

Planning method for a reversing single joint tractor-trailer system

Ofa Ismail

Master of Science Thesis in Electrical Engineering

Planning method for a reversing single joint tractor-trailer system

Ofa Ismail

LiTH-ISY-EX--21/5419--SE

Supervisor: **Theodor Westny**
ISY, Linköpings universitet
Fredrik Nordin
Scania Group
Daniel Peña
Scania Group

Examiner: **Erik Frisk**
ISY, Linköpings universitet

*Division of Vehicular Systems
Department of Electrical Engineering
Linköping University
SE-581 83 Linköping, Sweden*

Copyright © 2021 Ofa Ismail

Abstract

This thesis investigates the design of a local planning method for a reversing single joint tractor-trailer system that can be used in a sampling-based motion planner. The motion planner used is a Rapidly-exploring Random Tree (RRT) developed by Scania.

The main objective of a local planning method is to generate a feasible path between two poses, which is needed when expanding the search tree in an RRT. The local planning method described in this thesis uses a set of curves, similar to Reeds-Shepp curves, feasible for a single joint tractor-trailer system. The curves are found by solving a constrained optimization problem that adheres to the kinematic model of the system. The reference for the tractor is generated by discretizing the path between curves. The reference for the trailer is generated by simulating the mission backwards where the curve radiuses are used as input. Simulating the mission backwards circumvents the instability of the system when reversing.

The generated references are then compared to references generated by a lattice-based motion planner. The length of the references generated by the RRT are smaller than those generated by the lattice-based motion planner in simple open environments. The RRT had issues finding a path in cases where the environment was complex while the lattice-based motion planner found a path in every scenario. The computational time was significantly lower for the RRT in all simulations. The RRT generates all references between any two given poses while the lattice-based motion planner approximate the start and goal poses to the closest vertex in the search-space.

The references generated by the RRT did not perform optimally when small turns were needed along the curves due to the distance needed for maneuvering the tractor. Therefore, a new optimization problem has to be defined for which the small turns are considered.

Acknowledgments

I would like to thank my examiner Erik Frisk and supervisors Theodor Westny, Fredrik Nordin and Daniel Peña for showing great interest in this thesis and contributing with help and advice throughout the work.

I would also like to thank my family for their love and support during the time of my studies. I would not have made it here without them.

Stockholm, Juni 2021
Ofa Ismail

Contents

1	Introduction	1
1.1	Background	1
1.2	Problem formulation	2
1.2.1	Relaxations	2
1.2.2	Research questions	2
1.3	Related work	3
1.3.1	Kinematic model	3
1.3.2	Motion planner	3
1.3.3	Reversing truck and trailer system	3
1.3.4	Planning methods	4
1.4	Thesis outline	4
2	Vehicle modeling	7
2.1	Single joint tractor-trailer configuration	7
3	Motion planner	11
3.1	Rapidly-exploring random tree	11
3.1.1	Sampling strategies	12
3.1.2	Connecting samples to the search-tree	14
3.1.3	Planning methods	15
3.2	Lattice-based motion planner	18
4	Tractor-trailer curves	21
4.1	Calculating optimal path between start and goal poses	22
4.2	Generating reference path from path sets	28
4.2.1	Generating reference for forward motion	29
4.2.2	Mapping of trailer curve to reversing tractor curves.	30
4.3	Calculating optimal curve radiuses for reversing maneuvers	35
5	Results	37
5.1	Simple scenarios in an open environment	37
5.2	Cage obstacle in an open environment	40
5.3	Large open environment	42

6 Conclusion	45
6.1 Choosing optimal curve radius	45
6.2 Comparison to the state lattice planner	45
6.3 Further development	46
Bibliography	49

1

Introduction

This thesis is written in collaboration with Scania where a planning method for a reversing single joint tractor-trailer system used in a Rapidly-exploring Random Tree (RRT), is researched and developed. The work will provide an alternative approach to the path planning problem than previous solutions consisting of a state lattice and Closed-loop RRT [5][16].

1.1 Background

Autonomous vehicles has become a prominent research topic during the last decade. Everything from systems that assist the human driver to fully autonomous vehicles are researched on. Advanced driver-assistance systems in modern vehicles is now the norm rather than the exception [2] while full-scale autonomous truck systems are still in development. Initial trials of full-scale autonomous truck systems driving on freeways have been conducted [3], however, fully automated systems that can meet the everyday demands, both for passenger cars and heavy duty vehicles, is not a reality today. The world transportation industry, estimated to be worth \$6.2 Trillion, rely on drivers who do not have the best working conditions [13]. Automating the transportation industry or at least some parts of it would be beneficial since the pressure on the drivers would decrease, the safety on the road would not depend on the physical state of the driver. Driving a full-scale heavy duty vehicle comes with additional challenges. For example, the reversing maneuvers that drivers do when docking a trailer or fitting into a parking space is a demanding task even for the most proficient drivers. The same maneuvers are difficult to automate since a reversing tractor-trailer system is unstable and can end up in a jack-knifing state.

1.2 Problem formulation

This thesis investigates the possibility to define a set of curves feasible for a reversing single joint tractor-trailer system similar to Reeds-Shepp curves [18]. The method will be evaluated by studying the computational time and the length of generated paths. The computational time of the method will be compared to the computational time of the lattice-based motion planner described in the work by Ljungqvist [16]. The length of the generated paths will be compared to the length of paths generated by the lattice-based motion planner. The goal is to design a local planning method, that can be used in Scania's current motion planner, capable of planning in an open environment between two general poses without the use of a controller.

1.2.1 Relaxations

The path planning problem has been simplified by introducing three initial relaxations to the problem:

- assume a joint angle, β , of zero between the tractor and trailer in the initial and final state of each path, i.e., the tractor and trailer are aligned with the same heading. Assuming $\beta = 0$ simplifies the process of connecting two local paths since the final state of a path will always have the same configuration as a start state of an other path.
- only use the Curve-Straight-Curve (CSC) set of the Reeds-Shepp curves [18]. The CSC path is a good initial approach to the problem since many of the transitions and maneuvers needed to define a set of new curves are included in the path. The difficulty of expanding to more paths is dependent on the complexity of the maneuvers conducted by the tractor.
- only use paths where all segments have the same gear, i.e., only driving forward or only reversing. This limits the amount of maneuvers needed for defining the new paths. However, a full path can still include both forward driving and reversing by connecting two local CSC paths with different gears, one driving forward and one reversing.

1.2.2 Research questions

For evaluation, a set of research questions are used as foundation.

- Which turning radius should be used when reversing the tractor-trailer system to guarantee not ending up in a jack-knifing state?
- How does the performance, with regard to computational time and plan length, of the motion planner compare to the well documented lattice-based motion planner [14][16][17]?
- How can the motion planner ensure that the paths generated are feasible for a single joint tractor-trailer system?

- Can the motion planner generate a path between any two random poses in an open environment?
- Can the local planning method be integrated into Scania's current motion planner?

1.3 Related work

The work presented in this thesis is built on previous research in the field. This section will give a short overview of related work.

1.3.1 Kinematic model

In [1], Altafini proposed a kinematic model of a general n-trailer and truck system that holds under a no slip assumption. The single joint tractor-trailer system will drive with a low speed when reversing, which minimizes the slip, making the kinematic model valid for this thesis.

1.3.2 Motion planner

LaValle introduced a sample based motion planning algorithm named Rapidly-exploring Random Tree (RRT) [9]. The idea is to build a tree using any desired planning method between randomly sampled nodes. The initial RRT does not guarantee that the generated path will be optimal. However, further development resulted in a planning algorithm named RRT* which converges towards the optimal solution when the node count increases [6].

There are other motion planners that has been used when planning for a truck and trailer system, one of which is the lattice-based motion planner [16]. The idea is to generate a discretized state space and connecting the nodes in the state space by using some heuristic or control law, resulting in a directed graph. Search algorithms are then applied on the graph for generating the desired path. Since the solution is dependent on search algorithms, the computational time can be quite extensive if the state space is large. One other property that can affect the computational time is the discretization of the state lattice. The solution will always be generated between two states in the state lattice, meaning that the discretization distance has to be small enough to generate good enough solutions, however, not too small since the search problem increase in size.

1.3.3 Reversing truck and trailer system

The problem of path planning for a reversing truck and trailer system has been investigated extensively. Proposed solutions include:

- application in a closed loop system [4][15][8][5].
- high computational power, e.g. use of a controller to guarantee feasibility [15][8][5].

- planning to an approximate pose rather than generating a path to a specific pose [4][15].

When only a truck is used, the system is stable and the issues listed can be solved by using e.g. Reeds-Shepp curves [18]. A Reeds-Shepp solution would allow path planning between two true random poses rather than approximating the goal pose, which the solutions listed does. In this thesis, the designed curves will be used to connect random sampled poses with the tree in an RRT [10]. Regarding computational power, planning a path using a pre-defined set of curves will always require the same amount of calculations for each path segment in the tree, no matter the length of the segment. The calculations are pure geometric, e.g., calculating the intersection of a curve and a line. However, the full path planning problem will require more computational power when the complexity of the mission increases, making it interesting to compare the proposed solution to the fast state lattice solution described in [16].

1.3.4 Planning methods

Researching planning methods is the main objective of this thesis. The proposed solution in this thesis is build on Reeds-Sheep curves that are designed for vehicles obeying by the kinematic bicycle model [18]. Reeds and Shepp designed combinations of curves and straight lines that generate an optimal path between any two poses. One important optimality criteria is that the turn radius of the curves has to be equal to the minimal turn radius of the vehicle used. A more detailed description of the Reeds-Shepp curves is given in Section 3.1.3.

1.4 Thesis outline

The thesis is divided into chapters where a short overview of each chapter is given below:

Chapter 2 - Vehicle modeling: The kinematic vehicle model used throughout the thesis is presented.

Chapter 3 - Motion planner: This chapter will give a basic introduction to motion planning using a randomly-exploring random tree, RRT. Previous research on sampling strategies and the importance of using a good heuristic is given. A more detailed description of planning methods using Dubins and Reeds-Sheep will also be given. A short overview of a lattice-based motion planner is also introduced.

Chapter 4 - Tractor-trailer curves: This chapter will give a detailed description of the planning method that an RRT can use when planing for the tractor-trailer system. The chapter starts with describing how the different candidate paths are constructed followed by how the paths are used for generating a reference that can be given to a controller. Lastly, the optimization problem for choosing the properties of the curves is formulated.

Chapter 5 - Results: The test scenarios used for evaluating the results are given in this chapter followed by data and figures of the tests conducted.

Chapter 6 - Conclusions: In this chapter, a discussion on the result is given. Final conclusions are drawn and a list of possible future development in the subject is given.

2

Vehicle modeling

In this chapter, the general n-trailer model is given together with a description of how the kinematic model for a tractor-trailer system can be derived.

2.1 Single joint tractor-trailer configuration

The kinematic bicycle model is a widely used representation of ground vehicles in control applications. The model describes how the change in position and orientation of a ground vehicle is calculated at any given point in time. It is given by:

$$\dot{x}_1 = v \cos(\theta_1), \quad (2.1a)$$

$$\dot{y}_1 = v \sin(\theta_1), \quad (2.1b)$$

$$\dot{\theta}_1 = \frac{v}{L_1} \tan(\alpha), \quad (2.1c)$$

where x_1, y_1 , and θ_1 is the tractor's rear axle position and orientation, L_1 is the length of the tractor's wheelbase, v is the velocity, and α is the steering angle. When connecting trailers to the tractor, the behaviour of the trailers does not obey by the same kinematic model. However, the orientation and velocity are said to be known for any general trailer connected to the tractor by the orientation and velocity:

$$\dot{\theta}_{i+1} = \frac{v_{i+1} \tan(\theta_i - \theta_{i+1})}{L_{i+1}} - \frac{M_i}{L_{i+1} \cos(\theta_i - \theta_{i+1})} \dot{\theta}_i, \quad (2.2a)$$

$$v_{i+1} = v_i \cos(\theta_i - \theta_{i+1}) + M_i \cos(\theta_i - \theta_{i+1}) \dot{\theta}_i, \quad (2.2b)$$

where L_i is the length of the wheelbase of trailer i , M_i is the distance from the rear axle of trailer i to the hitch point[1]. The model assumes that the hitch connection of trailer $i + 1$ is located behind the rear axle of trailer i . However, the single joint tractor-trailer system used in this thesis is modelled with positive off-hitching, such that the hitch point is located in front of the rear axle of the tractor, as shown in Figure 2.1. This is modified in Equation (2.2) by changing the sign of M_j . Furthermore, the indexing of M is dropped for brevity.

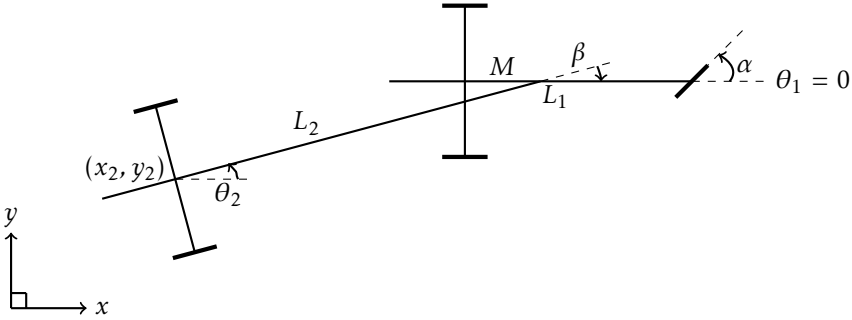


Figure 2.1: Schematic view of the single joint tractor-trailer system. A description of the states and parameters shown is listed in Table 2.1.

For the trailer in the tractor-trailer system shown in Figure 2.1, the derived orientation becomes:

$$\begin{aligned}\dot{\theta}_2 &= \frac{v_2 \tan(\theta_1 - \theta_2)}{L_2} + \frac{M}{L_2 \cos(\theta_1 - \theta_2)} \dot{\theta}_1 \\ &= \frac{v_2 \tan(\theta_1 - \theta_2)}{L_2} + \frac{vM \tan(\alpha)}{L_1 L_2 \cos(\theta_1 - \theta_2)},\end{aligned}\quad (2.3)$$

where

$$\begin{aligned}v_2 &= v \cos(\theta_1 - \theta_2) - M \sin(\theta_1 - \theta_2) \dot{\theta}_1 \\ &= v \cos(\theta_1 - \theta_2) - \frac{vM \sin(\theta_1 - \theta_2) \tan(\alpha)}{L_1}.\end{aligned}\quad (2.4)$$

Equation (2.3) can be simplified to:

$$\dot{\theta}_2 = \frac{v}{L_2} \left(\sin(\theta_1 - \theta_2) + \frac{M \cos(\theta_1 - \theta_2) \tan(\alpha)}{L_1} \right).\quad (2.5)$$

The joint angle β can be calculated using the relationship it has with θ_1 and θ_2 shown in Figure 2.1:

$$\beta = \theta_1 - \theta_2,$$

resulting in the model function:

$$\begin{aligned}
 \dot{\beta} &= \dot{\theta}_1 - \dot{\theta}_2 \\
 &= \frac{v}{L_1} \tan(\theta_1) - \frac{v}{L_2} \left(\sin(\theta_1 - \theta_2) + \frac{M \cos(\theta_1 - \theta_2) \tan(\alpha)}{L_1} \right) \\
 &= \frac{v}{L_1 L_2} \left(L_2 \tan(\alpha) - L_1 \sin(\beta) - M \cos(\beta) \tan(\alpha) \right).
 \end{aligned}$$

Using the derived equations above, the single joint tractor-trailer kinematic model can be expressed as:

$$\dot{x}_1 = v \cos(\theta_1), \quad (2.6a)$$

$$\dot{y}_1 = v \sin(\theta_1), \quad (2.6b)$$

$$\dot{\theta}_1 = \frac{v}{L_1} \tan(\alpha), \quad (2.6c)$$

$$\dot{x}_2 = v_2 \cos(\theta_2), \quad (2.6d)$$

$$\dot{y}_2 = v_2 \sin(\theta_2), \quad (2.6e)$$

$$\dot{\theta}_2 = \frac{v}{L_2} \left(\sin(\beta) + \frac{M \cos(\beta) \tan(\alpha)}{L_1} \right), \quad (2.6f)$$

$$\dot{\beta} = \frac{v}{L_1 L_2} \left(L_2 \tan(\alpha) - L_1 \sin(\beta) - M \cos(\beta) \tan(\alpha) \right), \quad (2.6g)$$

where

$$v_2 = v \cos(\theta_1 - \theta_2) - \frac{vM \sin(\theta_1 - \theta_2) \tan(\alpha)}{L_1}.$$

A description of the states and parameters in Equation (2.6) are described in Table 2.1.

Table 2.1: Description of states and model parameters used in a single joint tractor-trailer system.

States	Description
x_1	The x-coordinate of the tractor's rear axle.
y_1	The y-coordinate of the tractor's rear axle.
θ_1	The orientation of the tractor's rear axle with respect to the global frame.
x_2	The x-coordinate of the trailer's rear axle.
y_2	The y-coordinate of the trailer's rear axle.
θ_2	The orientation of the trailer's rear axle with respect to the global frame.
β	The relative joint angle between the tractor and the trailer.
Parameters	Description
L_1	The length of the tractor's wheelbase.
L_2	The length of the trailer's wheelbase.
M	The hitch distance between the tractor's rear axle and the hitch connection.

3

Motion planner

The main objective of a motion planner is to generate a feasible path a vehicle can follow. There exists numerous research and proposed solutions on the topic of motion planning. This thesis work will mainly consider the use of a sampling-based motion planner, namely the rapidly-exploring random tree (RRT) algorithm, originally proposed by LaValle [9]. An overview of the method, together with recently proposed extensions, are provided in this chapter. Additionally, a more detailed presentation of planning methods using Dubins and Reeds-Shepp is given. Lastly, a brief description of the lattice-based motion planner is given for future reference.

3.1 Rapidly-exploring random tree

The RRT algorithm provides a complete motion planning framework and falls under the category of sampling-based motion planners. The method relies on a sampling-method that incrementally adds new nodes thus yielding a tree-like structure. Each node in the tree structure corresponds to a vehicle state, x^k . When the number of nodes in the tree increases, the resolution improves without setting any resolution parameter. However, the new nodes can only be connected to the tree if there exists a path between the new node and a node in the tree that is dynamically feasible for a system. The framework may be modified to solve many motion-planning problems mainly by adjusting the connection-strategy or local planning method which is the topic of this work. The strategies behind sampling and deciding how to connect the nodes have been researched extensively [7][12][18]. A brief overview of sampling and heuristics strategies is provided in this section. The tree is built according to Algorithm 1 [11]. An example of an RRT tree is shown in Figure 3.1.

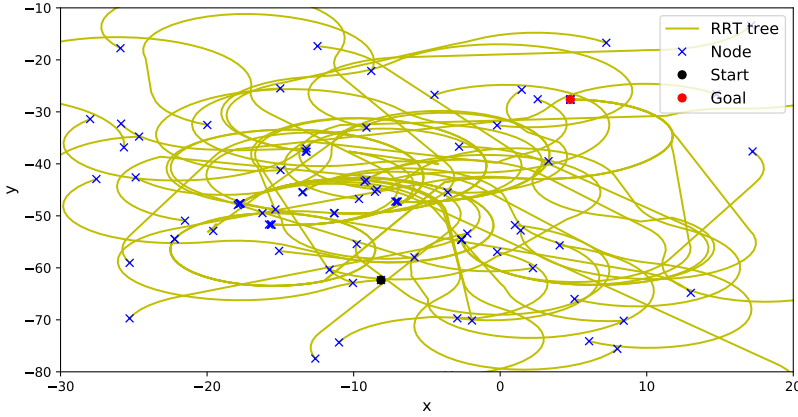


Figure 3.1: Example of an RRT tree structure.

Algorithm 1: Rapidly-exploring random tree.

input : Initial state q_{init} , goal state q_{goal} and number of nodes N .

output: The RRT graph \mathcal{G} .

$\mathcal{G}.init(q_{init});$

for $n = 0$ **to** N **do**

$q_{rand} = \text{sample_new_random_node}(q_{goal});$

$q_{near} = \text{find_nearest_node_in_tree}(q_{rand}, \mathcal{G});$

$\varepsilon = \text{generate_path_between_nodes}(q_{rand}, q_{near});$

if $\text{obstacle_free}(\varepsilon)$ **then**

$\mathcal{G}.add_node(q_{rand});$

$\mathcal{G}.add_edge(\varepsilon);$

end

end

return \mathcal{G}

3.1.1 Sampling strategies

Sampled nodes might end up in locations that will not benefit the given mission when using a random sampling, resulting in unused nodes. Each additional sample increases the computational time since all nodes in the tree are considered when expanding the tree. Therefore, an efficient sampling strategy could improve real-time performance. One strategy is to sample in the direction given by the orientation of the states in the search-tree [7]. This strategy is sensible for car-like vehicles since their movement is limited by differential constraints. One important property of the RRT is the randomness that allows for flexibility in the search-space compared with other planners such as the state lattice planner which is bound to a set of states. To maintain the element of randomness, a

Gaussian distribution, $\mathcal{N}(\mu, \sigma)$, is used in the directed sampling strategy. When the desire is to sample in the direction θ_0 at a position r_0 meters away, some deviation occurs due to the added distribution. The sampled pose can be generated using the equation:

$$\begin{bmatrix} x_{\text{rand}} \\ y_{\text{rand}} \\ \theta_{\text{rand}} \end{bmatrix} = \begin{bmatrix} x_0 \\ y_0 \\ 0 \end{bmatrix} + \begin{bmatrix} r \cos(\theta) \\ r \sin(\theta) \\ \theta \end{bmatrix} \quad \text{with} \quad \begin{cases} r = \sigma_r |\mu_r| + r_0 \\ \theta = \sigma_\theta \mu_\theta + \theta_0 \end{cases}, \quad (3.1)$$

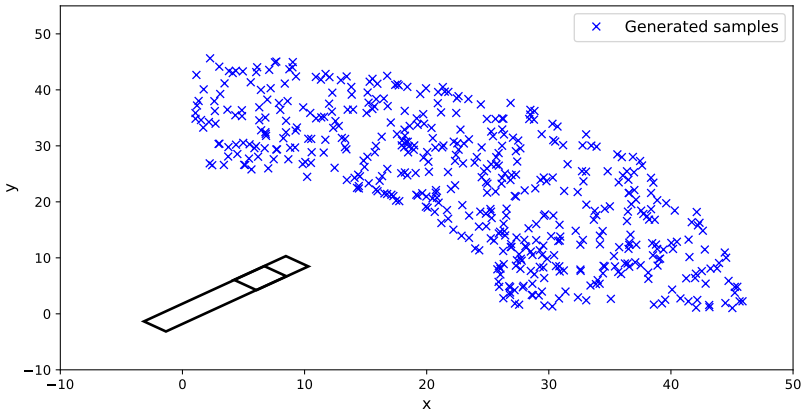
where x_0 , y_0 , and θ_0 refer to states in the search-tree and r_0 is an offset with respect to x_0 and y_0 . Figure 3.2 shows how the directed sample in Equation (3.1) is visualized. To enforce that the plan converges towards the goal region, common practice is to occasionally sample points within the goal region [11]. The frequency may be controlled by sampling the majority according to a general sampling strategy, as the one described in Equation (3.1), and the rest at the goal pose, thus biasing the sampling towards the goal region. Algorithm 2 shows the logic of general sampling strategy.

Algorithm 2: Draw sample.

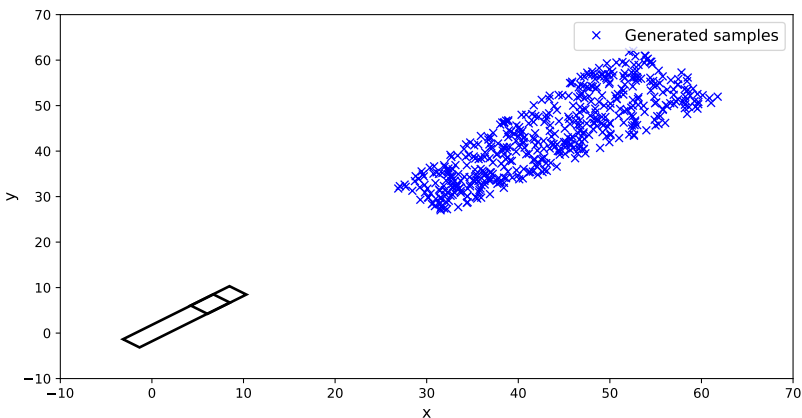
```

input : Goal state  $q_{\text{goal}}$ .
output: Random state  $q_{\text{rand}}$ .
val = random value between 0 and 1;
if val > 1 - bias then
  | return  $q_{\text{rand}} = q_{\text{goal}}$ ;
end
while True do
  |  $q_{\text{rand}} = \text{sample\_random\_pose}$ ;
  | if obstacle_free( $q_{\text{rand}}$ ) then
  | | return  $q_{\text{rand}}$ ;
  | end
end

```



(a) Directed sampling simulating a vehicle with low speed.



(b) Directed sampling simulating a vehicle with high speed.

Figure 3.2: Directed sampling using Equation (3.1).

3.1.2 Connecting samples to the search-tree

When a new node has been sampled, a decision has to be made on how to connect it in the tree. One approach is to find the node in the tree that is closest in distance to the new node. However, the closest distance does not guarantee that the node is best fitted considering the nonholonomic constraints. Consider a vehicle obeying by the kinematic bicycle model, one planning method that take its drivability into account is Dubins path [11]. Dubins path provides the optimal path for a kinematic bicycle model between two poses. More details are provided in Section 3.1.3. Figure 3.3 shows the different paths chosen by the different heuris-

tics with each corresponding length; note that the nearest point depend on the heuristic. If the wrong heuristic is chosen, the tree can be built in an undesirable manner. For best practice, a well performing heuristic is a planning method applicable to the kinematic vehicle model used.

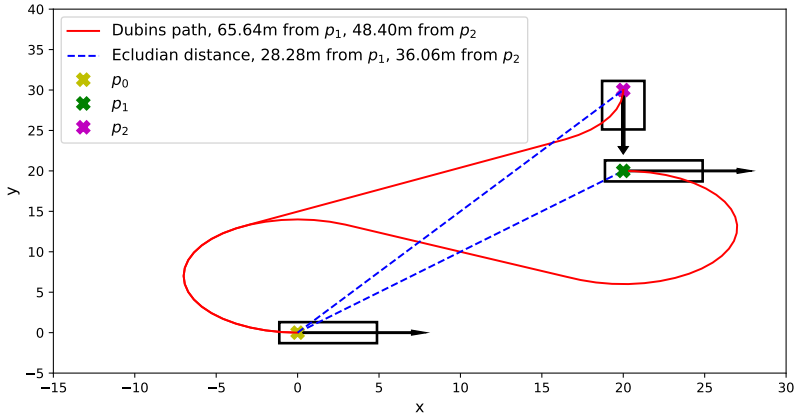


Figure 3.3: Different heuristics used to determine where to connect a new node, p_0 in the figure. It is shown that the Euclidian distance favours p_1 while Dubins path favours p_2 by viewing the lengths of each path.

3.1.3 Planning methods

The main objective of a planning method is to generate a feasible path between two states. The planning method that this thesis focuses on is described in Chapter 4, however, the idea is inspired by an optimal planning method for vehicles obeying by the kinematic bicycle model that uses Reeds-Shepp curves [11]. Furthermore, the Reeds-Shepp curves are based on curves used in Dubins path. The main difference between the curves in Reeds-Shepp and Dubins is that Reeds-Shepp curves allow reverse motion while Dubins curve only allow forward motion.

Dubins

Dubins path is generated by either following curves or a straight line. All curves are the same size and one important property is that the curve radius is the same as the minimal turn radius of the vehicle. Each curve can differ in direction, either right or left, denoted by \mathcal{R} and \mathcal{L} respectively. A straight line is denoted by \mathcal{S} . The segments in Dubins path only allow forward motion. Dubins showed that optimal paths can be constructed by following a combination of these segments. However, not all combinations are included in the optimal set, only

$$\{\mathcal{LRL}, \mathcal{RLR}, \mathcal{LSL}, \mathcal{LSR}, \mathcal{RSL}, \mathcal{RSR}\}$$

are used. These are a combination of three curves or two curves and a straight, visualized in Figure 3.4.

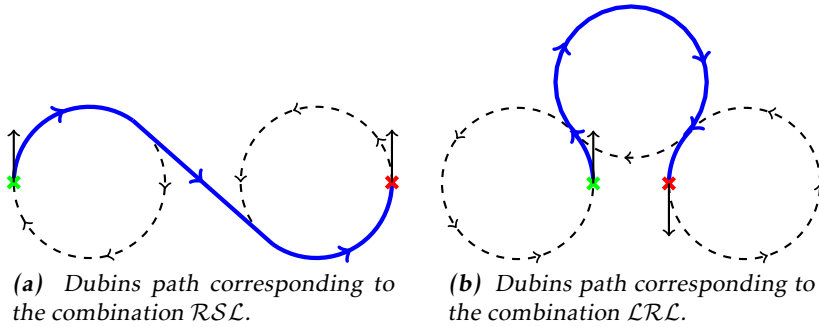


Figure 3.4: Dubins path for combinations consisting of three curves or of two curves and a straight. Start point on green mark and goal point on red mark. The vehicle orientation at each point is illustrated with a black arrow.

Dubins path is generated between two given poses. The given pose with its corresponding curves is called a *configuration*. These configurations include all possible curves, not only the one used. A visualization of a configuration is shown in Figure 3.5

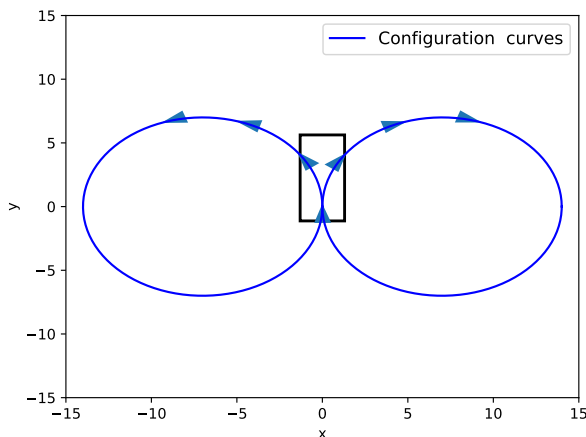


Figure 3.5: Tractor configuration at the pose $(0, 0, \frac{\pi}{2})$ with its possible configuration curves for Dubins path.

Reeds-Shepp

Compared to Dubins path, Reeds-Shepp allow reverse motion when generating a path. Comparing the path in Figure 3.4b with the path in Figure 3.6 where the start and goal positions are the same, the generated path is shorter when Reeds-Shepp curves are used.

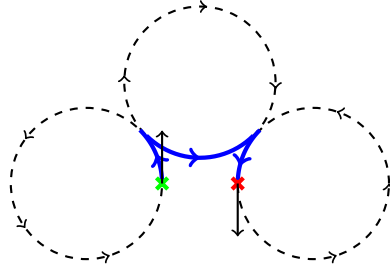


Figure 3.6: Reeds-Shepp path for an \mathcal{LRL} path. Start point on green mark and goal point on red mark. The vehicle orientation at each point is illustrated with a black arrow. Note that the vehicle is moving in the opposite direction of the curve arrows. This is due to the fact that the direction of a curve is defined as the forward motion of a vehicle, not reversing motion as in this case.

Since reverse motion is allowed, the curves used for generating a path can differ in gear as well as direction: forward and reverse. An updated visualization of the configuration is shown in Figure 3.7.

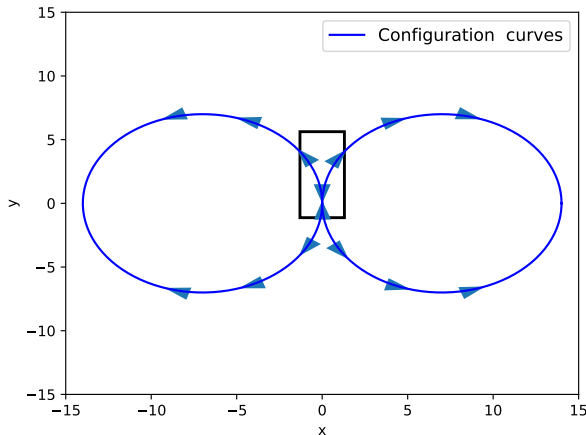


Figure 3.7: Tractor configuration at the pose $(0, 0, \frac{\pi}{2})$ with its possible configuration curves for Reeds-Shepp path.

The notation has to be updated from Dubins path to include the new combinations in Reeds-Shepp. The symbol “|” is used to indicate a change of gear. Compared to the six combinations to describe Dubins path, Reeds-Shepp has 48 combinations. To simplify the notation, \mathcal{C} is used to indicate a curve. The direction of a curve is indicated by the symbols “+” and “-” where “+” is right and “-” is left. When the direction is not specified in a combination, the combination refer to a path set, i.e. \mathcal{CSC} can refer to the paths $\mathcal{C}^+S\mathcal{C}^-$, $\mathcal{C}^+S\mathcal{C}^+$, etc. Reeds-Shepp defined the combinations:

$$\{\mathcal{C}|\mathcal{C}|\mathcal{C}, \mathcal{C}\mathcal{C}|\mathcal{C}, \mathcal{C}|\mathcal{C}\mathcal{C}, \mathcal{C}S\mathcal{C}, \mathcal{C}\mathcal{C}|\mathcal{C}\mathcal{C}, \mathcal{C}|\mathcal{C}\mathcal{C}|\mathcal{C}, \mathcal{C}|\mathcal{C}S\mathcal{C}, \mathcal{C}S\mathcal{C}|\mathcal{C}, \mathcal{C}|\mathcal{C}S\mathcal{C}|\mathcal{C}\}$$

as the optimal path sets. The $\mathcal{LR}\mathcal{L}$ path in Figure 3.6 can be written as $\mathcal{C}^-|\mathcal{C}^+|\mathcal{C}^-$ and is part of the $\mathcal{C}|\mathcal{C}|\mathcal{C}$ path set.

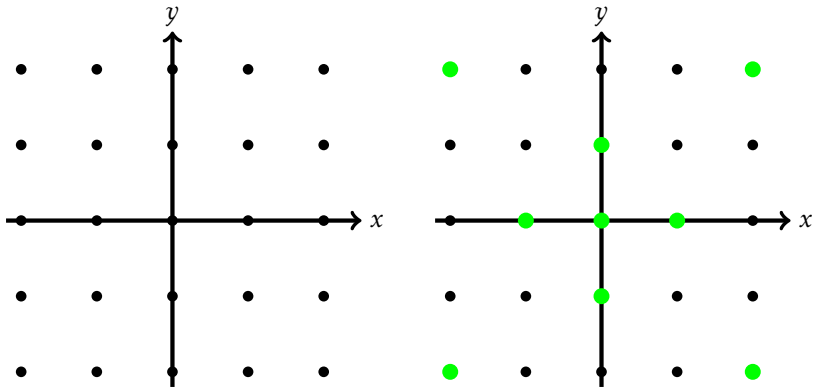
3.2 Lattice-based motion planner

The search space of a lattice-based motion planner is discretized given a resolution, thus yield a state lattice. When using a state lattice planner, search algorithms are used in a directed graph, \mathcal{G} , to generate a path. One advantage of a state lattice solution is that the components needed for the directed graph are pre-calculated offline, saving computational time. The directed graph consists of vertices, \mathcal{V} , and edges, \mathcal{E} . Each vertex, $v_k \in \mathcal{V}$, represents a vehicle state, x_k , and each edge, $e_k \in \mathcal{E}$, is a motion primitive, connecting two vertices. All available motion primitives are generated by solving the optimal control problem in Equation (3.2), where J is the objective functional and L is a cost function dependent on the vehicle state and control signal [16]. Note that Equation (3.2) is formulated in such a manner that it minimizes the cost of travelling between two points, x_i and x_f , while satisfying the nonholonomic constraints of the system, f , similar to the objective of a planning method. Meaning that the underlying principle is similar for the two planners described, however, the planning method is generated offline in a state lattice planner compared to the RRT.

$$\begin{aligned} \underset{u(\cdot), t_f}{\text{minimize}} \quad & J = \int_0^{t_f} L(x(t), u(t)) dt \\ \text{subject to} \quad & \dot{x} = f(x(t), u(t)), \\ & x(0) = x_i, \quad x(t_f) = x_f, \\ & x(t) \in \mathbb{X}, \quad u(t) \in \mathbb{U} \end{aligned} \tag{3.2}$$

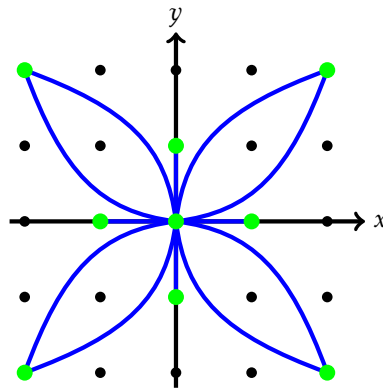
The offline construction of the state lattice is shown in Figure 3.8. Note that the performance of the state lattice is dependant on the resolution used when discretizing the state space. A more dense state space results in better performance. \mathcal{G} is generated by connecting all motion primitives available in the state space. The motion primitives can reach more points when using a dense state space. When \mathcal{G} has been generated, the graph can be searched through online to find

a path [16]. The length of the generated path depends on the search algorithm used. For example, Dijkstra's algorithm is a search algorithm that guarantees optimality, meaning that the shortest path will always be found if used. However, the computational time can be high if the search-space is large. A* is another search algorithm that guarantees optimality if an admissible heuristic is used within the algorithm. The computational time for an A* search is correlated to the dimensions of the search-space as well, however, it is faster than Dijkstra.



(a) Discretize the vehicle state space with the desired resolution.

(b) Select the states that the vehicle can travel between when considering its reachability.



(c) Generate the motion primitives by solving Equation (3.2).

Figure 3.8: The offline construction of the state lattice.

4

Tractor-trailer curves

The Reeds-Shepp planning method is mainly applicable on systems obeying by the kinematic bicycle model. However, similar curves can be defined for a tractor-trailer system. For forward motion, the path is planned similar to Reeds-Shepp, explained in section 3.1.3. For reverse motion, the configuration curves are used for planning a path for the trailer instead of the tractor. However, the reference path that the planner sends the controller will contain the states x_1 , y_1 , θ_1 , and β , meaning that a path has to be created for the tractor that allows the trailer to follow its configuration curves.

In this chapter, a planning method that uses curves and straight lines to generate a reference for a tractor-trailer system is presented. The process of generating a reference starts with connecting all possible configuration curves and straight lines to the given start and goal poses. The configuration curves of the start and goal poses are then connected by a straight line, creating a set of candidate paths. The shortest path from the set of candidate paths is considered to be the optimal path. Section 4.1 describes this process in more detail. Furthermore, Section 4.2 describes how the optimal path is used to generate a reference that a controller can follow. Section 4.2.1 shows how the optimal path is discretized into a reference containing sampled poses in the case where the gear of the optimal path is forward. The configuration curves in the optimal path has to be mapped from trailer poses to corresponding tractor poses before discretized into a reference in the case where the optimal path is reversing. That is described in Section 4.2.2. The radiuses of the tractor curves used when reversing and the length for that each tractor curve is followed is given by a constrained optimization problem stated in Section 4.3.

4.1 Calculating optimal path between start and goal poses

When planing a local path, a start pose and a goal pose are given. The path sets considered for a tractor-trailer system when reversing is \mathcal{CSC} while \mathcal{CS} and \mathcal{CSCS} are used when driving forward. Note that an initial and a final curve can be neglected by following the curve for zero meters; meaning that the \mathcal{CS} path set is a \mathcal{CSCS} path set where the final two segments are neglected. The radiuses of the curves are determined in Section 4.3.

A set of candidate paths are created by generating all possible configuration curves and connecting them to the start and goal poses. The first segment of all path sets is a curve. The position of the curve depends on the gear. For forward motion, the initial curve is located on the rear axle of the tractor, meaning that the curve is used in the reference and can be followed directly. For reverse motion, the configuration curves are located behind the rear of the trailer and can not be followed directly. This is because the vehicle needs additional leeway to adjust trailer orientation for adequate path tracking. This distance is denoted $x^{\text{offset}}, y^{\text{offset}}$ in Equation (4.1). Figure 4.1a shows the start configuration. For each curve in the start configuration, its center position, direction and gear is saved. The center position is calculated according to Equation (4.1) where r is the radius of the curve and $(x_1^{\text{start}}, y_1^{\text{start}}, \theta_1^{\text{start}})$ is the given start pose of the tractor.

$$x_{\text{forward}}^{\text{center}} = x_1^{\text{start}} \pm r \cos(\theta_1^{\text{start}} - \frac{\pi}{2}) \quad (4.1a)$$

$$y_{\text{forward}}^{\text{center}} = y_1^{\text{start}} \pm r \sin(\theta_1^{\text{start}} - \frac{\pi}{2}) \quad (4.1b)$$

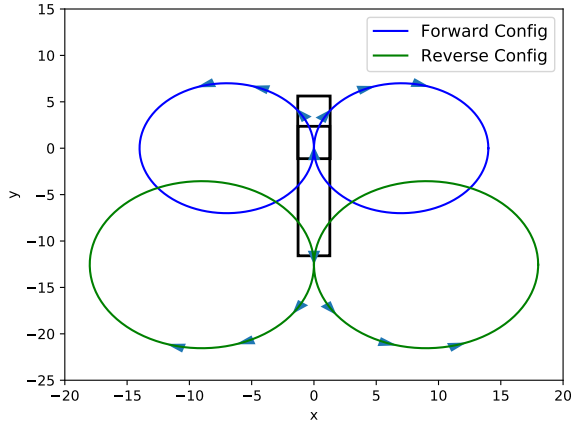
$$x_{\text{reverse}}^{\text{center}} = x_1^{\text{start}} - x^{\text{offset}} \cos(\theta_1^{\text{start}}) \pm y^{\text{offset}} \sin(\theta_1^{\text{start}}) \pm r \sin(\theta_1^{\text{start}}) \quad (4.1c)$$

$$y_{\text{reverse}}^{\text{center}} = y_1^{\text{start}} - x^{\text{offset}} \sin(\theta_1^{\text{start}}) \mp y^{\text{offset}} \cos(\theta_1^{\text{start}}) \mp r \cos(\theta_1^{\text{start}}) \quad (4.1d)$$

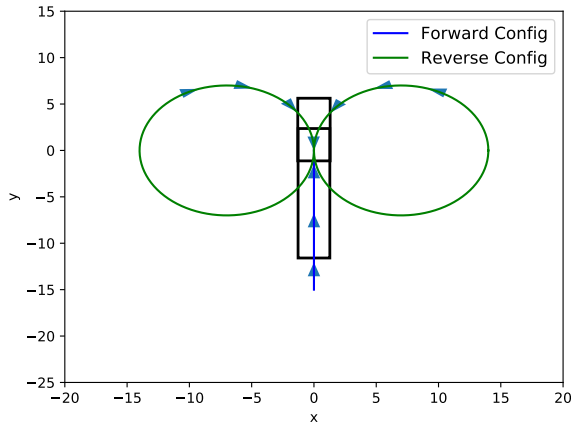
The final segment for the different path sets differ depending on the gear, curve for reverse motion and straight for forward motion. The configuration curves are located in front of the goal pose when reversing since the tractor-trailer system needs some leeway to straighten out. This distance is denoted $x^{\text{offset}}, y^{\text{offset}}$ in Equation (4.2). The straight segment used for forward motion end at the given goal pose. Figure 4.1b shows the goal configuration. The center position, direction and gear is saved for the configuration curves while the gear and end pose is saved for the straight. The center position of a curve in the goal configuration is calculated according to Equation (4.2).

$$x_{\text{reverse}}^{\text{center}} = x_1^{\text{goal}} - x^{\text{offset}} \cos(\theta_1^{\text{goal}}) \pm y^{\text{offset}} \sin(\theta_1^{\text{goal}}) \pm r \sin(\theta_1^{\text{goal}}) \quad (4.2a)$$

$$y_{\text{reverse}}^{\text{center}} = y_1^{\text{goal}} - x^{\text{offset}} \sin(\theta_1^{\text{goal}}) \mp y^{\text{offset}} \cos(\theta_1^{\text{goal}}) \mp r \cos(\theta_1^{\text{goal}}) \quad (4.2b)$$



(a) Start configuration with curves for both gears.



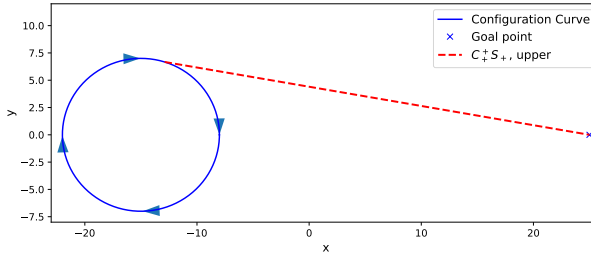
(b) Goal configuration with curves for reverse motion and straight for forward motion.

Figure 4.1: Start and goal configurations with the given start and goal poses located on the rear axle of the tractor. The arrows point in the direction of the motion.

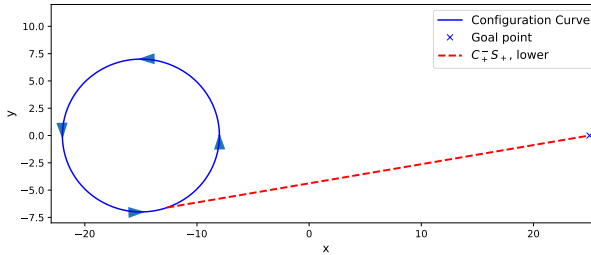
All curves in the configurations can differ in gear and direction while the straights can only differ in gear. Each path set contains all different combinations of curves and straights. However, to satisfy the constraint that each local path has to have the same gear for all segments, the combinations where the gear differs within the same path are disregarded. Furthermore, as described earlier, the path sets are used for different gears. The paths listed in Table 4.1 remain.

Table 4.1: Specification of paths in each path set used, noted by $\mathcal{O}_{gear}^{direction}$ where \mathcal{O} is the type of segment. Direction and gear are written as + and - where + is right and forward. The direction of a curve is defined as the forward motion of the tractor following the curve, meaning that the arrows in Figure 4.1 for the reverse configurations are pointing in the opposite direction of the curve direction.

Set	Paths
\mathcal{CS}	$\mathcal{C}_+^+ \mathcal{S}_+$ $\mathcal{C}_-^- \mathcal{S}_+$
\mathcal{CSC}	$\mathcal{C}_+^+ \mathcal{S}_- \mathcal{C}_+^+$ $\mathcal{C}_-^- \mathcal{S}_- \mathcal{C}_-^-$ $\mathcal{C}_-^- \mathcal{S}_- \mathcal{C}_+^+$ $\mathcal{C}_-^- \mathcal{S}_- \mathcal{C}_-^-$
\mathcal{CSCS}	$\mathcal{C}_+^+ \mathcal{S}_+ \mathcal{C}_+^+ \mathcal{S}_+$ $\mathcal{C}_+^+ \mathcal{S}_+ \mathcal{C}_-^- \mathcal{S}_+$ $\mathcal{C}_-^- \mathcal{S}_+ \mathcal{C}_+^+ \mathcal{S}_+$ $\mathcal{C}_-^- \mathcal{S}_+ \mathcal{C}_-^- \mathcal{S}_+$

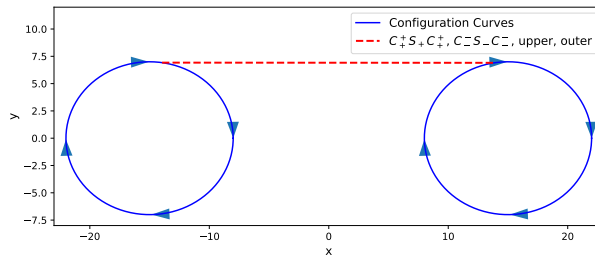


(a) Motion on curve is directed to the right.

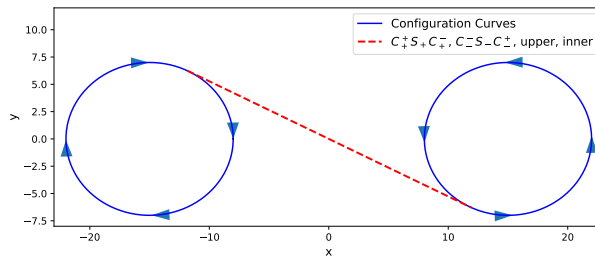


(b) Motion on curve is directed to the left.

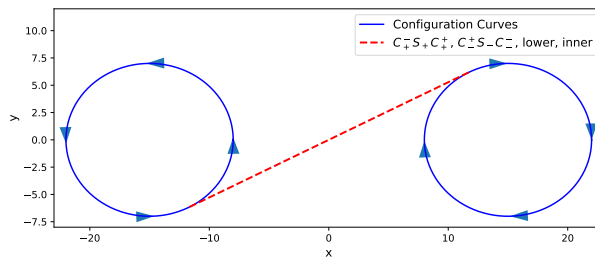
Figure 4.2: Straight options for connecting start and goal configurations, creating the \mathcal{CS} paths listed in Table 4.1.



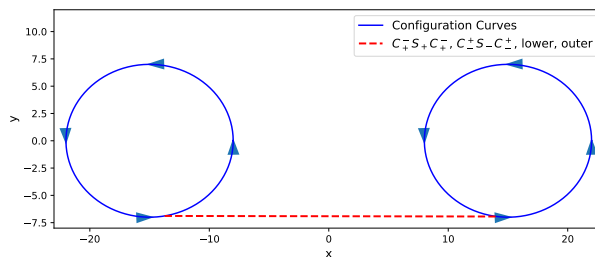
(a) Motion on both curves is directed to the right.



(b) Motion on start curve is directed to the right and goal curve to the left.



(c) Motion on start curve is directed to the left and goal curve to the right.



(d) Motion on both curves is directed to the left.

Figure 4.3: Straight options for connecting start and goal configurations, creating the CSC paths listed in Table 4.1.

The connection of start and goal configuration depend on the gear and direction of each combination listed in Table 4.1. The straight can be connected on different points along the curves. Figures 4.2 and 4.3 show the different options of mounting points, i.e., the points on the curves that the straight is mounted on. If the start and goal configuration curves have the same direction, one of the outer options are chosen. When deciding for upper or lower path, both direction and gear of the start curve has to be taken into consideration. If the direction of the start curve is right, an upper path is chosen if the gear is forward and vice versa. The logic behind choosing mounting points is described in Algorithm 3.

Algorithm 3: Principle for connecting start and goal configurations.

Result: The resulting straight is a combination of *outer/inner* and *upper/lower*. *Outer* meaning one of the paths shown in Figures 4.3a and 4.3d is chosen. *Upper* meaning that the start point of the straight is located on the upper most point of the start curve, as shown in Figures 4.3a 4.3b. Note that only *upper/lower* is needed for deciding the straight for the \mathcal{CS} set shown in Figure 4.2.

```

if start curve direction equals goal curve direction then
  | choose outer straight
else
  | choose inner straight
end
if start curve direction is right and gear is forward then
  | choose upper straight
else if start curve direction is left and gear is forward then
  | choose lower straight
else if start curve direction is left and gear is reverse then
  | choose upper straight
else if start curve direction is right and gear is reverse then
  | choose lower straight
end

```

The start and end points of the straight are calculated as tangent points on each corresponding curve. In the case where an outer straight is chosen in Algorithm 3, the orientation of the vehicle between the tangent points on both curves are the same as the direction of a vector from the center point of a curve in the start configuration to the center point of a curve in the goal configuration. The vector between the center points of the curves is calculated with the equations:

$$\Delta x = x_1^{\text{goal}} - x_1^{\text{start}}, \quad \Delta y = y_1^{\text{goal}} - y_1^{\text{start}}.$$

To normalize the vector, the length has to be calculated using the equation:

$$l = \sqrt{(\Delta x)^2 + (\Delta y)^2}.$$

A unit vector is created from the center point of the start curve towards the center

point of the goal curve:

$$\hat{x} = \frac{\Delta x}{l}, \quad \hat{y} = \frac{\Delta y}{l}.$$

To calculate the tangent point on the start curve, the unit vector is rotated 90° , as shown in Figure 4.4, in the direction given by *upper* or *lower* from Algorithm 3.

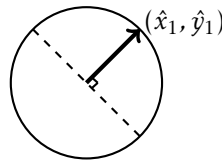


Figure 4.4: Unit vector in the start curve towards the center point of the goal curve when outer straight is chosen in Algorithm 3. The dashed lines end in the possible start points of the straight, depending on the choice of *upper* or *lower* in Algorithm 3.

In the case where an inner straight is chosen in Algorithm 3, the angle for which the unit vector is rotated has to be calculated. Figures 4.3b and 4.3c show that there is a distance of two times the radius between the end points of the straight when choosing an inner straight instead of an outer straight. A vector is created with the length of $2r$ from the center point of the goal curve. Another vector is created from the center point of the start curve without a specified length. The direction of the vectors are chosen so that they intercept perpendicularly, creating two legs in a right-angled triangle. The hypotenuse of the triangle is a vector between the center points of the curves. Figure 4.5 shows an overview of the described geometry.

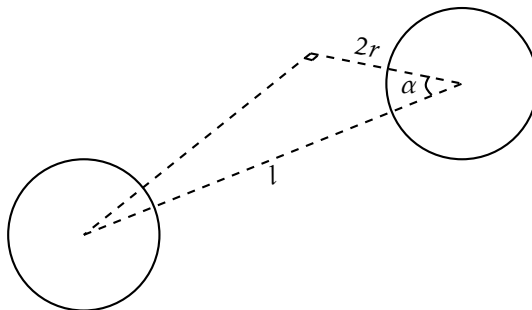


Figure 4.5: Geometric overview for deciding tangent points on curves that are the end points of the straight, in the case where an inner straight is chosen in Algorithm 3.

The cos and sin of the desired rotation are calculated using the equations:

$$\cos(\alpha) = \frac{2r}{l}, \quad \sin(\alpha) = \sqrt{1 - \cos^2(\alpha)} = \sqrt{1 - \frac{4r^2}{l^2}}.$$

Rotate the unit vector towards the tangent point by using the equations:

$$n_x = \cos(\alpha)\hat{x} \mp \sin(\alpha)\hat{y}, \quad n_y = \cos(\alpha)\hat{y} \pm \sin(\alpha)\hat{x},$$

where the sign depend on the choice of upper or lower in Algorithm 3. The tangent points are calculated using Equations (4.3) and (4.4).

$$\mathcal{S}^{\text{start}} = \left(x_{\text{start}}^{\text{tangent}}, y_{\text{start}}^{\text{tangent}} \right) = \left(x_{\text{start}}^{\text{center}} + rn_x, y_{\text{start}}^{\text{center}} + rn_y \right) \quad (4.3)$$

$$\mathcal{S}^{\text{end}} = \left(x_{\text{goal}}^{\text{tangent}}, y_{\text{goal}}^{\text{tangent}} \right) = \left(x_{\text{goal}}^{\text{center}} \pm rn_x, y_{\text{goal}}^{\text{center}} \pm rn_y \right) \quad (4.4)$$

In the \mathcal{CS} path set, the end point of the straight will be the same as the given goal point. In the \mathcal{CSCS} path set, the start point of the second curve will start where the straight ends, given in Equation (4.4). However, the end point of the second curve will be located behind the given goal pose with respect to the orientation, calculated using the equation:

$$\mathcal{C}_{\text{second}}^{\text{end}} = \mathcal{S}_{\text{second}}^{\text{start}} = \left(x_1^{\text{goal}} - \mathcal{S}_{\text{second}}^{\text{length}} \cos(\theta_1^{\text{goal}}), y_1^{\text{goal}} - \mathcal{S}_{\text{second}}^{\text{length}} \sin(\theta_1^{\text{goal}}) \right),$$

and the end point of the second straight will be the given goal point. The optimal path depends on the metric that defines the optimality criteria. Logically, the metric will be correlated to the length of the paths. Since the start and end position of all segments are stored as well as the center position of all curves, the length of each path can easily be calculated. The length of a curve arc is calculated using the equation:

$$\mathcal{C}_{\text{length}} = |\theta_{\text{diff}}|r,$$

where r is the radius of the curve and θ_{diff} is the angle difference along the curve. The start and end angle of a curve that the tractor follows is calculated given the center position of a curve $(x^{\text{center}}, y^{\text{center}})$ and a point on a curve $(x^{\text{curve}}, y^{\text{curve}})$:

$$\theta = \tan^{-1} \left(\frac{|y^{\text{curve}} - y^{\text{center}}|}{|x^{\text{curve}} - x^{\text{center}}|} \right). \quad (4.5)$$

The length of a straight is calculated using the Euclidian distance formula:

$$\mathcal{S}_{\text{length}} = \sqrt{(x_{\text{end}} - x_{\text{start}})^2 + (y_{\text{end}} - y_{\text{start}})^2}.$$

4.2 Generating reference path from path sets

The process of generating a reference from the optimal path calculated in Section 4.1 differs for forward and reverse motion. The configuration curves are

discretized and used as reference for forward motion. The configuration curves in a reversing path set is the path that a trailer follows. However, the reference that is sent to a controller is a reference path for the tractor. Therefore, the configuration curves are mapped to the corresponding tractor curves. The process of generating a reference for forward motion is described in Section 4.2.1. The process of mapping and generating a reference for a reversing tractor is described in Section 4.2.2.

4.2.1 Generating reference for forward motion

For forward motion, the reference path used by the controller is created by discretizing the optimal path chosen from Table 4.1. The configuration curves in the optimal path are discretized between the start and end angles with a discretization step calculated using the equation:

$$d_{s_{\text{curve}}} = \frac{\theta_{\text{diff}}}{\lceil \theta_{\text{diff}} \rceil}, \quad (4.6)$$

yielding the angles:

$$\theta^n = \theta^{\text{start}} + nd_{s_{\text{curve}}}, \quad n = 0, 1, \dots, \lceil \theta_{\text{diff}} \rceil. \quad (4.7)$$

A reference for the pose (x_1, y_1, θ_1) in a curve is generated by using the equation:

$$(x_1, y_1, \theta_1) = \left(x^{\text{center}} + r \cos(\theta^n), y^{\text{center}} + r \sin(\theta^n), \theta^n \pm \frac{\pi}{2} \right), \quad (4.8)$$

where r is the curve radius and θ^n refer to the angles calculated in Equation (4.7). The reference for the joint angle, β , can not be calculated geometrically since it depends on the trailer orientation which is not known. Instead, the vehicle model is simulated forward to generate the joint angle reference. The kinematic model in Equation (2.6g) is re-written as a function of the curve radius instead of steering angle, shown in Equation (4.9). Note that the initial beta reference, β^0 , will always be zero.

$$\beta^{n+1} = d_{s_{\text{curve}}} r \left(\pm \frac{1}{r} - \frac{\sin(\beta^n)}{L_2} \mp \frac{M \cos(\beta^n)}{L_2 r} \right), \quad n = 0, \dots, \lceil \theta_{\text{diff}} \rceil - 1 \quad (4.9)$$

The sample distance for a straight segment is calculated by using the equation:

$$d_{s_{\text{straight}}} = \frac{S_{\text{length}}}{\lceil S_{\text{length}} \rceil}. \quad (4.10)$$

The reference orientation is constant in a straight, calculated according to the equation:

$$\theta_{\text{straight}} = \tan^{-1} \left(\frac{y_{\text{straight}}^{\text{end}} - y_{\text{straight}}^{\text{start}}}{x_{\text{straight}}^{\text{end}} - x_{\text{straight}}^{\text{start}}} \right). \quad (4.11)$$

A reference for the pose (x_1, y_1, θ_1) in a straight is generated by using the equation:

$$\begin{pmatrix} x_1 \\ y_1 \\ \theta_1 \end{pmatrix} = \begin{pmatrix} x_{\text{straight}}^{\text{start}} + nds_{\text{straight}} \cos(\theta_{\text{straight}}) \\ y_{\text{straight}}^{\text{start}} + nds_{\text{straight}} \sin(\theta_{\text{straight}}) \\ \theta_{\text{straight}} \end{pmatrix}, \quad n = 0, 1, \dots, \lceil \mathcal{S}_{\text{length}} \rceil.$$

Equation (4.9) is used when calculating the joint angle reference. However, $r \rightarrow \infty$ for a straight line, resulting in the equation:

$$\beta^{n+1} = -ds_{\text{straight}} \frac{\sin(\beta^n)}{L_2}, \quad n = 0, \dots, \lceil \mathcal{S}_{\text{length}} \rceil - 1. \quad (4.12)$$

Note that the initial joint angle reference, β^0 , can be dependent on the last joint angle reference in the previous segment. For reverse motion, the process of creating a reference is not as simple since the reference is made for the tractor while the optimal path calculated in Section 4.1 is made for the trailer.

4.2.2 Mapping of trailer curve to reversing tractor curves.

To make the trailer follow a curve given by the optimal reversing path calculated in Section 4.1, the tractor has to perform a set of maneuvers using the curves in Figure 4.6. The tractor has to make an initial maneuver to adjust the orientation of the trailer in order to have it properly follow $\mathcal{C}_{\text{rev,trailer}}$. The starting maneuver is shown in Figure 4.7.

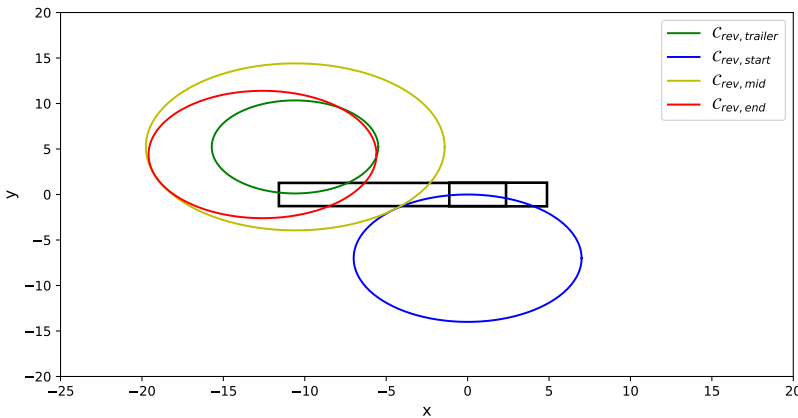
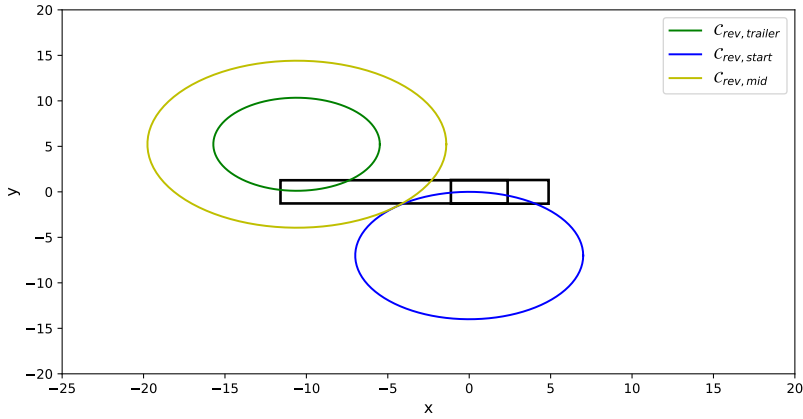
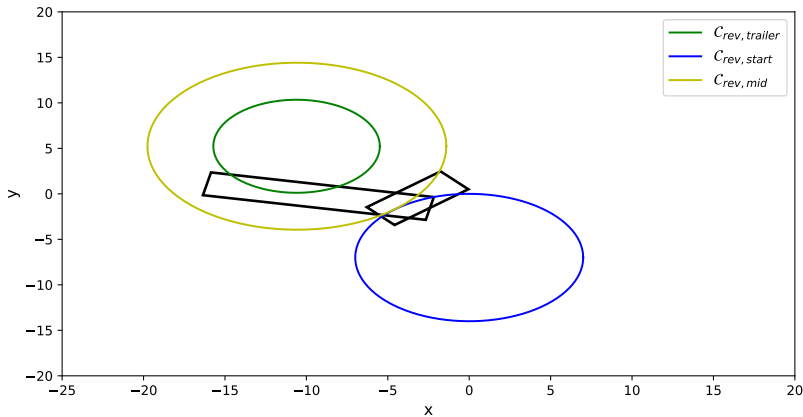


Figure 4.6: The curves that are used to properly follow the configuration curves, shown as $\mathcal{C}_{\text{rev,trailer}}$ in the figure.



(a) Start position of the tractor and trailer system where one of the reversing curves from the start configuration, as shown in Figure 4.1a, is chosen.



(b) Position of tractor and trailer system after the starting maneuver where the orientation of the trailer is turned to be able to follow the trailer curve.

Figure 4.7: The starting maneuver includes following the curve $C_{rev,start}$ until the orientation of the trailer is adjusted to properly follow $C_{rev,trailer}$.

As shown in Equations (4.7) and (4.8), the center position of each curve as well as the angles that the tractor start and end following the curves are needed for generating a reference. The center position of the tractor curve surrounding the configuration curve is the same as the center position of the configuration curve. The center position of $C_{rev,start}$ is perpendicular to the rear axle of the tractor and

its start orientation, meaning that the start angle on $\mathcal{C}_{\text{rev,start}}$ is:

$$\theta_{\mathcal{C}_{\text{rev,start}}}^{\text{start}} = \theta_1^{\text{start}} \mp \frac{\pi}{2}.$$

Since the curve radius is pre-defined for all curves, the length, $l_{\mathcal{C}_{\text{rev,start}}}$, that the tractor has to follow the curve $\mathcal{C}_{\text{rev,start}}$ to reach the desired trailer orientation is pre-defined as well, resulting in the end angle:

$$\theta_{\mathcal{C}_{\text{rev,start}}}^{\text{end}} = \theta_{\mathcal{C}_{\text{rev,start}}}^{\text{start}} \mp \frac{l_{\mathcal{C}_{\text{rev,start}}}}{r},$$

where r is the curve radius and the sign depends on the direction of $\mathcal{C}_{\text{rev,start}}$. Figure 4.7 shows that the end angle of curve $\mathcal{C}_{\text{rev,start}}$ and the start angle of curve $\mathcal{C}_{\text{rev,start}}$ are of opposite direction:

$$\theta_{\mathcal{C}_{\text{rev,mid}}}^{\text{start}} = \theta_{\mathcal{C}_{\text{rev,start}}}^{\text{end}} \pm \pi.$$

The center position of $\mathcal{C}_{\text{rev,start}}$ can be calculated using the equation:

$$\mathcal{C}_{\text{rev,start}}^{\text{center}} = \left(x_1^{\text{start}} - r \cos(\theta_{\mathcal{C}_{\text{rev,start}}}^{\text{start}}), y_1^{\text{start}} - r \sin(\theta_{\mathcal{C}_{\text{rev,start}}}^{\text{start}}) \right).$$

The center position of $\mathcal{C}_{\text{rev,mid}}$, the end position of curve $\mathcal{C}_{\text{rev,start}}$, and the start position of curve $\mathcal{C}_{\text{rev,mid}}$ can be calculated using the center position of $\mathcal{C}_{\text{rev,start}}$:

$$\mathcal{C}_{\text{rev,mid}}^{\text{center}} = \begin{pmatrix} x_{\mathcal{C}_{\text{rev,start}}}^{\text{center}} + (r_{\mathcal{C}_{\text{rev,start}}} + r_{\mathcal{C}_{\text{rev,mid}}}) \cos(\theta_{\mathcal{C}_{\text{rev,start}}}^{\text{end}}) \\ y_{\mathcal{C}_{\text{rev,start}}}^{\text{center}} + (r_{\mathcal{C}_{\text{rev,start}}} + r_{\mathcal{C}_{\text{rev,mid}}}) \sin(\theta_{\mathcal{C}_{\text{rev,start}}}^{\text{end}}) \end{pmatrix}^T,$$

$$\mathcal{C}_{\text{rev,start}}^{\text{end}} = \mathcal{C}_{\text{rev,mid}}^{\text{start}} = \begin{pmatrix} x_{\mathcal{C}_{\text{rev,start}}}^{\text{center}} + r_{\mathcal{C}_{\text{rev,start}}} \cos(\theta_{\mathcal{C}_{\text{rev,start}}}^{\text{end}}) \\ y_{\mathcal{C}_{\text{rev,start}}}^{\text{center}} + r_{\mathcal{C}_{\text{rev,start}}} \sin(\theta_{\mathcal{C}_{\text{rev,start}}}^{\text{end}}) \end{pmatrix}^T.$$

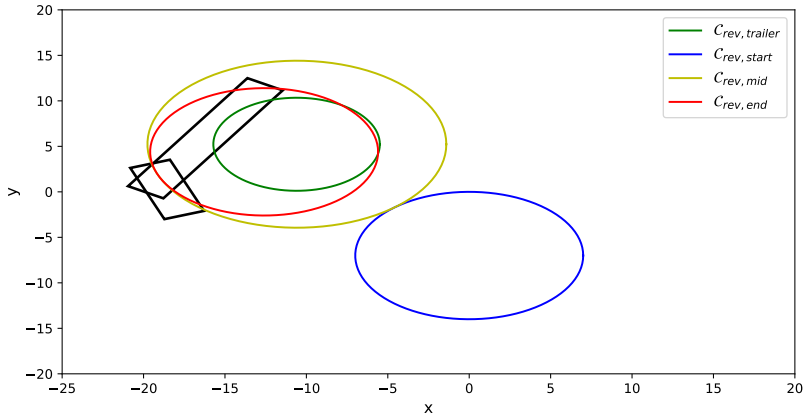
The goal of the finishing maneuver is to straighten out the tractor-trailer system and align its orientation with that of the goal. That is done by following a smaller curve with the tractor until the system is straightened out, as shown in Figure 4.8. The calculations start at the end point when deciding the needed properties of $\mathcal{C}_{\text{rev,end}}$. As shown in Figure 4.8b, the orientation of the goal is perpendicular to the end angle of the curve $\mathcal{C}_{\text{rev,end}}$:

$$\theta_{\mathcal{C}_{\text{rev,end}}}^{\text{end}} = \theta_1^{\text{goal}} \pm \frac{\pi}{2},$$

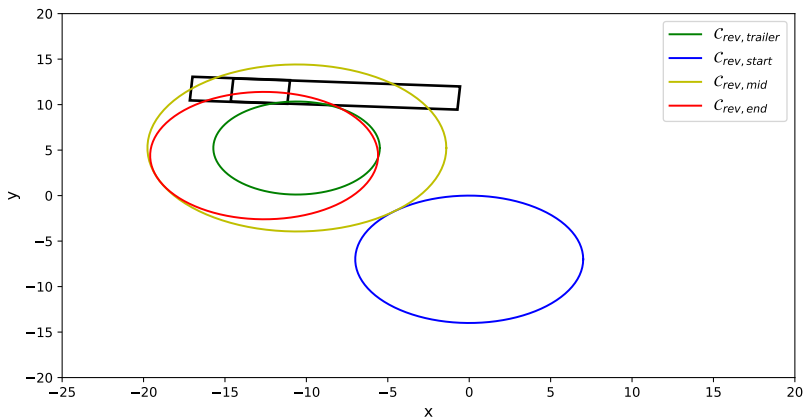
yielding the start angle:

$$\theta_{\mathcal{C}_{\text{rev,end}}}^{\text{start}} = \theta_{\mathcal{C}_{\text{rev,mid}}}^{\text{end}} = \theta_{\mathcal{C}_{\text{rev,end}}}^{\text{end}} \mp \frac{l_{\mathcal{C}_{\text{rev,end}}}}{r_{\mathcal{C}_{\text{rev,end}}}},$$

where $l_{\mathcal{C}_{\text{rev,end}}}$ is the length for that the curve $\mathcal{C}_{\text{rev,end}}$ is followed. The center position of curve $\mathcal{C}_{\text{rev,end}}$, end position of curve $\mathcal{C}_{\text{rev,mid}}$, and start position of curve



(a) Position of the tractor and trailer system when the finishing maneuver starts.



(b) Position of tractor and trailer system after the finishing maneuver where the orientation of the trailer is turned back to align with the tractor orientation.

Figure 4.8: The finishing maneuver when $C_{rev,end}$ is followed until the tractor and trailer system has been straightened out. The curve radius of $C_{rev,end}$ has to be greater or equal to the minimal turning radius of the tractor.

$C_{rev,end}$ can be calculated using the center position of curve $C_{rev,mid}$:

$$C_{rev,end}^{center} = \begin{pmatrix} x_{C_{rev,mid}}^{center} + (r_{C_{rev,mid}} - r_{C_{rev,end}}) \cos(\theta_{C_{rev,mid}}^{end}) \\ y_{C_{rev,mid}}^{center} + (r_{C_{rev,mid}} - r_{C_{rev,end}}) \sin(\theta_{C_{rev,mid}}^{end}) \end{pmatrix}^T,$$

$$C_{rev,mid}^{end} = C_{rev,end}^{start} = \begin{pmatrix} x_{C_{rev,mid}}^{center} + r_{C_{rev,mid}} \cos(\theta_{C_{rev,mid}}^{end}) \\ y_{C_{rev,mid}}^{center} + r_{C_{rev,mid}} \sin(\theta_{C_{rev,mid}}^{end}) \end{pmatrix}^T.$$

Equations (4.6), (4.7), and (4.8) are used to generate the reference for x_1 , y_1 and θ_1 since the center positions and relevant angles of the curves $\mathcal{C}_{\text{rev,start}}$, $\mathcal{C}_{\text{rev,mid}}$, and $\mathcal{C}_{\text{rev,end}}$ are known. An example of such a reference is shown in Figure 4.9.

Since the tractor-trailer system is unstable when reversing, using Equation (4.9) for generating the joint angle reference would require a very high resolution and many iterations. Therefore, the joint angle reference is generated by reversing the mission. Equation (4.9) is used to simulate the joint angle from the goal pose towards the start pose. The first state in the simulation corresponds to the last state in the joint angle reference and vice versa.

For the reversing straight segment, Equations (4.10) and (4.11) can be used for calculating the sample distance and orientation. However, since the vehicle is reversing, the orientation of the tractor is opposite to that of the straight, resulting in the reference pose:

$$\begin{pmatrix} x_1 \\ y_1 \\ \theta_1 \end{pmatrix} = \begin{pmatrix} x_{\text{straight}}^{\text{start}} + nds_{\text{straight}} \cos(\theta_{\text{straight}}) \\ y_{\text{straight}}^{\text{start}} + nds_{\text{straight}} \sin(\theta_{\text{straight}}) \\ -\theta_{\text{straight}} \end{pmatrix}, \quad n = 0, 1, \dots, \lceil \mathcal{S}_{\text{length}} \rceil.$$

Since each $\mathcal{C}_{\text{rev,end}}$ curve end with a joint angle of zero and each $\mathcal{C}_{\text{rev,start}}$ curve start in the same straightened system, the reference for the joint angle is zero for the straight.

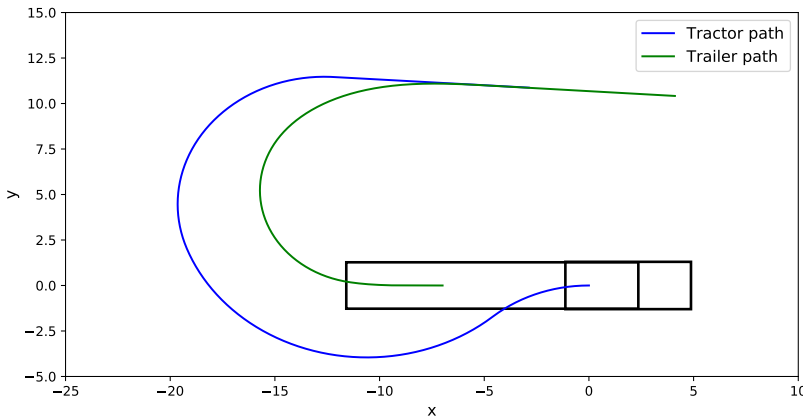


Figure 4.9: Reference path when reversing and following a configuration curve.

4.3 Calculating optimal curve radiuses for reversing maneuvers

The aim of the method is to choose the radiuses that generate the shortest path. From Reeds-Shepp, optimality is guaranteed when the minimal turn radius are used on the curves. However, in the tractor trailer case, that is not applicable since $C_{\text{rev,end}}$ must have a shorter turn radius than $C_{\text{rev,mid}}$ to be able to straighten out the system. Therefore, the different curve radiuses have to be included in the objective function. When optimizing, the true distance travelled along each curve will not be considered. Instead, a length dependent on the radius is chosen, according to the function:

$$\int_{t_n}^{t_{n+m}} R_C dt.$$

To incorporate the optimality criteria of Reeds-Shepp, the curves that do not have any constraints on the radius, are set with a radius corresponding to the minimal turn radius of the tractor, R_0 . These curves are $C_{\text{rev,start}}$ and $C_{\text{rev,end}}$, resulting in the objective function:

$$J = \int_0^{t_1} R_0 dt + \int_{t_1}^{t_2} R_{C_{\text{rev,mid}}} dt + \int_{t_2}^{t_f} R_0 dt, \quad (4.13)$$

where t_1 , t_2 , and t_f are time instants decided by the solver and $R_{C_{\text{rev,mid}}}$ is the radius of $C_{\text{rev,mid}}$. The constraint on the optimization problem adheres to the constraints of the kinematic model, described in Equation (2.6). However, the joint angle model is the only one used. The joint angle, as described by the kinematic model has been re-formulated such that it depends on the curve radius in Equation (4.9). Since all curves in the tractor maneuver vary in radius and direction, the constraint differs depending on which curve is being followed. With the objective function in mind, Equation (4.13), the constraint become:

$$\beta^{t+1} = -0.01 \left(-\frac{1}{R_0} - \frac{\sin(\beta^t)}{L_2} + \frac{M \cos(\beta^t)}{L_2 R_0} \right), \quad 0 \leq t \leq t_1 \quad (4.14a)$$

$$\beta^{t+1} = -0.01 \left(\frac{1}{R_{C_{\text{rev,mid}}}} - \frac{\sin(\beta^t)}{L_2} - \frac{M \cos(\beta^t)}{L_2 R_{C_{\text{rev,mid}}}} \right), \quad t_1 < t \leq t_2 \quad (4.14b)$$

$$\beta^{t+1} = -0.01 \left(\frac{1}{R_0} - \frac{\sin(\beta^t)}{L_2} - \frac{M \cos(\beta^t)}{L_2 R_0} \right), \quad t_2 < t \leq t_f. \quad (4.14c)$$

By studying the tractor and trailer system in Figures 4.7b and 4.8a, the joint angle is constant when $C_{\text{rev,mid}}$ is followed. Therefore, the change in the joint angle should be minimal when following that $C_{\text{rev,mid}}$; meaning that the constraint in Equation (4.14b) is limited on the right hand side of the equation. The final constraint on the system are the static constraints of the joint angle. The system has to start and end in a straightened system with a β value of zero and does not enter

any jack-knifing state by limiting the joint angle to a valid state, β^{\max} . The optimization problem may then be written as shown in Equation (4.15). Example 4.1 show one example of what the solver provides.

— **Example 4.1: Sample solution of optimization problem in Equation (4.15)** —

Given: $R_0 = 7.0m$, $\beta^{\max} = 60^\circ$, $M = 0.66m$, $L_2 = 7.65m$.

Result: $R = 9.18m$, $\beta^{\text{mid}} = 52.12^\circ$, $t_1 = 498s$, $t_f - t_2 = 1393s$

$$\text{minimize } J = \int_0^{t_1} R_0 dt + \int_{t_1}^{t_2} R_{C_{\text{rev,mid}}} dt + \int_{t_2}^{t_f} R_0 dt \quad (4.15a)$$

$$\text{subject to } \beta^{t+1} = -0.01 \left(-\frac{1}{R_0} - \frac{\sin(\beta^t)}{L_2} + \frac{M \cos(\beta^t)}{L_2 R_0} \right), \quad 0 \leq t \leq t_1 \quad (4.15b)$$

$$\beta^{t+1} = -0.01 \left(\frac{1}{R_{C_{\text{rev,mid}}}} - \frac{\sin(\beta^t)}{L_2} - \frac{M \cos(\beta^t)}{L_2 R_{C_{\text{rev,mid}}}} \right) \leq 0.05, \quad t_1 < t \leq t_2 \quad (4.15c)$$

$$\beta^{t+1} = -0.01 \left(\frac{1}{R_0} - \frac{\sin(\beta^t)}{L_2} - \frac{M \cos(\beta^t)}{L_2 R_0} \right), \quad t_2 < t \leq t_f \quad (4.15d)$$

$$R \geq R_0 \quad (4.15e)$$

$$\beta^{\text{init}} = 0 \quad (4.15f)$$

$$\beta^{\text{final}} \leq 0.05 \quad (4.15g)$$

$$\beta \leq \beta^{\max}. \quad (4.15h)$$

5

Results

This chapter will present references generated by an RRT with a planning method that uses the tractor-trailer curves presented in Chapter 4. The length and computational time of the generated references will be compared to references generated by a lattice-based motion planner. The lattice-based motion planner used is a modified version of the state lattice planner developed in the course TSRT10 given at Linköping University ¹. It was originally developed for a miniature LEGO 2-trailer system but has been modified to plan for a full scale single joint tractor-trailer system. Furthermore, the state lattice planner uses a heuristic look-up table and a fairly high resolution of 0.5m since decreasing the computational time is a prioritized objective of the planner.

5.1 Simple scenarios in an open environment

A reference between two random poses in an open environment without obstacles was simulated 672 times. The RRT generated a shorter reference than the state lattice planner in 624 simulations. The average difference in reference length was 25.3% shorter in favor of the RRT. The average number of nodes in the RRT tree was 10.2, however, each generated reference only had two nodes in all simulations; the start and goal pose. The average computational time for the RRT was 0.267s while the state lattice planner had an average computational time of 0.875s. Figure 5.1 shows the result of one simulation. Note that the state lattice planner only provides a reference for the trailer; meaning that the trailer references were used when calculating the length. A summary of all simulations is listed in Table 5.1

¹http://www.isy.liu.se/edu/projekt/tsrt10/2020/rev_truck/

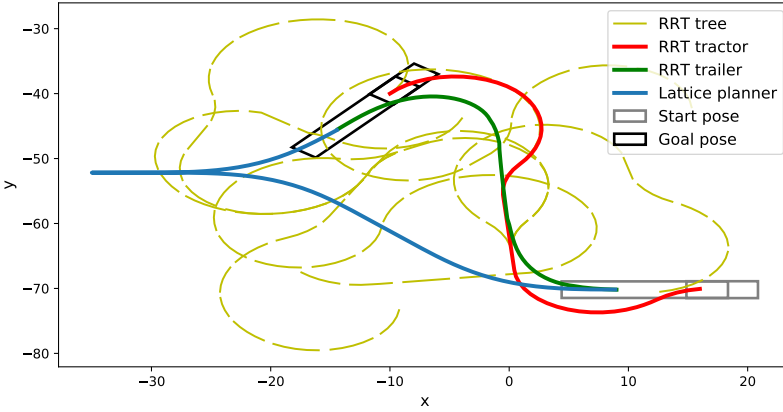


Figure 5.1: An example of a generated reference between two random poses in an open environment without obstacles. The length of the RRT reference is 47.7m and the state lattice reference is 72.4m. The RRT had a computational time 0.237s and 12 nodes in its tree. The state lattice planner had a computational time of 0.783s.

Table 5.1: Summary of simulations in an open environment without obstacles. The reference length and computational time values are comparisons to the references generated by the state lattice planner.

Length [%]	Time [%]	Nodes in tree [-]	Nodes in reference [-]
-25.3	-69.5	10.2	2

520 simulations were made between a random pose and a pose 30m in front of it with the same heading and a 40m wide obstacle between. The RRT outperformed the state lattice planner here as well. The average difference in reference length was 12.6% shorter in favor of the RRT. The average number of nodes in the RRT tree was 36.1 and the average number of nodes in the generated reference was 3.4. The average computational time for the RRT was 0.292s while the state lattice planner had an average computational time of 1.076s. Figure 5.2 shows the result of one simulation. It is shown that the first and second curve of the RRT reference are parts of one local \mathcal{CSC} path, however, the local path following is only a \mathcal{SC} path. Here, the local planing method neglected the first curve of the path since the local start pose already had the optimal heading. The behavior of the reference generated by the state lattice planner was identical for all simulations. A summary of all simulations is listed in Table 5.2

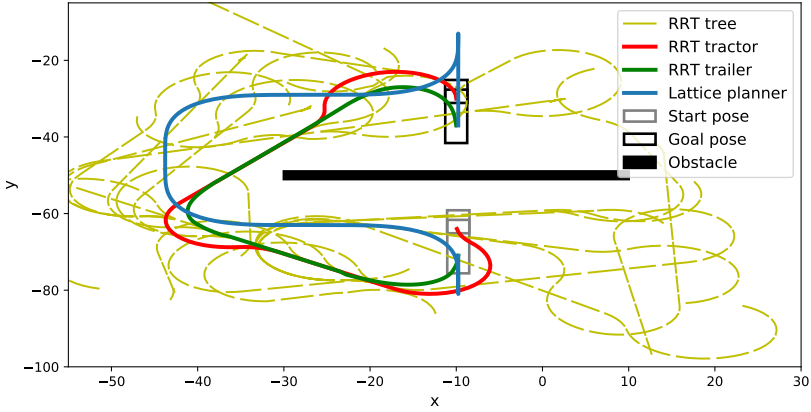


Figure 5.2: An example of a generated reference between a random pose and a pose 30m in front of it with the same heading and a 40m wide obstacle between. The length of the RRT reference is 99.0m and the state lattice reference is 152.1m. The RRT had a computational time 0.281s and 41 nodes in its tree. The state lattice planner had a computational time of 0.983s.

Table 5.2: Summary of simulations in an open environment with a 40m wide obstacle separating the start and goal pose. The reference length and computational time values are comparisons to the references generated by the state lattice planner.

Length [%]	Time [%]	Nodes in tree [-]	Nodes in reference [-]
-12.6	-72.9	36.1	3.4

20 simulations were made between a random pose and a pose 20m behind with opposite orientation. Both the start and goal pose are trapped in a 20m wide parking spot. The average difference in reference length was 4.1% shorter in favor of the RRT. The average number of nodes in the RRT tree was 41.7 and the average number of nodes in the generated reference was 3.8. The average computational time for the RRT was 0.301s while the state lattice planner had an average computational time of 1.205s. Figure 5.3 shows the result of one simulation. A summary of all simulations is listed in Table 5.3

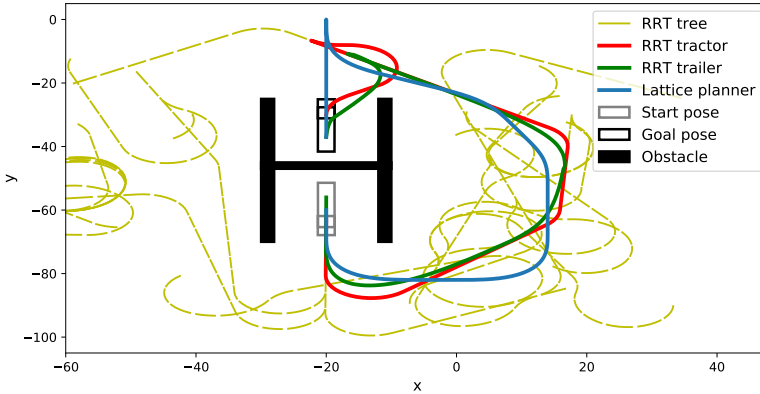


Figure 5.3: An example of a generated reference between a random pose trapped in a 20m wide parking space and a trapped pose 20m behind with opposite orientation. The length of the RRT reference is 163.9m and the state lattice reference is 181.1m. The RRT had a computational time 0.271s and 37 nodes in its tree. The state lattice planner had a computational time of 1.131s.

Table 5.3: Summary of simulations in an open environment where the start and goal poses are trapped in a 20m wide parking space. The reference length and computational time values are comparisons to the references generated by the state lattice planner.

Length [%]	Time [%]	Nodes in tree [-]	Nodes in reference [-]
-4.1	-75.0	41.7	3.8

5.2 Cage obstacle in an open environment

20 simulations were made between a pose trapped in a cage with an opening of 10m, 15m, and 20m respectively to a random pose outside the cage. The RRT was only able to find a path in 25% of the simulations where the opening was 10m wide, 85% of the simulations where the opening was 15m wide and 100% of the simulations where the opening was 20m wide. The state lattice planner was able to find a path in every simulation. The average difference in reference length was 11.0% shorter in favor of the state lattice planner. The average number of nodes in the RRT tree was 67.9 and the average number of nodes in the generated reference was 6.1. The average computational time for the RRT was 0.472s while the state lattice planner had an average computational time of 1.756s. Figure 5.4 shows the result of one simulation. A summary of all simulations is listed in Table 5.4

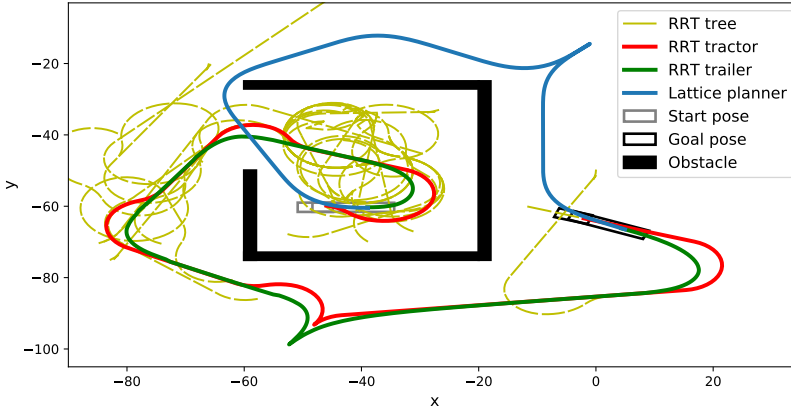


Figure 5.4: An example of a generated reference between a pose trapped in cage with an opening of 20m and a random pose outside the cage. The length of the RRT reference is 222.6m and the state lattice reference is 177.8m. The RRT had a computational time 0.395s and 53 nodes in its tree. The state lattice planner had a computational time of 1.404s.

Table 5.4: Summary of simulations in an open environment between a pose trapped in a cage with an opening of 10m, 15m, and 20m respectively and a random pose outside of the cage. The reference length and computational time values are comparisons to the references generated by the state lattice planner.

Length [%]	Time [%]	Nodes in tree [-]	Nodes in reference [-]
+12.4	-73.1	69.7	6.1

The RRT managed to find a path more frequently when the goal was to enter a cage instead of exiting. The RRT found a path in 50% of the simulations when the opening was 10m, 100% of the simulations when the opening was 15m and 20m. The state lattice planner was able to find a path in every simulation here as well. The average difference in reference length was 10.1% shorter in favor of the state lattice planner. The average number of nodes in the RRT tree was 65.7 and the average number of nodes in the generated reference was 5.4. The average computational time for the RRT was 0.422s while the state lattice planner had an average computational time of 1.687s. Figure 5.5 shows the result of one simulation. A summary of all simulations is listed in Table 5.5

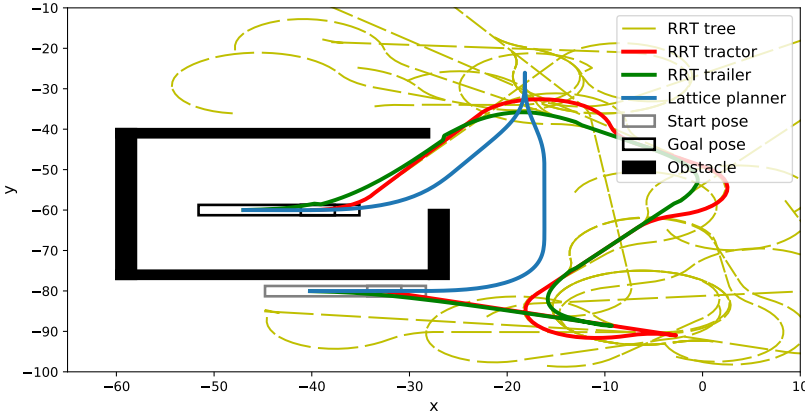


Figure 5.5: An example of a generated reference between a random pose outside of a cage and a pose trapped in cage with an opening of 20m. The length of the RRT reference is 141.5m and the state lattice reference is 125.3m. The RRT had a computational time 0.425s and 70 nodes in its tree. The state lattice planner had a computational time of 1.118s.

Table 5.5: Summary of simulations in an open environment between a random pose outside of a cage with an opening of 10m, 15m, and 20m respectively and a pose trapped in the cage. The reference length and computational time values are comparisons to the references generated by the state lattice planner.

Length [%]	Time [%]	Nodes in tree [-]	Nodes in reference [-]
+11.2	-75.0	65.7	5.4

5.3 Large open environment

100 simulations were made between two random poses separated by approximately 100m in an open environment without any obstacles. The RRT behaved similarly as when the smaller environment without obstacles was used. All references generated by the RRT only had two nodes; start and goal pose. The average number of nodes in the RRT tree was 12.2. The average difference in reference length was 18.7% shorter in favor of the RRT. The average computational time for the RRT was 0.273s while the state lattice planner had an average computational time of 1.201s. Figure 5.6 shows the result of one simulation. A summary of all simulations is listed in Table 5.6

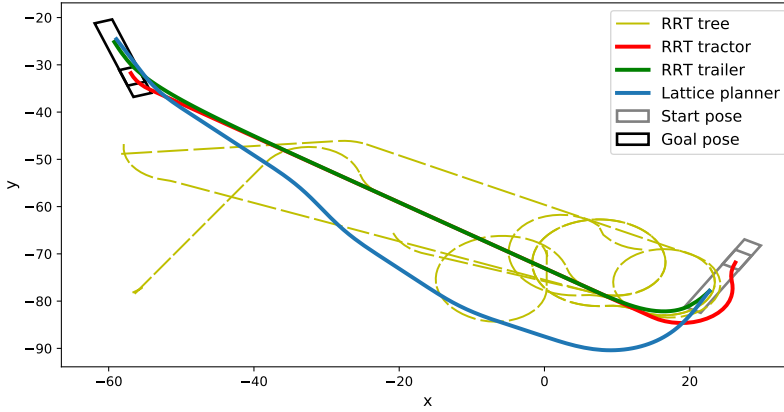


Figure 5.6: An example of a generated reference between two random poses in a large open environment without obstacles. The length of the RRT reference is 104.0m and the state lattice reference is 116.9m. The RRT had a computational time 0.237s and 12 nodes in its tree. The state lattice planner had a computational time of 1.267s.

Table 5.6: Summary of simulations in a large open environment without any obstacles. The reference length and computational time values are comparisons to the references generated by the state lattice planner.

Length [%]	Time [%]	Nodes in tree [-]	Nodes in reference [-]
-18.7	-77.3	12.2	2

Obstacles were introduced into the environment and 100 simulations with the same start and goal poses were made. The average difference in reference length was 3.7% shorter in favor of the RRT. The average number of nodes in the RRT tree was 39.2 and the average number of nodes in the generated reference was 4.2. The average computational time for the RRT was 0.342s while the state lattice planner had an average computational time of 1.502s. Figure 5.7 shows the result of one simulation. A summary of all simulations is listed in Table 5.7

Table 5.7: Summary of simulations in a large open environment with obstacles between the start and goal poses. The reference length and computational time values are comparisons to the references generated by the state lattice planner.

Length [%]	Time [%]	Nodes in tree [-]	Nodes in reference [-]
-3.7	-77.2	39.2	4.2

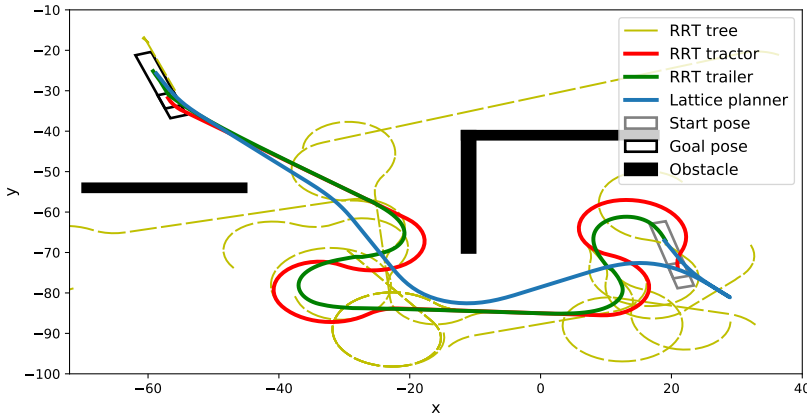


Figure 5.7: An example of a generated reference between two random poses in a large open environment without obstacles. The length of the RRT reference is 165.8m and the state lattice reference is 138.5m. The RRT had a computational time 0.327s and 33 nodes in its tree. The state lattice planner had a computational time of 1.598s.

Figure 5.8 shows the relationship between the number of nodes in the RRT tree and the computational time for all simulations.

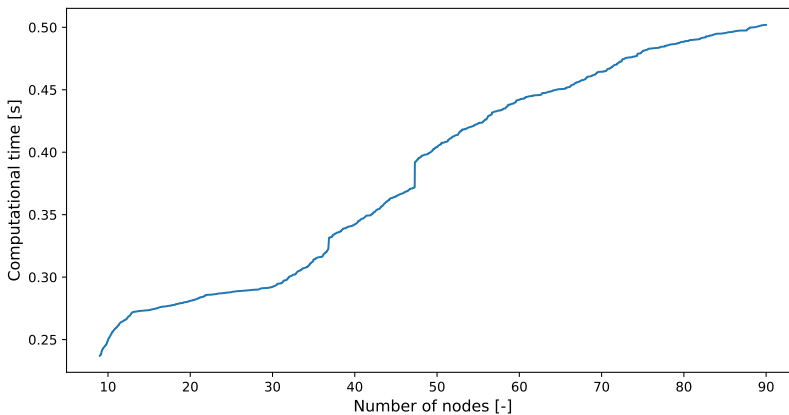


Figure 5.8: Relationship between the number of nodes in the RRT tree and the computational time of the RRT planner.

6

Conclusion

In this chapter, answers to the research questions stated in Section 1.2.2 are given using the method described in Section 4 and conclusions drawn from the results in Section 5.

6.1 Choosing optimal curve radius

The optimal curve radiuses are chosen by solving the optimization problem stated in Equation (4.15). They depend on the minimal turning radius of the desired vehicle and the largest allowed joint angle between the tractor and trailer. The optimization problem share a constraint on the joint angle with the tractor-trailer kinematic model that guarantees that the given turn radiuses are feasible. Since the vehicle will only turn according to these turning radiuses or reverse in a straight line, the generated references are ensured to be feasible.

6.2 Comparison to the state lattice planner

Simulations presented in Section 5 clearly show that the RRT planner with a planning method using the tractor-trailer curves is faster than the state lattice planner when a path can be found. This is reasonable since the tractor-trailer planning method is generating the reference purely geometrically while the state lattice planner solves a graph problem. The computational time of the RRT planner depends on the number of nodes in the RRT tree, that is clearly shown in Figure 5.8. The length of the generated references does not noticeably effect the computational time. That is shown by observing the average computational time of the open environment without obstacles scenario for the larger and smaller environ-

ment. This is a major advantage compared to the state lattice planner where the dimensions of the graph problem increases when the environment is larger.

The difference in length of the generated paths depend on the obstacles in the environment. The RRT generates nearly optimal paths when no obstacles are present since the reference only contain the start and goal poses and the vehicle only turns according to optimal curve radiuses. The only scenarios where the state lattice planner out-performed the RRT planner was when the optimal reference was a straight reversing path. The reason for this is that the tractor-trailer curves does not perform optimally when turning less than the maneuver require. However, the length of the generated references never differed more than 0.5m when reversing straight.

The generated references are far from optimal when obstacles are introduced into the environment. The RRT still generate shorter references than the state lattice planner. The environment where a cage obstacle is present show the importance of sampling strategy in an RRT. The state lattice planner has the luxury of always being able to find a path in a complex environment as long as there exist a path and the resolution of the search space is smaller than the smallest opening in the environment. The RRT does not have a similar guarantee since it heavily depends on its sampling strategy. However, the cage openings in the simulations are all large enough for the RRT to find a path if the sampling strategy is improved. Simulations were made where the RRT planner was allowed to run for several minutes without finding a path. The sampling strategy of the RRT was not improved during this thesis since it was deemed to be outside of the thesis scope. The sampling strategy used is a general sampler developed by Scania.

Another advantage of the planning method that uses the tractor-trailer curves is that it can generate a reference between any two random poses. Figures 5.2 and 5.6 clearly show that the goal pose differ between the two references. The reason for this is that the state lattice planner approximate the start and goal pose to closest vertex in the directed graph. Issues can occur depending on the resolution of the state space.

6.3 Further development

A needed improvement is to advance the sampling strategy. As mentioned earlier, the quality of the generated references heavily depends on it. The current sampling strategy did not find a path every time in the cage simulations even if it was allowed to generate more samples. A sampling strategy where a high number of random samples are generated could improve the references since the sampling method is fast.

A major issue with the tractor-trailer curves is when a reference for straight reversing is needed. One solution is to simplify the optimization problem stated in equation (4.15) to only consider $C_{rev,start}$ and $C_{rev,end}$. The length for that the

configuration curve is followed can be used for t_f and the solution would be t_1 :

$$\text{minimize } J = \int_0^{t_1} R_0 dt + \int_{t_1}^{t_f} R_0 dt \quad (6.1a)$$

$$\text{subject to } \beta^{t+1} = -0.01 \cdot \left(-\frac{1}{R_0} - \frac{\sin(\beta^t)}{L_2} + \frac{M \cos(\beta^t)}{L_2 \cdot R_0} \right), \quad 0 \leq t \leq t_1 \quad (6.1b)$$

$$\beta^{t+1} = -0.01 \cdot \left(\frac{1}{R_0} - \frac{\sin(\beta^t)}{L_2} - \frac{M \cos(\beta^t)}{L_2 \cdot R_0} \right), \quad t_1 < t \leq t_f \quad (6.1c)$$

$$\beta^{\text{init}} = 0 \quad (6.1d)$$

$$\beta^{\text{final}} \leq 0.05. \quad (6.1e)$$

The method of using curves for generating a reference for tractor-trailer systems can be expanded to include more trailers in the system since the majority of the calculations are done offline. The general n-trailer kinematic model that is stated in Section 2 can be used to derive the constraints for an extended optimization problem, similar to the one given in Equation (4.15).

Bibliography

- [1] C. Altafini. The general n-trailer problem: conversion into chained form. In *Proceedings of the 37th IEEE Conference on Decision and Control (Cat. No.98CH36171)*, volume 3, pages 3129–3130 vol.3, 1998. doi: 10.1109/CDC.1998.757991.
- [2] Sebastian Blanco. Advanced driver-assistance systems. what the heck are they anyway? *Forbes*, Accessed May 31 2021, Updated May 2020. URL <https://www.forbes.com/wheels/advice/advanced-driver-assistance-systems-what-are-they/>.
- [3] Gordon Dickson. Driverless trucks are already on texas roads with more to come. *the Fort Worth Star-Telegram*, Accessed May 31 2021, Updated November 23 2020. URL <https://www.govtech.com/fs/automation/driverless-trucks-are-already-on-texas-roads-with-more-to-come.html>.
- [4] N. Evestedt, O. Ljungqvist, and D. Axehill. Motion planning for a reversing general 2-trailer configuration using closed-loop rrt. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3690–3697, 2016. doi: 10.1109/IROS.2016.7759544.
- [5] Olov Holmer. Motion planning for a reversing full-scale truck and trailer system. Master’s thesis, Linköping University, Automatic Control, 2016.
- [6] Sertac Karaman and Emilio Frazzoli. Sampling-based algorithms for optimal motion planning. *The International Journal of Robotics Research*, 30(7):846–894, 2011. doi: 10.1177/0278364911406761. URL <https://doi.org/10.1177/0278364911406761>.
- [7] Yoshiaki Kuwata, Justin Teo, Gaston Fiore, Sertac Karaman, Emilio Frazzoli, and Jonathan P. How. Real-time motion planning with applications to autonomous urban driving. *IEEE Transactions on Control Systems Technology*, 17(5):1105–1118, 2009. doi: 10.1109/TCST.2008.2012116.

- [8] Karl Kvarnfors. Motion planning for parking a truck and trailer system. Master's thesis, KTH, School of Electrical Engineering and Computer Science (EECS), 2019.
- [9] S. LaValle. Rapidly-exploring random trees : a new tool for path planning. *The annual research report*, 1998.
- [10] Steven Lavalle and James Kuffner. Rapidly-exploring random trees: Progress and prospects. *Algorithmic and computational robotics: New directions*, 01 2000.
- [11] Steven M. LaValle. *Planning Algorithms*. Cambridge University Press, USA, 2006. ISBN 0521862051.
- [12] Steven M. LaValle. *Motion Planning*, page 63–65. Cambridge University Press, 2006. doi: 10.1017/CBO9780511546877.004.
- [13] Nova Lines. Truck driver work conditions on the job. *Nova Lines*, Accessed May 31 2021, Updated July 18 2019. URL <https://novalines.com/truck-driver-work-conditions-on-the-job/#>.
- [14] O. Ljungqvist, N. Evestedt, M. Cirillo, D. Axehill, and O. Holmer. Lattice-based motion planning for a general 2-trailer system. In *2017 IEEE Intelligent Vehicles Symposium (IV)*, pages 819–824, 2017. doi: 10.1109/IVS.2017.7995817.
- [15] Oskar Ljungqvist. Motion planning and stabilization for a reversing truck and trailer system. Master's thesis, Linköping University, Automatic Control, Faculty of Science & Engineering, 2015. URL <http://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-119773>.
- [16] Oskar Ljungqvist. *Motion planning and feedback control techniques with applications to long tractor-trailer vehicles*. PhD thesis, Linköping University, 2020.
- [17] Oskar Ljungqvist, Niclas Evestedt, Daniel Axehill, Marcello Cirillo, and Henrik Pettersson. A path planning and path-following control framework for a general 2-trailer with a car-like tractor. *Journal of Field Robotics*, 2019. doi: 10.1002/rob.21908. Funding Agencies|Strategic vehicle research and innovation programme (FFI); Scania CV [2017-01957].
- [18] J. A. Reeds and L. A. Shepp. Optimal paths for a car that goes both forwards and backwards. *Pacific J. Math.*, 145(2):367–393, 1990. URL <https://projecteuclid.org:443/euclid.pjm/1102645450>.