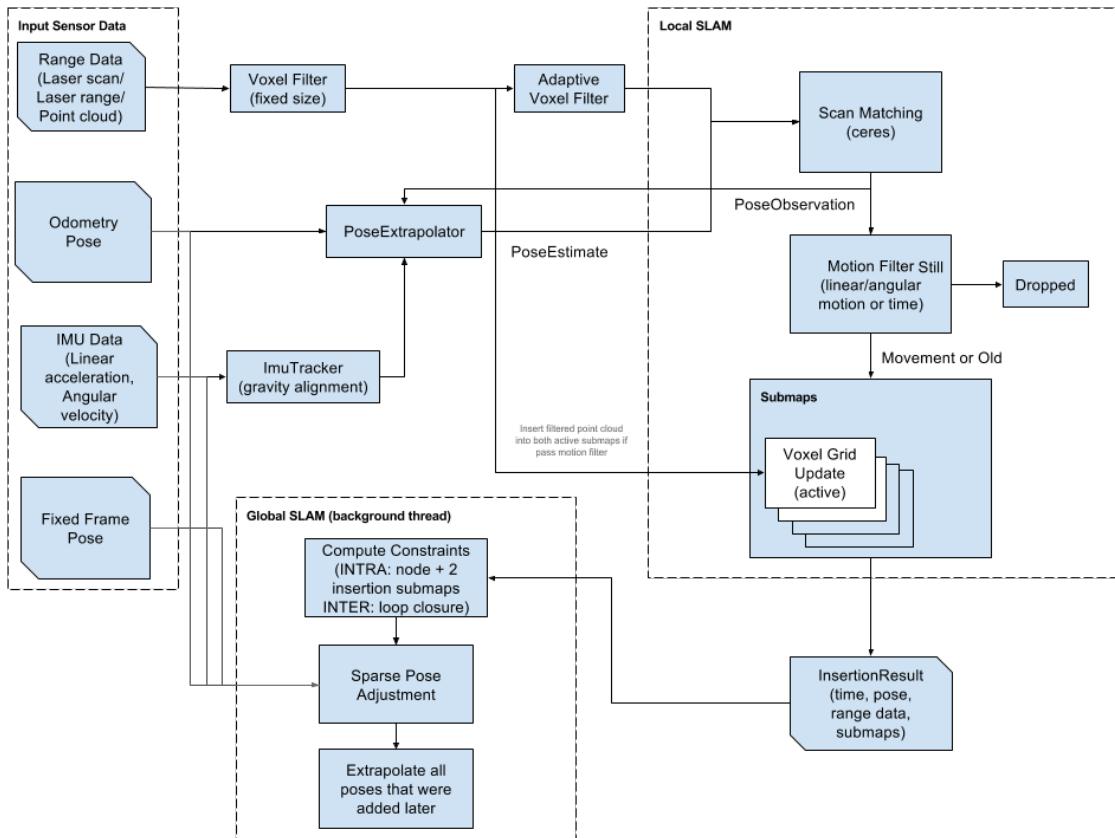


Figure 8: High level system overview describing how Google Cartographer performs SLAM. Reproduced with permission from [75], © 2016 IEEE.

been used to map an environment, it is possible for it to operate in purely localization mode, i.e., where Cartographer uses the generated map to calculate its pose.

Path Planning and Following

ROS Navigation [78] was used to provide path planning and following, it was selected due to its widespread usage and maturity. There are two main sub-systems that comprise ROS Navigation, a global and a local planner (see Figure 9). Both these sub-systems operate on a cost map that uses an occupancy grid of the robot's surroundings and inflates costs based on user specifications such as the robot's dimensions. The standard functionality

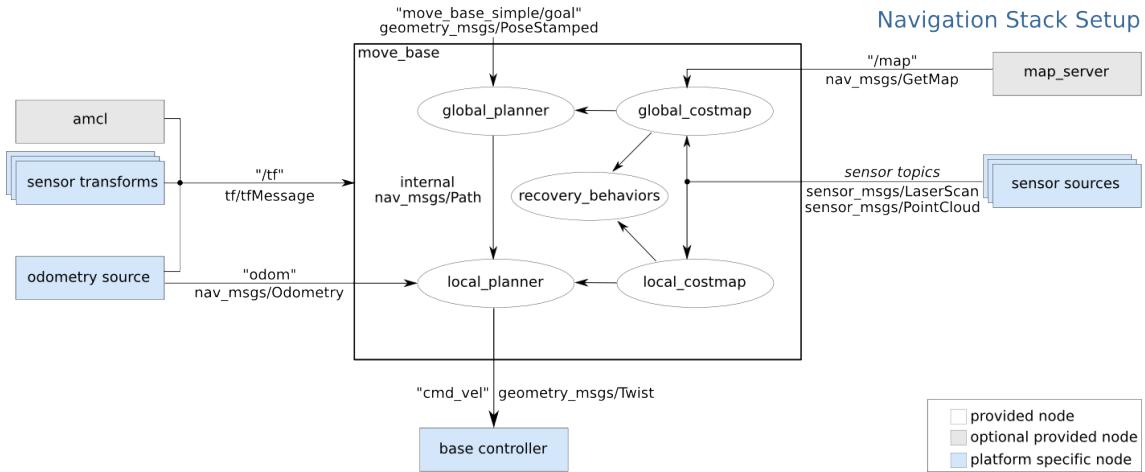


Figure 9: High level system overview of ROS Navigation. Reproduced with permission http://wiki.ros.org/move_base, CC BY 3.0

of ROS Navigation's global planner will use the cost map to generate a path using a shortest path algorithm (e.g., Dijkstra's [79], or A* [80]). This path, while collision-free, will not necessarily adhere to the kinematic constraints of the robot. The local planner will then operate on this global plan, attempting to follow it as closely as possible, while respecting the kinematic constraints of the vehicle. ROS Navigation offers two standard local planners, Dynamic-Window Approach (DWA) [81], and Timed Elastic Band (TEB) [82], [83]. The approach proved by default using ROS Navigation is well suited for reactive obstacle avoidance, however, it is ill-suited for the pallet engagement task due to the precise path tracking performance required. The inherent handicap in the design of the default path following algorithm is the fact that the module which generates the motor commands gets isolated information about the path it is tracking. The algorithm, therefore, does not consider the kinematic constraints of the entire path, so is constantly forced to re-plan as new information is received.

A different approach was considered in order to adhere to the strict path following requirements. The global plan is created based on a geometric model to consider the kinematic constraints of the vehicle and to also ensure a collision-free path. This path could then be used by a path following algorithm which in turn, attempted to follow the path as closely as possible. Although this change required modifying both the ROS Navigation global and local planner, the framework had several distinct merits:

- It allowed for the custom approach to be easily tested other conventional approaches.
- It has a robust and well-known interface.
- It provided several useful helper functions, such as creating the cost maps.

Open and Closed Loop Control

The APES used two types of control, open loop and closed loop, the difference being the closed-loop controller's output does not depend on the control action. Closed-loop control was achieved through a standard Proportional-Integral-Derivative (PID) controller. This controller was implemented using a standard ROS package [84] and offered several advantageous features:

- Dynamic reconfiguration of gains.

- Limiting output of the plant.
- Limiting windup.
- Ability to use angular measurements as an input.

The closed-loop controllers used for fork control were position controllers. In order to achieve a specified position, velocities were commanded to the internal RM6000 control software (see Figure 10). Two open-loop controllers were used to control the forklift's wheel speed and angle. These values were collectively referred to as traction. The final controller – referred to as “Rotate on Spot” – was used to control the forklift's angular position relative to the global localization frame.

The controllers were designed as actions that adhered to the actionlib interface. This meant that if a node wished to actuate the tines of the forklift, it would send a request to the server with the specified position, which in turn would use the PID package to move the forks to the specified position. Using actionlib allowed for a robust and reliable method of processing goals from different nodes as it allowed incoming goals to pre-empt current goals without failure or disruption. This was preferable when engaging with the pallet as although the pallet was static, its perceived position would change regularly as the pallet detection sensor received new information.

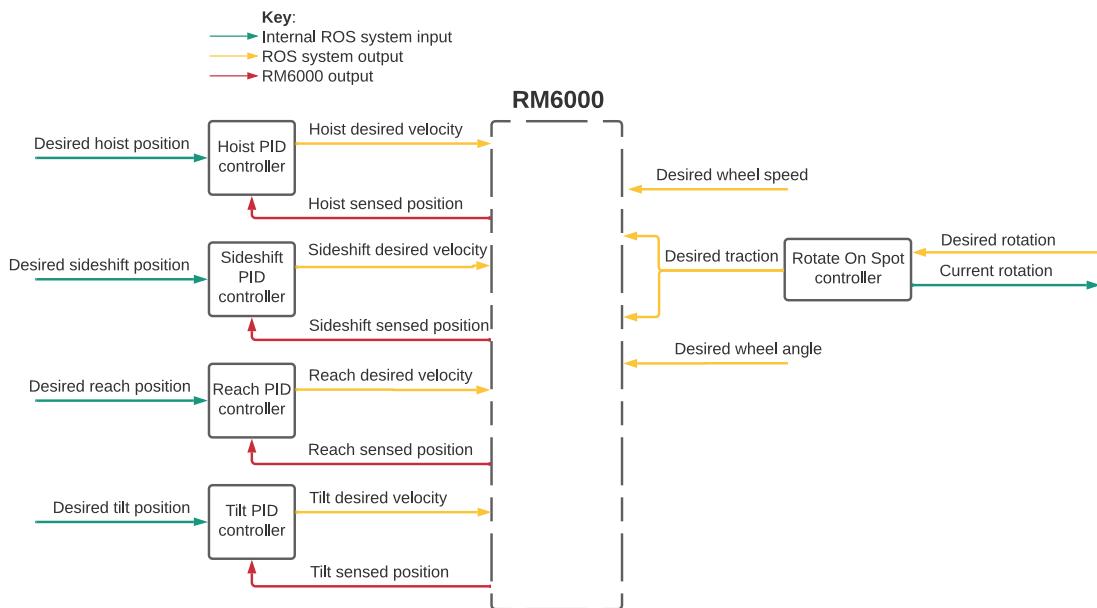


Figure 10: Overview of PID controllers.

Finite-state Automaton:

Creating complex robotic behaviour is not a trivial task as a robust system needs to be adapted to all reasonable changes in the environment. Designed by Boren et al., SMACH is a popular package for defining executable state machines in Python [85]. With SMACH, state machines can be designed programmatically where states are realized as classes, with a specific function for execution and an option to view the designed state machine diagram. Flexbe by Schillinger et al., [86] builds upon the work of SMACH to allow the state machine behaviours to be designed through a graphical tool, while the behaviour of each state is still defined programmatically.

The initial state machine designed for pallet engagement is shown in Figure 11, which shows both a combination of state and of containers which have nested state machines. There are three types of Flexbe containers:

- State machine container: This container acts like a standard hierarchical state machine. It can be used to define repeated sequences of states, e.g., a robot hand picking up a can.
- Parallel state container: This container allows multiple states to run concurrently. The signal for exiting out of the container can be simple (e.g., when all states are finished, or any state finishes), or complex. This container is useful when two modular states can be run in parallel in certain instances, e.g., a robot hand picking up a can while simultaneously whilst driving to a bin.
- Priority state container: If this container becomes active, it will prevent all other states from being active while it is active. For example, where the main state machine runs in one container and an error checking state runs concurrently. If an error is raised, the state transitions to a priority container to handle the error.

Information can be communicated to states in one of several ways. Behaviour parameters can be specified before run-time, and subsequently accessed by all states. Alternatively states can take inputs from other states. Or parameters can be specified for states. There are two concurrent states shown in the top-level state machine (see Figure 11); “Adjust forks for travel” and “Adjust forks for engagement”. The container “Adjust forks for travel” contains the state for hoisting and tilting the forks, where the specific instance of the states had parameters to specify the height and tilt the forks should move to.

The overall state machine is comprised of four basic types of state, however, there are several instances of each state. The functionality of each base state and the list of their instances are listed in Table 9.

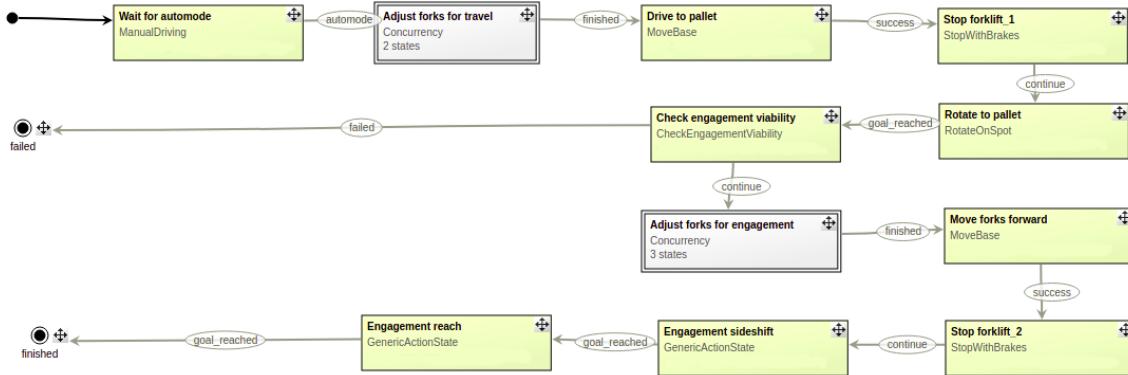


Figure 11: Top level state machine for pallet engagement. The figure is a screenshot of the graphical interface provided by the Flexbe software [86].

Pallet Pose Tracking

Although a pallet detection system was not implemented as part of the project, the APES was designed around the idea of initially being given a pose for the pallet, which would be updated as the forklift got within range of the PDS. This process was implemented as a ROS server, with the interface shown in Figure 12. The pallet pose tracker server started by reading the ground truth pallet position from a configuration file. For some simulations tested, an initial pallet position would then be generated by adding X, Y and κ error (see

	Generic State	State Instances
ManualDriving	Monitors the CAN messages and waits until the forklift is in auto mode.	“Wait for auto-mode”
MoveBase	Calls the ROS Navigation implementation of an action, to move the forklift from its current position to the pallet’s position. Parameters can specify the direction of travel, as well as if the forklift should drive to the front of the pallet, or to an offset point perpendicular to the pallet.	“Drive to pallet”, “Move forks forward”
StopWithBrake	Sends CAN message to apply brake and monitors the forklifts speed until the forklift’s speed is below a set threshold.	“Stop forklift_1”, “Stop forklift_2”
RotateOnSpot	Calls an action to rotate the forklift relative to the localized coordinate system. The rotation can be a relative angle (i.e., with respect to the forklift’s current position), or an absolute angle.	“Rotate to pallet”
CheckEngagementViability	Checks if pallet engagement appears feasible based on the forklift’s relative yaw to the pallets pose.	“Check engagement viability”
GenericActionState	Calls an action to run a PID controller on one of the fork actuators until the controller error is within a specified threshold. The target position can be a static value, or can dynamically track the current known position of the pallet.	“Travel hoist”, “Travel tilt”, “Hoist adjustment”, “Tilt adjustment”, “Side-shift adjustment”

Table 9: Description of all the base states used to create the state machine and a list of all the implemented states. Some of the state instances are not shown in the overview diagram as they were nested within concurrency containers.

the CS from Figure 3) to the ground truth position, otherwise the correct pallet pose would be used. During the engagement process clients, such as the “move_base” action node, could request and receive the pallet position.

A node would monitor the distance between a simulated virtual pallet detection system and the pallet (i.e., the actual pose, not the tracked pose) to simulate a pallet detection sensor. Before the virtual pallet detection system (VPDS) would return the pallet pose, three conditions had to be met:

1. The pallet must be within the PDS max measuring distance, specifically:

$$\sqrt{x_r^2 + y_r^2 + z_r^2} \leq \text{PDS max measuring distance}$$

2. The pallet must be within the vertical field-of-view (FOV) of PDS, specifically:

$$\text{abs}(\arctan\left(\frac{z_r}{x}\right)) \geq \text{PDS vertical FOV}$$

3. The pallet must be within the horizontal FOV of PDS, specifically:

$$\text{abs}(\arctan\left(\frac{y_r}{x_r}\right)) \geq \text{PDS horizontal FOV}$$

4. The pallet must be within the PDS detection threshold, specifically:

$$\kappa_r \leq \text{PDS detection } \kappa$$

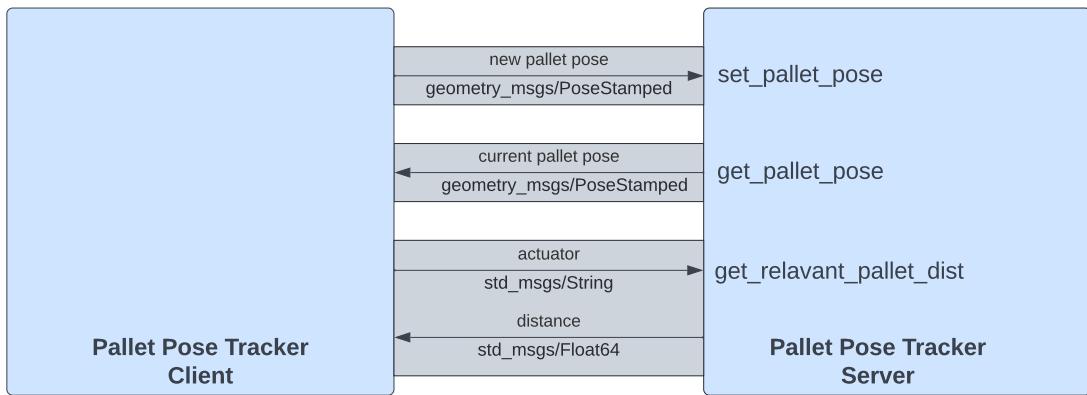


Figure 12: Pallet Pose Tracker Interface.

4.2.4 Simulation

Although there exist several simulation packages, Gazebo [87] is most seamlessly integrated with ROS, so was used for this project. Although simulation can allow for the rapid and low-cost development of ideas, there exists a risk of divergent results between simulation and reality. Therefore, to help avoid this pitfall, the interface (see Figure 13) between the APES software and Gazebo, and the interface between the APES software and the RM6000 software was kept as consistent as possible. Gazebo used the “ros_control” package, to simulate the forklift’s internal controllers [88].

Once the interface between Gazebo and the APES was established, the next step was to describe the physical world in simulation (see Figure 14). The forklift was described using the Unified Robotic Description Format (URDF) standard [89]. The following are some of the characteristics of the robots which can be described:

- Sensors (e.g., camera, laser range-finder)
- Links (e.g., inertia, and collision characteristics)

- Transmissions (e.g., between what actuator/joint, what type)
- Joints (e.g., fixed, resolute, piston)
- Overall kinematic/dynamic description

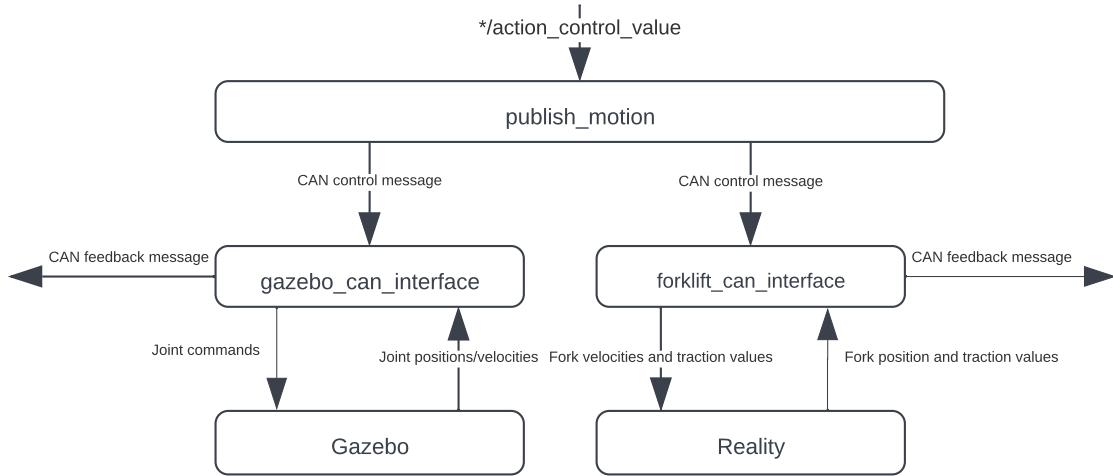
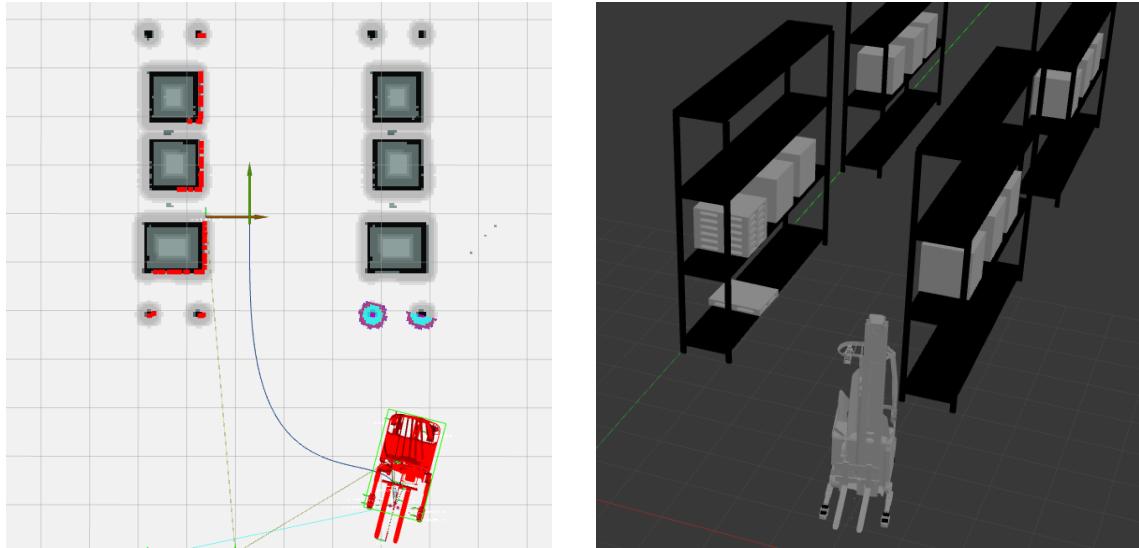


Figure 13: Interface between APES, the Gazebo simulation and reality.



(a) Visualization from Rviz. The green arrow is the APES initial navigation goal point, and the brown arrow is the pallet's current position. The red dots intersecting with the racking are caused by the simulated laser scanner.

(b) Visualization from Gazebo.

Figure 14: Visualization from Rviz [90] and Gazebo [87] during a simulated pallet engagement.

4.2.5 Test Infrastructure

Out of the options considered (see Table 6), for test infrastructure hardware, the virtual reality motion tracking system was selected. The specific type used to implement the test

infrastructure was the HTC Vive’s lighthouse tracking system (referred hereafter as the Vive). The Vive is comprised of two or more base-stations, with one or more trackers and can record the pose of the trackers (relative to the base stations) with six degrees of freedom. The tracking system relies on optical transmission, where the two base stations emit synchronized flashes (blinking light-emitting diodes) and two infrared sweeps (one vertical and one horizontal). The tracking devices are equipped with inertial measurement devices for finding dead-reckoning position. Although this position is relatively accurate for a very short duration it suffers from drift so has to be corrected. To correct this drift, the tracking devices are also equipped with photo-diodes that detect the infrared sweep and LED flash. Based on the duration between these events, the Vive is able to triangulate and update the position of its trackers [91]. Several studies have investigated the accuracy of the Vive and these have been summarized in Table 10.

Ref	Static analysis	Ground truth system	Intended application
[92]	$\sigma_{lin} < 0.5 \text{ mm}$, $\sigma_{ang} < 0.006^\circ$	Astrobee robot	Robotics
[93]	$e_{EA} : 0.63 \pm 0.27 \text{ mm}$, $0.66 \pm 0.60^\circ$	UR5 robot with VICON MOCAP	Clinical research
[94]	RMSE: $< 0.066 \text{ mm}$, 0.053°	Manual measurements with string and chalk	VR for scientific research
[95]	RMSE: $6.8 \pm 3.2 \text{ mm}$, $1.64 \pm 0.18^\circ$	Vicon MOCAP	Lumbar postural change
[96]	RMSE: $< 1.05 \text{ mm}$, $< 1.124^\circ$	UR5 robot with Optitrack MOCAP	Tracking medical devices
[97]	$\sigma < 0.3 \text{ mm}$	ABB-IRB 140 robot	Automated testing of industrial robots

Table 10: Comparison of state-of-the-art HTC Vive evaluation studies, based on the work by Ikbal et al., [98]. RMSE refers to the root mean squared error.

These results indicate that the Vive has a high likelihood of fulfilling the accuracy test requirements, however, due to the significant amount of variance in the evaluation studies, independant analysis on the specific setup was completed below. To construct the test infrastructure, two base stations were placed behind the pallet which was to be engaged with, one tracker was placed on the fork carriage, and one tracker placed on the pallet. By recording the pose of both the fork carriage and the pallet, then taking the difference, the relative engagement error could be calculated. The calibration procedure used to calibrate the Vive, and to therefore, calculate the engagement error is shown in Section 4.3.5.

4.3 Implementation

This section is a more in-depth presentation of selected features to provide clarity for researchers wishing to extend the presented work. First, Section 4.3.1 describes the techniques used to tune the PID controllers and to validate the control system output. Next, Section 4.3.2 gives the specific algorithms used for path planning and following

as well as any alterations made to the standard algorithms to adapt them to the task of pallet engagement. Finally, Section 4.3.4 describes what steps were taken to scale up the project simulation capabilities and ensure it could meet the requirements of the project.

4.3.1 PID Controller Tuning

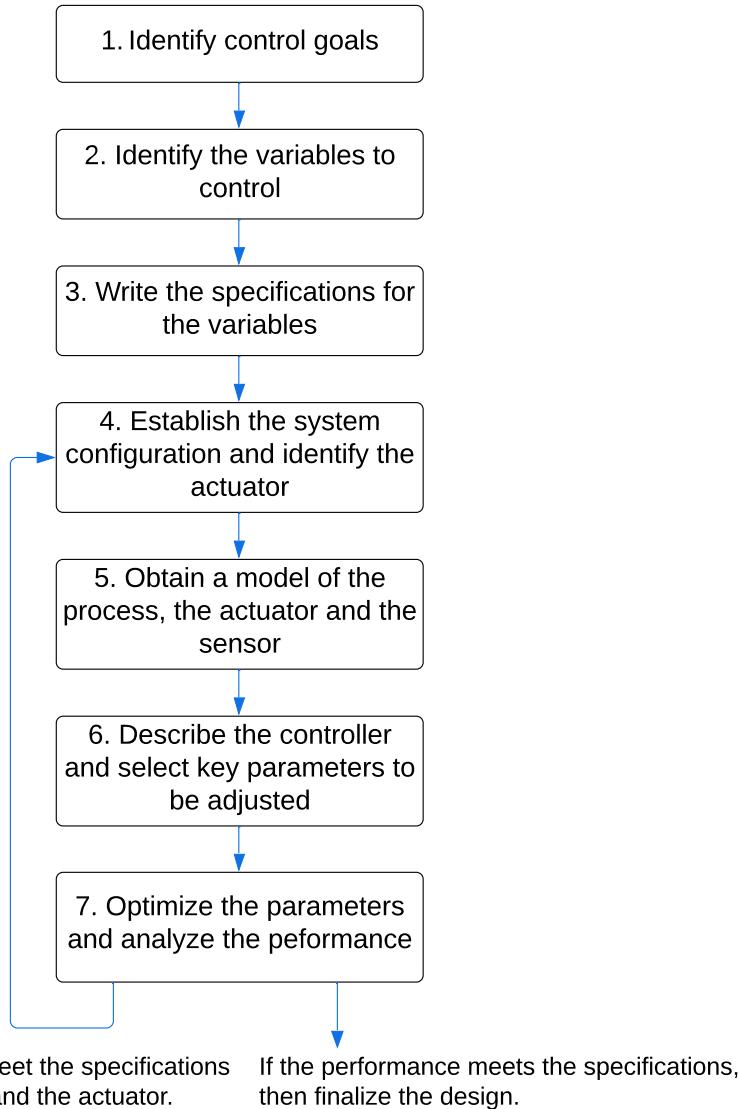


Figure 15: PID design process from “Modern Control Systems, Global Edition, 13th Edition”, reproduced with permission from [99], © 2017)

To implement the PID controllers to control the forklift's tines, the standard procedure from Richard et al, in “Modern Control Systems, Global Edition, 13th Edition” [99] was used. An overview of the procedure is shown in Figure 15. The controllers were a form of position controller where they command a velocity output in the attempt to move the tines to the desired position. The specifications for the PID controllers were those prescribed in Table 2 and the system and actuator configuration are described in Figure 10. There was an additionally closed loop controller that could actuate the forklift's drive wheel such that a specific yaw could be achieved (the yaw being relative to the localization's coordinate system).

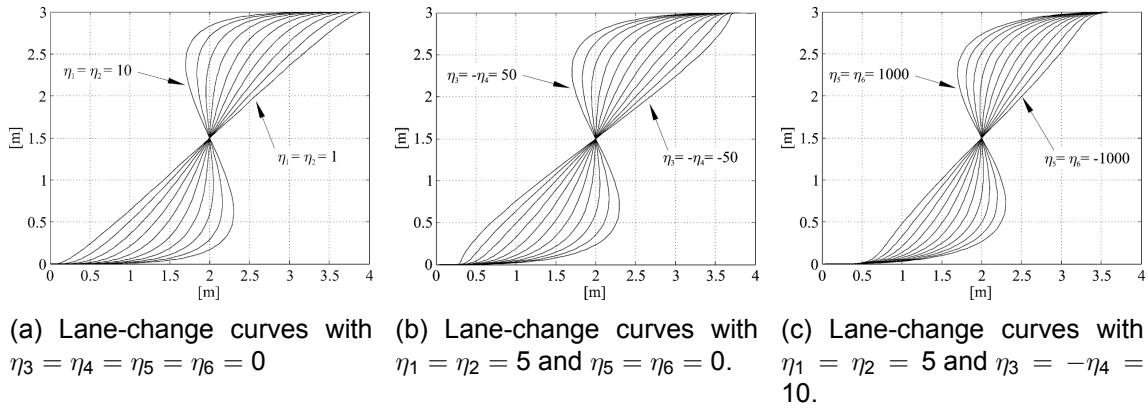


Figure 16: Illustration of the impact of altering the spline parameters on the resulting curve interpolation. Reproduced with permission from [102], © 2007 IEEE

To obtain a model of the process, actuator and the sensor (i.e., the system), for the fork PID controllers, a unit step velocity input was first presented to the velocity controllers, with the corresponding displacement output recorded. The velocity was chosen such that it would be within the normal operating range of the forklift. The period where the velocity command was non-zero was selected such that the actuator would not exceed its limits. The period where the velocity command was zero was selected such that the transient effect from the previous motion was negligible. MATLAB [100] was then used with the System Identification Toolbox [101] to approximate the system. Once an approximation of the system was calculated, the Control System Toolbox [101] was then used to test different gains, to see which best met the requirements (see Table 2).

4.3.2 Path Planning

A custom path planner and follower are needed to fulfill the requirements of this project, specifically the need to highly accurately navigate to a goal-point. The path planner algorithm used was a type of η -spline geometric interpolation (specifically η^3 -spline interpolation) which means the resulting path will be a G^3 type path (see Table 1). Therefore, for any path created, there exist initial conditions and continuous acceleration commands to allow the robot to follow the specified path. Path planning is done using a Cartesian primitive, referred to as a η^3 -spline, which is a seventh order polynomial that allows for the interpolation between any two Cartesian points.

This problem is formally defined by Piazz et al., [102] as: *Determine the minimum order polynomial curve that interpolates two points $\mathbf{p}_A = [x_A, y_A]^T$ and $\mathbf{p}_B = [x_B, y_B]^T$ which with related unit tangent vectors defined by θ_A and θ_B , scalar curvatures κ_A and κ_B and curvature deviations $\dot{\kappa}_A$ and $\dot{\kappa}_B$. Assume that the resulting interpolation $\mathbf{p}_A, \mathbf{p}_B \in \mathbb{R}^2$, $\theta_A, \theta_B \in [0, 2\pi]$, and $\dot{\kappa}_A, \dot{\kappa}_B \in \mathbb{R}$ can be arbitrarily assigned. Therefore, the equations of the seventh order polynomial curve $\mathbf{p}(u) = [\alpha(u)\beta(u)]^T$, $u \in [0, 1]$ can be defined as the following:*

$$\begin{aligned}\alpha(u) &:= \alpha_0 + \alpha_1 u + \alpha_2 u^2 + \alpha_3 u^3 + \alpha_4 u^4 + \alpha_5 u^5 + \alpha_6 u^6 + \alpha_7 u^7 \\ \beta(u) &:= \beta_0 + \beta_1 u + \beta_2 u^2 + \beta_3 u^3 + \beta_4 u^4 + \beta_5 u^5 + \beta_6 u^6 + \beta_7 u^7\end{aligned}$$

By considering the problem definition, the coefficients can be solved for (see [102] for

full solution). This results in two equations which depend on a six dimension vector of real parameters, $\eta := [\eta_1, \eta_2, \eta_3, \eta_4, \eta_5, \eta_6]^T$, the start and end points, $\mathbf{p}_A = [x_A, y_A]^T$ and $\mathbf{p}_B = [x_B, y_B]^T$, the unit tangent vectors defined by θ_A and θ_B , scalar curvatures κ_A and κ_B and curvature deviations $\dot{\kappa}_A$ and $\dot{\kappa}_B$. The η parameters can be used to alter the curve resulting between the specified start and end goals. This is shown in Figure 16, where the systematic alteration of pairs of the η parameters generates a systematically altered interpolation curve. Modifying parameters η_1, η_3, η_5 will change the curve at the start point, whereas, parameters η_1, η_3, η_5 will influence the point at the end. The parameters η_1, η_2 and η_3, η_4 and η_5, η_6 can be interpreted as influencing the curves “velocity”, “acceleration”, and, “jerk” respectively.

One of the main disadvantages of using the geometric interpolation method is its limited ability to plan obstacle-free paths. To address this, the following methodology was undertaken to ensure an obstacle-free path was planned where paths would be optimized based on resulting kinematic properties and obstacle avoidance. Firstly, a three-dimensional sample space was constructed, which could be used to generate three values, $\eta_{12}, \eta_{34}, \eta_{56}$. These in turn would define the η parameters, specifically, $\eta = [\eta_{12}, \eta_{12}, \eta_{34}, -\eta_{34}, \eta_{56}, \eta_{56}]$. The randomly sampled η parameters would be used to generate paths that were evaluated based on their merits, specifically: their curvature, their distance to obstacles (based on the ROS Navigation cost map), and their path length. This process is shown in more detail with Figure 17.

4.3.3 Path Following

Once a path has been generated it is the responsibility of the path following algorithm to generate motor commands to make the APES follow the path. The selected path planner is the pure-pursuit algorithm, which is a path following algorithm that was first introduced in [103], before being improved upon in [104] and [105].

The algorithm works by first calculating a look-ahead point based on a look-ahead distance L . A semi-circle is fitted between the robots tracking point (in this case the midpoint between the two load wheels of the RM6000) and the curvature (κ) calculated with the following:

$$\kappa = \frac{2 \sin(\alpha)}{L}$$

The commanded heading rate can then be calculated (v_r is the vehicle speed):

$$\omega = \frac{2v_r \sin(\alpha)}{L}$$

In the above equation α is calculated by transforming the look-ahead point into the coordinate system of the robot, specifically (θ is the vehicles heading):

$$\alpha = \arctan \left(\frac{y_{\text{ref}} - y_r}{x_{\text{ref}} - x_r} \right) - \theta.$$

To allow the pure pursuit to perform best on the engagement task several additional changes were made. The look-ahead distance was replaced with a look-ahead time T , which the look-ahead distance was based on, i.e., $L = T \cdot v_r$. The next improvement was to dynamically modulate the vehicle speed based on the following:

- Distance to nearby obstacles.

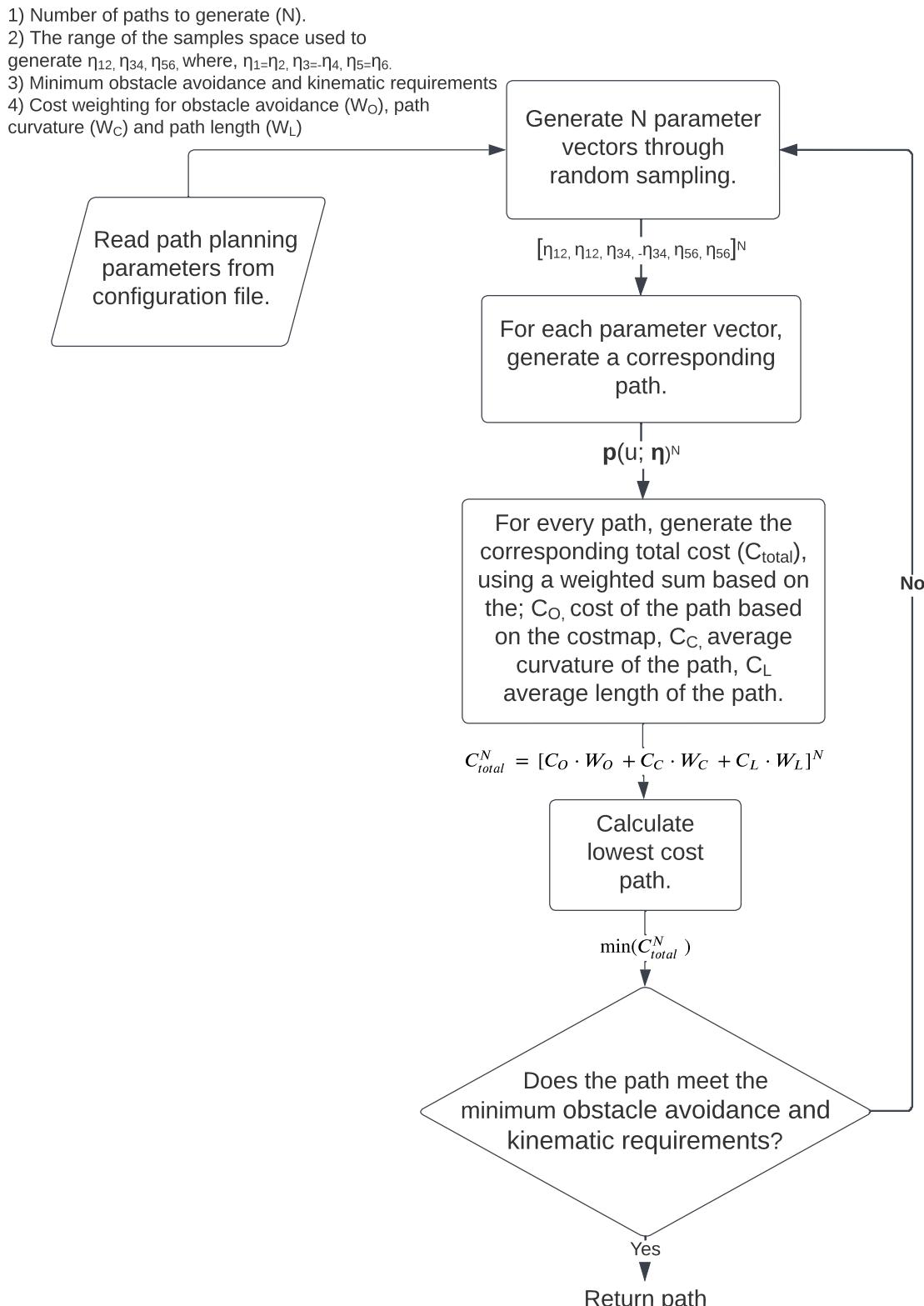


Figure 17: Path planning algorithm.

- Distance to the goal-point
- Current path curvature
- Current heading error
- Acceleration limit
- Deceleration limit
- Current direction of travel

The pure pursuit algorithm would control the forklift's wheel angle and speed. When the forklift was in motion, its velocity would be monitored, alongside the proximity of objects to avoid obstacles. This would be done by projecting the APES's current motion for a set period of time (i.e., assuming no external forces and the current velocities), and checking that this would not result in any collisions. This check was important when navigating in the aisle due to the tight navigational tolerances, as even minor deviation from the path (i.e., due to insufficient path following), could be enough for the APES to collide with adjacent racking. No sensor was used to check for unexpected or moving obstacles as this task is often taken care of by an independent safety system. Such systems monitors for nearby obstacles (often with laser range-finders), and generally stop the vehicle when they are within a set distance of the vehicle. These systems are required to be rigorously tested to a high-level, before they are validated, therefore, this process was not within the scope of the project.

Two conditions are evaluated to check when the path following algorithm has reached the goal-point.

$$X_{threshold} > \text{abs}(X_{dist\ to\ goal})$$

$$Yk_{threshold} > Y_{dist\ to\ goal} + \text{forklength} * \sin(\kappa)$$

The first condition simply checks if the error along the X axis is below a threshold (50 mm). However, the second condition is a blend of both the yaw and Y error (error to the goal-point from the forklift). These error requirements were blended to better represent the path following requirements when the forklift is driving FF towards the pallet as it is the combined action of the Y_e and the κ_e which determine if the engagement will be successful.

4.3.4 Simulation

Section 4.2.4, detailed how the simulation environment was created and interfaced with the APES. This section will first give more detail as to how experiments were run in simulation, then the usage of high-performance computing and finally the Singularity containerization software [106] will be discussed.

Simulation Experiments:

To run experiments in simulation, engagement trials needed to be run repeatedly without requiring manual starting or stopping. To facilitate this, a specific module would first start the Gazebo environment (and the static object such as the pallet within it), before spawning the forklift in a random position (governed by the random seed). Google cartographer would be seeded with the forklift's current position, which would enable it to track its position correctly from the start. Next, the APES's state machine would receive a start command which would allow it to begin its pallet engagement sequence. Once either the

state machine had completed successfully, or an error was detected during engagement (e.g., there was a collision with the pallet), results on the trial would be recorded to a spreadsheet. These results included:

- Outcome (e.g., pass, pallet collision, restart error)
- Engagement duration
- Engagement error (x, y, z, κ)
- Cross track error (mean, max and standard deviation (std.))
- Heading error (mean, max and std.)

It was necessary to shut down and restart the APES software between every engagement trial. Additionally, specific modules had to be monitored, unloaded and reloaded to ensure stable operation across multiple trials. By performing simulation across a large number of trials it was possible to get more data on the impact of different design decisions on the engagement outcome. For example, a large number of simulations could be run with different configurations of a virtual laser scanner (i.e., for localization), to determine which led to better performance.

Computational Capacity:

To allow the number of trials to be scaled up, high-performance computing was used. This was achieved through a project grant (“uoa03445”) from the New Zealand eScience Infrastructure (NeSI). NeSI is a collaboration between NIWA, the University of Auckland, University of Otago, and Manaaki Whenua - Landcare Research. NeSI’s services include high-performance computing facilities (i.e., their computing cluster), and consulting. The project grant allowed for engagement experiments to be run with more computational power and therefore, shorter durations, as well as the ability to run multiple experiments simultaneously without any decrease in performance (i.e., by running the different experiments on different cores). To enable experiments to be run on the NeSI cluster, some changes had to be implemented; primarily the usage of Singularity [106] for containerization and Gzweb [107] for visualization.

Singularity:

Singularity is an open-sourced operating-system-level virtualization software (i.e., containerization). Singularity containers are defined by a definition file, which defines the operating system and packages needed. These definition files can then be compiled to an image which can be used for containerization. If programs execute within the container, they should run identically regardless of if they run locally (i.e., on a laptop) or on a high-performance computer. Singularity is different to virtual machines as it shares the operating system kernel. This is advantageous in high-performance computing applications as it helps remove the need for elevated privileges during the execution of programs, which allows them to be safely used on shared computing platforms.

Singularity was needed to execute virtual pallet engagement experiments on NeSI as not all software packages were available on NeSI by default. As elevated privileges were not allowed on NeSI, so software had to be defined and built locally (i.e., with elevated privileges) before being executed on NeSI without elevated privileges. To ensure consistent software versioning between local and remote experiments, the workflow depicted in Figure 18 was used. This workflow allowed for one repository of files, one definition file, and one results folder (i.e., where data from experiments was written to). These files were kept updated by syncing folders (i.e., locally and on NeSI) using the rsync utility [108].

Both ROS and Gazebo used a specific default master port for communication. This configuration was problematic if two singularity containers running an APES experiment were run simultaneously as both containers would be attempting to communicate over the same range of ports. To avoid this issue, when containers were started, the environment variables that defined the ROS and Gazebo master ports were assigned to random valid ports.

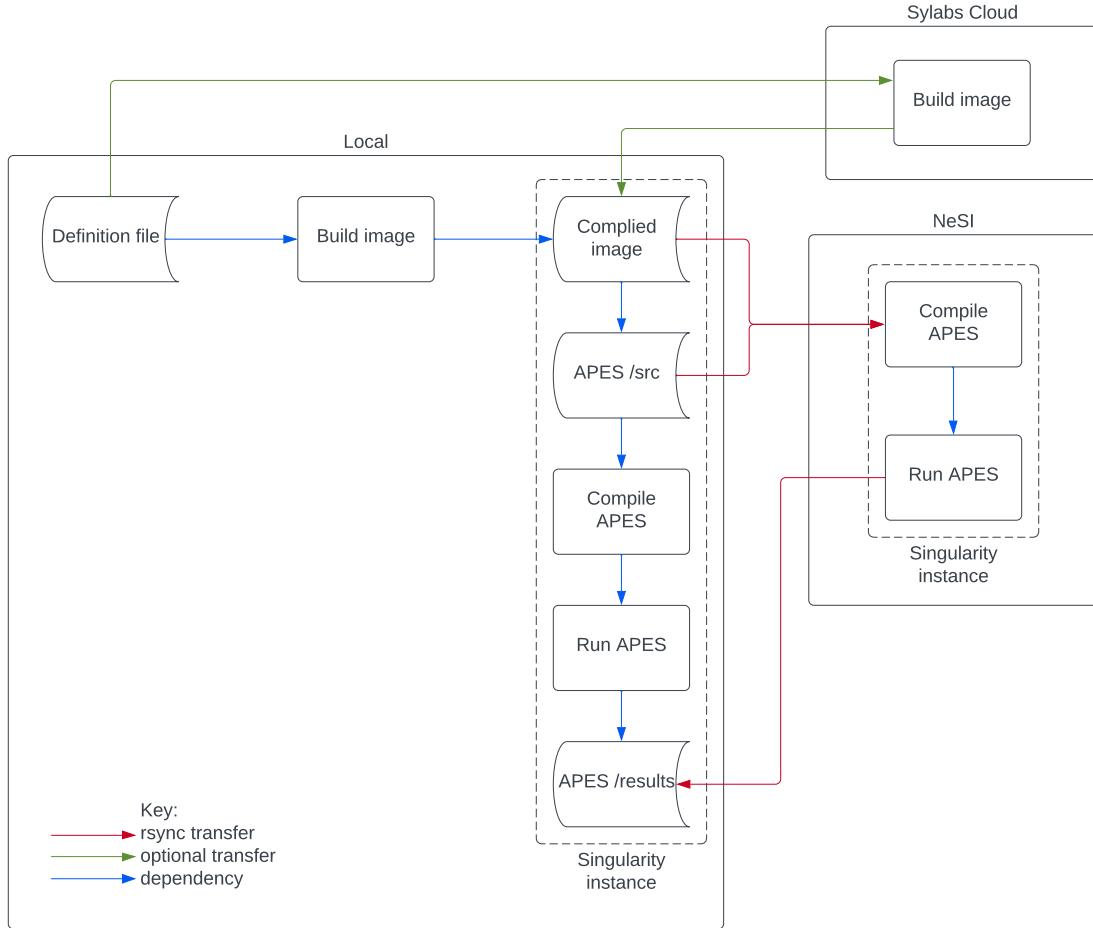


Figure 18: Containerization workflow, between local and remote compilation. The optional link to the “Build image” process within the Sylabs Cloud container represents the option of remotely building the image.

GzWeb:

To visualize the engagement sequence when running remotely, the program Gzweb [107] was used to display the Gazebo world on a local web browser. This was necessary as occasionally discrepancies between the outcomes of the simulation running locally and remotely would occur (e.g., due to greater latency locally), which would need to be investigated while running the APES remotely. To allow Gzweb to run, port forwarding had to be enabled across the Secure Shell (SSH) connection to the NeSI cluster, allowing traffic forwarded from the local SSH terminal to NeSI’s SSH cluster. On NeSI’s cluster, whilst an APES experiment was running, GzWeb could be started, which would forward traffic through the open port to the local machine. This would enable the simulation to be viewed and paused/played from a local machine.

4.3.5 Test Infrastructure

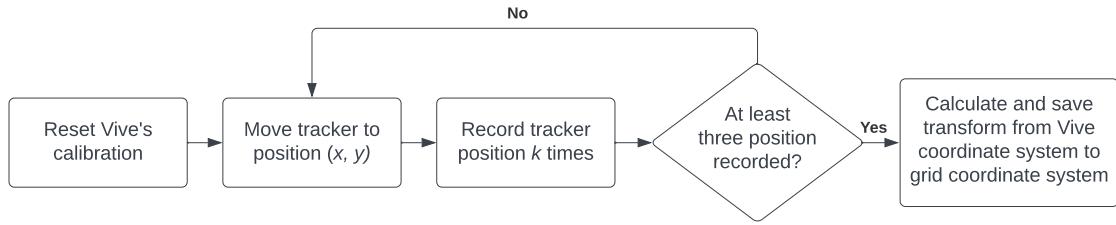


Figure 19: Overview of the procedure for calculate the transform between the HTC Vive coordinate system, and the grid coordinate system.

This section will describe the implementation of the test infrastructure. First, an overview of the system will be given, followed by a description of how the system was calibrated and then a description of the procedure used to validate the accuracy of the hardware used in the test infrastructure.

Mounting Hardware:

The test infrastructure used the Vive for finding the relative pose between the APES and the pallet. The two base stations were placed behind the pallet which was to be engaged with, one tracker was placed on the pallet, and one tracker was placed on the fork carriage. Figure 20 shows how the two trackers were mounted. The tracker mounted on the fork carriage used a magnetic L-shape bracket (with tripod mounting hardware) to adhere it to the fork carriage. The tracker had to be repeatedly placed in the same place (i.e., so the rigid transform from the tracker was valid). For this reason the perimeter around the bracket's mating surface was marked. The Vive tracker used the standard tripod connection [109], so a mounting plate was fashioned out of an aluminium plate using a countersunk M10 screw, machined to length as mounting hardware.

Calculating Vive Transform:

When the Vive system initialized, it was calibrated to a coordinate system based on the position and orientation of the base stations. This meant a transform needed to be calculated between the Vive's coordinate system and one relative to the pallet's coordinate system. First, a 1 m by 1 m grid was drawn on the ground nearby the pallet, which was used for engagement testing. A laser grid (Bosch, Gerlingen, Germany) and tripod were used to align the grid alongside the pallet's y-axis (see Figure 20). This grid was then used to collect three or more poses relative to the grid's coordinate system and the Vives coordinate system. These poses were then used to calculate the transform between the Vive coordinate system and the grid coordinate system. The following method was used to calculate the transform between the set of points [94]:

Defining the set of points:

$$A = [p_1^a, p_2^a, p_n^a] = [(x_1^a, y_1^a), (x_2^a, y_2^a), (x_n^a, y_n^a)]$$

$$B = [p_1^b, p_2^b, p_n^b] = [(x_1^b, y_1^b), (x_2^b, y_2^b), (x_n^b, y_n^b)]$$

Finding centroid for each set of points:

$$COG^a = \frac{1}{n} \cdot \sum_{i=0}^{i=0} p_i^a \quad COG^b = \frac{1}{n} \cdot \sum_{i=0}^{i=0} p_i^b$$

Finding relative translation between set of points:

$$A' = A - COG^a \quad B' = B - COG^b$$

Estimating rotation between set of points, using the Kabash algorithm [110], within the SciPy package [111].

$$L(C) = \frac{1}{2} \sum_{i=1}^n w_i ||a_i - Cb_i||^2, \text{ minimize } L(C)$$

Finding the transformation from A to B:

$$A \rightarrow B = -C * A' + B$$

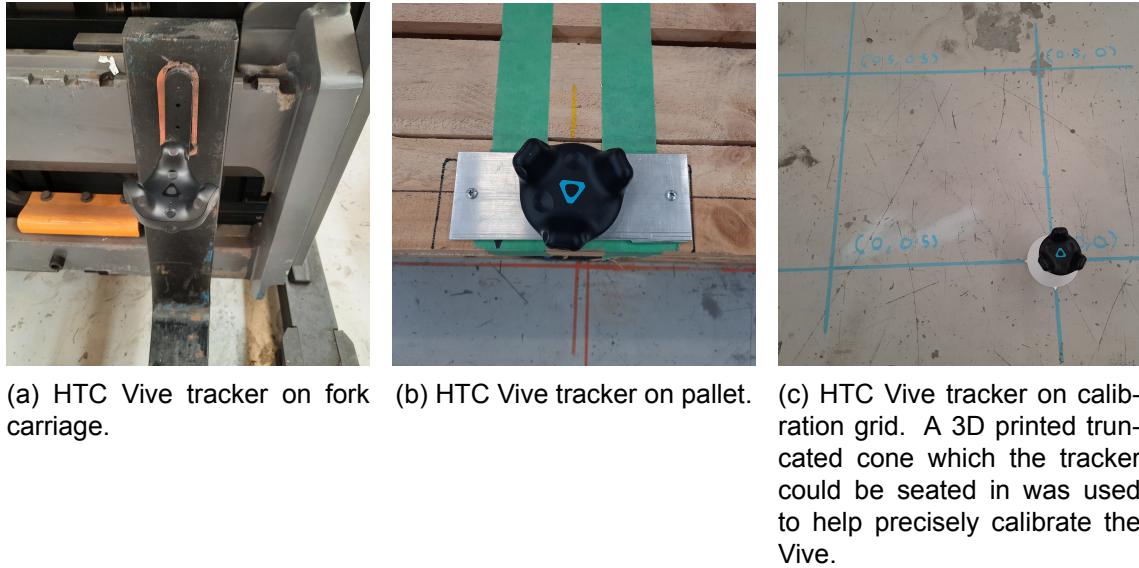


Figure 20: Vive mounting and calibration hardware.

Calibrating Test Infrastructure

Once the transform between the Vive's and the grid's coordinate system was established, the next step was to establish the transforms between the trackers and the points they were tracking. These points were the face of the pallet, and a point on the tines 20 mm down from the top surface of the tines and 350 mm from the base of the tines. These transforms were rigid and could be manually measured using a measuring tape, a laser grid (Bosch, Gerlingen, Germany) and a tripod.

Once these transforms were calculated, it was possible to use the poses of the trackers (rel. to the Vive's coordinate system), to find the engagement error. A diagram showing the different transforms is presented in Figure 21. The Kinematics and Dynamics Library by the Orocosp Project [112], was used to calculate the relative transform from the point of interest on the forks and the pallet, using the other transforms.

Parameter Tuning:

When configuring the APES, many parameters could collectively influence the engagement performance, such as the position and orientation of the PDS. One approach to configure the APES is to follow a “guess and check” approach. The limitation of this

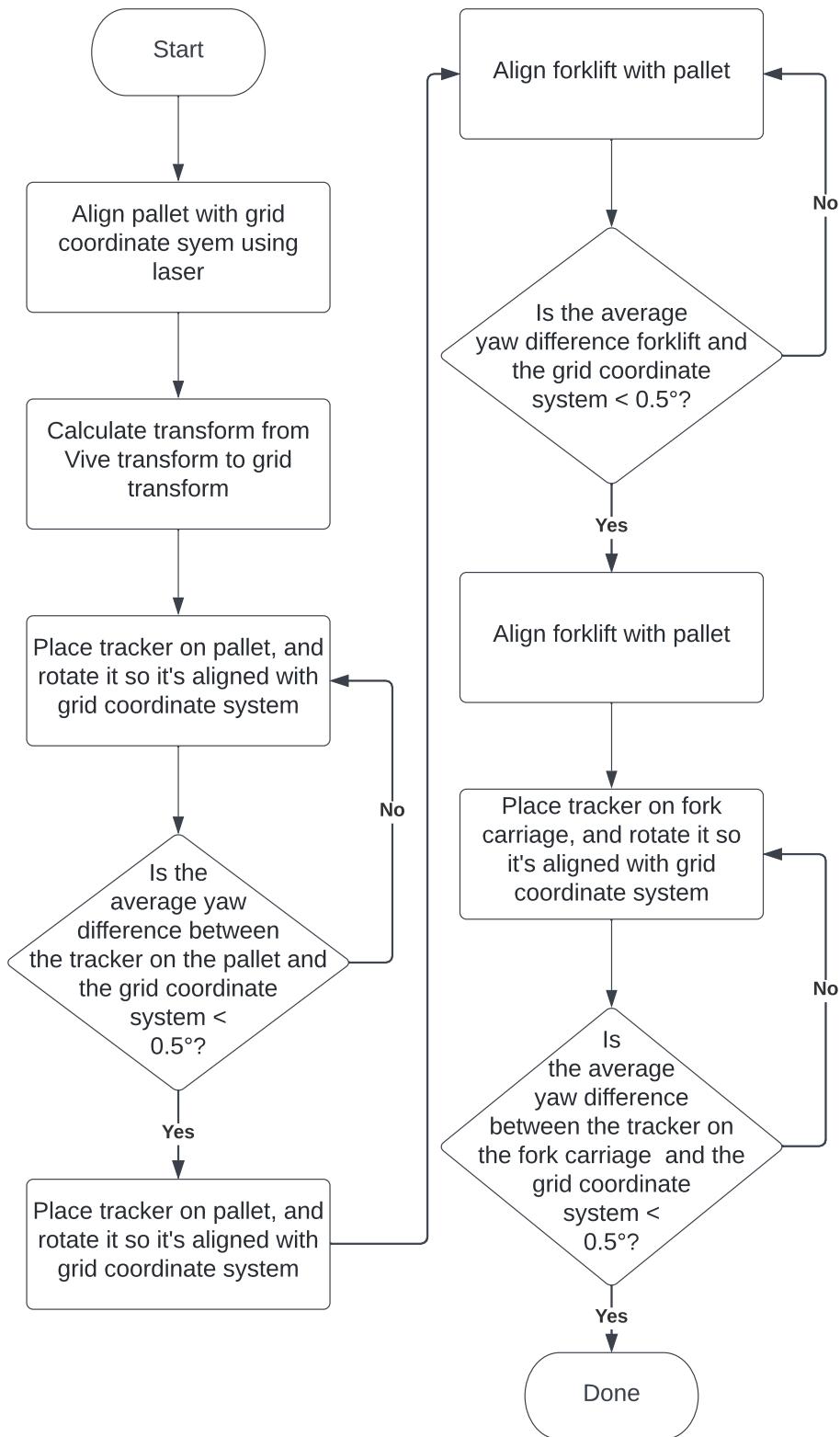


Figure 21: Overview of the procedure for calibrating the proposed test infrastructure.

approach is that there may be inter-dependency between parameters, so although it's feasible to test individual parameters in isolation, the combined performance of the system may not be as optimal as if the parameters are tuned together. A common approach

in the domain of machine learning when tuning the parameters that influence the neural network's training is to use an optimization algorithm to automatically sample, test, and find the best combination of parameters [113], [114]. A similar approach was used to test different parameters which would influence the engagement process to find an optimal combination of values.

The parameters chosen for optimization were:

- The position of the PDS (four possible positions on the APES)
- The orientation of the PDS
- The relative x and y position of the goal-point the APES will initially navigate to (relative to the pallets pose)

Although there were significantly more parameters that could be used for optimization, the more parameters added, the longer it would have taken for the optimization algorithm to converge (find a stable optima). The parameters were chosen based on which would most likely impact the engagement performance the trials the most significantly. It may be an interesting direction for future research to explore additional parameters for optimization. For each iteration of the tuning algorithm, a combination of values for these four parameters was specified. To adjust the position and orientation of the PDS, the URDF file of the APES was to be modified before each engagement trial. This involved reading the master copy of the URDF file, and modifying the appropriate values before saving a copy to a temporary location. The relative x and y position of the APES initial goal-point were represented as a ROS parameter (overriding the default value was sufficient to modify these parameters). A metric derived from the engagement performance was used as the feedback metric during tuning. Given X_e , Y_e , X_e , κ_e are the relative distances and yaw from the pallet to the tines of the forklift (i.e., the engagement error), and, X_l , and Y_l , κ_l were the range of possible acceptable values, the engagement error E was given as:

$$E = \frac{X_r}{X_l} * x_w + \frac{Y_r}{Y_l} * y_w + \frac{Z_r}{Z_l} * z_w + \frac{\kappa_r}{\kappa_l} * \kappa_w$$

The weighting between each engagement error could be adjusted to prioritize different aspects of the engagement error, for example to minimize engagement error for yaw (κ_e). However, for this set of experiments, they were all set to one.

A common algorithm for parameter tuning is Bayesian optimization. Bayesian optimization works by trying to model processes through building a probabilistic model based on Bayes theorem [115]. This attempts to both search effectively for parameters and ultimately find the optimal solution. Bayesian optimization is generally divided into two stages. First, the search space (i.e., the range of possible parameter values) is randomly sampled. Then based on the information from these initially sampled parameters, values likely to contain the optima are sampled. Because Bayesian optimization is able to conduct an efficient parameter search with no prior knowledge, it is often used to optimize complex and computationally expensive processes.

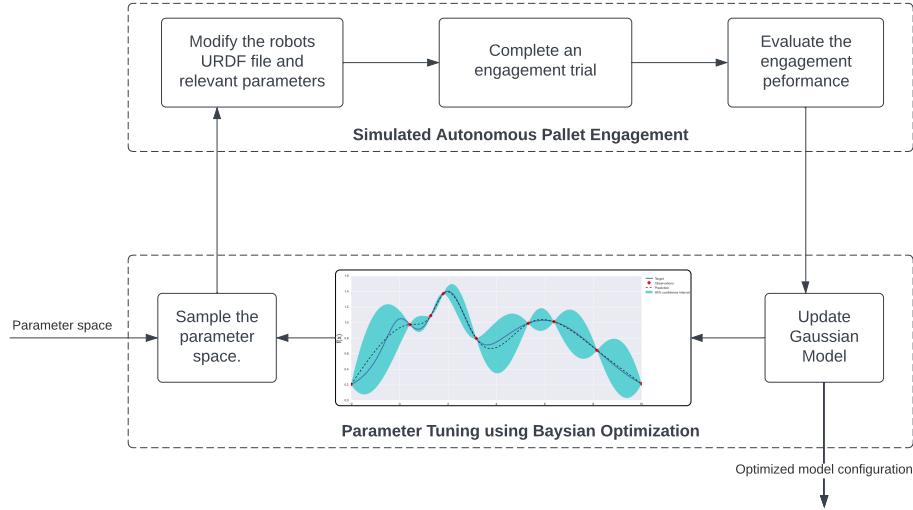


Figure 22: Overview of the procedure used to tune selected parameters of the APES.

The developed process used for parameter tuning is shown in Figure 22. The fmfn library provided the implementation of the Bayesian optimization algorithm [116].

5

Evaluation

The developed system will be analysed and held up against the requirements, and overall goals of the projects to allow conclusions to be formed about the suitability of the developed system. A number of experiments were conducted to form these conclusions. Sections 5.1 outlines the design of each experiment conducted and Section 5.2 provides the results collected from these experiments.

5.1 Experimental Design

This section will give an overview of the experiments conducted during the project, the parameters investigated and the experimental procedure. The experiments conducted can be broken into two major categories; “Simulation Testing” and “Real-world Testing”. Within these categories, there are some similar experiments conducted.

5.1.1 Simulation Testing

Deterministic:

This experiment will evaluate if the simulation framework is deterministic and if not, to what extent the output varies between trials with the same inputs. Gazebo [87] did not provide a method for setting the simulation’s random seed when it was being used with ROS. So although the simulated environment was not deterministic, the APES and experiment software (e.g., the software randomly determining the forklifts starting pose) were created to be deterministic. To test if the developed software was deterministic, thirty trials were run in Gazebo, where the random seed was iterate once every three trials and provided the APES, and experiment software. Each trial that had the same random seed was then grouped, and each group’s standard deviation (std.) was calculated (i.e., three trials per group), across the engagement metrics. Table 11 summarizes the median intragroup std. across all the engagement performance metrics.

These results show that the developed system is unable to be labelled deterministic, owing to the variation between the engagement metrics across different runs of the same random seed. This variance was significant for some engagement metrics, such as X_e or the heading error FF. However, the variance was extremely small for other metrics such as Z_e . When considering the motive behind the requirement for a deterministic system (i.e., to aid in debugging and trial reproducibility), the variance across trials with the same random seed appears adequate.

	Duration (s)	X_e (mm)	Y_e (mm)	Y_e (mm)	κ_e (°)	CT error PUF (mm)	Heading error PUF (°)	CT error FF (mm)	Heading error FF (°)
Median of intragroup std.	14.17	26.30	5.34	0.09	0.17	0.68	1.36	0.40	2.43

Table 11: Deterministic evaluation of the developed APES in simulation.

Simulated Engagement Testing One:

This experiment sought to test if the developed APES met the core requirements of the standard ASTM F3499 [24]. The experiment involves 29 repeated engagement trials

(see Section 3.1 for explanation of selection of the number of trials) of a pallet from a fixed starting location. The forklifts starting position was offset –8 m, 1.6 m along the y-axis, with a 90° (rel. the pallets CS). The exact pallet’s location was provided to the APES at the start of each trial. A random seed (corresponding to the trial number), was used for each trial. Several metrics were collected after each trial, including:

- Trial outcome.
- Engagement duration.
- Engagement error (X_e , Y_e , Z_e , κ_e).
- Path following mean, max and standard deviation cross-track track error.
- Path following mean, max and standard deviation heading error.

The engagement error and trial outcome was then used to assess if each trial, and therefore, the APES met the requirements of the ASTM F3499 standard [24]. The engagement error was calculated in each trial by querying the poses of the forklift and pallet from Gazebo. The standard ASTM F3499 [24] used the accuracy of the test infrastructure to determine the number of trials. This was because the number of trials was based on the desired confidence in success and in the test infrastructure (e.g., a lower number of trials were needed if less confidence in the test infrastructure was desired). Although reading the engagement error directly from simulation can be considered to have negligible measurement error, the 5 % measurement error was used in later testing and therefore, 29 trials (i.e., based on 5 % measurement error) were used for all simulation experiments.

The trial outcome could be one of either:

- Success
- Contact with pallet
- Contact with racking
- Timeout
- Software failure
- Infeasible engagement

In order for a trial to be deemed a success, the state machine must have been executed to completion, and returned a “finished” outcome. However, if other conditions were met during the engagement trial, an alternative outcome was logged, and the trial was stopped. During simulated trials, the total magnitude of the linear and angular velocity of the pallet and the nearby racking was monitored. This was done to check that the forklift did not push the pallet or racking at any point. If the pallet or racking was pushed, this was considered a failure, and the trial outcome would be either “contact with pallet” or “contact with racking” based on the object in question. During the project 0.001 m/s and 0.001 °/s were used to indicate if the movement was significant enough to be considered contact.

The “timeout” outcome occurred when the state machine had not reached an outcome within a set period (specifically within 10 minutes) or no other outcome conditions had been met. During a trial, ROS packages could log messages, where each log message

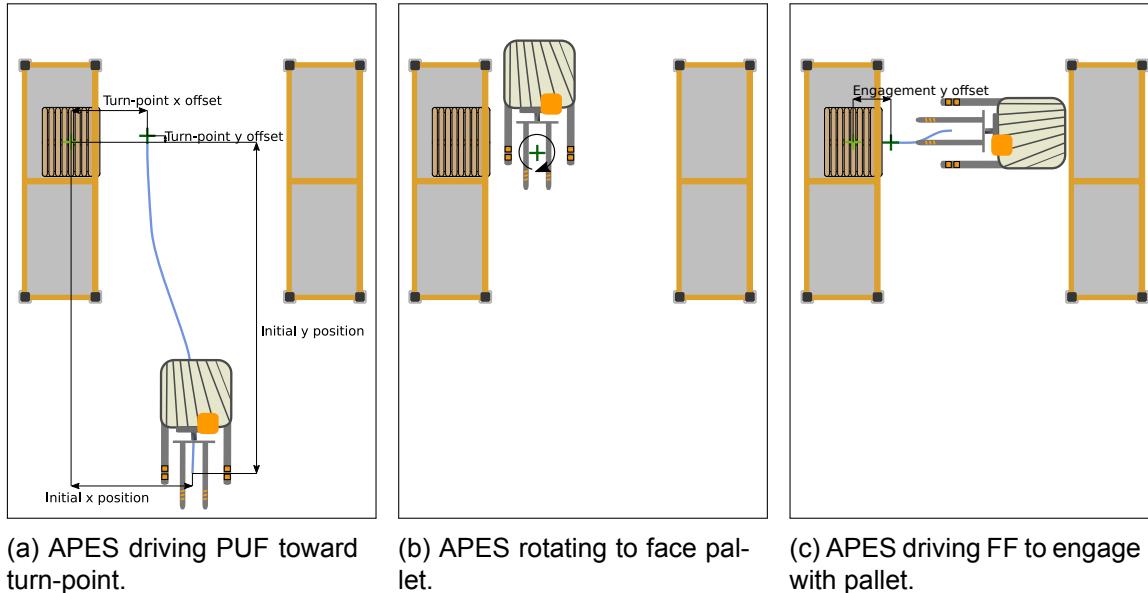


Figure 23: Pallet engagement process.

had five different verbosity levels ranging from “Debug” to “Fatal”. Usually, if an external ROS package (i.e., a publicly available package not explicitly created for this project) encountered an error, or if it unexpectedly stopped, this resulted in a message being logged with the verbosity of “Error” or higher. For ROS packages creating during this project, the same principle was used. For example, if the path planning algorithm could not find a feasible path (e.g., if the path to the pallet was obstructed), the path planning package raised an error message. If at any time during a engagement trial, a message was logged with verbosity of “Error” or higher, this was classified as a software failure and the trial outcome a “software failure”. Before the APES drives FF towards the pallet, it had a state to check if the engagement was feasible (based on the relative yaw from the forklift to the pallets pose); if this state determined the engagement was unfeasible, the engagement trial terminated, and the outcome labelled “infeasible engagement”.

Simulated Engagement Testing Two:

This experiment was similar to the experimental design of the previous section. However, instead of the forklift starting from a fixed pose and being provided with the correct pallet pose, the forklift was started within a range of starting poses. Additionally, the pallet pose provided had artificial error added to it, and therefore, the virtual pallet detection sensor had to be used to determine the correct pallet pose.

The starting location of the forklift was based on the starting position of the previous test ($Y = -8\text{ m}$, $X = 1.6\text{ m}$, rel. to pallet CS). However, there was some random variation added to the pose. The variation was along the forklifts X and Y axis and relative yaw (relative to the localization coordinate system). The variations values used during this experiment were: $X_a = Y_a = \pm 4\text{ m}$, $\kappa_a = \pm 89.95^\circ$. The pallet engagement algorithm used the orientation of the forklift (relative to the pallet CS) to determine which direction the forklift would travel down the aisle. To avoid events where the forklift would be parallel with the pallet, and the ambiguity this could introduce, the yaw variation added (κ) was slightly less than 90° .

When the APES was within detection distance of the pallet, it was provided with an updated pose for the pallet. However, artificial error was added to the pose reported from

the VPDS, to mimic a real PDS. The requirements for when the pallet was within detection distance of the PDS are described in Section 4.2.3. Specifically, four parameters govern when the pallet is in detection distance:

- PDS max measuring distance
- PDS vertical FOV
- PDS horizontal FOV
- PDS detection κ

For this experiment the maximum measuring distance was set to be 10 m, and the PDS vertical, horizontal and detection κ was set to 45°. Additionally, the time delay between when the VPDS would report pallet poses was set to 4 seconds. These values were roughly based on the TOF camera's produced by IFM (IFM, Essen, Germany). This particular type of PDS was selected as a blueprint as it has been used in previous studies for autonomous pallet engagement [50]. Once the pallet was within detection range, it reported the pallet's pose. The ground truth pallet pose was read from a configuration file, then the ROS transform library [117], the pose relative to the PDS pose was calculated. The pose then had artificial error added to the X, Y, and Z axis and to the relative yaw reported (rel. to PDS CS). The range of artificial error was determined by randomly sampling from the following ranges: $X_a = Y_a = Z_a = \pm 10$ mm, $\kappa_a = \pm 5^\circ$.

Tuned APES versus Baseline APES:

This experiment compared an APES which had certain parameters automatically tuned against an APES with a hand-picked configuration. The tuned APES would be subject to the same testing conditions as Section 5.1.1. The Bayesian t-test [118] was used to compare configurations to evaluate if the tuned APES configuration performed differently to the hand-picked or “baseline” configuration.

The APES was tuned across 90 trials, using 90 different configurations of parameters. The parameters selected for tuning were:

- The PDS position.
- The PDS yaw change.
- The turn point x offset (rel. to pallet CS, see Figure 23)
- The turn point x offset (rel. to pallet CS, see Figure 23)

There were three possible positions for the PDS (see Figure 24). These positions were selected as being likely, and feasible places to mount a PDS. The PDS orientation yaw change refers to an additional alteration of the PDS yaw from the orientations shown in Figure 24.

Out of the 90 trials, 25 of these trials comprised the “random sampling” segment of the Bayesian optimization algorithm, and 65 of the trials were used to hone in on the optima. This process was run on NeSI’s high-performance computing cluster. It was possible to run this optimization process locally, however, it took significantly longer and prohibited normal functionality, as the simulation would use the majority of the computing resources.

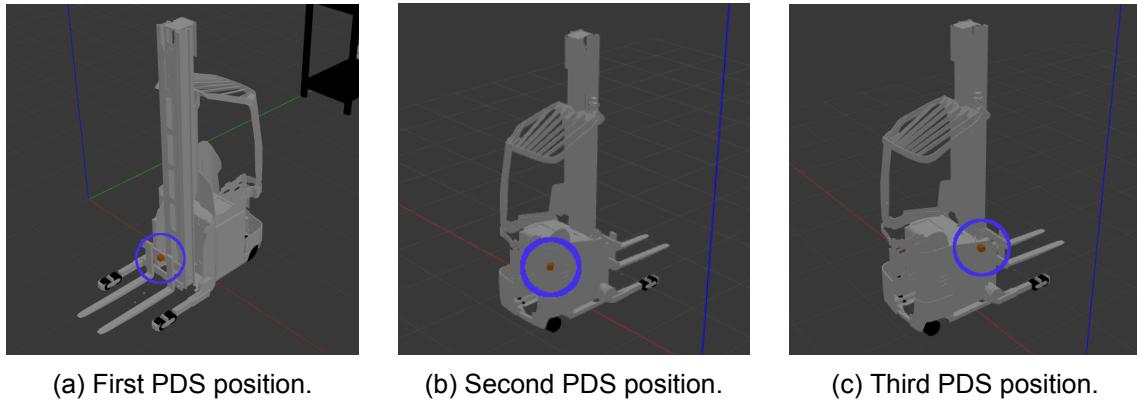


Figure 24: The PDS is shown in orange.

5.1.2 Real-world Testing

Vive Performance Evaluation:

The Vive was evaluated to check if it was sufficient to allow the test infrastructure to fulfil the measurement precision and bias set by ASTM F3499 [24]. By default, the standard does not allow the precision and bias of the test infrastructure to exceed 5 % of the maximum error margins of the task performance margins of each axis. Table 12 summarizes the calculated measurement error for each axis of the Vive. The Vive was calibrated using a coordinate system marked on the ground using a laser level, tripod and rule. To evaluate the measurement accuracy of the Vive, points were measured from the grid coordinate system and the position reported from the Vive was compared to the ground-truth position (i.e., found by measuring from the grid coordinate system). Two sets of points were recorded: points which had been used to initially calibrate the Vive and points that were separate from the Vive's calibration.

Because the two Vive's were fixed on mounts, their only degree of movement was rotation around the Z axes (i.e., yaw). Therefore, the calibration error would only impact the measurements of the engagement error of the X, Y and κ axes, but not the Z axes. When calibrating the Vive, there were three sources of possible yaw error (tracker and pallet, pallet and forklift, and tracker and forklift). The maximum these errors were allowed to be individually was 0.5°. This standard was maintained by: taking the average of 100 points between the trackers to ensure the relative yaw was within 0.5 % of the desired value, and by using a laser measure (Leica, Wetzla, Germany) to find the relative distance, and therefore, angle between the pallet and forklift. Using this process the maximum possible overall calibration error reported within the engagement error was 1.5°.

When reaching to engage the pallet, the controller actuating the reach mechanism had an allowable error band, i.e., where if the position of the actuator was within the band for a set amount, this constituted reaching the set-point. Once the controller registered it had met this requirement, the state-machine would produce a “finished” output and the trial would end. The worst-case-scenario for miscalibration occurred when the reach actuator was at the extreme of its allowable error band (range of non-zero error values which are programmed to produce no actuator output to avoid jitter). As the error band for reach was ± 80 mm, this could result in up to 2.1 mm of calibration along the y-axes, and 0.05 mm of error along the x-axes (based on the geometry of transforms).

The Vive's measurement error was quantified using a similar method as Niehorster et al. [94], using a measurement grid for calibration and for quantifying measurement error. Six

points were used to ascertain the measurement error, the four points used for calibration and two additional points.

To provider reference to the engagement errors reported in the following sections, Figure 25, shows the forklift in three positions relative to the pallet. For each position, the engagement error measured using the Vive is provided.



(a) $X_e = 0.4 \text{ m}$, $Y_e = 0.0 \text{ m}$, $Z_e = 0 \text{ m}$, and $\kappa_e = 0.0^\circ$. (b) $X_e = 0.4 \text{ m}$, $Y_e = 0.5 \text{ m}$, $Z_e = 0 \text{ m}$, and $\kappa_e = 0.0^\circ$. (c) $X_e = 0.4 \text{ m}$, $Y_e = 0.0 \text{ m}$, $Z_e = 0 \text{ m}$, and $\kappa_e = 5^\circ$.

Figure 25: Illustration of different error values reported from the Vive.

Real World Engagement Testing One and Two:

These experiments were similar to the experimental setup performed in simulation to evaluate if the developed APES met the requirements set out in the standard ASTM F3499 [24], except they used real hardware. To allow the APES to be tested safety, it was not tested within racking, but in an open area with the pallet to be engaged with being placed on a stack of two other pallets. Cardboard boxes were placed inline, and opposite the pallet and were used to represent the aisle of racking. This experimental setup is shown in Figure 26. The primary differences between simulation and real-world experimentation was how each trial's outcome was recorded, how the engagement metrics were calculated, and how the experiment was set up.

The total number of possible trial outcomes possible was consistent between simulation and real-world testing. These outcomes were:

- Success
- Contact with a pallet
- Contact with racking
- Timeout
- Software failure
- Infeasible engagement

The determination of all these outcomes followed the same methodology, as was described above in Section 5.1.1, except for the outcomes: “Contact with a pallet” and “Contact with racking”. If contact was observable during a trial (defined as observable movement), then this trial would be noted and its outcome modified based on if the contact occurred with a pallet or the cardboard box racking.



Figure 26: Real-world experimental setup.

Another primary difference was how the engagement metrics were calculated. The engagement errors, i.e., X_e , Y_e , Z_e and, κ_e were calculated using the test infrastructure described in Section 4.3.5. Determination of the path following metrics was done based on the real-world localization data. Additionally, the Gazebo simulation was run in parallel to the real-world experimentation and would simulate the outcomes of the actuator commands provided from the APES. This was used to compare the engagement errors calculated from the test infrastructure against that of the simulation (see Figure 28).

5.2 Results

5.3 Vive Validation

	X (mm)	Y (mm)	Z (mm)	κ ($^{\circ}$)
Task performance margins	120.00	101.50	30.50	7.50
Max test infrastructure precision and bias	6.00	5.08	1.53	n/a ^a
RMSE (total)	1.10	2.96	1.97	0.95
RMSE (calibration points)	0.64	1.91	0.24	0.57
RMSE (other points)	1.68	4.35	3.40	0.83

Table 12: Vive evaluation results. ^a There was no set measurement error requirement for yaw using ASTM F3499.

Several conclusions can be drawn from the Vive evaluation results. The results found are broadly consistent with those reported in other studies (see Table 12). If the aggregated results are used, the Vive's measurement error was within the required 5 % measurement

error threshold. However, if we look only at the measurement error of the points which did not form part of the calibration, the reported measurement error for Z, does not meet the 5 % requirement. A possible reason for the difference is that the transform from the Vive's coordinate system into the grid/pallet coordinate system contained some calibration error.

5.4 Simulation Testing

Engagement Experiment One:

As described above, this experiment tested the suitability of the development with respect to standard ASTM F3499 [24]. To validate this, a summary of the engagement performance the APES achieved is in Table 13.

	Duration (s)	X_e (mm)	Y_e (mm)	Z_e (mm)	κ_e (°)	CT. error PUF (mm)	HD. error PUF (°)	CT. error FF (mm)	HD. error FF (°)
Mean	75.94	23.91	- 33.43	0.82	-3.3	42.6	0.20	5.15	0.69
std.	14.27	10.05	26.59	0.21	1.78	0.75	0.10	1.21	0.26
TPMs	n/a	120	101.5	30.5	7.5	n/a	n/a	n/a	n/a

Successful trials = 29/29, trials where the engagement error is within the TPM's=29/29

Table 13: Engagement performance summary for engagement test one in simulation. The X_e , and Z_e are the adjusted errors.

The results show that the APES was able to complete the 29 engagements successfully and that all the engagements were within the prescribed task performance requirements.

An interesting feature of the results was the significant difference in cross tracking and heading error between driving PUF and FF. The cross-track error was significantly higher when driving PUF. However, the HD error was significantly lower when driving FF. One explanation could be that when the APES is driving FF, its maximum allowed speed was reduced. This in turn, likely improved the cross-track error as the path following algorithm had more time to respond. This, however, raises the question of why the heading error was significantly worse for driving FF. The likely explanation is that although the path following algorithm had more time to respond when driving FF, the average curvature change for the FF segment was significantly higher than the PUF. This, therefore, would put more stress on the path following algorithm, resulting in a higher heading error. Although further reducing the forklifts speed when driving FF's may improve the heading error, the current mean heading error FF is still relatively low (i.e., 0.69°).

Engagement Experiment Two:

This test explored the suitability of the developed system in a more realistic scenario than that tested above. For each trial, the APES's starting location varied, and it was provided with an inaccurate pallet pose, which was updated by a virtual pallet detection system (VPDS).

In contrast to the results from the first engagement experiment, only 21 engagement trials were successful, with all the unsuccessful trials being caused by contact with a pallet.

	Duration (s)	X_e (mm)	Y_e (mm)	Z_e (mm)	κ_e (°)	CT. error PUF (mm)	HD. error PUF (°)	CT. error FF (mm)	HD. error FF (°)
Mean	101.18	25.49	- 29.77	1.01	4.3	46.64	0.49	5.833	39.01
std.	37.83	5.31	19.01	0.35	3.63	18.42	1.31	5.98	27.76
TPM's	n/a	120	101.5	30.5	7.5°	n/a	n/a	n/a	n/a
Successful trials	n/a	21/29 ^a	29/29	29/29	29/29	n/a	n/a	n/a	n/a

Successful trials = 21/29^a, trials where the engagement error is within the TPM=21/29.

Table 14: Engagement performance summary for engagement experiment two in simulation. ^a All the unsuccessful trials were a result of contact with a pallet.

Generally, this contact with the pallet would occur when the APES was driving FF and attempting to correct misalignment, or reaching out when not sufficiently aligned. Although the APES had a state before driving FF to check the engagement viability (based on the current position of the forklift and pallet), this state was intended to stop obviously unfeasible engagements (e.g., in the case of a software bug in real-world testing), so allowed engagement attempts with a good possibility of success. The observed decrease in engagement success rate can likely be attributed to a combination of: the updated (i.e., from the PDS) pallet pose not being received soon enough, or the path planning and following algorithms not reacting sufficiently when the updated pallet pose was received.

Additionally, there was a large difference (specifically a change from 75.74s to 101.18s) in the duration it took to complete an engagement. The difference in starting poses most likely accounted for this difference. For the first engagement experiment, the forklift was started in essentially the most advantageous position for engagement (i.e, facing directly down the aisle). Therefore, for the second engagement experiment where the forklift was started at random poses and orientations (within a predefined range), this would result in a greater amount of distance having to be travelled before it reached the pallet, therefore, resulting in a greater duration to engage with the pallet.

Tuned APES versus Baseline APES:

This experiment tested if the method of tuning the APES's parameters would produce a better result (as judged against the requirements of the project), than without tuning. Two sets of results are shown to evaluate this: a statistical comparison of the engagement performance achieved during the second engagement test (see Table 15) and the selected parameters for the tuned system versus the holistically chosen system (see Table 16).

Several engagement metrics were statistically different with the two configurations: trial duration, and all the path following metrics. The mean duration was observed to decrease when using the tuned system parameters. One explanation could be that the tuned parameters allowed the forklift to be in better alignment with the pallet after it had turned on the spot (i.e., in anticipation of driving FF). This improved alignment would likely result in a quicker engagement as the pure pursuit controller would modulate the forklift speed if it was not correctly following the planned path. The tuned configuration also showed an improvement of the path following cross track and heading error when

	Duration (s)	X_e (mm)	Y_e (mm)	Z_e (mm)	κ_e (°)	CT. error PUF (mm)	HD. error PUF (°)	CT. error FF (mm)	HD. error FF (°)
Simulated Mean ^a	119.88	21.13	42.20	1.00	4.94	53.90	10.47	5.55	38.67
Tuned Mean ^a	75.87	23.63	38.04	0.88	3.05	42.17	0.06	4.72	7.00
Mean difference	44.01	-2.50	4.17	0.12	0.41	11.72	10.41	0.83	31.66
HDI (%)	100.0	74.8	89.5	68.4	85.8	110.0	99.9	99.7	100.0

Table 15: Statistical analysis of tuned APES parameters versus heuristically tuned parameters. Statistically significant results (if the HDI > 95%) are highlighted. CT error refers to the cross track error and HD error refers to the heading error. ^a These were not the arithmetic means of the data (as reported previously), but the pooled empirical mean calculated during the Bayesian t-test.

	PDS posi- tion	PDS angle °	Turn point x offset mm	Turn point y offset mm
Manually selected parameters	0	0	0.9	0.0
Tuned parameters	2	-5.64	1.08	0.16

Table 16: Comparison of the parameters selected manually (i.e., “guess-and-check”) and through the Bayesian optimization method.

navigating PUF and FF. This improvement may have been caused by a better selection of the turn point position and earlier detection of the pallet, which in turn may have reduced the amount of correction needed during the engagement.

5.4.1 Real-world Testing

PID Performance Results:

This experiment set out to determine if the closed-loop PID controllers meet the requirements set out in Section 2. The PID controllers were presented with a step position input to generate the results. The position input was based on the range of the actuator. For example, for the host controller, which has a range of positions between 264 mm and 2520 mm, the position input stepped between 750 mm and 2250 mm.

The timestamps, position set-point and actual value were recorded for several input cycles, and the PID performance calculated using MATLAB’s [100], Control System Toolbox [119]. The following performance metrics were used:

- Rise time (seconds)
- 2% Settling time (seconds)
- Overshoot (%)

	Overshoot		Rise time		Settling time	
	Req.	Actual	Req.	Actual	Req.	Actual
Hoist	10 mm	2.6 mm	2s	1.48s	5s	2.67s
Reach	30 mm	0.40 mm	2s	1.71s	5s	3.52s
Side shift	30 mm	0.0 mm	2s	1.22s	5s	3.44s
Tilt	1.0°	0.67°	2s	1.82s	3s	2.63s

Table 17: Comparison of PID performance set out in the requirements versus, the achieved PID performance.

Table 17 outlines the time-domain response characteristics of the tuned PID controllers. All the characteristics were within the prescribed requirements, which was expected. Following the process set out in Figure 15, if a controller after tuning did not meet the preset requirements, their PID gains would be adjusted, and in certain instances, the requirements would be adjusted if they were found to be unreasonable.

Engagement Experiment One:

This was the first real-world test conducted with the APES and set out to test if the

	Duration (s)	X_e (mm)	Y_e (mm)	Z_e (mm)	κ_e (°)	CT. error PUF (mm)	HD. error PUF (°)	CT. er- ror FF (mm)	HD. error FF (°)
Mean	52.15	-27.34	14.78	11.69	0.19	41.108	-0.55	-12.22	0.90
std.	7.23	8.10	17.04	10.86	4.08	1.97	0.07	6.01	0.78
TPM's	n/a	120	101.5	30.5	7.5°	n/a	n/a	n/a	n/a

Successful trials = 29/29, trials within TPM's = 26/29^a.

Table 18: Engagement performance of real-world testing from a constant starting pose. ^a All the trials that exceeded the prescribed TPM's were a result of a κ_e of greater magnitude than 7.5°.

developed system met the core requirements of the standard ASTM-F3499 [24]. This test resulted in the APES completing 29/29 successful engagements, with the engagement performance summarized in Table 18. However, three of the trials exceeded the prescribed TPM's for κ_e . This is interesting as the path following algorithm is designed around meeting the TPM. For the path following algorithm to finish, the following condition must have been met (see Section 4.3.3):

$$Yk_{threshold} > Y_{dis\ to\ goal} + forklength * \sin(\kappa)$$

Where the $Yk_{threshold} = 0.1$, and the $forklength = 0.914$. Therefore, even if the distance to the goal along the Y axis was zero, the maximum allowable yaw error to the goal point was 6.28° . As this is lower than the TPM of 7.5° , it stands to reason that in the instances where the yaw TPM was exceeded this may have been caused by calibration error introduced by the test infrastructure.

Engagement Experiment Two:

Because the APES will not always begin engagement while starting at the same ideal starting location, it was necessary to test its performance with a variety of starting locations. The first set of results presented is the engagement performance achieved when starting the APES from a variety of starting locations (see Table 19). The engagement performance achieved is similar with that achieved during the first real-world engagement experiment. All the means of the engagement metrics were within one standard deviation. This, therefore, indicates that the forklift was capable of starting at a variety of starting poses without a significant change in performance.

The second set of results show the relationship between the path following performance and the starting pose of the forklift. When the forklift is in driving PUF there appears to be little relationship between the starting pose and the resulting cross track and heading error. One reason for this may be due to the continuous curvature nature of the paths generated, i.e., even when the forklift is started at a moderate angle to the intended direction, the generated path is smooth enough to have negligible impact on the mean cross track and heading error. When observing the data when the forklift is driving FF, it appears that paths with a smaller cross track error are grouped, with the paths falling on either side having a larger cross track error. These groupings may correlate to the paths which start driving FF and need minimal adjustment for alignment, compared to the paths which begin with the forklift misaligned with the pallet.

	Duration (s)	X_e (mm)	Y_e (mm)	Z_e (mm)	κ_e ($^\circ$)	CT. error PUF (mm)	HD. error PUF (mm)	CT. error FF (mm)	HD. error FF (mm)
Mean	53.32	-8.85	-1.06	16.40	1.50	48.77	-0.22	-	-0.26
								12.07	
std.	4.92	43.30	17.47	16.65	7.76	9.91	1.08	4.92	0.92
TPM's	n/a	120	101.5	30.5	7.5°	n/a	n/a	n/a	n/a

Successful trials = 19/20^a, trials within TPMs = 18/20 ^b.

Table 19: Engagement performance of real-world testing from a moving starting location.^a The single failure was caused by contact with the pallet.^b One of the trial that exceeded the TPM's exceeded the prescribed κ_e , and the other exceeded both the κ_e , and Z_e .

5.4.2 Simulation Versus Real-World Results

Because it was possible to run the Gazebo simulation during a real-world experiment, there would be two sets of engagement errors recorded for each real-world trial; the error's reported from the Vive test infrastructure and the errors reported by the simulation. Therefore, to aid in the comparison between the real-world result and the simulation res-

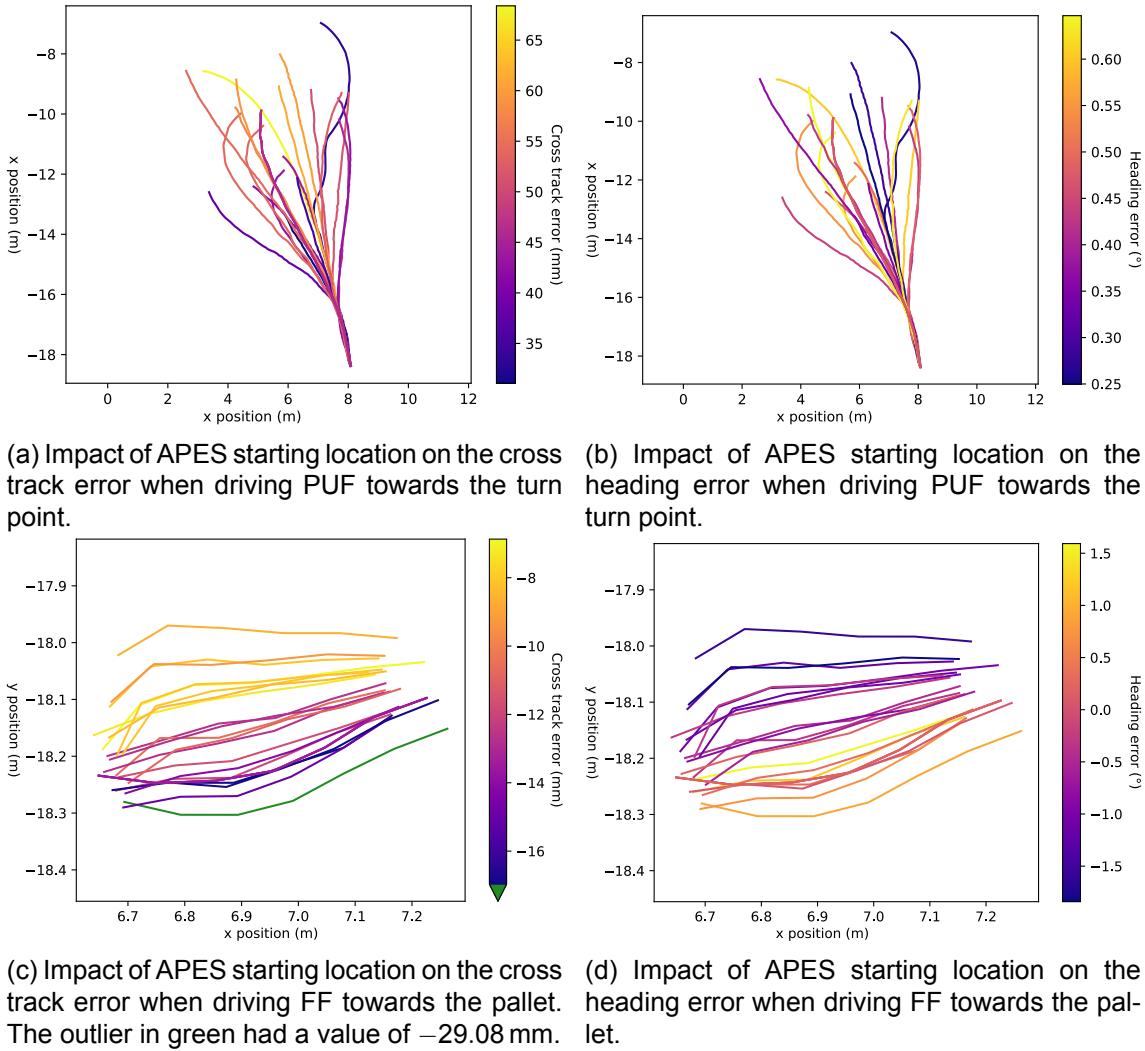


Figure 27: Impact on altering the starting location of the APES on the path following characteristics.

ults, the errors reported for all the trials in the second experiment were plotted (see Figure 28). These plots show that there is a degree of systematic difference between the error reported along the X, Y, and Z axis. However, the yaw reported by both the Vive and the simulation appear similar. A primary source of systematic error may be caused by miscalculation of the transforms used to calculate the engagement error (see Section 4.3.5).

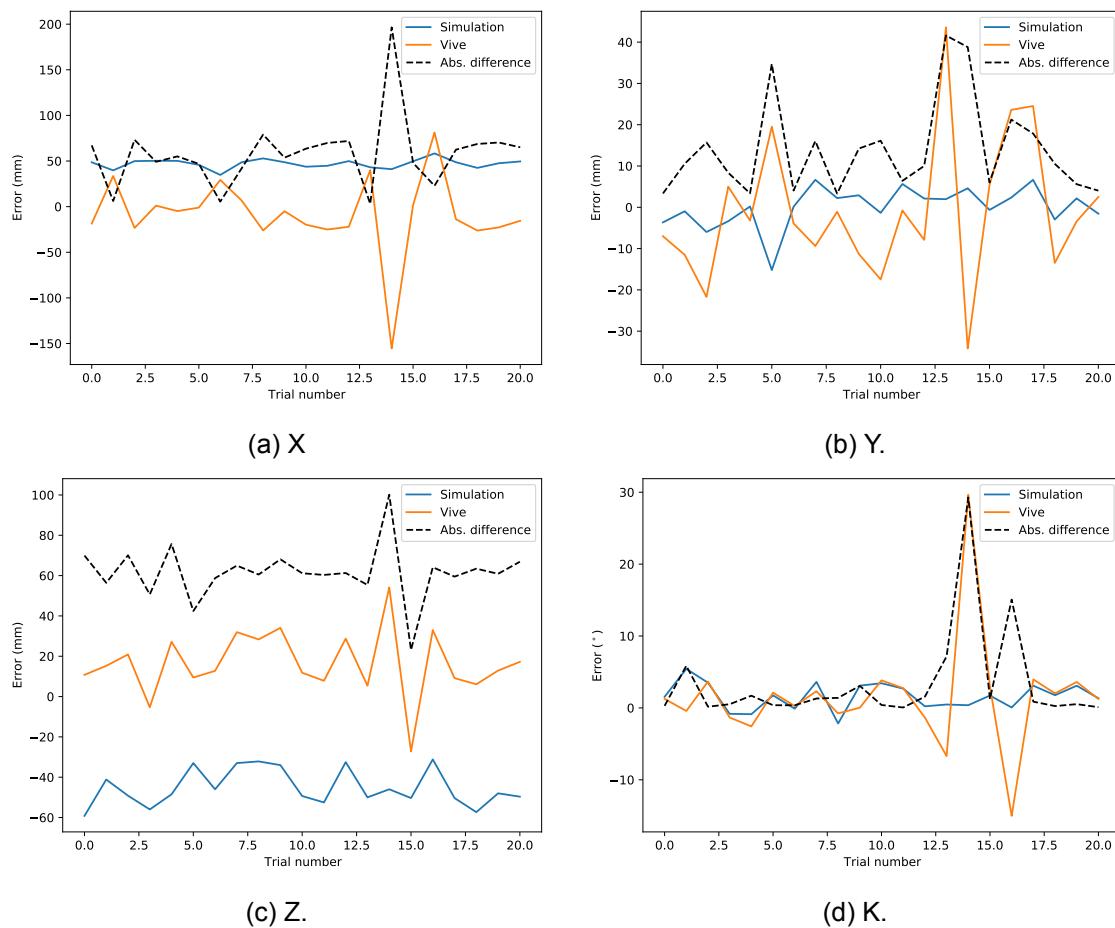


Figure 28: Difference in engagement error reported by the Vive and Gazebo.

5.5 Summary

In developing this project, the following requirements were created:

- *Simulation*: To help develop the APES, there should be a deterministic simulation which is capable of modifying experimental variables across repeated experiments. The simulation results should be able to be compared with real-world test results.
- *Obstacle Avoidance*: The APES must be capable of generating obstacle-free paths and engaging with the pallet while in aisles as narrow as 3.2 m.
- *Engagement Performance*: The APES should be capable of engaging with the pallet with the following maximum error margins: X<120 mm, Y<101.5 mm, Z<30.5 mm, k<7.5°.
- *Reliability and Robustness*: The APES must complete 29 repeated engagement based on the failure conditions of standard ASTM F3499 [24].

Because there was a significant degree of dependency between these requirements, they were not tested in isolation. During the project, the simulation environment using Gazebo was used to test and develop an APES. Gazebo could not be configured to be deterministic. However, the APES and experimental software were designed to be deterministic, which for the most part allowed for trials to be reproducible (i.e., for debugging and comparing between configurations) when repeated with the same random seed. Through the use of the simulation, it was possible to compare the APES with different configurations across several trials. This principle was used to automatically tune several parameters relating to the APES, using the Bayesian Optimization Algorithm. The resulting tuned configuration converged to a set of parameters not dissimilar to that chosen manually. The tuned configuration had a different PDS position and angle, however, a relatively similar position for the turn-point. The tuned configuration of the APES performed similarly to its non-tuned counterpart, however, it achieved a slightly shorter trial duration, cross-track error (PUF), and heading error (PUF, FF), and a slightly worse engagement yaw error.

The developed APES was capable of engaging within 3.2 m aisles both in simulation and in real-world testing. During the experiments conducted both in simulation and in the real world, there was no contact with the aisles. During the experiments conducted within the simulated environment, the APES, was able to complete 29/29 engagements within the prescribed task performance margins when it was started from a fixed start location. When the same test was conducted with the real equipment, all the engagement trials were successful. However, three engagement trials exceed the yaw engagement requirement of 7.5°. The developed APES was tested in simulation using a virtual pallet engagement system. This experiment showed that the developed system was capable of reliably (i.e., 29/29 trials) engaging with pallets with inaccurate initial pose information which was later updated by an imperfect PDS. When the APES was tested in the real-world and was started from a variety of different poses, it was able to successfully engage with the pallet 20/20 times, while being within the TPM 18/20 times.

6

Conclusions

6.1 Conclusions

This thesis describes the design, implementation and experimental testing of an automated pallet engagement system (APES). The developed system was designed to use a pallet detection sensor that would aid in its flexibility, and ability to work alongside driver-operated forklifts. The motivation for this work is driven by a predicted increase in demand for autonomously guided vehicles and the need to develop such vehicles capable of operating alongside driver-operated vehicles. This project focused on engagement of pallets within a confined space such as a narrow aisle. In the pursuit of achieving this goal, the following project requirements were used:

- *Simulation*: To help develop the APES, there should be a deterministic simulation which is capable of modifying experimental variables across repeated experiments. The simulation results should be able to be compared with real-world test results.
- *Obstacle Avoidance*: The APES must be capable of generating obstacle-free paths and engaging with the pallet whilst in aisles as narrow as 3.2 m.
- *Engagement Performance*: The APES should be capable of engaging with the pallet with the following maximum error margins: X<120 mm, Y<101.5 mm, Z<30.5 mm, k<7.5°.
- *Reliability and Robustness*: The APES must complete 29 repeated engagements based on the failure conditions of standard ASTM F3499 [24].

The simulation environment was used to offer an expedited process of testing the APES, and collecting results. Gazebo [87] was used for simulation. Although basic simulation of the forklift is trivial using off-the-shelf software, several steps were made to allow the simulation to easily and efficiently test and modify experimental variables. Software tooling was developed to allow the simulation to be safely started/stopped and experiment variables (such as the position of the PDS) to be modified. To allow the reproducibility of experiments, the developed APES was made to use a constant random seed, and the simulation and APES were run on a high-performance computer to allow for a large number of trials and parallel experiments. Because this infrastructure was developed, it was then possible to use an optimization algorithm (specifically Bayesian optimization [116]) to automatically test and optimize experimental parameters.

Because the focus of the project was on engagement within confined environments such as aisles, the APES needed to engage with pallets while in-between aisles as narrow as 3.2 m. There is a large amount of literature and work on path planning algorithms. However, because of the precise navigation requirements, a geometric method was selected for path planning, specifically η -spline interpolation. Geometric methods are a common method when planning paths for pallet engagement [22], [37], [48]. However, they lack the ability to consider obstacles (i.e., nearby racking), when planning paths. Therefore, the existing methods were built upon to create a custom path planner, which tests a set of paths by setting the η -spline parameters to randomly sampled values. The generated paths would then be tested for obstacle avoidance, curvature, and length to find an optimal path. The generated path would then be followed using the standard pure pursuit, which had been modified for use alongside a pallet engagement algorithm.

To test the developed system, the project took the lead from the standard ASTM F3499 [24] to help develop a test methodology. There were two fundamental tenants to the

test criteria; the task performance requirements and the repeatability criteria. The task performance requirements were a set of the minimum expected performance of the developed system. The reliability criteria were a minimum number of successful trials (specifically, 29), which had to be completed to confirm reliability confidence in both the developed APES and the test infrastructure used to evaluate it.

The first result presented was an evaluation of the Vive [91], which formed part of the test infrastructure. This test indicates that when the Vive was evaluated against poses which were used in its calibration it was capable of meeting the 5 % measurement error requirement. However, when additional points were analyzed, the Vive did not meet the 5 % measurement error requirement for the Z axis. The APES was then tested in simulation through two primary experiments; from a fixed starting pose with knowledge of the pallet pose and from a variety of starting poses and using a virtual pallet detection system and with imperfect knowledge of the pallet pose. These experiments showed that the developed system met the reliability requirement of 29 trials and completed all engagements within the task performance margins. There are several experimental variables controlling the behaviour of the APES, such as the position and orientation of the PDS. Initially, these parameters were chosen through a “guess-and-check” methodology. However, this process is time-intensive and unlikely to result in a globally optimal solution. Therefore, the APES in simulation was interfaced with a Bayesian optimization algorithm. This algorithm was able to select parameters relating to the engagement process, monitor the outcome, and find what is interpreted as the optimal combination of parameters. When the APES with the tuned parameters was compared against that selected manually, the results were similar. However, the tuned configuration improved the engagement duration and path following characteristics slightly.

Next, the APES was evaluated with real hardware. Two tests were conducted: engagement with a pallet from a fixed starting pose and from a variety of starting poses. When tested from a fixed pose, the APES successfully completed 29/29 trials, yet three of these trials exceeded the yaw task performance margin. When evaluated using a variety of starting locations, the APES completed 19/20 trials successfully and 18/20 trials were within the task performance margins.

The presented work extends the state of the literature in several ways; it presents an efficient method for selecting the parameters of an autonomously guided system, it provides a practical method of planning obstacle-free paths which have kinematically optimal properties, and it uses off-the-shelf hardware to test the developed system against a relevant industry standard.

6.2 Existing Work

The project builds on existing work by Crown. Communication between the RM6000 forklift was established through a bridge library and hardware provided by Crown. Crown also provided a skeleton Gazebo simulation of the forklift where had the fork actuators were simulated, however, the simulation of the steer/drive wheel for driving was added during this project. Existing work included a description of the forklift using the URDF standard [89], however, work was done to improve the description of the forklifts dynamic behaviour, and to add a simulated pallet detection system and laser scanner. The 3D printed truncated cone (see Figure 20c), was provided by Crown and used to calibrate the Vive for the testing infrastructure.

6.3 Future Work

This section presents options for further developing the presented work.

Pallet Detection Sensor: Although the APES was tested in simulation, it would be interesting to use either off the shelf hardware (e.g., an IFM PDS (IFM, Essen, Germany) [120]), or develop a pallet detection sensor and test it alongside the proposed APES.

Blending: “Blending” is a technique where actuator inputs are operated simultaneously and is commonly used by experienced forklift operators. A form of blending was used during the project, i.e., simultaneously actuating the fork controls. However, this idea could be extended further. For example by blending actions such as raising the forks and turning towards the pallet. Real-world testing would be useful to determine the time saved by blending additional actions.

Parameter Tuning: The parameter tuning strategy can be extended in several ways. The error function used for tuning could be adjusted to account for more than just the engagement error, i.e., the cross-track and heading error. Additionally, only four parameters were tuned. However, a number of additional parameters could be tuned.

Optimization with Parallel Simulations: When tuning the APES, a computational constraint was the sequential execution of simulations. If simulations were run in parallel (i.e., with the simulations having different parameter configurations), it would significantly reduce the time to tune a system. This process would take significantly more computational capacity, however, this could be provided by a high-performance computation cluster (e.g., NeSI). Message Passing Interface [121], may be suitable to implement simultaneous simulations.

References

- [1] B. Trebilcock, *Let's remember mac barrett, father of the agv*, 2010.
- [2] M. M. R. P. LTD., *Autonomous forklift market: Global industry analysis and forecast (2021-2027) by component, tonnage, application, level of autonomy, and region*. Jan. 2022.
- [3] F. B. Insights, *Automated guided vehicle market size, share & covid-19 impact analysis, by type (tow vehicles, automated forklift trucks, underride/tunnelling vehicles, assembly line vehicles, and others), by navigation technology (laser guided, magnetic guided, vision guided, and others), by application (transportation & distribution, storage & assembly, and packaging), by industry (automotive, food & beverage, e-commerce, and others), and regional forecast, 2022-2029*. Mar. 2022.
- [4] C. G. P. W. B. D. H. A. M. L. M. C. outray, *2018 deloitte skills gap and future of work in manufacturing study*, 2018.
- [5] J. J. Bartholdi and S. T. Hackman, *Warehouse & Distribution Science: Release 0.89*. Supply Chain and Logistics Institute Atlanta, 2008.
- [6] E. H. Frazelle, *World-class warehousing*. Logistics Resources International, 1996.
- [7] J. Gu, M. Goetschalckx and L. F. McGinnis, 'Research on warehouse operation: A comprehensive review', *European journal of operational research*, vol. 177, no. 1, pp. 1–21, 2007.
- [8] R. Bostelman, R. Bostelman and T. Hong, *Review of Research for Docking Automatic Guided Vehicles and Mobile Robots*. US Department of Commerce, National Institute of Standards and Technology, 2016.
- [9] T. Lozano-Perez, 'Spatial planning: A configuration space approach', in *Autonomous robot vehicles*, Springer, 1990, pp. 259–271.
- [10] M. Zucker, N. Ratliff, A. D. Dragan et al., 'Chomp: Covariant hamiltonian optimization for motion planning', *The International Journal of Robotics Research*, vol. 32, no. 9-10, pp. 1164–1193, 2013.
- [11] L. E. Kavraki, P. Svestka, J.-C. Latombe and M. H. Overmars, 'Probabilistic roadmaps for path planning in high-dimensional configuration spaces', *IEEE transactions on Robotics and Automation*, vol. 12, no. 4, pp. 566–580, 1996.
- [12] S. M. LaValle et al., 'Rapidly-exploring random trees: A new tool for path planning', 1998.
- [13] B. Paden, M. Čáp, S. Z. Yong, D. Yershov and E. Frazzoli, 'A survey of motion planning and control techniques for self-driving urban vehicles', *IEEE Transactions on intelligent vehicles*, vol. 1, no. 1, pp. 33–55, 2016.
- [14] S. Karaman and E. Frazzoli, 'Optimal kinodynamic motion planning using incremental sampling-based methods', in *49th IEEE conference on decision and control (CDC)*, IEEE, 2010, pp. 7681–7687.
- [15] M. Otte and E. Frazzoli, 'Real-time motion planning/replanning for environments with unpredictable obstacles', in *Algorithmic Foundations of Robotics XI*, Springer, 2015, pp. 461–478.

- [16] S. Karaman, M. R. Walter, E. Frazzoli and S. Teller, ‘Closed-loop pallet engagement in an unstructured environment’, in *In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2010, pp. 5119–5126.
- [17] J. Pages, X. Armangué, J. Salvi, J. Freixenet and J. Martí, ‘A computer vision system for autonomous forklift vehicles in industrial environments’, in *Proc. of the 9th Mediterranean Conference on Control and Automation*, 2001, pp. 1–6.
- [18] L. Baglivo, N. Biasi, F. Biral *et al.*, ‘Autonomous pallet localization and picking for industrial forklifts: A robust range and look method’, *Measurement Science and Technology*, vol. 22, no. 8, p. 085502, 2011.
- [19] M. Seelinger and J.-D. Yoder, ‘Automatic visual guidance of a forklift engaging a pallet’, *Robotics and Autonomous Systems*, vol. 54, no. 12, pp. 1026–1038, 2006.
- [20] B. Molter and J. Fottner, ‘Semi-automatic pallet pick-up as an advanced driver assistance system for forklifts’, in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, IEEE, 2019, pp. 4464–4469.
- [21] U. Behrje, M. Himstedt and E. Maehle, ‘An autonomous forklift with 3d time-of-flight camera-based localization and navigation’, in *2018 15th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, IEEE, 2018, pp. 1739–1746.
- [22] M. Hentschel, D. Lecking and B. Wagner, ‘Deterministic path planning and navigation for an autonomous fork lift truck’, *IFAC Proceedings Volumes*, vol. 40, no. 15, pp. 102–107, 2007.
- [23] S. Thrun, M. Montemerlo, H. Dahlkamp *et al.*, ‘Stanley: The robot that won the darpa grand challenge’, *Journal of field Robotics*, vol. 23, no. 9, pp. 661–692, 2006.
- [24] A. International, *Standard test method for confirming the docking performance of a-ugvs*, Jan. 2021.
- [25] ISO, *Robotics - performance criteria and related test methods for service robots part 2: Navigation*, 2019.
- [26] L. E. Dubins, ‘On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents’, *American Journal of mathematics*, vol. 79, no. 3, pp. 497–516, 1957.
- [27] M. Quigley, K. Conley, B. Gerkey *et al.*, ‘Ros: An open-source robot operating system’, in *ICRA workshop on open source software*, Kobe, Japan, vol. 3, 2009, p. 5.
- [28] G. Garibotti, S. Masciangelo, M. Ilic and P. Bassino, ‘Robolift: A vision guided autonomous fork-lift for pallet handling’, in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems.*, IEEE, vol. 2, 1996, pp. 656–663.
- [29] R. Tsai, ‘A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses.’, *Robotics and Automation, IEEE Journal of*, vol. 3, pp. 323–344, Sep. 1987. DOI: 10.1109/JRA.1987.1087109.
- [30] G.-z. Cui, L.-s. Lu, Z.-d. He *et al.*, ‘A robust autonomous mobile forklift pallet recognition’, in *2010 2nd International Asia Conference on Informatics in Control, Automation and Robotics (CAR 2010)*, IEEE, vol. 3, 2010, pp. 286–290.
- [31] S. Byun and M. Kim, ‘Real-time positioning and orienting of pallets based on monocular vision’, in *2008 20th IEEE International Conference on Tools with Artificial Intelligence*, IEEE, vol. 2, 2008, pp. 505–508.

- [32] M. Seelinger and J.-D. Yoder, 'Automatic pallet engagement by a vision guided forklift', in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, IEEE, 2005, pp. 4068–4073.
- [33] M. M. Aref, R. Ghacheloo and J. Mattila, 'A macro-micro controller for pallet picking by an articulated-frame-steering hydraulic mobile machine', in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2014, pp. 6816–6822.
- [34] J. Nygards, T. Hogstrom and A. Wernersson, 'Docking to pallets with feedback from a sheet-of-light range camera', in *International Conference on Intelligent Robots and Systems*, IEEE, vol. 3, 2000, pp. 1853–1859.
- [35] S. Wang, A. Ye, H. Guo, J. Gu, X. Wang and K. Yuan, 'Autonomous pallet localization and picking for industrial forklifts based on the line structured light', in *2016 IEEE International Conference on Mechatronics and Automation*, IEEE, 2016, pp. 707–713.
- [36] R. Cucchiara, M. Piccardi and A. Prati, 'Focus based feature extraction for pallets recognition.', in *BMVC*, 2000, pp. 1–10.
- [37] D. Lecking, O. Wulf and B. Wagner, 'Variable pallet pick-up for automatic guided vehicles in industrial environments', in *2006 IEEE Conference on Emerging Technologies and Factory Automation*, IEEE, 2006, pp. 1169–1174.
- [38] L. Baglivo, N. Bellomo, G. Miori, E. Marcuzzi, M. Pertile and M. De Cecco, 'An object localization and reaching method for wheeled mobile robots using laser rangefinder', in *2008 4th International IEEE Conference Intelligent Systems*, IEEE, vol. 1, 2008, pp. 5–6.
- [39] Z. He, X. Wang, J. Liu, J. Sun and G. Cui, 'Feature-to-feature based laser scan matching for pallet recognition', in *2010 International Conference on Measuring Technology and Mechatronics Automation*, IEEE, vol. 2, 2010, pp. 260–263.
- [40] I. S. Mohamed, 'Detection and tracking of pallets using a laser rangefinder and machine learning techniques', Ph.D. dissertation, European Master on Advanced Robotics+(EMARO+), University of Genova, Italy, 2017.
- [41] S. Hsu, S. Acharya, A. Rafii and R. New, 'Performance of a time-of-flight range camera for intelligent vehicle safety applications', in *Advanced Microsystems for Automotive Applications 2006*, Springer, 2006, pp. 205–219.
- [42] J. Suarez and R. R. Murphy, 'Hand gesture recognition with depth images: A review', in *2012 IEEE RO-MAN: the 21st IEEE international symposium on robot and human interactive communication*, IEEE, 2012, pp. 411–417.
- [43] F. Endres, J. Hess, J. Sturm, D. Cremers and W. Burgard, '3-d mapping with an rgb-d camera', *IEEE transactions on robotics*, vol. 30, no. 1, pp. 177–187, 2013.
- [44] C. Beder, B. Bartczak and R. Koch, 'A comparison of pmd-cameras and stereovision for the task of surface reconstruction using patchlets', in *2007 IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, 2007, pp. 1–8.
- [45] S. Kleinert and L. Overmeyer, 'Using 3d camera technology on forklift trucks for detecting pallets', in *Proceedings of the Distributed Intelligent Systems and Technologies Workshop (DIST)*, 2012, pp. 55–62.
- [46] S. Kleinert and L. Overmeyer, 'Integration von 3d-kamerasyystemen am gabelstapler', *Logistics Journal*, vol. 2013, no. 10, 2013.

- [47] J. Xiao, H. Lu, L. Zhang and J. Zhang, 'Pallet recognition and localization using an rgb-d camera', *International Journal of Advanced Robotic Systems*, vol. 14, no. 6, p. 1729 881417737799, 2017.
- [48] B. Molter and J. Fottner, 'Real-time pallet localization with 3d camera technology for forklifts in logistic environments', in *2018 IEEE International Conference on Service Operations and Logistics, and Informatics (SOLI)*, IEEE, 2018, pp. 297–302.
- [49] G. Beach, C. J. Cohen, D. Haanpaa, S. Rowe and P. Mahal, '3d camera identification for enabling robotic manipulation', in *2015 IEEE Applied Imagery Pattern Recognition Workshop (AIPR)*, IEEE, 2015, pp. 1–6.
- [50] D. Haanpaa, G. Beach and C. J. Cohen, 'Machine vision algorithms for robust pallet engagement and stacking', in *2016 IEEE Applied Imagery Pattern Recognition Workshop (AIPR)*, IEEE, 2016, pp. 1–8.
- [51] E. L. F, E. E. R and S. M. J, 'Method for tracing line from possible corner of pallet for material handling vehicle used in warehouse, involves determining average pixel value nearest current pixel location, and tracing current pixel location.', 2014.
- [52] H. L. F, E. E. R, J. E. L and S. M. J, 'Computer-implemented method for imaging and locating pallet to be picked by vehicle for use in warehouse, involves acquiring grey scale image having pallets, where horizontal gradient image is determined using computer.', 2014.
- [53] T. L. F, E. E. R, S. M. J and Y. J. S, 'Method for evaluating possible center stringer of pallet, involves providing image comprising pixels that may generally correspond to orthogonal distance from origin point to one or multiple possible vertical left lines.', 2014.
- [54] A. L. F, E. E. R, S. M. J and Y. J. S, 'Method for identifying scored candidate objects correspond to actual pallets in gray scale image, involves removing subset of first scored object candidate by computer from list to create updated list for second scored object candidate.', 2014.
- [55] B. L. F, E. E. R, S. M. J and Y. J. S, 'Method for evaluating possible pallet object/candidate object in gray scale image, involves calculating composite score based on confidence scores, where each composite score represents chance that object corresponds to pallet structure.', 2014.
- [56] C. L. F, E. E. R, S. M. J and Y. J. S, 'Method for evaluating rectangles that correspond to pallet in image screen, involves providing gray scale image with multiple possible lines, and calculating height based on estimated upper left and upper right corner locations.', 2014.
- [57] D. L. F, E. E. R, S. M. J and Y. J. S, 'Method for identifying one or multiple scored candidate objects, involves identifying multiple scored candidate objects in gray scale image by computer, and starting list of scored candidate objects.', 2014.
- [58] I. L. F, E. E. R, J. E. L and S. M. J, 'Method for determining whether possible line is most likely to be actual line passing through corner of pallet, involves defining lines passing through corner, where lines are oriented at angle within range of angles to axis of image.', 2014.
- [59] G. L. F, E. E. R, J. E. L, S. M. J and Y. J. S, 'Method for tracking scored candidate objects in sequence of image frames acquired by camera of materials handling vehicles, involves determining and analyzing location for each new scored candidate object in next image frame by computer.', 2014.

- [60] F. L. F, E. E. R, J. E. L *et al.*, ‘Programmable computer-implemented method for finding possible corners of pallet, involves determining set of pixels corresponding to possible corner of pallets using set of horizontal and vertical cross correlation results.’, 2014.
- [61] H. S, ‘Cargo handling pallet position and orientation estimation apparatus for use with forklift, has pallet detector which generates captured image data with frame in which frame line surrounding one of pallet holes at pallet front is designated.’, JP2021024718-A, 2021.
- [62] V. L. Barry D. Douglas, ‘Pallet localization systems and methods’, CA3032531A1, 2017.
- [63] I. G. Hemal Shah, ‘Pallet tracking during engagement and disengagement’, US10640347B2, 2020.
- [64] P. J. R. J, T. E. C, P. R. J, T. E. C and P. R. J, ‘Load handling module for material handling vehicle, has controller that is configured to autonomously control movement of fork, and camera field of view which is configured to cover portion of bottom of load that is supported.’, EP3718951-A1; US2020317483-A1; CA3077881-A1; CN111792582-A; AU2020202364-A1, 2019.
- [65] E. Frazelle, ‘World-class warehousing and material handling. mcgraw-hill’, New York, 2002.
- [66] R. G. Sargent, ‘Verification and validation of simulation models’, in *Proceedings of the 2010 winter simulation conference*, IEEE, 2010, pp. 166–183.
- [67] F. Ke, Z. Kaibin and Z. Yuchen, ‘Validation method for simulation models with cross iteration’, *Journal of Systems Engineering and Electronics*, vol. 30, no. 3, pp. 555–563, 2019.
- [68] P. I. Corke, ‘Visual control of robot manipulators—a review’, in *Visual Servoing: Real-Time Control of Robot Manipulators Based on Visual Sensory Feedback*, World Scientific, 1993, pp. 1–31.
- [69] T. Chong, X. Tang, C. Leng, M. Yogeswaran, O. Ng and Y. Chong, ‘Sensor technologies and simultaneous localization and mapping (slam)’, *Procedia Computer Science*, vol. 76, pp. 174–179, 2015.
- [70] E. Olson, ‘Apriltag: A robust and flexible visual fiducial system’, in *2011 IEEE international conference on robotics and automation*, IEEE, 2011, pp. 3400–3407.
- [71] P. Merriaux, Y. Dupuis, R. Boutteau, P. Vasseur and X. Savatier, ‘A study of vicon system positioning performance’, *Sensors*, vol. 17, no. 7, p. 1591, 2017.
- [72] J. S. Furtado, H. H. Liu, G. Lai, H. Lacheray and J. Desouza-Coelho, ‘Comparative analysis of optitrack motion capture systems’, in *Advances in Motion Sensing and Control for Robotic Applications*, Springer, 2019, pp. 15–31.
- [73] 2-d-lidar-sensor, OMD10M-R2000-B23-V1V1D, Pepperl+Fuchs Group, Mar. 2022.
- [74] M. A. Eitan Marder-Eppstein Vijay Pradeep. ‘Actionlib’. (2021), [Online]. Available: <http://wiki.ros.org/actionlib> (visited on 20th Feb. 2022).
- [75] W. Hess, D. Kohler, H. Rapp and D. Andor, ‘Real-time loop closure in 2d lidar slam’, in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2016, pp. 1271–1278.

- [76] R. Yagfarov, M. Ivanou and I. Afanasyev, ‘Map comparison of lidar-based 2d slam algorithms using precise ground truth’, in *2018 15th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, 2018, pp. 1979–1983. DOI: 10.1109/ICARCV.2018.8581131.
- [77] K. Krinkin, A. Filatov, A. y. Filatov, A. Huletski and D. Kartashov, ‘Evaluation of modern laser based indoor slam algorithms’, in *2018 22nd Conference of Open Innovations Association (FRUCT)*, 2018, pp. 101–106. DOI: 10.23919/FRUCT.2018.8468263.
- [78] E. Marder-Eppstein, E. Berger, T. Foote, B. Gerkey and K. Konolige, ‘The office marathon: Robust navigation in an indoor office environment’, in *2010 IEEE international conference on robotics and automation*, IEEE, 2010, pp. 300–307.
- [79] E. W. Dijkstra *et al.*, ‘A note on two problems in connexion with graphs’, *Numerische mathematik*, vol. 1, no. 1, pp. 269–271, 1959.
- [80] P. E. Hart, N. J. Nilsson and B. Raphael, ‘A formal basis for the heuristic determination of minimum cost paths’, *IEEE transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, 1968.
- [81] D. Fox, W. Burgard and S. Thrun, ‘The dynamic window approach to collision avoidance’, *IEEE Robotics & Automation Magazine*, vol. 4, no. 1, pp. 23–33, 1997.
- [82] C. Rösmann, W. Feiten, T. Wösch, F. Hoffmann and T. Bertram, ‘Trajectory modification considering dynamic constraints of autonomous robots’, in *ROBOTIK 2012; 7th German Conference on Robotics*, VDE, 2012, pp. 1–6.
- [83] ———, ‘Efficient trajectory optimization using a sparse model’, in *2013 European Conference on Mobile Robots*, IEEE, 2013, pp. 138–143.
- [84] P. B. Andy Zelenak. ‘Pid’. (2015), [Online]. Available: <http://wiki.ros.org/pid> (visited on 20th Feb. 2022).
- [85] J. Bohren and S. Cousins, ‘The smach high-level executive [ros news]’, *IEEE Robotics & Automation Magazine*, vol. 17, no. 4, pp. 18–20, 2010.
- [86] P. Schillinger, S. Kohlbrecher and O. von Stryk, ‘Human-Robot Collaborative High-Level Control with an Application to Rescue Robotics’, in *IEEE International Conference on Robotics and Automation*, Stockholm, Sweden, May 2016.
- [87] O. S. R. Foundation. ‘Robot simulation made easy.’ (2019), [Online]. Available: <http://gazebosim.org/> (visited on 20th Mar. 2022).
- [88] S. Chitta, E. Marder-Eppstein, W. Meeussen *et al.*, ‘Ros_control: A generic and simple control framework for ros’, *The Journal of Open Source Software*, vol. 2, no. 20, pp. 456–456, 2017.
- [89] W. Garage. ‘Universal robot description format (urdf)’. (2009), [Online]. Available: <http://wiki.ros.org/urdf/> (visited on 5th Jan. 2022).
- [90] H. R. Kam, S.-H. Lee, T. Park and C.-H. Kim, ‘Rviz: A toolkit for real domain data visualization’, *Telecommunication Systems*, vol. 60, no. 2, pp. 337–345, 2015.
- [91] A. Yates. ‘Alan yates on the impossible task of making valve’s vr work’. (2016), [Online]. Available: <https://www.youtube.com/watch?v=75ZytcYANTA> (visited on 16th Feb. 2022).
- [92] M. Borges, A. Symington, B. Coltin, T. Smith and R. Ventura, ‘Htc vive: Analysis and accuracy improvement’, in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2018, pp. 2610–2615.

- [93] T. A. Jost, G. Drewelow, S. Koziol and J. Rylander, 'A quantitative method for evaluation of 6 degree of freedom virtual reality systems', *Journal of Biomechanics*, vol. 97, p. 109 379, 2019.
- [94] D. C. Niehorster, L. Li and M. Lappe, 'The accuracy and precision of position and orientation tracking in the htc vive virtual reality system for scientific research', *i-Perception*, vol. 8, no. 3, p. 2 041 669 517 708 205, 2017.
- [95] S. M. van der Veen, M. Bordeleau, P. E. Pidcoe, C. R. France and J. S. Thomas, 'Agreement analysis between vive and vicon systems to monitor lumbar postural changes', *Sensors*, vol. 19, no. 17, p. 3632, 2019.
- [96] T. Ameler, K. Blohme, L. Brandt *et al.*, 'A comparative evaluation of steamvr tracking and the optitrack system for medical device tracking', in *2019 41st annual international conference of the IEEE engineering in medicine and biology society (EMBC)*, IEEE, 2019, pp. 1465–1470.
- [97] K. Sletten, 'Automated testing of industrial robots using htc vive for motion tracking', M.S. thesis, University of Stavanger, Norway, 2017.
- [98] M. S. Ikbal, V. Ramadoss and M. Zoppi, 'Dynamic pose tracking performance evaluation of htc vive virtual reality system', *IEEE Access*, vol. 9, pp. 3798–3815, 2020.
- [99] R. C. Dorf and R. H. Bishop, *Modern control systems*. Pearson Prentice Hall, 2008.
- [100] MathWorks®. 'Matlab®'. (2021), [Online]. Available: <https://au.mathworks.com/products/matlab.html> (visited on 22nd Feb. 2022).
- [101] MathWorks. 'System identification toolbox'. (2021), [Online]. Available: <https://au.mathworks.com/products/sysid.html> (visited on 22nd Feb. 2022).
- [102] A. Piazzì, C. Guarino Lo Bianco and M. Romano, ' η^3 -splines for the smooth path generation of wheeled mobile robots', *IEEE Transactions on Robotics*, vol. 23, no. 5, pp. 1089–1095, 2007. DOI: 10.1109/TRO.2007.903816.
- [103] R. S. Wallace, A. Stentz, C. E. Thorpe, H. P. Moravec, W. Whittaker and T. Kanade, 'First results in robot road-following.', in *IJCAI*, 1985, pp. 1089–1095.
- [104] O. Amidi, 'Integrated mobile robot control', CARNEGIE-MELLON UNIV PITTSBURGH PA ROBOTICS INST, Tech. Rep., 1990.
- [105] R. C. Coulter, 'Implementation of the pure pursuit path tracking algorithm', Carnegie-Mellon UNIV Pittsburgh PA Robotics INST, Tech. Rep., 1992.
- [106] G. M. Kurtzer, V. Sochat and M. W. Bauer, 'Singularity: Scientific containers for mobility of compute', *PLoS one*, vol. 12, no. 5, e0177459, 2017.
- [107] O. S. R. Foundation. 'Gzweb - web client for gazebo'. (2014), [Online]. Available: <https://gazebosim.org/gzweb.html> (visited on 22nd Feb. 2022).
- [108] W. D. Andrew Tridgell Paul Mackerras. 'Rsync'. (1996), [Online]. Available: <https://linux.die.net/man/1/rsync> (visited on 22nd Feb. 2022).
- [109] 'Photography - tripod connections', International Organization for Standardization, Standard, Jul. 2010.
- [110] W. Kabsch, 'A solution for the best rotation to relate two sets of vectors', *Acta Crystallographica Section A: Crystal Physics, Diffraction, Theoretical and General Crystallography*, vol. 32, no. 5, pp. 922–923, 1976.
- [111] P. Virtanen, R. Gommers, T. E. Oliphant *et al.*, 'SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python', *Nature Methods*, vol. 17, pp. 261–272, 2020. DOI: 10.1038/s41592-019-0686-2.

- [112] O. P. Ruben Smits Erwin Aertbelien. ‘Kinematics and dynamics library’. (2014), [Online]. Available: <http://wiki.ros.org/kdl> (visited on 15th Jan. 2022).
- [113] L. D. Barnes, K. Lee, A. W. Kempa-Liehr and L. E. Hallum, ‘Detection of sleep apnea from single-channel electroencephalogram (eeg) using an explainable convolutional neural network’, *bioRxiv*, 2021.
- [114] T. Nagarajah and G. Poravi, ‘A review on automated machine learning (automl) systems’, in *2019 IEEE 5th International Conference for Convergence in Technology (I2CT)*, IEEE, 2019, pp. 1–6.
- [115] J. Joyce, ‘Bayes’ theorem’, 2003.
- [116] F. Nogueira, *Bayesian Optimization: Open source constrained global optimization tool for Python*, 2014–. [Online]. Available: <https://github.com/fmfn/BayesianOptimization> (visited on 18th Feb. 2022).
- [117] T. Foote, ‘Tf: The transform library’, in *Technologies for Practical Robot Applications (TePRA), 2013 IEEE International Conference on*, ser. Open-Source Software workshop, Apr. 2013, pp. 1–6. DOI: 10.1109/TePRA.2013.6556373.
- [118] J. K. Kruschke, ‘Bayesian estimation supersedes the t test.’, *Journal of Experimental Psychology: General*, vol. 142, no. 2, p. 573, 2013.
- [119] MathWorks. ‘Control system toolbox’. (2021), [Online]. Available: <https://au.mathworks.com/products/control.html> (visited on 22nd Feb. 2022).
- [120] G. Place, *Modernize warehouse operations with pds*. [Online]. Available: https://www.ifm.com/binaries/content/assets/us/pdfs/resources/white-papers/robotics_ifm_techpaper.pdf (visited on 19th Apr. 2022).
- [121] W. Gropp, W. D. Gropp, E. Lusk, A. Skjellum and A. D. F. E. Lusk, *Using MPI: portable parallel programming with the message-passing interface*. MIT press, 1999, vol. 1.