Dedication

Acknowledgement

Abstract

Keywords:

Contents

List of Figures

List of Tables

Glossary of Acronyms

NN RNN FL: fuzzy Logic DWA MP RRT GA SA PSO

-
-

•

General Introduction

The latest events of the current decade have highlighted the challenges that manufacturers, suppliers, and end customers face during fluctuations in logistics and supply chain processes. Living in a VUCA world—Volatile, Uncertain, Complex, and Ambiguous—requires us to continuously adapt to changes and anticipate future events by preparing our developed environments and scaling our solutions. Simultaneously, it is crucial to maintain high standards that ensure productivity, enhance work safety, and optimize ergonomics.

In this context, the primary objective of intralogistics is to optimize, integrate, automate, and manage internal logistical flows of material and information within distribution centers, warehouses, or manufacturing plants. This subfield focuses on increasing operational efficiency by employing new technologies, such as autonomous robots.

Modernizing industrial environments through intralogistics offers significant potential for companies that adopt and adapt to it. However, convincing potential customers of the efficiency and impact of intralogistics robots presents challenges. These limitations include high training and implementation costs, changes to work routines, and the need for space and process adaptations.

A recent study from CBRE, the world's largest real estate services provider, revealed that European industrial and logistics investments increased by 16% in Q1 of 2024 compared to Q1 of 2023. Despite this, many warehouses are old, repurposed buildings that are unorganized due to the nature of their daily tasks. These brownfield warehouses are expensive to maintain and digitalize but represent ideal grounds for developing and utilizing fully autonomous systems. Unlike AGVs, autonomous vehicles possess the intelligence and capability to plan and execute their plans efficiently. They are designed to adapt to uneven terrains and unorganized working environments given the revolutionary technologies that they hold.

In this context, STILL, a KION group company, has been developing smart intralogistics solutions since its establishment more than a 100 years ago, successfully integrating automation into logistics. STILL offers a wide variety of products that cater to industries ranging from food retail to automotive manufacturing and chemical sectors. Their solutions address various customer challenges, such as reaching high shelves, order picking, palletizing, fleet management, and providing consulting services. Trusted by leading German companies like Siemens, STILL's products and services are renowned for their reliability and efficiency.

The STILL Autonomous Robots department focuses on developing and enhancing smart vehicles. These autonomous robots, with minimal cost-effective input from the warehouse environment, can perceive their surroundings, estimating their positions, efficiently planning future tasks, controlling their movements to reach destinations, executing desired actions, and making corrections if necessary. This focus on smart, autonomous vehicles demonstrates STILL's commitment to pushing the boundaries of intralogistics and automation.

In light of this, this thesis aims to contribute to the process of palletizing by optimizing a local path planning approach applied in the warehouse's stations near the shelves or spots where pallets are located for picking or in free placing areas. The developed approach seeks to plan the near-field path optimally while simultaneously avoiding obstacles.

The objective is to create predictable, repeatable, and explainable vehicle behaviors, demonstrating the autonomous vehicle's ability to generate effective solutions tailored to each specific scenario. By focusing on optimal, pattern-based near-field path planning, this thesis addresses the challenge of navigating complex intralogistics environments, ensuring maximum efficiency and safety in operations. This approach not only enhances the vehicle's performance but also showcases the potential of autonomous technology in transforming modern intralogistics.

This work encloses 4 chapters:

- **Chapter 1** gives a deep insight about the host company's structure, activities and products. Then it dives into the project context and it motivations, the studied problematic, the fundamental aspects of the work, the thesis specifications and, the work methodology.
- **Chapter 2** delves into the state of the art of the work area, then goes through a review of the literature that served as a base of the thesis and gave an overview of the existing solutions. Finally, it presents milestones followed in the course of the thesis work.
- **Chapter 3** explains the development steps of the approach: it presents the mathematical aspect of splines and their implementation in robotic path planning, explains the geometric division of the stations into transition zones, discusses the studied path discrimination approaches, and finally it explores the optimization approaches for the local path planning problem.
- **Chapter 4** explicits the steps it takes to implement the developed approach in the RACK framework, test them in the RACK simulation system, then on the automated vehicle, run different test scenarios and states the obtained results.

Chapter 1
Host company and Project context

Chapter 1

Host company and Project context

Introduction

This chapter is reserved to present STILL GmbH as the host company, its organizational structure, the mother company KION group. It will then proceed to describe the range of products that the company produces. The second part is dedicated to set the project context by explaining the problem statement, the motivation behind this thesis project, and its specifications. The final part will emphasize the work methodology adopted to carry out this project.

1 Host company: STILL GmbH

This section introduces the host group and company through their activities, products, and activities

1.1 KION Group and STILL GmbH

STILL GmbH, based in Hamburg, Germany, is a leading manufacturer of intralogistics solutions with 14 locations in Germany and a global sales network spanning 246 locations. Operating under the KION Group, Europe's largest forklift truck manufacturer, STILL boasts over 100 years of experience. The company develops highly efficient, client-tailored products, serving businesses of all sizes with a wide range of forklift trucks—from manually driven forklifts to high-reach trucks and fully automated vehicles—alongside consultancy services and software solutions.

STILL prioritizes smart logistics and energy optimization while maintaining award-winning product quality, catering to industries such as food and retail, automotive, and electronics. Employing over 9,000 people across departments like sales and marketing, research and development, production, mechatronics, and quality assurance, STILL remains at the forefront of intralogistics innovation.

KION Group is one of the global leaders in the fields of industrial trucks and supply chain solutions. It is the mother company of: Linde, Dematic Baoli, OM, Fenwick, and STILL who produce the goods and services of the group as detailed in Figure 1.1.

Present in 4 continents and hiring more than 42000 employees, KION's startegy is to ensure profitable ans sustainable growth while focusing on Automation and robotics deplyment as one of the main leaders of this growth.

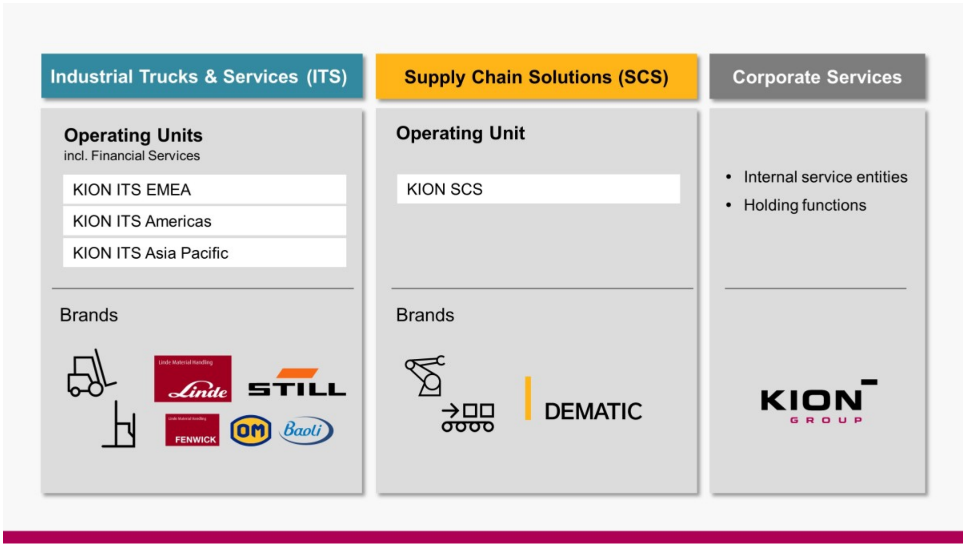

Figure 1.1: KION segment services and companies [1]

1.2 KION Management Hierarchy

The company is composed of departments managing the operations in all companies that are divided by scope of interest like R&D, Management, finances, etc.. Figure 1.2 illustrates the different areas of responsibility of the Executive Board. The Autonomous vehicles team belongs to the Mobile Automation department under CTO.

1.3 STILL Products

The 2017-established Autonomous vehicles team aims to develop fully automated solutions that leverage novel technologies to create innovative services delivered through forklift trucks. The vehicles are developed while keeping safety and high-performance as the main priorities.

iGo neo shown in Figure 1.6 is one of the main products developed by the department, it is a low level order picker transformed into the agent's autonomous assistant. Functioning in autonomous or semi-autonomous modes, it can follow the operator and their pace while avoiding obstacles and perceiving their surroundings as well as pick and place pallets in designed areas. Its added value is in preserving ergonomics of the operators by preventing heavy load carrying for long distances and decreasing the driving ascents and descents by 75% thus increasing the personal and collective performances [3].

Figure 1.2: KION Executive Board responsibilities as of 01.2024 [2]

As STILL specializes in forklift trucks, it counts many other products. Trucks are either Diesel or Gas fueled, or electric trucks that use Li-Ion batteries. Depending on the client's warehouse type, they can choose from a vast range of reach trucks Figure 1.3, hand pallet trucks Figure 1.4, double stacker trucks Figure 1.5, and Automated industrial Trucks Figure 1.6 [4].

Figure 1.3: STILL reach truck

Figure 1.4: STILL hand truck

Figure 1.5: STILL reach truck

Figure 1.6: STILL hand truck

Despite the impressive capabilities of the iGo neo and similar autonomous vehicles, the implementation of such advanced technology brings up several challenges, particularly in ensuring reliable and predictable behavior under all operating conditions. This leads to a key motivation for further investigation and improvement in the field.

2 Graduation project Motivation and Specifications

2.1 Motivation and Problem statement

While autonomous vehicles can be highly reliable and efficient in carrying out various tasks, their behavior is not always predictable or easily explained. The output often exhibits a stochastic nature. For example, an obstacle-avoiding solution planned by the autonomous vehicle may be safe and correct but might follow an unusually shaped path.

Such stochastic behaviors can lead to a lack of trust and interest in robotized forklift trucks from a customer's perspective. This unpredictability can cause customers to question the system's repeatability, fearing that it may not perform consistently in critical situations. Moreover, the unexpected nature of these behaviors can make it difficult for operators to understand and anticipate the vehicle's actions, further reducing confidence.

Adding to these concerns, many autonomous systems, particularly in the intralogistics sector, require significant commissioning efforts before they can be implemented in a new environment and begin their service. Whether it's a required software, sensors, or measurements, these systems demand substantial time and financial investment—two crucial resources that we aim to optimize.

As engineers, we are committed to developing optimal solutions that are easy to commission in a new environment. These so-called "plug-and-play" solutions reduce the effort required and allow customers to start benefiting from the autonomous features with just the physical truck on-site and some basic input from the warehouse map, the rest, is online recognition and processing. This approach significantly enhances the impact and convenience of the technology.

To address these issues, the autonomous vehicles department is dedicated to creating solutions that are not only reliable and efficient but also transparent and understandable. By focusing on explainable autonomous systems, the department aims to build greater trust with customers, ensuring they feel confident in the technology and are more likely to adopt and utilize these advanced robotic solutions.

This thesis discusses one such possible application of autonomous vehicles. The use case involves solving the following problem: as illustrated on Figure 1.7, after the vehicle enters the station- a limited area inside the warehouse where the shelf stands to palletize/depalletize, in predefined positions, it faces the following problematics:

- The vehicle's forks are not facing the destination shelf but rather the opposite direction, so a driving direction change is needed.

- The vehicle is heavy (1200 to 1500 KG) with an overall length of 2500 to 4000 mm which makes it both challenging and dangerous to change directions: turning on the spot or navigating in highly curved paths[5].
- The pallet docking process has to be very precise to avoid shifts and mistakes.

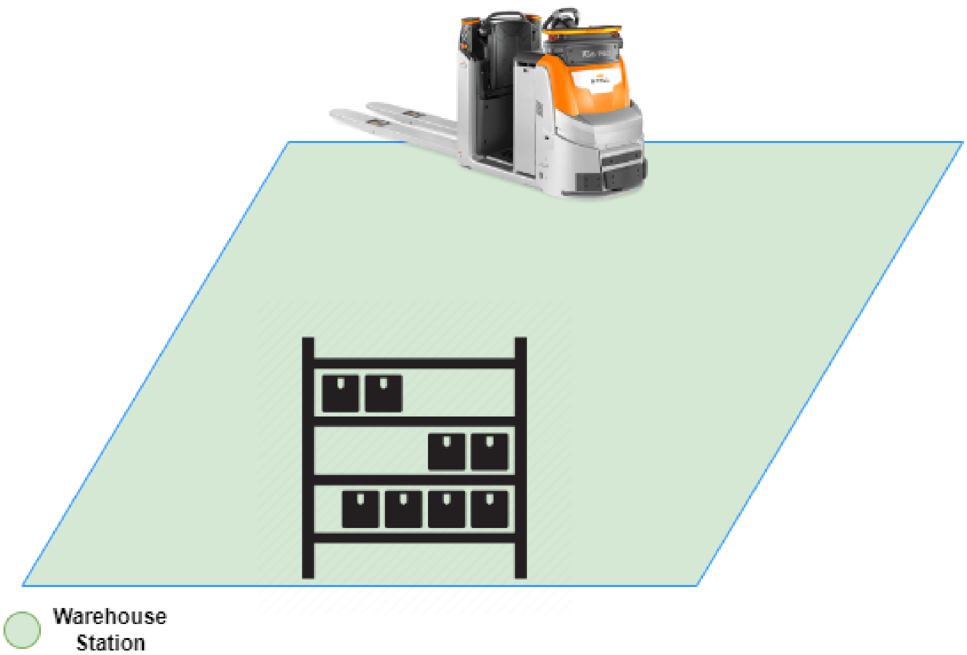

Figure 1.7: Vehicle setting in the station

The first inspiration for the proposed solution was the forklift drivers themselves. The experienced drivers all agree to solve the problematic – if it was to be solved manually, in the same way: to drive in an arc shape to a point, then to change the driving direction and orienting the vehicle to the destination position.

We would like to solve this problematic in a manner that:

- Imitates the manual driving process to pick a pallet when in the same situation (facing backward of the pallet).
- Implements a predictable local path planning algorithm for the vehicle operations inside a station.
- Reduce the computational expense used with the global dynamic path planning algorithms implemented through a pattern
- Chooses an optimal path out of the various paths that can be driven.

2.2 Project specifications

The aim of this thesis is to devise and assess a methodology for automated path planning of a mobile robot, taking into account environmental obstacles. The proposed path should optimally connect the robot to a nearby target pose, whereby optimality is defined as achieving maximum speed while adhering to the robot's kinematic constraints. Moreover, the path planning process should employ an online pattern-based approach, ensuring that the robot's behavior is explainable, and that all environmental recognition information is utilized during path planning. The work should build on existing work in the Robotics Application Construction Kit (RACK) and satisfy real-time processing capabilities. Currently, existing methods in the literature shall be taken into consideration so that the functionality is sufficient for typical intralogistics applications and available computing power on the mobile robot itself.

The thesis consists of the following steps:

- getting acquainted with the current topic area with subsequent clustering and discrimination of available approaches based on a scientific literature review.
- design methods for evolving B-Spline based path planning approaches under consideration of typical patterns used by manual driven trucks in intralogistics.
- design methods for determining optimization approaches and related metrics to enable real-time capable path planning.
- implement the derived approaches in the RACK framework and evaluate the performance of the developed methods.
- verify and evaluate the developed robotic application on a mobile robot under consideration of intralogistics boundary conditions.

3 Work Structure and Methodology

Our team adopts an Agile Scrum methodology showcased in Figure 1.8 to ensure efficient and flexible project management. Here's how we approach our work:

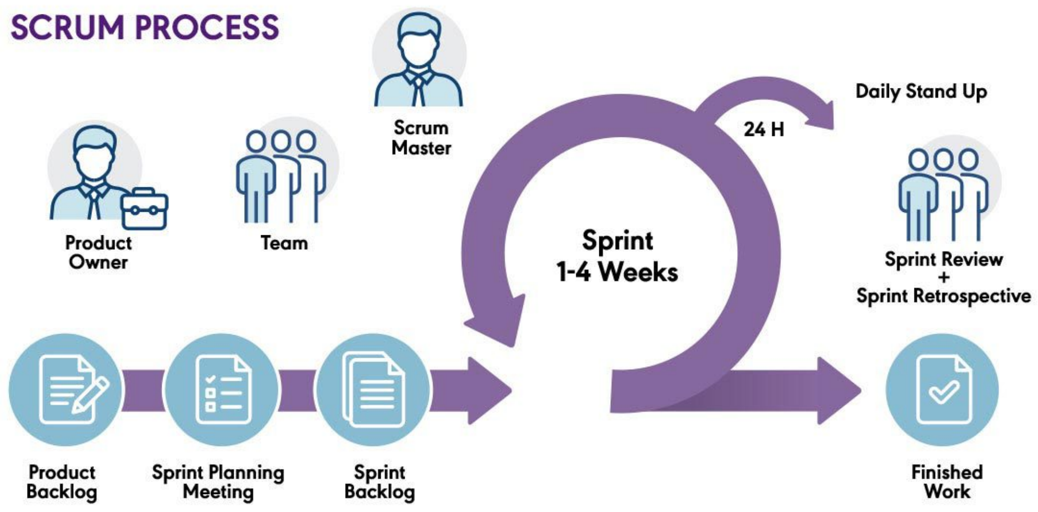

Figure 1.8: Agile Scrum Process [6]

3.1 Agile Scrum Framework

- **Jira:** We use Jira to organize and track our tasks and progress. Jira allows us to create and manage tickets, which are detailed records of tasks, bugs, or features that need attention. Each ticket is assigned to team members and tracked through its development stages until completion.
- **Sprints:** Our work is organized into 2-week sprints. Each sprint is a focused period where we aim to complete a set of predefined tasks. At the start of each sprint, we hold a meeting to review the previous sprint: every team member presents their completed tickets, and communicates the changes or blockers that appeared during the process and plan for the next sprint: decide which tasks will be tackled during the sprint. This helps us maintain a steady pace and regularly deliver increments of our project.
- **PI Planning:** Every quarter, we engage in Program Increment (PI) planning with the mobile automation teams. The PI happens in two phases: each team prepares their planning for the next 3 months, then it is discussed and tailored again in a bigger round. This planning session helps us align our goals and strategies for the upcoming quarter. We review progress, set objectives, and coordinate with other teams to ensure that our work is aligned with broader project goals and company vision.
- **Daily Standups:** We hold 15 minutes long daily standup meetings to keep everyone on the same page. During these meetings, team members share

updates on their progress, discuss any challenges they are facing, and outline their plans for the day. This practice promotes transparency, communication and quick problem-solving through collaboration.

3.2 Version Management

- **GitHub:** We use GitHub for version control and code management. GitHub allows us to collaborate on code, track changes, and manage different versions of our project. Each team member can contribute to the codebase, and we use pull requests to review and integrate new features.

3.3 Communication and Collaboration:

- **Microsoft Teams:** We use Microsoft Teams for real-time communication and collaboration. Teams provides a platform for chatting, video calls, and sharing files, facilitating smooth and efficient interactions among team members.
- **Microsoft Outlook:** Outlook is used for email communication and scheduling. It helps us manage meetings, track important messages, and coordinate tasks and deadlines.

By integrating these tools and practices, we ensure a structured yet flexible workflow, enabling us to adapt to changes, communicate effectively, and deliver high-quality results.

Chapter 2
State of the Art

Chapter 2

State of the Art: Optimal path planning for autonomous robots

Introduction

The intralogistics sector is undergoing a significant transformation due to the integration of digital tools such as connected industries and robotics. The cluttered and highly dynamic nature of this environment makes it challenging to successfully implement robotic solutions.

The first steps in digitalizing logistic processes with robots involved the use of Automated Guided Vehicles (AGVs). First introduced in 1955, AGVs perform tasks like material handling. AGVs are managed by top-level software that handles task planning, providing the vehicles with intermediate waypoints to navigate from start to end points [7].

However, the decentralized decision-making approach of AGVs makes them unresponsive to changes in their environment. For example, AGVs localize themselves using specific and precise anchors physically located in their workspace.

This localization mechanism helps them follow the assigned path. As a result, even a simple environmental change of the dispositionsrequires an update of the measurements and maps on which the planning is based. Another example is that an AGV is unable to plan and execute a solution if it encounters an obstacle on its way to the goal or it arrives at a shifted destination. In such cases, the AGV's reaction is to stop and wait for top-level instructions [8]. This behavior decreases productivity and disrupts the planned sequence of tasks until an intervention is managed.

To overcome these disadvantages of AGVs, Autonomous Mobile Robots (AMRs) were introduced. AMRs are equipped with a decentralized control system, enhanced perception of their surroundings through more complex hardware, and advanced software to manage and integrate these hardware additions (Figure 2.1).

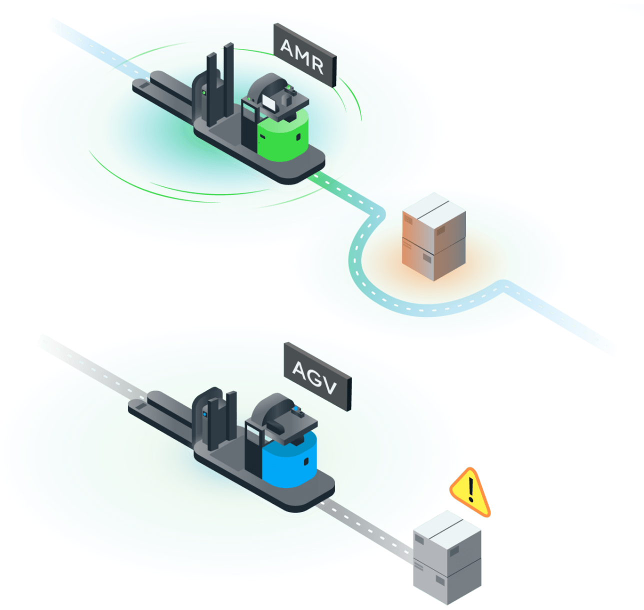

Figure 2.1: AMR and AGV behaviors at presence of an obstacle [9]

In addition, they grant a fast integration into new environments due to smart perception and recognition technologies. AMRs use this data as input to smart algorithms that enable it to plan its missions, navigate to its destination, avoid collisions, and execute missions all while collaborating with humans in the working space. The Autonomous vehicles do not require an automation expert or a roboticist's presence to be configured or to deal with complex situations. It is able to solve such situations individually due to the scalable software that it has on board.

On the one hand, this evolution enables AMRs to navigate more dynamically and adaptively within intralogistics environments, improving overall efficiency and responsiveness. On the other hand, the level of flexibility that AMRs brings important safety and efficiency considerations [7].

One of the main challenges is developing the navigation mechanisms. In a logistics environment, it is crucial to comply with the nature of the workspace. The AMR should recognize the location of materials to be handled and be able to navigate to and from those locations to next missions. Building a flexible and efficient solution requires a deep understanding of the situations and special cases that the vehicle may encounter and a study about how to create scalable solutions for such events.

Before diving into the methodologies and technologies used to address this thesis' topic, it is relevant to examine the related works. This research will serve later in the thesis as a guiding outline. Literature helps to investigate the level of progress that other researchers reached in similar topics, prevents re-invention of existing concepts, and pushes to ethically exploit the developed technologies. For this thesis, it is important to deeply examine the research papers and scientific resources to build a solid foundation of knowledge, identify the gaps in some studies, and propose novel solutions.

In this context, this state-of-the-art report, delves into pathplanning and near-field path planning for mobile robots in general and for intralogistics AMRs specifically. Then, it examines possible solutions for path creation and generation. Afterwards, it studies the decision-making science approaches that can be used to evaluate and optimize path suggestions. Finally, this chapter outlines the methodology to be followed throughout this scientific work(Figure 2.2).

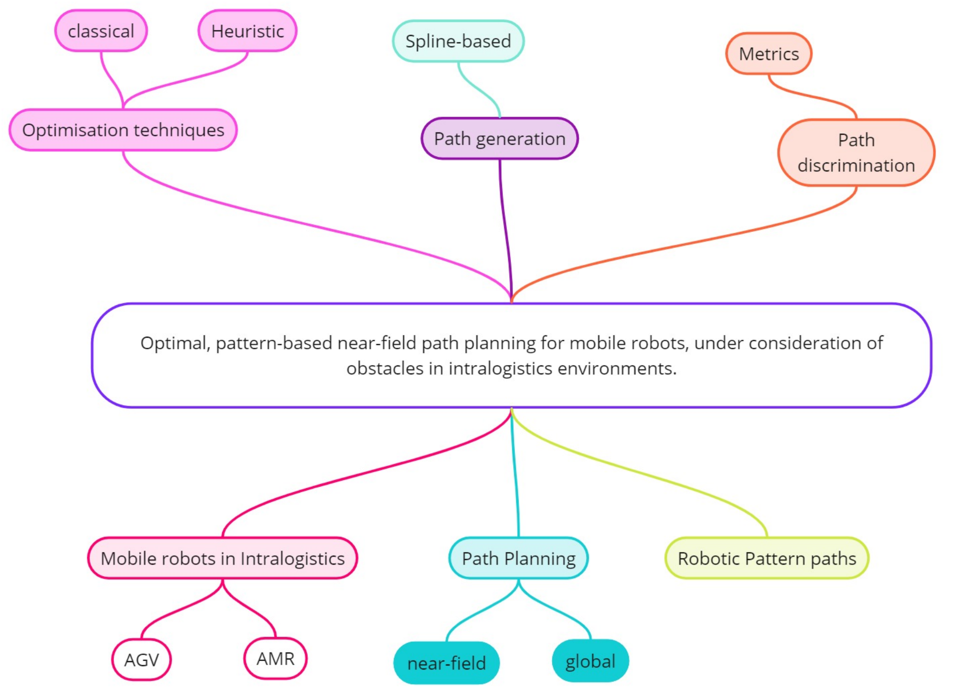

Figure 2.2: Mind map of the key topics

1 Path Planning

This section will dive into the path planning state of the art: presenting its promising opportuniities and current challenges, detailing the different approaches, and discussing their efficiency and compatibility with the problem statement.

1.1 Path Planning for autonomous robots: Opportunities and challenges

Path planning for mobile robots involves creating and generating efficient routes for the robot to travel from a starting position (A) to a target position (B), ensuring minimal time and travel distance while avoiding collisions with nearby objects [7].

In complex environments, it is challenging for robots to move randomly and fulfill their missions. However, with appropriate input from various sensors, such as laser scanners, cameras, and LiDAR, robots can perceive their surroundings and plan the right path accordingly. The innovation in hardware made navigation flexibility and autonomous recovery after failure possible[7].

The application areas of path planning for autonomous mobile robots are factories and warehouses, healthcare institutions, hotels and restaurants, and domestic areas. The relevance of efficient path planning, mainly in the intralogistics sector, is derived from the constant need of optimizing material flow, productivity rates, and cost effectiveness. Path planning is very promising to reduce travel time and distances when moving goods and thus, improving overall operational costs.

In those dynamic environments, operators and employees are moving around, goods and pallets are being transported or stocked, and materials must be handled safely and carefully. Flexibility in routing and planning, enables the autonomous vehicles to always drive the optimal path based on the space settings. The benfits here include:

- Independence from Human intervention -AMRs do not require assistance, unike AGVs [7].
- Respect of safey standards through developed vision and recognition: critical personnel safety protection, preserving the handled goods, manufacturing systems, storage material or other vehicles.
- Reduced energy consumption thanks to optimal: smooth and short paths that respect the vehicles' kinematics and allocated task and travel time.
- Robustness and responsiveness thanks to decentralized decison making: enables fast recovery after failure [7].

- Alignment with real-world applications and the growing demand for automation in logistics due to e-commerce growth and supply chain complexities.

Efficient path planning is essential for ensuring the safe handling of objects in the environment. In intralogistics, the materials being handled—whether goods, manufacturing systems, storage shelves, or other vehicles—are valuable and must be treated with care. Moreover, safety is a critical concern, especially when human safety and protection are involved. To pass quality tests, autonomous forklifts must meet personal safety requirements. These requirements include, but are not limited to, maintaining a safe distance from both static and dynamic objects and people, as well as detecting surroundings at any height and time. By following an efficient path, overall productivity increases since operations are repeated many times throughout the day. An efficient path optimizes both length and smoothness, thereby reducing travel time. Additionally, efficient path planning conserves the truck's energy, reducing the frequency of recharging.

Autonomous robots, as the name suggests, are standalone systems that must compile and process such input and generate, through algorithms, efficient paths. Path planning serves as the crucial link between the robot's sensor input and its motion control [10].

Literature and scientific studies differentiate between two main types of path planning: global and local path planning. Global planning involves finding an optimal path from the start to the target position based on sensor input within a known, static environment, whereas local planning focuses on real-time obstacle avoidance, typically used while moving to avoid dynamic obstacles [11].

More that 60 years have elapsed since "Shaky" the first wheeled robot was running it first tests in Stanford University' labs. However, most of the robotic related topic are still being researched and improved. Dealing with all the aspects and challenges that robotics comes with can be very intricate. One of the major topics posing challenges to researchers is path planning. In a research by S. H. Tang et al. [20], the authors reviewed recent path planning approaches and challenges in dynamic unkown environments. Saftey in path planning was the main concern for 29 % of the reviewed studies. Navigating efficient paths while avoiding collisions is challenging to accomplish. Collision avoidance is tightly related to perception input through sensors, analysis and use of the data. While it may seem simple for the robots to correctfully analyze and recognize the objects around them, in reality, detailed understanding is not possible [21]. Issues are related to the accuracy of the sensors and the robustness of the algorithms. It is expected from the robot to preceive of the obstacles just like humans do, recognizing 3d shapes, dimensions, depth, direction and velocity, but from the robot's perspective, it is only able to recognize the outer surface that reflects the sensor's signals as shown in figure 2.3.

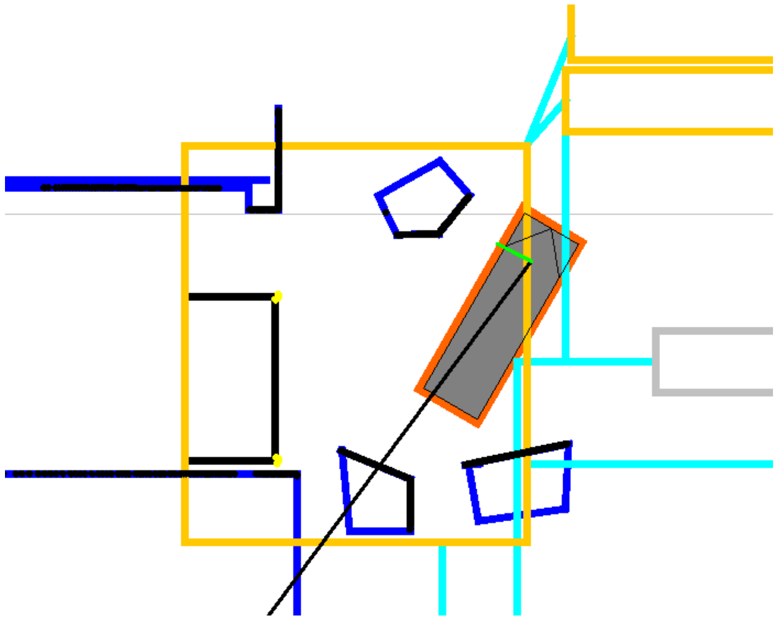

Figure 2.3: Simulation of the robot's perception of surrounding obstacles.
Yellow rectangle: Station limits
Black rectangle: Shelf limits: where to pick or to place pallets
In gray: Robot in simulation
Blue polygones: Simulated obstacles
Black lines: Perceived obstacle points

As a result, algorithms are to compensate this ambiguity by enforcing safety measures like keeping the vehicle at a safe distance from the obstacles and deccelerating at the proximity of static and dynamic objects to avoid collisions In addition, this input contains noises and shadows caused by the sensors reflections of other objects or inaccuracies. This makes it challenging to interpret the input and make safe decisions as the built algorithms have to deal with the given data in all cases and detect the inaccuracies and noises. In [22], the authors investigate a collision avoidance approach Beyond that, it is complex to generate feasible solutions in all types of environments, some planners and algorithms risk stagnating in a local minima and not converging to the optimal routes.

Besides safety, computational cost is an interesting topic that researches are looking at. It is considered as a key metric by scientists to judge if an algorithm will be able to accomplish certain tasks within the needed time frame. It accounts for the time and resources consumed by the algorithm while computing [24]. In an experiment to compare global motion planners, Heiden et al. use different metrics on which they based their comparison. Computation time is one of the considered factors used to discrminate approaches. They study closely the difference in computation time in different environment scenarios besides the effect of certain improvements introduced to the algorithms on the computation time. The computational cost is justified by optimizations running on the computed paths like path smoothing, computing steer functions, and checking for collision possibilities [23]. In a real time context, it is important to synchronize different tasks, analyze massive amounts of data generated by sensors and camera like point clouds and 2d/3d images, and compute the required decisions in a reliable and accurate way. As the complexity of the environment increases, so does the computational burden, often leading to longer processing times or the need for more powerful hardware [23].

While managing computational costs is crucial, it is equally important to ensure that the paths generated are not only computationally efficient but also smooth and short, as these factors significantly influence the robot's overall performance and energy consumption. Given the kinematics of a robot, the destination's location and the clutter in the environment, path planners unusually render rough paths. Cusps, which are sudden and sharp direction changes, and high curvatures of the path are unusual path properties that are hard to drive, energy and time inefficient, and require continuous decelerations and accelerations. Long paths are also not favored. While they can be necessary to avoid obstacles or to create a smooth path, longer distances result in time consumption and extensive energy usage. Some path planner include post-smoothing methods that modify the paths after its creation and intervene by shortening and smoothing paths areas while obstacles. In [23], Heiden et al. present various ways to implement like using splines, and short-cuts. They conclude that different methods deal with certain improvements areas differently. For example, While Splines are outperformed in the curvature and cusps areas, they are efficient when it comes to path-smoothning computation time.

In conclusion, tackling robot path planning requires looking at several improvement areas at the same time. Through an analysis of [20], we can notice that most of the studied approaches are effective in particular aspects, but lack optimization in others. This further emphasizes that these challenges remain under research and are not yet fully addressed in the literature. While these challenges highlight areas needing further exploration, various path planning approaches offer different strategies to address them. Understanding these approaches can provide insights into potential solutions and advancements in overcoming the existing limitations.

1.2 Comprehensive Path Planning Approaches

1.2.1 Global Path Planning

At the beginning of the research phase for this thesis, it was important to look at the available types of path planning methods. Science generally distinguishes between three general approaches of Global Path Planning all having the same aim to plan the path from a start to a goal position within a pre-mapped environment: grid-based methods, sampling-based methods, and artificial intelligence-based methods [13].

Grid-based methods involve discretizing the environment into a grid of cells, where each cell represents a small, discrete part of the space. Then, the algorithm visits the grids to decide which cells will be used for the path based on the occupancy or cost and calculates the fitness function from start to the reached point in A* algorithm for example or from the reached point to the destination for Dijkstra Algorithm. While this method seems easy to understand and quite simple to implement, it is computationally intensive because of the number of grids to evaluate and the repetitiveness and constrained in direction (Figure 2.4)

Sampling-based algorithms do not discretize the entire environment into a grid but instead randomly sample the space to construct a path: the samples are distributed in a manner that they avoid the obstacles and serve as a roadmap for to create a start-to-end path [15]. The exact approach to transform the roadmap into a continuous path depends on the algorithm like PRM or RRT.

Sampling-based algorithms present advantages like handling big environments and complex situations, however, they can also be computationally intensive and demanding in such situations. Figure 2.5 shows the difference in the resulting optimal path in a cluttered environment where with 1 core the algorithm successfully finds a feasible path, and, as the cores double, the computed path improves its quality as the search tree expands.

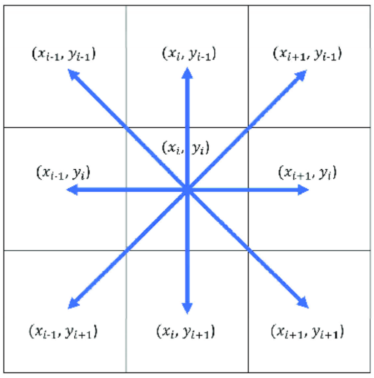

Figure 2.4: 8 direction possibilities to move from the current cell to the next cell [14]

As for artificial intelligence-based methods, also known as heuristic and metheuristic approaches, leverage the use of neural networks, machine learning and deep learning, and evolutionary algorithms. These methods learn from experience, adapt to changes, and optimize paths in complex environments. They are good at handling dynamic and unpredictable scenarios, making them suitable for tasks like autonomous driving, robot navigation, and logistics.

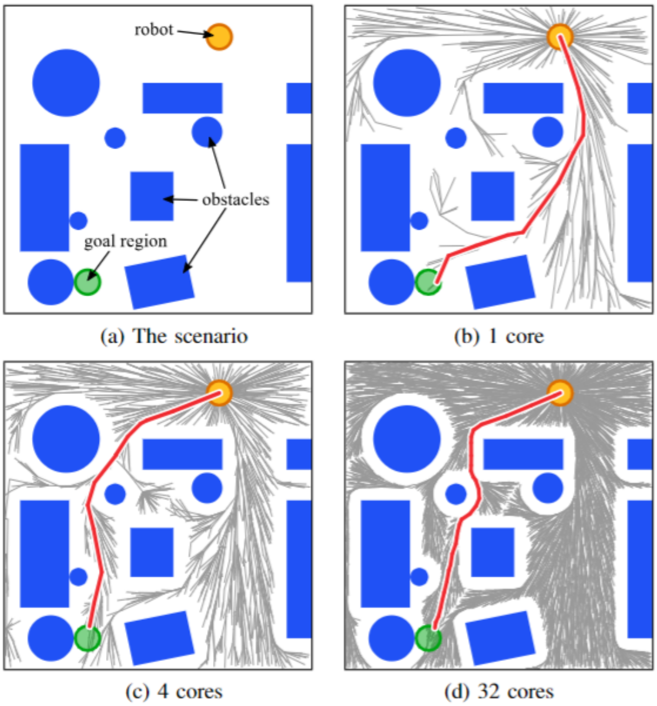

Figure 2.5: 8 direction possibilities to move from the current cell to the next cell [16]

To start with, Neural networks (NN) are inspired from the biological harmonious connection of the brain neurons that gives it the ability to process big amounts of data and generate ideas and decisions and solve problems. The NN is built in a way that enables it to get its input in a form of data and process it by learning, improving and adjusting the output to the desired results. NN are able to perform Parallel processing: the information is transmitted in two directions to the neuron in the case of Recurrent Neural Networks (RNN) that allows for learning fron current and past inputs to the NN. This approach improves the computation time and overall performance. The NN is then able to process complex solutions and create paths for difficult environments. It is well suited for unpredictable situations as it is built to adapt to the available input and the desired output. However, it presents a practicality challenge. Althugh the performance an results can be impressive, yet, in a real-time context, it is hard to rely on solutions that require extensive computational efforts and need long durations to process solutions. In addition, the number of neurons and layers have to be scalable and depends closely on the level of complexity that the vehicle is to deal with [12].

Fuzzy Logic (FL), on the other hand, is a way of thinking that mimics how humans make decisions, especially when things are unclear or uncertain. Instead of working with exact numbers, it uses "fuzzy" terms like "good", "average" or "bad" to make decisions. Input numbers are clustered following the fuzzy sets or intervals and assigned a value using membership functions. Later the values are interfered and defuzzified to genearte the ouput as a command value. In robotics and path planning, FL helps robots navigate by allowing them to handle uncertain situations, like avoiding obstacles or finding the best route, even when the environment is not fully known. Instead of needing exact data, the robot can employ fuzzy terms like "close" or "far" to understand its surroundings. For example, if an obstacle is "somewhat close," the robot can smoothly adjust its path to avoid it. FL also helps the robot choose the best route by weighing various factors like distance and safety, even when the information is not perfect. This makes the robot better at handling unpredictable environments and making flexible and optimal decisions. However, one disadvantage is that it can be tricky to create the right rules for the robot to follow, and the system can become complicated as more rules are added. It may present a scalability problem because in unpredictable and dynamic environments it is not simple to decide about fixed fuzzy sets that would be practical in all of the cases [12].

While Neural Networks and Fuzzy Logic offer powerful approaches for handling uncertainty and complex decision-making, another key area in artificial intelligence for path planning is the use of meta-heuristic algorithms. Meta-Heuristoc algorithms are inspired from biological and natural processes for evolution and survival. Unlike NN and FL, that rely on data as input to generate decisions, meta-heuristic algorithms explore the solution space by evolving random solutions for the problem and optimizing them by rounds until a stopping criterion or set of criteria is satisfied (see more in section ...: Optimization Algorithms). Approaches like Genetic Algorithms (GA) have been used for path planning by Ahmed Elshamli et al. [17]. The The robusteness of their solution is its adaption to dynamic environments. They evolve their GA using variable size chromosomes, where each node represents a waypoint of the path, then they measure the quality of each path using an evaluation function. A modified Genetic Approach is then applied to each population: Crossover, mutation, **Repair** infeasible paths, **Shorten**, then **Smoothen** feasible paths while dynamically checking for new obstacles. The approach is tested in static and dynamic environments and has achieved proven efficiency when it comes to local optima challenges and survivig the best elements of the population. While the use of the GA itself is robust and successful, it can be challenging to recreate the algorithm and to add the improvements. In addition, the tests the ran are limited to a simple simulation with unrealistic situations and obstacle setting as displayed in figure 2.9.

1.2.2 Local Path Planning

Local path planning has as the role to make real-time decisions about how to move safely and efficiently in their surroundings. Unlike global path planning, which focuses on finding the best overall route from a starting point to a destination based on a static map, local path planning deals with navigating the environment directly

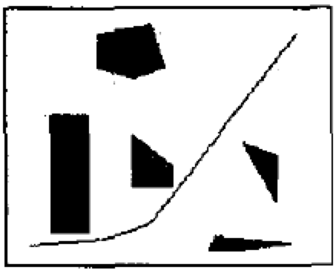

Figure 2.6: Simple obstacle environment

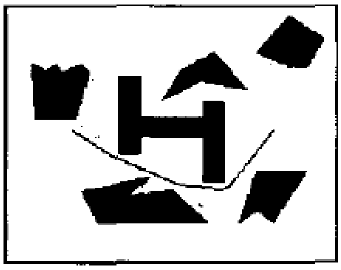

Figure 2.7: Intermediate obstacle environment

Figure 2.8: Complex obstacle environment

Figure 2.9: GA Test scenarios [17]

around the robot while considering real-time data input from sensors without prior knowledge about the surroundings[18]. This involves quickly responding to obstacles that appear or changes in the terrain, ensuring the robot can continue moving without collisions. Unlike Global Path planning, it functions without prior localization and mapping of the surrounding environment. Local path planning is crucial for the safe and effective operation of robots, especially in dynamic and unkown environments. For example, in busy spaces with moving people or vehicles, a robot needs to be able to make quick adjustments to avoid accidents. This type of planning allows the robot to react immediately to new information, making it essential for applications where unpredictability and quick responses are vital, such as in autonomous vehicles or service robots in warehouses and factories.

Common Techniques involve Reactive methods which are approaches in local path planning that focus on real-time obstacle avoidance and online adjustements on the path that are processed while the robot is navigating. Techniques like the Potential Fields, Dynamic Window Approach (DWA), and Bug Algorithms are commonly used. In their research paper [25], Buniyamin, N. et al. propose the point to point Bug algorithm to navigate in unkown environments. Bug algorithms are based on range sensors input. The robot at the starting point, plans to move directly to the target. Then it rotates scanning for obstacle points. If it encounters a sudden point, it navigates in its direction. The robot rotates in the target's direction until it is able to resume its direct path to the target based on the constant search for the shortest distance from the current standpoint or find the next obstacle. While this approach is simple from the computational point of view and the hardware used, it may not be optimal considering other metrics. Its path length optimality depends on the nature of the environments and its effciency decreases if the complexity of the test area increases. Local path planning approaches are effective in navigating through cluttered environments, but when it comes to long-range planning, a hybrid approach that integrates both local and global path planning strategies can provide more robust and efficient results.

1.2.3 Hybrid Path Planning

In recent years, path planning approaches that combine both global and local techniques have gained popularity. By integrating global and local path planning approaches, robots and autonomous vehicles can take advantage of the strengths of both methods. This hybrid approach is a promising area of research, with the potential to benefit a wide range of applications. Hybrid path planning approaches are becoming increasingly popular due to their ability to handle complex environments and dynamic obstacles.

In [19], Liu et al. used Djikstra ALgorithm for Global path planning and the DWA as the local path planner for sudden unkown obstacles that could appear for smart cars while following the global path. It works by evaluating different possible movements the robot could make within a short time frame and choosing the one that avoids obstacles while also moving towards the goal. The "window" refers to a limited set of possible velocities of the velocity space the robot can use based on its current speed and capabilities. Figure 2.10 details the flowchart of the DWA. The combined algorithms were tested on 3 spaces with 3 stages of environment complexity ranging from simple to complex. The tests showed that the hybrid path planning approach was efficient for navigating the robot and avoiding collisions. However, the tests were effected with dynamic obstacles only inside the simulation.

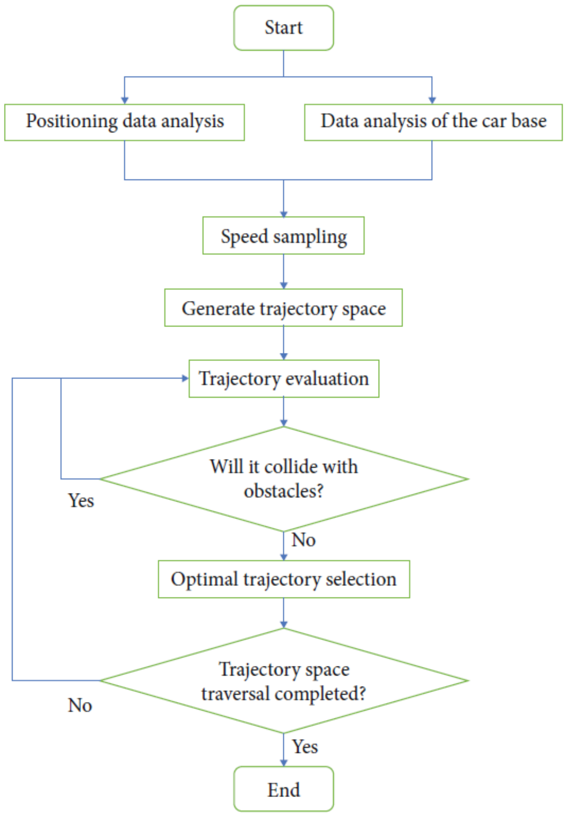

Figure 2.10: Flowchart of the DWA [19]

Sensor-based approaches, on the other hand, rely on data collected by the robot's sensors, such as LIDAR, sonar, or cameras, to make decisions about where to go next. These sensors help the robot create a map of its surroundings, which can then be used to identify obstacles and safe paths. Occupancy grids and Point cloud processing are techniques used to interpret the sensor data and guide the robot's movements. This approach allows the robot to have a detailed understanding of its immediate environment, making it more capable of avoiding obstacles and navigating complex spaces.

1.3 Discussion

While traditional path planning methods have proven effective across various applications and usecases, they often face challenges when applied to highly structured, repetitive environments like those found in industrial automation. To overcome these limitations, the research in the previous years became rather oriented to investigating and using heuristic approaches. In [26], Masehian et al. conducted a chronological review about the followe approaches in Motion Planning (MP). The results testify that in 30 years, The application of Heuristic approaches in MP went from 0% to 54% from 1977 to 2007 as shown on Figure 2.11.

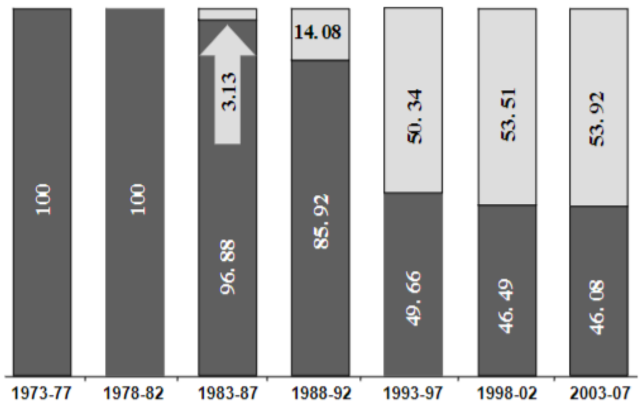

Figure 2.11: Application of classic and heuristic approaches in MP [26]
Dark gray: Classic approaches
Light gray: Heuristic approaches

The classic algorithms have the drawback of traps in local minima and increased complexity in sophisticated environments. In practice, the conditions where the robot operates are cluttered and complex than we can imagine which makes the classic approaches impractical. Consequently, it is necessary to tackle these issues in a different way. Although probabilistic methods proved high computational capabilities, scientists tend to combine 2 algorithms to benefit from the classic and heuristic advantages simultaneously. In [27], Chen et al. delve into a hybrid approach for online motion planning. Their approach is based on space exploration and heuristic search to determine optimal path linking a start point to an end goal. Tested on both low and high speed scenarios, the algorithm ensured a low planning time: it outperformed RRT and had similar results in terms of planning time to Hybrid A*.

To draw to a close, even though researches confirm that the heuristic approach prevails the classic one: [20] and [25], these approaches are not without their limitations. Heuristic methods, while good at navigating complex environments and avoiding local minima, sometimes fall short in finding the best possible solutions, especially in high-dimensional spaces. In contrast, classical methods, though computationally expensive, provide a more systematic and theoretically solid approach to path planning. To address these trade-offs, there is growing interest in hybrid methods that combine the strengths of both approaches. These hybrids use the precision and reliability of classical methods along with the flexibility and speed of heuristics. By doing so, they reduce the weaknesses of each approach, offering a more balanced and effective solution for path planning in complex environments. As a result, hybrid methods hold promise for future research, potentially leading to more efficient and dependable path planning algorithms.

2 Spline based Paths

This section will closely examine splines, their different types, their application in robotics and the importance of their integration and detail the challenges and limitations related to the use of Splines for robotic motion planning, and conclude by a discussion around their relevance to this work.

2.1 Definitions and Basic Concepts

In mathematics and computer graphics, a **curve** is a continuous and smooth flowing line without sharp angles. It can be defined parametrically or implicitly and represents a path that can be traced by a moving point. A curve can be defined using a parameter u, where $u$varies over an interval, and the coordinates of the curve are given by functions of:

$$C(u) = (x(u), y(u)) \quad \text{given} \quad a \leq u \leq b \tag{2.1}$$

The circle is an example of a curve where:

$$x(u) = \cos(u) \tag{2.2}$$

$$y(u) = \sin(u) \quad \text{given} \quad 0 \leq u \leq \frac{\pi}{2}[29] \tag{2.3}$$

whereas, the implicit form of the circle would be:

$$(x - h)^2 + (y - k)^2 = r^2 \tag{2.4}$$

where (h, k) is the center and r is the radius.

Parametric curves have a clear direction (from C(a) to C(b)), which implicit curves lack. This makes it simpler to create ordered sequences of points along a parametric curve. Additionally, the parametric form is more intuitive for designing and modeling shapes on a computer, as the coefficients in many parametric functions (such as Bezier and B-Splines) carry geometric significance. This results in user-friendly design techniques and algorithms that are numerically stable [28]. However, using one polynome-curves is inadequate as it is not possible to represent complex shapes, certain curve changes, and fitting the needed points. The solution is to use piecewise polynomial curves of degree $(n - 1)$ for example if we need to integrate n data points [29].

Polynomials are a commonly used type of function in robotics because they can be easily differentiated and are less prone to numerical rounding errors (known as floating point errors). Although they cannot represent every geometric curve, polynomials can usually approximate them with sufficient accuracy.

Using classical polynomial functions or their derivatives in Bézier form is mathematically equivalent. This means that any curve described using standard polynomials can be transformed into a Bézier curve, and vice versa. However, when it comes to geometric modeling—especially in applications like computer graphics or robotics—the Bézier form is often preferred. This preference is due to the fact that the coefficients in the standard polynomial form (also known as the power basis form) don't offer much information about the curve's actual shape, making it harder to intuitively control and adjust the curve. In contrast, the coefficients in the Bézier form -the control points- directly influence the shape of the curve, in a visually meaningful way, making it easier to understand and manipulate as shown in equation 2.5 [28].

$$C(u) = \sum_{i=0}^{n} B_{i,n}(u)\mathbf{P}_i \quad \text{with } 0 \leq u \leq 1 \tag{2.5}$$

The basis function $B_{i,n}(u)$ is called the Bernstein polynomial of degree n and follows the equation

$$B_{i,n}(u) = \frac{n!}{i!(n-i)!} \cdot u^i \cdot (1-u)^{n-i} \tag{2.6}$$

In this context, the Bernstein polynomial is a fundamental component. The basis function $B_{i,n}(u)$ is used to determine how much influence each control point $\mathbf{P}_i$ has on the shape of the Bézier curve.

The geometric coefficients $\mathbf{P}_i$ are known as control points, and they define the shape of the polynomial curve. In many applications, such as mapping trajectories, a large number of control points is needed [28]. Furthermore, in the Bézier form, the continuity of the curve depends on the placement of control points. This means if you want to change the shape of part of the curve while keeping the rest smooth, you can't easily do so because changing one part affects the whole curve [29].

Instead, a more effective curve representation can be expressed as:

$$C(u) = \sum_{i=0}^{n} N_i(u)\mathbf{P}_i \tag{2.7}$$

where:

- $\mathbf{P}_i$: the control points
- $N_i(u)$: the piecewise polynomial functions

The continuity of the curve is determined by these basis functions, allowing for flexible modification of control points without affecting the curve's smoothness [29].

The basis functions for B-splines are defined recursively. For a given degree p and a set of non-decreasing knot values $\{u_i\}$, the basis functions $N_{i,p}(u)$ can be defined as:

$$N_{i,0}(u) = \begin{cases} 1 & \text{if } u_i \leq u < u_{i+1} \\ 0 & \text{otherwise} \end{cases} \tag{2.8}$$

$$N_{i,p}(u) = \frac{u - u_i}{u_{i+p} - u_i} N_{i,p-1}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u) \tag{2.9}$$

With:

- u is the parameter along the curve.
- u_i are the knot values, which divide the parameter range into intervals.
- $N_{i,p}(u)$ are the B-spline basis functions of degree p.

In figure 2.12 stands an example of a NURBS spline generated through setting random control points using th NURBS generator [W1]

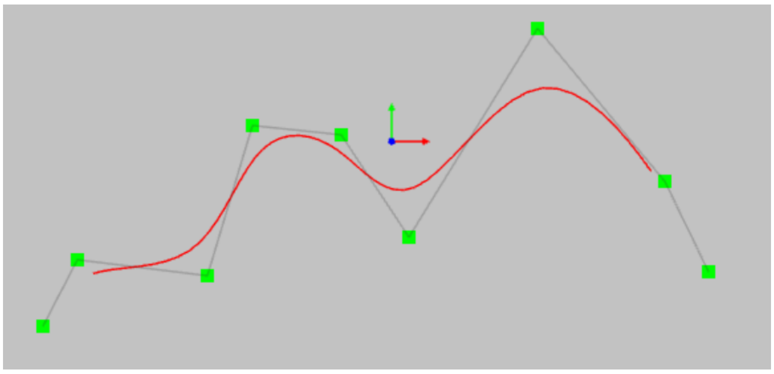

Figure 2.12: NURBS spline example
Green: Control points
Red: NURBS spline

Understanding these principles is essential as we move forward to explore the applications of splines in robotics. In the next section, we will examine how these spline concepts are applied to solve real-world problems in robotic path planning and motion control.

2.2 Applications of Splines in Robotics

2.2.1 Trajectory Planning

Robots like AMRs in the intralogistics sector usually have mission to carry heavy loads. This property makes it risky to operate abrupt motion changes like stopping or turning. These vehicles' motion planning requires precise control and detailed alignment to the kinematic constraints that they present [30]. Given the mathematical nature of B-Splines, they ensure continuity of the first and second derivatives. This continuity translates to smooth transitions of the resulting velocities and accelerations of the path that the robot will follow. By having continuous velocity and acceleration, we make sure that the robot achieves smooth transitions and speed changes. Furthermore, tracking, precise following and correct control interventions are guaranteed allowing for precise navigation [30].

Although simple, the connected waypoints approach comes with drawbacks like sub-optimal travel time, unoptimized long paths, and mechanical wear because of abrupt direction changes unlike curved paths [30]. The morphological difference in both paths is distinguished in figure 2.13

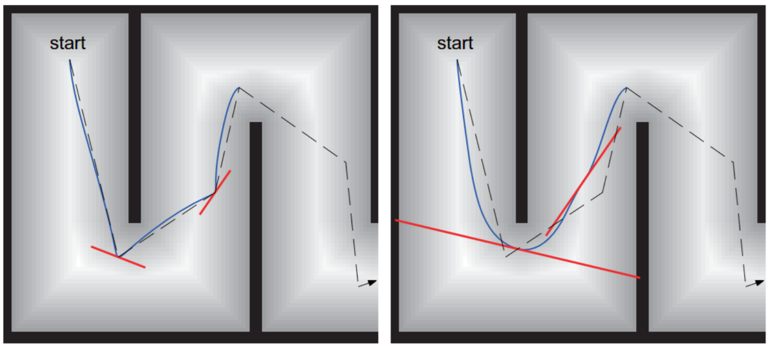

Figure 2.13: Morphological difference between a path connecting waypoints (left) and a curved path (right) [30]

Multiple approaches have been developed to introduce smooth transitions and turns into robot trajectories. One such method involves using Clothoid curves, which connect straight path segments with circular arcs to avoid abrupt changes in direction [31]. Originally introduced for designing highways and roads to ensure safe and comfortable driving for humans, Clothoids offer a gradual change in curvature.

However, when applied to robotics, Clothoid curves present certain drawbacks. These include the potential for discontinuities in curvature at the transitions, which can lead to jerky motions in robotic paths. Additionally, Clothoid-based paths can result in sub-optimal path lengths, as they may not be the shortest possible routes. Moreover, while circular arcs with constant radii are used in this approach, longer arcs lead to larger curvatures, which can be impractical for mobile robots that require tighter turns and more precise control [31].

Migrating from a path with sharp angles to one with smooth curves can be effectively accomplished by using splines. The interpolation of the control points of the original path creates a smooth, continuous curve. By incorporating a convex hull around these control points, we ensure that the path does not deviate excessively from the desired points, thus avoiding potential collisions with obstacles [33]. The convex hull acts as a boundary that keeps the spline inside the limits of the control points that prametrize it (see figure 2.14) and keeps the robot from straying too far, ensuring a safe and accurate trajectory [29].

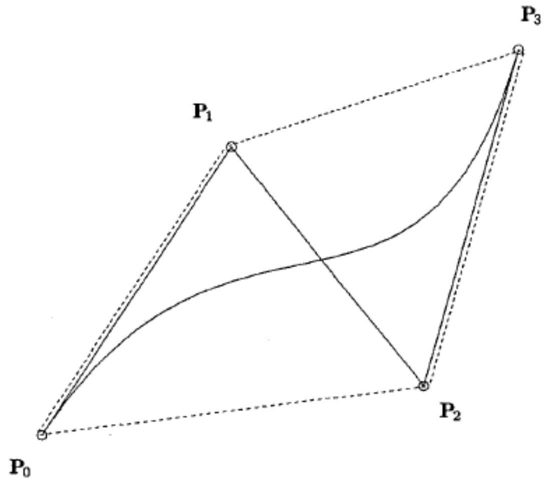

Figure 2.14: Convex hull contained Spline [29]

Splines are advantageous because they can accommodate various constraints such as curvature limits and minimum turning radii. These constraints are crucial in many applications, including robotics and vehicle navigation, where maintaining a smooth path is essential for stability and control. By adjusting the spline parameters, it is possible to design paths that meet specific requirements, ensuring that the resulting trajectory is not only smooth but also feasible given the physical and operational constraints [30].

In scenarios involving obstacle avoidance, splines can be blended or connected to create a seamless path while circumventing obstacles. This blending technique allows for the integration of different spline segments into a continuous trajectory, ensuring that the path remains clear of obstacles. By carefully adjusting the blend points and ensuring smooth transitions between splines, it is possible to navigate complex environments safely and efficiently. This approach combines the advantages of smooth path planning with effective obstacle avoidance.

A notable case study of splines integration in solving path planning problems, is a research conducted by B. Lau et al. [30] that aims to develop a time optimal solution while considering the kinodynamic properties of a mobile robot. They used a global planner to generate straight-line paths and minimize time of travel. Then, they integrated splines to join the resulting path segments in a smooth and continuous way and replan in case of unprecedented collisions. The results show that our motion planning system works well in both real and simulated settings. It created smooth and accurate paths, with an average positional error of about 1 cm and a velocity error of less than 2 cm/s. The optimization process improved travel time by 31% compared to initial paths. The system also handled dynamic environments, like crowded trade shows, and adjusted smoothly when localization errors occurred by updating the path as needed. Overall, it performed reliably, navigating precisely and adapting effectively to changes and errors.

2.3 Discussion

In general, Splines are an effective tool that centralizes advantages related to path smoothness, simplicity of constraining the path according to the kinodynamic properties and mechanical limitations of the mobile robot like limiting the curvature, and allowing for safe and smooth speed and direction changes. These properties accommodate simple manipulation of splines for path planning and tracking of path properties. As an illustration, it is possible to measure the spline's features like curvature or approximate its length. Figure 2.15, shows an example plot of the curvatures of 3 random splines.

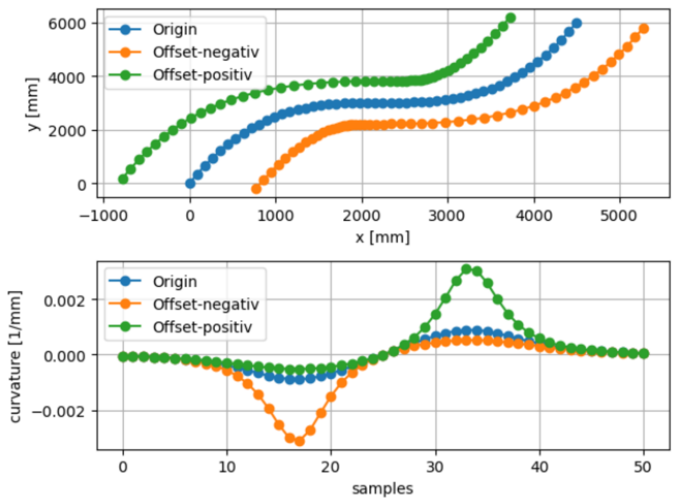

Figure 2.15: Curvature of 3 S-shaped splines

In the next section, we discuss the ways we can use such information about splines to measure paths quality.

3 Evaluating Path Efficiency: Key Metrics

When planning paths for robots in the Intralogistics or in any field in general, it is important to have the ability to quantify the quality of the resulting path. There exists different ways to judge the efficiency of the solution and to which extent it accomodates the goal and optimizes the gaps that we are willing to improve. This section studies the strategies that the literature followed while quantifying path quality in the robotics field. It will look at the different indicators, why, and in which case they were used.

3.1 Core Metrics for Path Evaluation

The key metrics that researchers focus on and try to improve can vary depending on the specific areas of optimization they are working on. For example, if the goal is to make a robot move faster, then time and speed might be the main metrics they study. On the other hand, if the focus is on making the robot navigate more safely, then the proximity to obstacles and collision avoidance would become the primary metrics. Therefore, the choice of metrics is closely linked to the particular goals that researchers aim to achieve in their optimization efforts.

According to Tang, S. H. et al. [20], from 2011 to 2015, the second most common topic in the path planning research scope was path length. Path length is a key metric that helps us understand other important factors like travel time and energy consumption. By measuring the length of a path, we can get insights into how long it will take to travel and how much energy the robot might use. In other words, path length often reflects the overall efficiency and effectiveness of the route. Heiden, E et al. [23], employed path length as of the metrics used to evaluate the length of the paths generated by the algorithms that they compared against each other. Their planners were configured to minimize path length, therefore, they measured the path lengths that were computed in 4 scenarios with applied to 17 path planners and 4 steering algorithms. They used path length, along other metrics, to quantify the quality of paths after palnning and smoothing. Besides, they compared the algorithms using the above-mentioned metrics in order to classify them under the benchmark and to justify the choice of the best algorithm.

In addition to path length, Path smoothness is crucial for robotics in general and especially in the intralogistics sector because it directly affects the efficiency and safety of robot operations. Smooth paths reduce the need for sudden stops or sharp turns, which helps robots maintain a steady speed and avoid unnecessary wear on their components. In intralogistics, where robots often work in busy environments with other machines and human workers, smooth paths also minimize the risk of accidents and product damage. Overall, ensuring path smoothness helps robots operate more reliably and effectively, leading to faster and more accurate material handling. Path smoothness reflects less cusps -sharp turns- and generally less curvature which allows for smooth driving and smaller scale steering. Curvature of a circle at a point $x(t)$ is defined according to Gary D. Knott, in his book "Interpolating Cubic Splines" [34], as the reciprocal of the radius at the same point as shown in equation 2.10.

$$\kappa = \frac{1}{R} \tag{2.10}$$

For a cubic spline $y = f(x)$, the curvature κ at a point x is given by:

$$\kappa = \frac{|f''(x)|}{\left(1 + (f'(x))^2\right)^{3/2}} \tag{2.11}$$

The unsigned curvature indicating whether the curve is concave or convexe is given by:

$$\kappa = \frac{f''(x)}{\left(1 + (f'(x))^2\right)^{3/2}} \tag{2.12}$$

Heiden, E et al. [23], also relied on curvature to determine path quality. They used it along path length and tracked it for the different scenarios and algorithms to measure their effectiveness. Liu, S. et al. integrated path smoothness metric to quantify path quality for their benchmark to compare motion planning algorithms[35]. Although the above-mentioned sources used the metrics to evaluate their optimized solutions, they did not mention the techniques followed to measure the metrics.

Path clearance is another significant metric used to evaluate how well a robot can safely navigate its environment. It measures the distance between the robot's position along the path and any obstacles it might encounter. A larger path clearance means the robot has more space around it, reducing the risk of collisions and making the path safer. This metric is especially useful when optimizing paths, as it helps ensure that the robot can move smoothly without bumping into objects or other robots. Elshamli, A. et al. [17] have used path clearance as one of the indicators of path feasibility. Clearance is measured as follows in 2.13:

$$\sum_{i=1}^{n-2} \exp\left(a \cdot (g_i - \tau)\right) \tag{2.13}$$

where g_i is the shortest distance from the i^{th} discrete position of the robot to the surrounding obstacles, and τ is the desired clearance distance that guarantees safety. By focusing on path clearance, we can design routes that not only get the robot from one point to another efficiently but also do so in a way that minimizes risks and improves overall safety.

3.2 Combinations of single metrics

Some applications focus solely on minimizing path length. They use this single value as the metric for evaluating and distinguishing path solutions or for optimization. However, other applications require the combination of several metrics. This combination invites us to look at possible ways to blend the measured value into one significant value that reflects path quality. Elshamli, A. et al. [17] used a weighted sum of the Distance between two nodes, or two waypoints, Path smoothness, and clearance as given by 2.14.

$$\text{eval}(p) = w_d \cdot \text{dist}(p) + w_s \cdot \text{smooth}(p) + w_c \cdot \text{clear}(p) \tag{2.14}$$

The outcome of this evaluation function of each path is used as the fitness function for that path. The fitness of each path serves later the evolution of the genetic algorithm.

Zhang B et al. [36], deployed a nonlinear optimization problem that integrates path length and curvature variation over a spline-based path. The problem is constrained by the curvature limits, the parametrized location of the spline's control points, and the distance to obstacles.

Combining metrics for path evaluation should be done with a simple approach. It is advisable to limit the number of combined metrics to 2 or 3. Using more metrics can make the optimization process complex and slow, as the optimizer has to handle more variables. Additionally, with too many metrics, it becomes challenging to understand how each one influences the results. By keeping the metrics to a manageable number, we ensure that the optimization remains efficient and the impact of each metric is clear and easy to interpret. Using a more metrics like computational time, number of cusps or direction changes, solvability percentage or predicted speed or travel time can be efficient for evaluating path planning algorithms and benchmarking some usecases as in [23].

4 Heuristic Approaches to Path Optimization in Robotics

This section will focus on the Application of heuristic algorithms for solving robotic path planning problematics.

4.1

Optimization refers to the decision making led by the machine with availability of certain circumstances and constraints thet either help or limit the outcomes of the optimizer [37]. In the case of path planning, optimization aims to modify some properties of the generated solutions to cope with the surrounding conditions. In stochastic environments, it is challenging to plan feasible paths. The increase in the complexity of the environment and the constant dynamics increase the number of variables involved in the mathematical representation of the problem. As a consequence, it raises the computation efforts that need to be deployed for the creation and optimization of the solution and decreases feasibility [7]. The integration of an optimizer that takes charge of the solution and fits it into the constraints that bound it is necssary. In order to get optimal results like smooth and short paths, finding a feasible path is not enough. The evaluation of the generated path and the optimization are helpful to ensure that the robot is driving the optimal path. There exists a range of planners and optimizers which are clustered as classic and Heuristic approaches [12]. Classic approaches include **Analytical** and **Enumerative** methods. The former rely on mathematical modeling, which can become increasingly complex as the navigation environment becomes more cluttered. Additionally, applying these models can be challenging in scenarios where the necessary models for different components are not readily available. The latter have the drawback of increased intricacy in bigger or more elaborate search spaces. On the other hand, Heuristic approaches can be sub-categorized into **Meta-Heuristic** and **Evolutionary algorithms**. Some of these methods can fall in the trap of local minima and sub-optimality [12] Alternatively, sebsequent to a review of the available planning and control approaches in intralogistics, Fragapane, G. et al. [7], concluded that nature-inspired algorithms can instill intelligence into planned paths. Elshamli, A. et al. explained in [17], that Meta-Heuristics are adaptive to the dynamics of the surroundings unlike classic approaches that proceed sequentially to generate solutions. The paths developed by classic approaches can thus become infeasible in a later stage if the environment changes. Metaheuristic approaches are based on parallel search mechanisms that can have updated input of the conditions and thus are more adaptive and practical in real-world scenarios.

4.2 Heuristic Optimization Algorithms : An Overview

In this section, we will dive into Meta-Heuristics which are defined by Bilal et al. as 'the optimization techniques mainly based on function evaluation and make little or no use of the properties of objective functions and constraints. Meta-heuristics are thus problem independent techniques not taking advantage of any specificity of the problem' in their review [37]. The authors later break down Meta-Heuristics into **Neighborhood-based Algorithms** and **Population-based Algorithms** as given by the tree graph in Figure 2.16

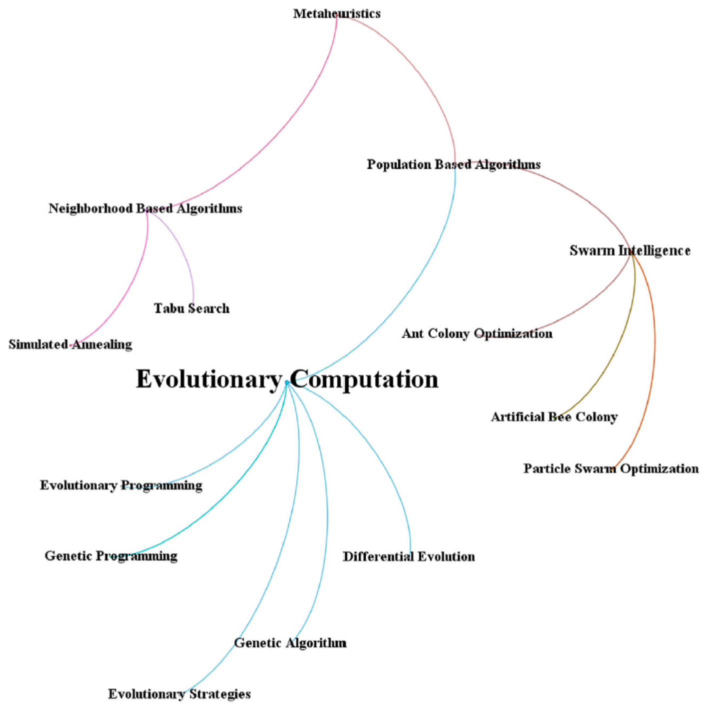

Figure 2.16: Tree of Meta-Heuristic Algorithms [37]

The Neighborhood-based algorithms benefit from the neighboring solutions by migrating from the current solution to the near surrounding solutions with the goal of improving the fitness function. The examples mentioned for this approach are Simulated Annealing (1979) and Tabu Search (1989). The Population-based algorithms are methods that start from a sample population and rely on their evolutions. They are inspired by the evolutionary processes observed in nature and are sub-categorized in this approach under Swarm intelligence and Evolutionary Algorithms. Swarm Intelligence approaches are analogous to the socio-cooperative practices of species like bees, ants, and birds. For instance, Ant Colony Optimization (1992)and Particle Swarm Optimization (1995) are examples of such algorithms. On the other hand, Evolutionary algorithms rather copy the species evolution theory, among which are Differential Evolution (1995) and Genetic Algorithm (1957) [37].

4.3 Application of Heuristic Optimization in Robotic Path Planning

These algorithms have been used multiple times for robotic path planning, as noted in the literature. Each algorithm has its own unique search methods and strengths, providing different benefits and areas for optimization.

Elshamli, A. et al. [17] have employed the Genetic Algorithm (See algorithm steps in figure 2.17) to solve the problem of path planning in dynamic environments. They represent their paths using chromosomes where each node of the chromosome represents a waypoint and develop a variable chromosome length approach that accommodates obstacle avoidance and efficiency in reaching the goal. The fitness of each chromosome is evaluated according to an objective function that includes path length, smoothness, and clearance to obstacles. A modified genetic Algorithm is applied where after crossover and mutation, paths are shortened and smoothed if it is possible. The implemented strategies guaranteed low convergence for the different complexity stages. Saving the best individuals generated by the GA was effective in delivering the overall best solutions. Tamilselvi, D. et al. also modified the GA by applying the elitism concept. They save the best solution S_{best} then they replace the worst element of the next population with S_{best}. This approach guarantees landing on a global optimum and was able to find feasible paths for environments with limited dynamic obstacles and integrate a motion prediction mechanism for dynamic obstacles based on the grid map [38].

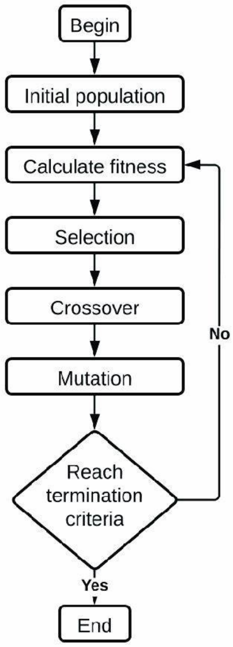

Figure 2.17: Flowchart of the GA [39]

Miao, H. et al. also worked on the path planning in dynamic environments problem. They crtic the GA to be computationally intensive and to require long planning time [40]. Instead they suggest the Simulated Annealing algorithm (SA) to solve the path finding problem.

Being a probabilistic meta-heuristic algorithm it is designed to guide the local optimum to a global optimum. The approach is based on the evaluation of the path length to discriminate path candidates: the shorter the path the better. It works by trying out different paths for the robot, beginning with a high "temperature" that lets the algorithm accept both good and bad paths. This helps it explore a wide range of options and avoid getting stuck in a local minima. As the process continues, the temperature slowly decreases, making the algorithm less likely to accept bad paths, and helping it to focus on finding the global minima as given by figure 2.18. algorithm was tested in four environments with different static and moving obstacles. The algorithm successfully found the best or nearly best paths and could quickly adjust to avoid collisions in real-time. Compared to a genetic algorithm, the SA method was 57% faster for complex environments and 74% faster for simple environments. However, the processing time increases exponentially for complex environments (13.57 seconds).

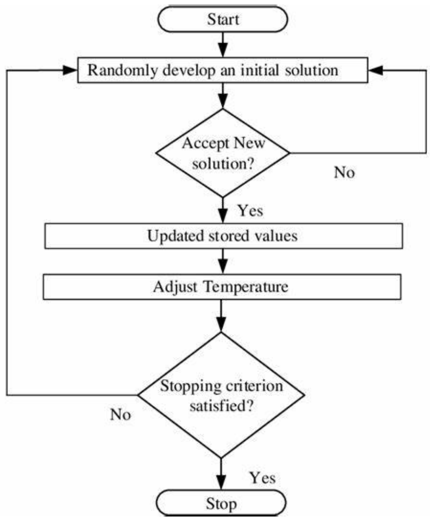

Figure 2.18: Flowchart of the SA [41]

Particle Swarm Optimization Algorithm (PSO), as well, is used for solving non-linear optimization problems like scheduling, power management and also robotic path planning. It is inspired from the cohesive behavior of bird and fish swarms traveling together.It works by having a group of particles (potential solutions) move around in the search space to find the best solution. Each particle in the PSO algorithm remembers the coordinates of the best solution it has found so far, known as its personal best (pbest). Additionally, the algorithm tracks the best solution found by any particle, referred to as the global best (gbest). The core idea of PSO is to guide each particle towards both its pbest and the gbest positions: from state a to c as given by figure 2.19, using a randomly weighted acceleration during each iteration. The algorithm iteratively updates the particles' positions and velocities until an optimal or near-optimal solution is found. Alaliyat, S. et al. have investigated dynamic environment navigation using PSO. They found out that the performance of the algorithm depends on parameters tuning. Overall, feasible paths were successfully generated. Planning time is rather high for simply organized environments (18.35s) and higher for the most complicated scenario (31.17s).

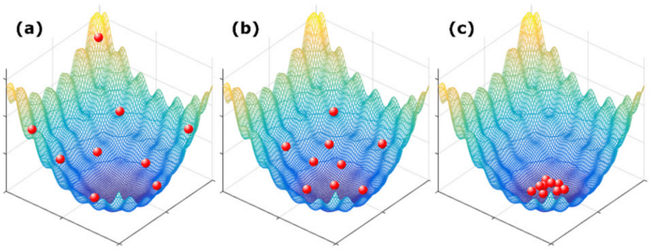

Figure 2.19: Particle Swarm Optimization [42]

Defferential Evolution (DE) has been used in a hybrid algorithm along with PSO by Tand, B. et al. [43] to leverage both of the algorithms benefits. DE is rarely solely applied to path planning problems. In their review [37] , of 20 years of research about DE, Bilal et al. did not mention robotics among the fields of application of DE. DE starts with a set of potential solutions. Each solution, or individual, is modified through mutation, crossover, and selection to improve over time. Mutation introduces random changes to create new candidates and avoid getting stuck in local optima. Crossover mixes parts of the new candidates with the original solutions to explore new possibilities. The selection process keeps only the better solutions for the next generation. This method effectively balances exploring new options and refining existing ones, making DE well-suited for complex optimization tasks. By integrating improved PSO and DE algorithms for enhancing particle diversity, Tand, B. et al. achieved better path quality compared to other methods. The results demonstrate that their approach outperforms traditional evolutionary algorithms in path optimality an optimize computation times by 1.38% [43].

Ant Colony Optimization (ACO) is known for being robust and good at parallel processing, but it can be slow and sometimes gets stuck in local optima solutions. To fix this, improvements like Ant Colony System (ACS) and Max-Min Ant System (MMAS) have been developed, which make ACO better at finding solutions but still have issues like being too rigid and converging too early. In mobile robot path planning, new strategies have been created to address these problems, such as adaptive methods that improve how pheromones are updated and balance exploration with finding the best path. Recent research has also introduced advanced versions like polymorphic ACO and multi-objective optimization, making ACO more effective for real-time and complex situations. However, these improvements also make the algorithm more complex and take more time to compute.

Meta-heuristic algorithms, such as Genetic Algorithms (GA), Particle Swarm Optimization (PSO), and Differential Evolution (DE), have become increasingly popular in robotic path planning. These algorithms are attractive because they can find good solutions to complex problems by mimicking natural processes like evolution or swarming behavior. However, despite their growing use in research and their potential efficiency in practical usecases, the application of meta-heuristic algorithms in real-world robotic path planning like the intralogistics field remains quite limited. One of the major issues is that most studies focus on simulations rather than testing in real environments. While simulations can help develop and adjust these algorithms, they do not fully cover the challenges robots face in the real world, such as varying conditions, surface types, or unexpected noises. Another major issue is how obstacles are represented in these studies. Often, obstacles are either unrealistically large compared to the robot or have overly simple shapes that don't accurately reflect real-world conditions. This lack of realism can result in algorithms that work well in simulations but fail when used in real environments. Additionally, because these algorithms are mostly tested in controlled, simulated settings, they may be too finely tuned to those specific conditions. This can make them less reliable when applied to the unpredictable conditions of the real world. In summary, while meta-heuristic algorithms show promise for robotic path planning, more real-world testing and realistic obstacle modeling are needed to ensure they work effectively in practical situations. Addressing these issues is key to advancing the use of robotics in intralogistics and other fields.

5 Predictability in intralogistics: why is it important?

6 Summary and Discussion

Chapter 3

Chapter 3

chapter name

Introduction

Chapter 4

Chapter 4

chapter name

Introduction

General conclusion

Bibliography

[1] KION group website. URL: https://www.kiongroup.com/en/About-us/Management/ consulted on 19/07/2024

[2] KION group website. URL: https://www.kiongroup.com/en/About-us/KION-at-a-glance/ consulted on 19/08/2024

[3] STILL website, iGo Neo page. URL: https://www.still.de/en-DE/trucks/new-trucks/order-pickers/opx-20-25-igo-neo.html

[4] STILL website, iGo Neo page. URL: https://www.still.de/en-DE/trucks/new-trucks.html

[5] iGo Neo datasheet. URL: data.still.de/assets/products/Vehicles/Order_Pickers/OPX_iGo_neo/pdfs/OPX_EN_TD.pdf?mod=1681891889&s=baf72412ac0d268f13b017603ca95ea2

[6] Agile Scrum article. URL: https://top20review.com/phuong-phap-agile/

[7] Giuseppe Fragapane, René de Koster, Fabio Sgarbossa, Jan Ola Strandhagen, Planning and control of autonomous mobile robots for intralogistics: Literature review and research agenda, European Journal of Operational Research, Volume 294, Issue 2, 2021, Pages 405-426, ISSN 0377-2217, https://doi.org/10.1016/j.ejor.2021.01.019.

[8] Moshayedi, Ata Jahangir & Liao, Liefa & Kolahdooz, Amin. (2021). Gentle Survey on MIR Industrial Service Robots: Review & Design. 10.

[9] AMR vs. AGV Figure, Vecna Robotics Website. URL: https://www.vecnarobotics.com/amr-vs-agv/, consulted on 28.07.2024

[10] Lixing Liu, Xu Wang, Xin Yang, Hongjie Liu, Jianping Li, Pengfei Wang, Path planning techniques for mobile robots: Review and prospect, Expert Systems with Applications, Volume 227, 2023, 120254, ISSN 0957-4174, https://doi.org/10.1016/j.eswa.2023.120254.

[11] Marín, Pablo & Hussein, Ahmed & Martín Gómez, David & de la Escalera, Arturo. (2018). Global and Local Path Planning Study in a ROS-Based Research

Platform for Autonomous Vehicles. Journal of Advanced Transportation. 2018. 1-10. 10.1155/2018/6392697.

[12] Mohd. Nayab Zafar, J.C. Mohanta, Methodology for Path Planning and Optimization of Mobile Robots: A Review, Procedia Computer Science, Volume 133, 2018, Pages 141-152, ISSN 1877-0509, https://doi.org/10.1016/j.procs.2018.07.018.

[13] Zhang B, Zhu D. A new method on motion planning for mobile robots using jump point search and Bezier curves. International Journal of Advanced Robotic Systems. 2021;18(4). doi:10.1177/17298814211019220

[14] H. -T. Lee, H. -M. Choi, J. -S. Lee, H. Yang and I. -S. Cho, "Generation of Ship's Passage Plan Using Data-Driven Shortest Path Algorithms," in IEEE Access, vol. 10, pp. 126217-126231, 2022, doi: 10.1109/ACCESS.2022.3225571.

[15] Sertac Karaman and Emilio Frazzoli. 2011. Sampling-based algorithms for optimal motion planning. Int. J. Rob. Res. 30, 7 (June 2011), 846–894. https://doi.org/10.1177/0278364911406761

[16] c. Ichnowski and c. Alterovitz, "Parallel sampling-based motion planning with superlinear speedup," 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, Vilamoura-Algarve, Portugal, 2012, pp. 1206-1212, doi: 10.1109/IROS.2012.6386194.

[17] A. Elshamli, H. A. Abdullah and S. Areibi, "Genetic algorithm for dynamic path planning," Canadian Conference on Electrical and Computer Engineering 2004 (IEEE Cat. No.04CH37513), Niagara Falls, ON, Canada, 2004, pp. 677-680 Vol.2, doi: 10.1109/CCECE.2004.1345203.

[18] Q. Jin, C. Tang and W. Cai, "Research on Dynamic Path Planning Based on the Fusion Algorithm of Improved Ant Colony Optimization and Rolling Window Method," in IEEE Access, vol. 10, pp. 28322-28332, 2022, doi: 10.1109/ACCESS.2021.3064831

[19] Liu, Li-sang, Lin, Jia-feng, Yao, Jin-xin, He, Dong-wei, Zheng, Ji-shi, Huang, Jing, Shi, Peng, Path Planning for Smart Car Based on Dijkstra Algorithm and Dynamic Window Approach, Wireless Communications and Mobile Computing, 2021, 8881684, 12 pages, 2021. https://doi.org/10.1155/2021/8881684

[20] S. H. Tang, F. Kamil, W. Khaksar, N. Zulkifli and S. A. Ahmad, "Robotic motion planning in unknown dynamic environments: Existing approaches and challenges," 2015 IEEE International Symposium on Robotics and Intelligent Sensors (IRIS), Langkawi, Malaysia, 2015, pp. 288-294, doi: 10.1109/IRIS.2015.7451627.

[21] Alterovitz, R., Koenig, S., & Likhachev, M. (2016). Robot Planning in the Real World: Research Challenges and Opportunities. AI Magazine, 37(2), 76-84. https://doi.org/10.1609/aimag.v37i2.2651

[22] Ohki, T., Nagatani, K., & Yoshida, K. (2012). Local Path Planner for Mobile Robot in Dynamic Environment based on Distance Time Transform Method. Advanced Robotics, 26(14), 1623–1647. https://doi.org/10.1080/01691864.2012.694648

[23] Heiden, E., Palmieri, L., Arras, K.O., Sukhatme, G.S., & Koenig, S. (2020). Experimental Comparison of Global Motion Planning Algorithms for Wheeled Mobile Robots. ArXiv, abs/2003.03543.

[24] Harris-Birtill , D & Harris-Birtill , R 2021 , Understanding computation time : a critical discussion of time as a computational performance metric . in A Misztal , P A Harris & J A Parker (eds) , Time in variance . The study of time , vol. 17 , Brill , Leiden , pp. 220-248 , The 17th triennial conference of the International Society for the Study of Time , California , California , United States, 2019. https://doi.org/10.1163/9789004470170_014

[25] Buniyamin, N., Ngah, W. W., Sariff, N., & Mohamad, Z. (2011). A simple local path planning algorithm for autonomous mobile robots. International journal of systems applications, Engineering & development, 5(2), 151-159.

[26] Masehian, Ellips & Sedighizadeh, D.. (2007). Classic and Heuristic Approaches in Robot Motion Planning - A Chronological Review. World Academy of Science, Engineering and Technology. 29.

[27] C. Chen, M. Rickert and A. Knoll, "Combining space exploration and heuristic search in online motion planning for nonholonomic vehicles," 2013 IEEE Intelligent Vehicles Symposium (IV), Gold Coast, QLD, Australia, 2013, pp. 1307-1312, doi: 10.1109/IVS.2013.6629647.

[28] Krüger-Basjmeleh, T. (2024). Befähigung autonomer, mobiler roboter zum one-shot-imitationslernen für Den vollständigen transport von paletten in lager- und produktionsbereichen Tino Krüger-Basjmeleh. Verlag Dr. Hut.

[29] Piegl, Les , and Wayne Tiller. The NURBS Book. 2nd ed., Springer Berlin, Heidelberg, 1995, https://doi.org/10.1007/978-3-642-59223-2

[30] B. Lau, C. Sprunk and W. Burgard, "Kinodynamic motion planning for mobile robots using splines," 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, St. Louis, MO, USA, 2009, pp. 2427-2433, doi: 10.1109/IROS.2009.5354805.

[31] Bui, Xuan-Nam, et al. "Shortest path synthesis for Dubins non-holonomic robot." Proceedings of the 1994 IEEE International Conference on Robotics and Automation. IEEE, 1994.

[W1] http://nurbscalculator.in/

[33] Elbanhawi, M., Simic, M. & Jazar, R.N. Continuous Path Smoothing for Car-Like Robots Using B-Spline Curves. J Intell Robot Syst 80 (Suppl 1), 23–56 (2015). https://doi.org/10.1007/s10846-014-0172-0

[34] Knott, G. D. (2000). Interpolating cubic splines. Birkhauser.

[35] Liu, S., Liu, P. Benchmarking and optimization of robot motion planning with motion planning pipeline. Int J Adv Manuf Technol 118, 949–961 (2022). https://doi.org/10.1007/s00170-021-07985-5

[36] Zhang B, Zhu D. A new method on motion planning for mobile robots using jump point search and Bezier curves. International Journal of Advanced Robotic Systems. 2021;18(4). doi:10.1177/17298814211019220

[37] Bilal, Millie Pant, Hira Zaheer, Laura Garcia-Hernandez, Ajith Abraham, Differential Evolution: A review of more than two decades of research, Engineering Applications of Artificial Intelligence, Volume 90, 2020, 103479, ISSN 0952-1976, https://doi.org/10.1016/j.engappai.2020.103479.

[38] D. Tamilselvi, S.M. Shalinie, A.F. Thasneem, S.G. Sundari, Optimal Path Selection for Mobile Robot Navigation Using Genetic Algorithm in an Indoor Environment, In: P.S. Thilagam, A.R. Pais, K. Chandrasekaran, N. Balakrishnan (eds), Advanced Computing, Networking and Security. ADCONS 2011, Lecture Notes in Computer Science, vol 7135, Springer, Berlin, Heidelberg, 2012, https://doi.org/10.1007/978-3-642-29280-4_31.

[39] Ziadia, Abdelhamid & Mohamed, Habibi & Kelouwani, Sousso. (2023). Machine Learning Study of the Effect of Process Parameters on Tensile Strength of FFF PLA and PLA-CF. Eng. 4. 2741-2763. 10.3390/eng4040156.

[40] H. Miao and Y. -C. Tian, "Robot path planning in dynamic environments using a simulated annealing based approach," 2008 10th International Conference on Control, Automation, Robotics and Vision, Hanoi, Vietnam, 2008, pp. 1253-1258, doi: 10.1109/ICARCV.2008.4795701.

[41] Lone, Tahir & Wadood, Abdul & Gholami Farkoush, Saeid & Yu, Jiangtao & Kim, Changhwan & Bong, Rhee. (2019). An Improved Optimal Solution for the Directional Overcurrent Relays Coordination Using Hybridized Whale Optimization Algorithm in Complex Power Systems. IEEE Access. PP. 1-1. 10.1109/ACCESS.2019.2925822.

[42] Septfons, Baptiste , Chehri, Abdellah , Chaibi, Hasna , Rachid, Saadane , Tigani, Smail. (2022). Swarm Robotics: Moving from Concept to Application. 10.1007/978-981-19-3455-1_14.

[43] Tang B, Zhu Z, Luo J. Hybridizing Particle Swarm Optimization and Differential Evolution for the Mobile Robot Global Path Planning. International Journal of Advanced Robotic Systems. 2016;13(3). doi:10.5772/63812

Annexes