BACHELOR

Design of a Graphical and Physical Representation of a Road Network for Path Planning of Autonomous Vehicles

Suzuki, Rikuto

Award date:
2022

Bachelor End Project - 4WC00

EINDHOVEN UNIVERSITY OF TECHNOLOGY

Design of a Graphical and Physical Representation of a Road Network for Path Planning of Autonomous Vehicles

Quartile 3-4 - 2021-2022

Full Name	Student ID	Study
Rikuto Suzuki	1413694	Mechanical Engineering

Report Number: DC 2022.063
Dynamics and Control Research Group

Eindhoven, June 28, 2022

Contents

List of Symbols

Table 1: List of Symbols

Quantity	**Symbol**
Set of road segment IDs to be connected	I_r
Length of a road segment	$l_{l\text{total}}$
Length of a parking lot	l_{pa}
Number of lanes of a road segment	n_l
Position of its center	p_c
Diverging point	p_{diverge}
Road segment position data	p_l
Merging point	p_{merge}
Radius of a roundabout	r
Outer radius of a roundabout	r_i
Inner radius of a roundabout	r_o
Start point of a parking lot with respect to the associated road segment length	s_{pa}
Traffic matrix	T
Angle of start and end points of outer boundary line of a roundabout	θ_b
Coefficient for moving the control point along the gradient of a lane	u_{shift}
Lane width of a road segment	w_l
Width of a parking lot	w_{pa}
Width of a circular virtual lane of a roundabout	w_r
Set of x coordinates of a road segment	x_{road}
Set of y coordinates of a road segment	y_{road}
Set of x coordinates of a drawn Bezier curve	x_b
Set of y coordinates of a drawn Bezier curve	y_b

Chapter 1

Introduction

Autonomous driving technology, which has developed rapidly in recent years, has been the focus of much attention in the automotive industry. Occupant safety is essential to autonomous driving, which requires accuracy at all levels of the decision-making process. This includes accurate road network information, path planning and real-time motion planning for obstacle avoidance, lane changes, and intersection negotiation, as well as vehicle positioning with robust control systems [10]. Road network data is all information about a road, including geometric information such as the length and width of road segments and attribute information such as traffic laws at intersections (e.g., whether U-turns are allowed). Such information is fundamental and important because all other levels are built upon it, and therefore must be sufficiently accurate to make appropriate driving decisions on a complex road network. Furthermore, it is also important to store the vast amount of data on the road network in a compact form that can be updated and downloaded wirelessly [6]. In concrete terms, for example, in the case of road segments, information such as the location of roads can be made into a function, so that instead of storing all the data points, a single function representing many data points can be stored, thereby making the information compact. This project focuses on how to efficiently accumulate road network data, generate graphical representations of the road network, and build algorithms to path planning based on this data.

In chapter 2, which road network elements are considered, how those elements are connected in a graphical representation, and how they are represented in a realistic visualization are explained. In chapter 3, the construction of paths using functions corresponding to lanes is explained. In chapter 4, the simulation results of the generated road network and path planning algorithm are discussed. Finally, chapter 5 summarizes the project, discusses areas for improvement, and offers suggestions for future work.

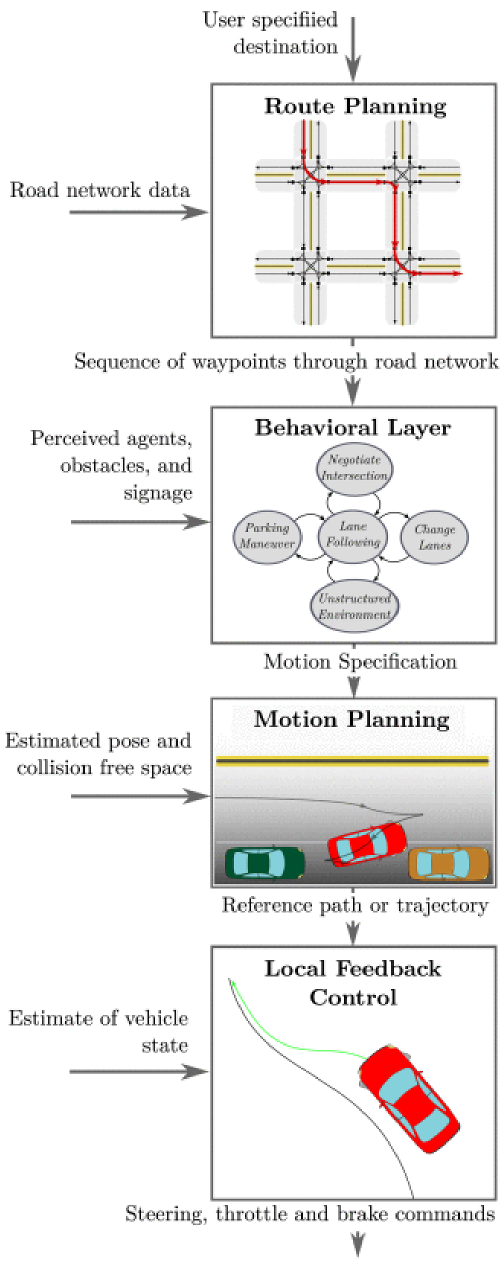


Figure 1.1: Illustration of the hierarchy of decision-making process. To safely perform autonomous driving, the system must have sufficient accuracy and a robust decision-making process. These include path planning, behavioral decision making such as obstacle avoidance, local motion planning, and vehicle position with robust control system. [10]

1.1 Project Description

The goal of this project is to design and implement a lane-level representation of the road network for path planning of autonomous vehicles. This project describes the design of a graphical representation of the road network, including single lanes, intersections, roundabouts, and traffic laws at intersections (e.g., U-turns, right turns allowed/not allowed). The graphical representation generated by the above method is used to find the shortest path between two nodes and combined with the corresponding pre-built realistic paths to generate the overall path to the destination.

In order to generate graphical representation of road network, the placement of nodes (event points) representing road network elements is first discussed. Next, the connectivity (edges) of the placed nodes is discussed, including the inter-connectivity of nodes belonging to different road network elements. Once the road network graph is generated, the exploratory path planning algorithm and the method used to calculate travel costs on the road network are discussed. Finally, methods for constructing realistic path and path conversion from the road network graph to the physical road network are discussed. Once the road network graph and path planning algorithm have been created, numerical simulations is performed to verify their performance.

1.2 Literature Review

In this section, existing literature related to road network generation and path planning algorithms is presented.

1.2.1 Autonomous Driving Techniques

Autonomous driving requires a certain level of safety and the safety task is a crucial in every single decision-making process [10]. This literature presents the current state of the art in planning and control algorithms and is useful to first understand what autonomous driving is, what path and trajectory planning is, and what vehicle dynamics is. It is also useful to analyze what path planning methods are used, their limitations and advantages.

1.2.2 Road Data Acquisition

There are several methods for extracting lane-level road geometry from road network data collected by sensors: trajectory-based, 3D point cloud-based, and vision-based [16]. The literature also introduces a mathematical modeling approach for lane-level road networks (Lanes at road segments and intersection).
As for data filtering, by performing morphological dilation and erosion on the collected data [15], outliers can be removed and the data can be augmented.

1.2.3 Road Network

High-definition (HD) maps, which are gaining attention in advanced automated driving technology, play an important role in automated vehicles. In addition to providing detailed map information to support HD positioning of smart cars, HD maps can help determine optimal driving routes using global path planning based on prior knowledge of maps and dynamic traffic information, effectively improving driving safety [17]. The literature presents a more in-depth method of generating HD road networks, including a typical line extraction method based on the PCA algorithm and a method of representing locations where road attributes change(event points).
When modelling a road network, following three requirements must be met [6];

- Centimeter-level accuracy:
  - The geometry of all lanes with accuracy at the centimeter level and the 3-D structure of roads to represent various types of roads, including sloped roads, overpasses, etc.
- Storage efficiency:
  - The geometry of the road should be expressed in a compact form such that the map can be downloaded and updated via wireless networks.
    - e.g. Reduce the number of functions representing roads and intersections
- Usability:
  - The road geometry should be expressed in an application-friendly format for autonomous decision making process

1.2.4 Path Planning

When constructing a global path to a certain destination, it consists of a set of individually generated paths. The global continuity and smoothness of the global path is important from the standpoint of comfort, so that the movement of the vehicle does not become jerky when transitioning to the next path. There are several types of functions that are used to generate paths and road networks; for example, Bezier curve [18], Clothoid, B-spline curves and several types of spline curves [6] [14] are proposed as an efficient parametric representation of smooth paths in literature in the history of this technological development so far.

1.2.5 Virtual Lanes

For road network elements that do not have physical lanes to mark travel routes, such as intersections, virtual lanes are used to construct driving path under these circumstances. Virtual lanes are widely used for autonomous navigation at intersections [8]. Driving lines can be used to abstractly represent virtual lanes with information on both the path and normal turning traffic [5]. An example of a virtual lane is shown in Figure 1.2.

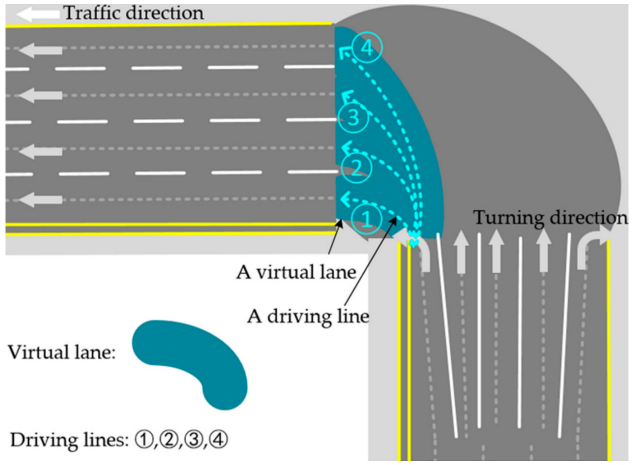


Figure 1.2: An example of the virtual lanes at an intersection [16]

Chapter 2

Road Network

One goal of this project is to establish a method for generating a graphical representation of a given road network. This section therefore presents the road network elements implemented in the graphical representation and the assumptions made about them and how the nodes are arranged. In addition, the nodes within the elements, the connectivity between the elements, how the nodes are connected, and how to detect pairs of nodes that need to be connected are discussed. In parallel, a realistic visualization based on information given about the road network (length, width, traffic laws, etc.) is also presented.

2.1 Graphical Representation of Road Network

Graphical representation of road network shows the spacial relations between two adjacent intersections such as how they are connected by road segments. There are many different ways of graphically representing the road network depending on literature.
Important information to be included in the graphical representation in general is as follows [9];

- Metric relations: take into account quantitative aspects of space
- Topological relations: give qualitative information about geometrical interactions between two spatial objects
- Order relations: specify the relative position of two spatial objects

The details of representation, such as the direction of travel on road network elements and connectivity of them, vary slightly from literature to literature. There are two main types of graphical representations of road networks: primal and dual graphs, as shown in Figure 2.1.

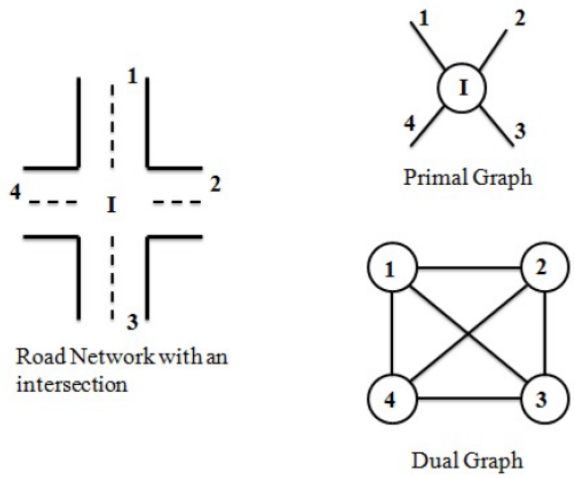


Figure 2.1: Primal and Dual graph representation [9]

In primal graph representation, a road network is represented by a graph whose nodes are the intersections and the edges are the roads segments joining the two adjacent intersections. This is the most commonly used method, found in various studies. In dual graph representation, road segments are represented by nodes instead of edges, and when two road segments intersect, an edge representing the intersection is added between the nodes.

Advantages of the primal graph is that it is more natural for human understanding because the attribute of the road segment such as the length can be associated with the edge of the graph. Furthermore, any types of intersections(T, X, roundabout etc.) can be expressed as a node, which improves the visibility of the graph. Advantages of the dual graph is that it can express road network which is much closer to the real setup where the primal graph is more abstracted version of it. Thus, the dual graph better represents the relationship between the two road segments, i.e., a more realistically represented traffic flow at the intersection. Since lane connectivity at intersections is important for path planning and physical path construction, dual graph representation is chosen for this project.

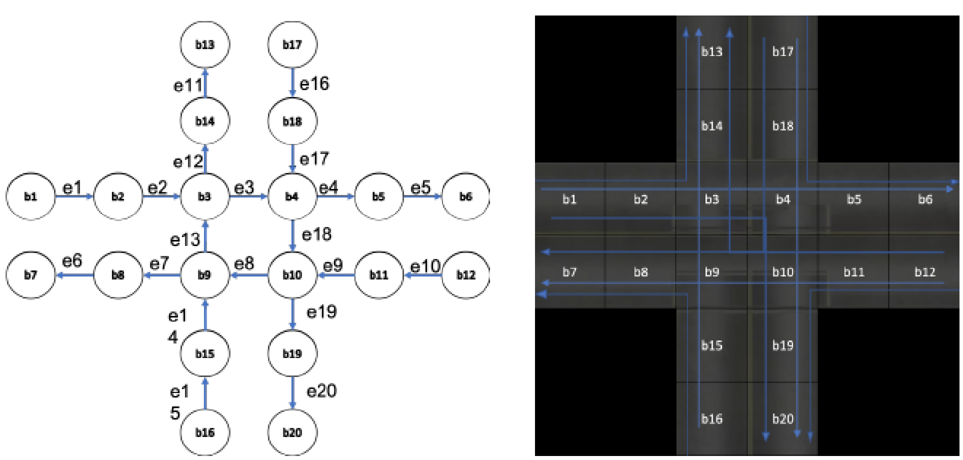


Figure 2.2: Example of dual graph representation of road network [11]

Figure 2.2 shows the example of incorporating the dual graph in the road network representation. In the example, the road segments are further divided into multiple sections. Nodes $b3$, $b4$, $b9$ and $b10$ are the end/ start of the road segments and they are connected by edges $e3$, $e8$, $e13$ and $e18$ to express the relations between the road segments at the intersection. The need for dual graphs depends on the application and whether it requires such detailed information about connectivity of lanes.

Table 2.1: Comparison between the primal and dual graph representation

	Primal Graph	Dual Graph
Advantages	• Simple and intuitive for human understandings • Easier to see which road segments are connected to which intersections	• More detailed visualization of the intersections – Connectivity of lanes – Whether U-turn is allowed
Disadvantages	• Lane connectivity at intersections are not visualized	• The intersections are less visible • Visualization of the road network becomes more complex

2.2 Road Network Elements

The road network refers to the spatial distribution state of the various road network elements. Functionally, it consists of various roads (highways, normal roads, and pedestrian roads), as well as intersections where roads intersect, runabouts, parking lots etc., and has a variety of information about the elements. From a micro perspective, each road network element includes traffic laws, such as the need for a stop and whether or not a U-turn is allowed at an intersection. By representing these road network elements as nodes and the edges connecting them, and storing their attribute information in a certain format, the physical road network can be abstracted to a graphical representation. This allows the shortest path to a certain destination to be calculated based on the length of the edges and converted into a physical road network path. The four road network elements implemented in the model in this project are road segments, intersections, roundabouts, and parking lots, which are the basic elements of the actual road network.

2.2.1 Road Segment

Road segment is defined as a road that connects two adjacent intersection, and it has a start/end points at the intersections. A road segment is defined as a road connecting two adjacent intersections, with a starting and ending point within those two intersections, respectively. A road segment may be divided into one carriageway (one-way) or two carriageways (multi-directional). There are several ways to express road segments in graphical representation of road network. If a road has lanes in opposite directions, whether it is represented by two arrows or not varies from literature to literature. Some literature do not indicate the direction of the road segment, while others use arrows or colors.

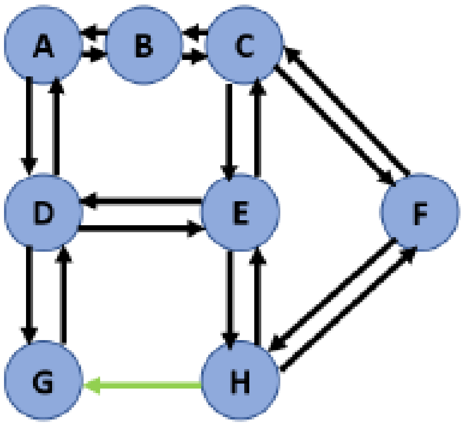


Figure 2.3: Graphical representation of road network expressing each lanes going different directions in road segments [18]

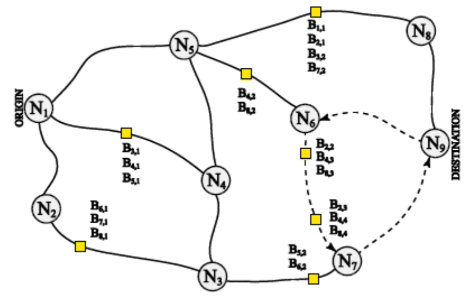


Figure 2.4: Graphical representation of road network expressing lanes going different directions as one line in road segments [3]

Figure 2.3 expresses a road segment with both direction as two lines with an arrow to the direction, and one with only one direction as a line with an arrow. On the other hand, Figure 2.4 expresses a road segment with lanes going both directions as a single line and the one with lanes going one direction as a dotted line with its direction.
The road network shown in Figure 2.3 is clearer and intuitive for human understandings because the directions of the road segments are clearly expressed with arrows, and the one-way road is expressed in a different color. On the other hand, in terms of simplicity, the one shown in Figure 2.4 is more preferable because there are less lines to show road segments and the difference between one-way and multi-way road segments is sufficiently visualized by the solid and dotted lines. To represent lane connectivity at intersections, lanes must be visualized on the graph representation. For this reason, the Figure 2.3 method is selected for lane visualization (Color is ignored).

2.2.2 Intersection

An intersection is an element that connects adjacent road segments, and the physical road network has a variety of shapes.

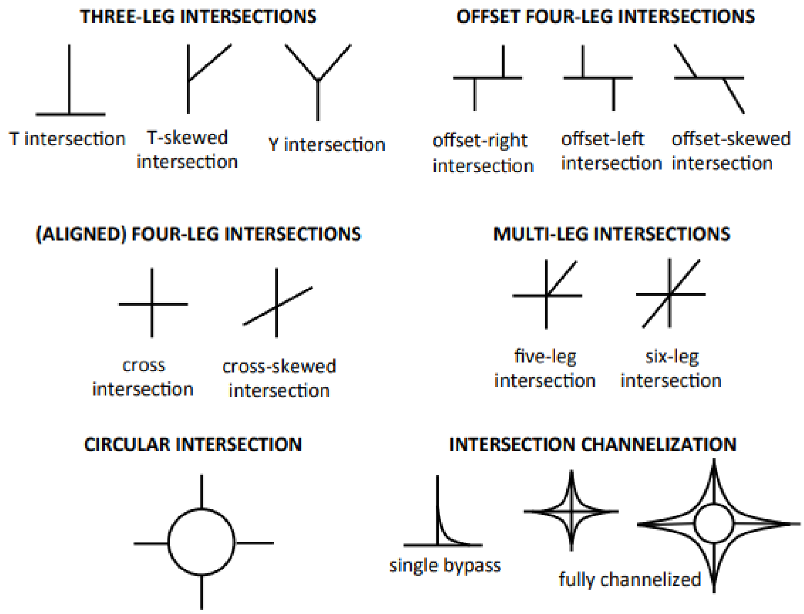


Figure 2.5: Basic intersection forms [4]

Figure 2.5 shows the basic intersection configuration in the physical road network. The roundabout in the lower left is identified as a roundabout. The channelized intersection in the lower right is omitted in the model.

2.2.3 Parking Lots(street parking)

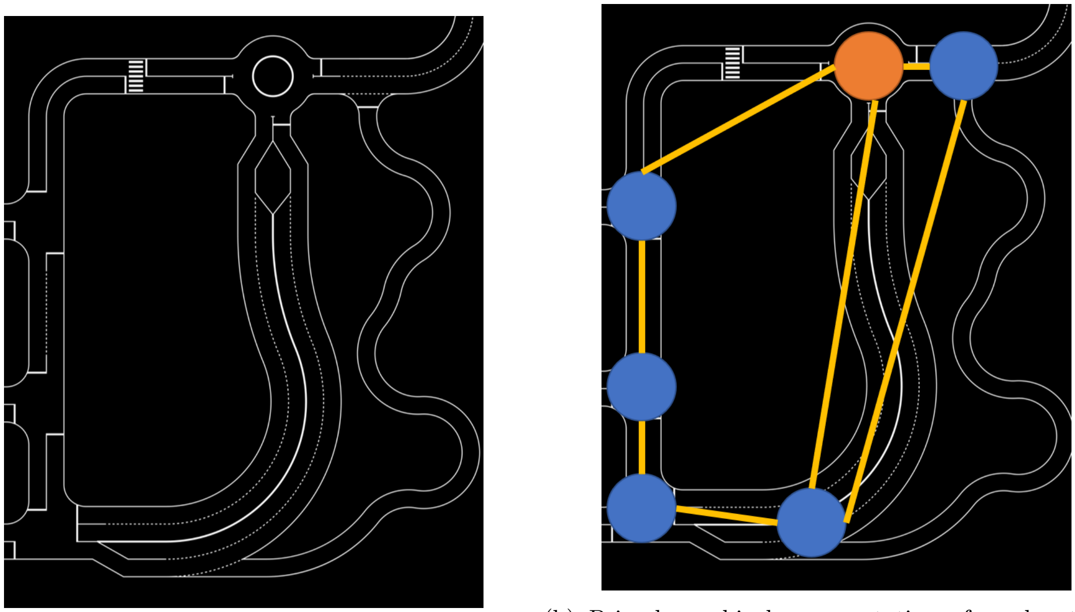
A parking lot is an area where vehicles can be parked. Since it is common in the Netherlands to have areas for parking vehicles on the side of roads, the model considers them as a subclass of road segments, and when defining road segments, the presence or absence of parking lot is also defined together.

2.2.4 Roundabout

A runabout is an element that has the same role as an intersection, connecting road segments with a ring road as shown in the lower left in Figure 2.5. The model assumption associated with them is that they have a constant radius. The value of this radius is taken to be the average value of runabouts existing in the Netherlands.

2.3 Road Network Connectivity

In order to graphically represent the road network, it is important to define the connectivity of nodes within and between elements, how nodes are connected, and how to detect pairs of nodes that need to be connected. This allows the relationships between the elements of the road network to be known and later used for search-based path planning. There are two levels of connectivity: the macro level shown in Figure 2.6b, which indicates the connectivity of road segments and intersections, roundabouts and other elements, and the lane-level shown in Figure 2.7, which indicates the connectivity of lanes, including virtual lanes. Namely, the macro road network is abstracted version of the lane-level road network. This section describes the procedure for creating a lane-level road network based on the given element information.
Connectivity can generally be defined by connecting a start point and end point. In lane-level, it can be defined by connecting the starting point of lanes in a class type to the ending point or vice versa. For example, the endpoint in the intersection is always connected to the start point of a lane of a road segment, and the start point of the lane is always connected to the endpoint in the intersection. There are also situations that lanes in a road segment join and split. The points where lanes merge or diverge shall also be the start and end points.



(a) BFMC track

(b) Primal graphical representation of road network (blue: intersections, orange: roundabout)

Figure 2.6: BFMC track in road network

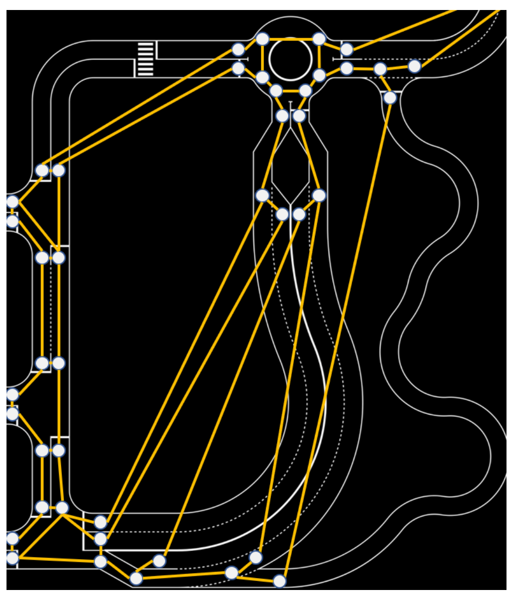


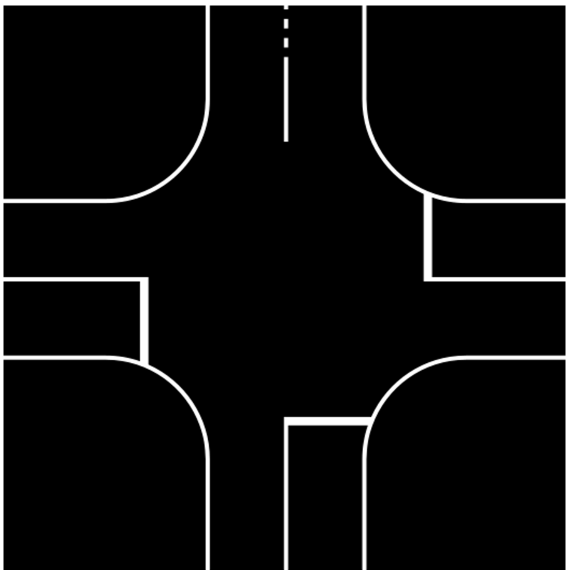
Figure 2.7: Dual graphical representation of road network

2.3.1 Event Point

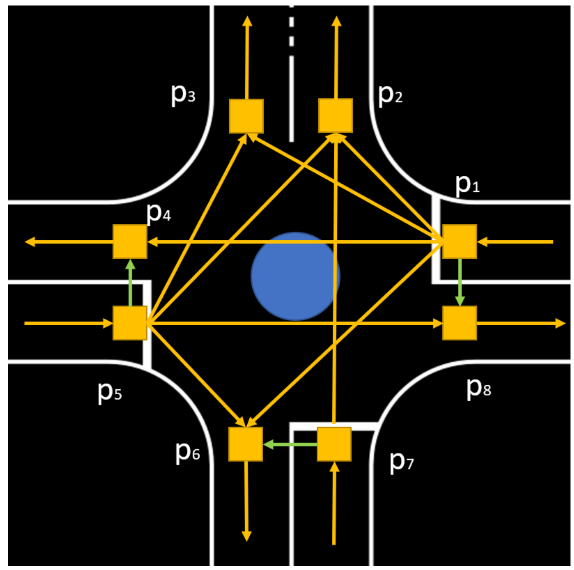
To clarify the definition of nodes in graphical representation of road network, the term ”event point” is used in this project. An event point is defined as a point on an element of the road network where its attributes change [17] and a vehicle must pass through. This makes it possible to represent complex situations such as roundabouts, or to add event points at locations where lane numbers change in the graphical representation to represent lane numbers. In detail, event points are placed in the following location;

- Start/end points of lanes
- Points where lane number changes
- Start/end points to access a parking lot
- Start/end points of parking lots
- Merging/diverging points in roundabouts

Based on the above definition of event points, event points in road network elements are expressed as follows;



(a) Standard intersection



(b) Expressed with nodes and edges

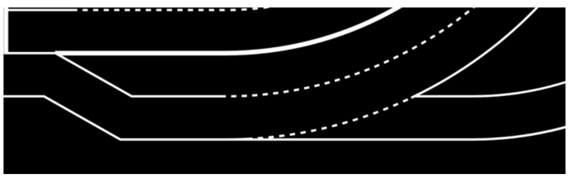
Figure 2.8: Example of standard intersection

Figure 2.8b shows the graphical representation of the standard 4-leg intersection. Event points are placed at the beginning and end of the road and they are connected by edges representing virtual lanes. The green lines connecting the event points next to each other represents the possibility and virtual lane of U-turn.The blue circle represents the center of the intersection, is not connected to any edge, and is placed only in this image to visualize the location of the intersection.

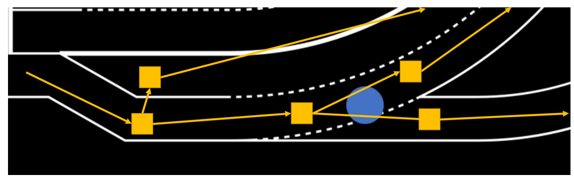
Traffic laws (lane connectivity) at intersections, such as whether U-turns, left turns and straight ahead are allowed, need to be defined. This can be done using a concept introduced as 'Traffic matrix' [14]. Table 2.2 shows the example of the traffic matrix corresponding to the intersection with the connections visualized in Figure 2.8b. Rows are start points and columns are end points. Numbers in the matrix, 1 indicates connectable, 0 indicates not connectable.

Table 2.2: Example traffic matrix

start/ end	p_1	p_2	p_3	p_4	p_5	p_6	p_7	p_8
p_1	0	1	1	1	0	1	0	1
p_2	0	0	0	0	0	0	0	0
p_3	0	0	0	0	0	0	0	0
p_4	0	0	0	0	0	0	0	0
p_5	0	1	1	1	0	1	0	1
p_6	0	0	0	0	0	0	0	0
p_7	0	1	1	1	0	1	0	1
p_8	0	0	0	0	0	0	0	0



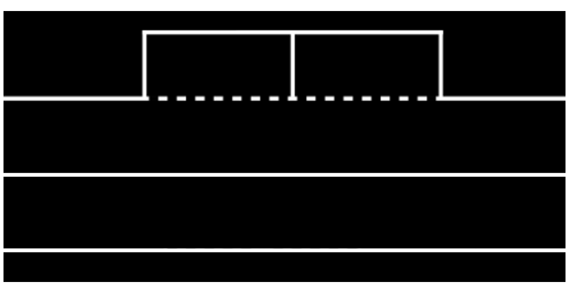
(a) Road segment



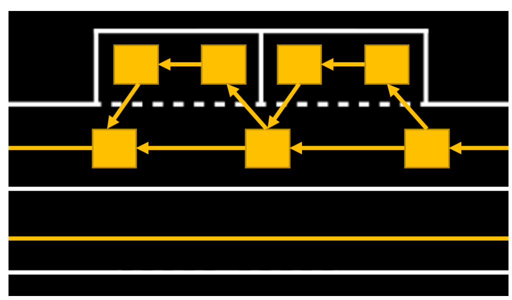
(b) Expressed with nodes and edges

Figure 2.9: Example of change in the number of lanes in the road segment and Y-shape intersection

Figure 2.9b shows an example of the graphical representation when the number of lanes in the road segment increases. Event points are located where lane number increases, at the beginning of a lane.



(a) Parking lot



(b) Expressed with nodes and edges

Figure 2.10: Example of a parking lot event points

An example of the placement of event points in a parking lot is shown in Figure 2.10b. Event points have been added along the lane for entering and exiting the parking lot. The location of the event point for entering must be set in advance by the user. Within the parking lot, there are two event points that indicate the start and end of the parking lot. The location of the event points within the parking lot is determined based on the length of the parking lot. In this model, the length of the parking lot is $6m$ and the event points are located at $2m$ and $4m$ along the parking lot.

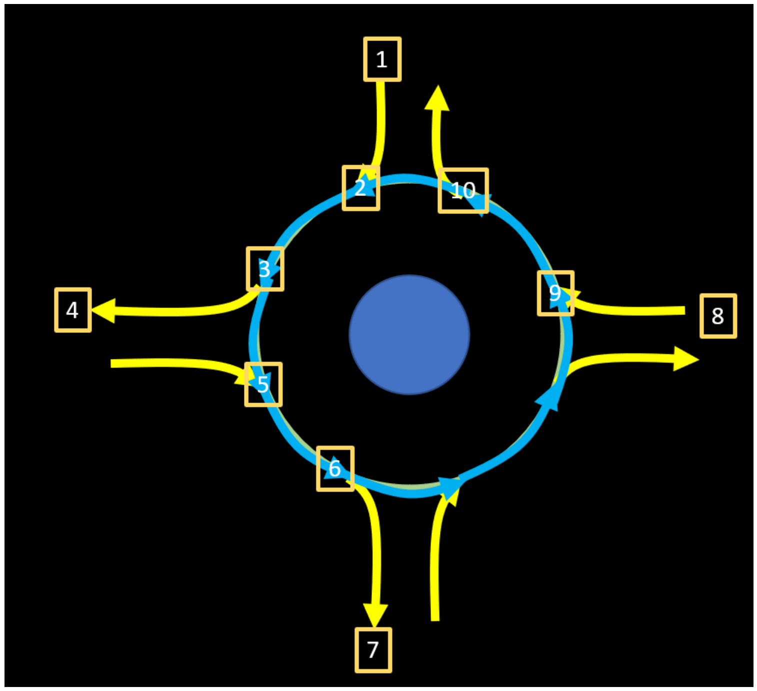
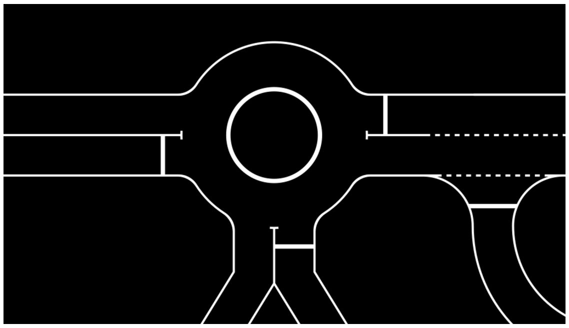
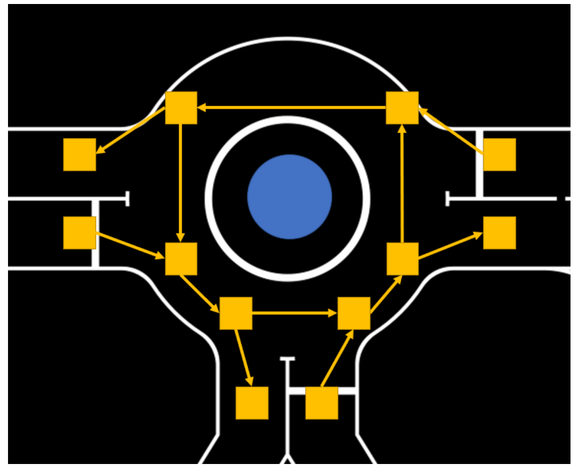


Figure 2.11: Potential paths; paths to enter/exit(yellow) and paths within the roundabout(cyan)

For roundabouts, there is a high degree of freedom in choosing where to place event points, and thus several methods are available. In this project, in addition to the starting and ending points of the lanes, n event points are used to represent the number of lanes connected to the roundabout, where n is the number of lanes. Figure 2.11 shows the potential paths that can be taken by a vehicle at the roundabout. Paths taken to enter/exit are indicated in yellow and the paths taken within the roundabout is shown in cyan. The reason for separating these two paths is that the paths within the roundabout are shared, and vehicles follow the same path no matter where they enter or exit the roundabout. For example, if a driver wants to go from node 1 to 7 in Figure 2.11, the shortest path is 1-2-3-5-6-7. If another driver wants to go from node 8 to 7, the path is 8-9-10-2-3-5-6-7. In this case, 2-3-5-6 identical paths would be followed within the roundabout. Therefore, dividing the roundabout with n (the number of lanes connected to the roundabout) event points makes constructing the path easier, since it can be done by a simple combination of corresponding paths to the edges passed. For example, if there are 6 lanes, the roundabout is represented by a graph like Figure 2.12b with 6 additional event points.



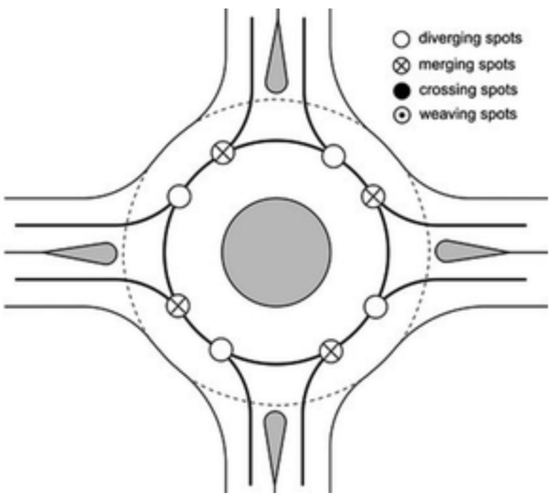
(a) Standard roundabout



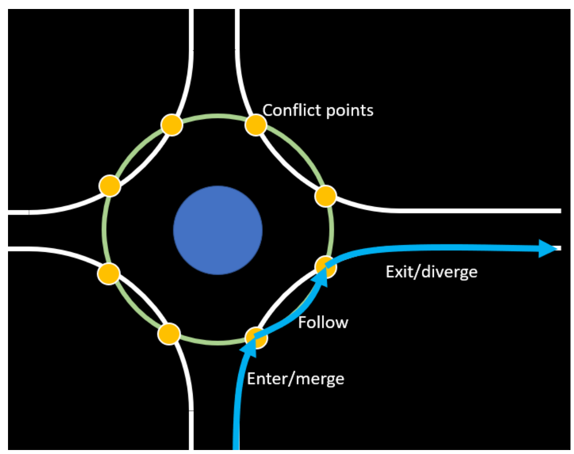
(b) graphical representation of the roundabout

Figure 2.12: Example of roundabouts

2.3.2 Event Points Detection in Roundabouts



(a) Conflict points [13]



(b) How conflict points are defined and example path(light blue)

Figure 2.13: Roundabout with conflict points

A conflict point is defined as the point where two different vehicle paths intersect, and the number of conflict points affects the potential risk of an accident [13]. The point where two virtual lanes begin to diverge is defined as the "diverging point," and the point where two virtual lanes merge is defined as the "merging point". Figure 2.13a shows the conflict points at standard roundabouts. In this case, the point where the virtual lane of the roundabout intersects the virtual lane between the two road segments is indicated as the conflict point. This concept, which defines the diverging and merging points of two virtual lanes at a roundabout, is useful in graph representation to split paths at the roundabout in an efficient way and to facilitate path construction from the graph and to know which points on the roundabout to pass through in path planning as shown in Figure 2.13b. Therefore, this idea of using conflict points (merging and diverging point) is incorporated in the graphical representation as event points in roundabouts.

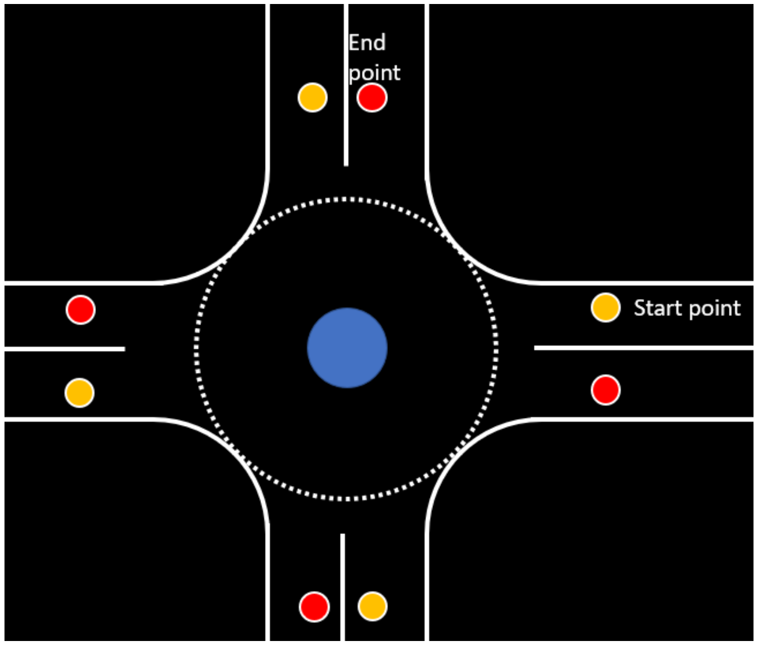
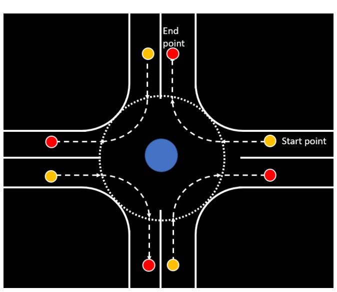


Figure 2.14: 4-leg roundabout

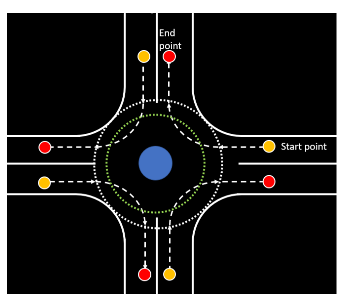
Figure 2.14 shows the regular 4-leg roundabout and this is used to demonstrate the proposed method to identify the event points. The detailed explanations of each step is as follows.

1. The roundabout is first considered the same as a normal intersection, and a virtual lane to take the nearest exit is drawn with a quadratic Bézier curve with a lane or road starting/ending point and the center of the intersection, as shown in Figure 6.8a.
2. A circular virtual lane with a radius corresponding to the roundabout is drawn with a Bézier curve as well (Figure 6.5b)
3. Finally, define the intersection points of the virtual lanes, as well as the start and end points as event points (Figure 2.15c)

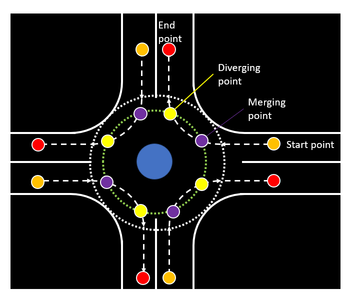
Quadratic Bézier curves are used because of their ease of use and accuracy. Details are explained in chapter 3.



(a) Step 1: Draw virtual lines connecting start/end points of lanes



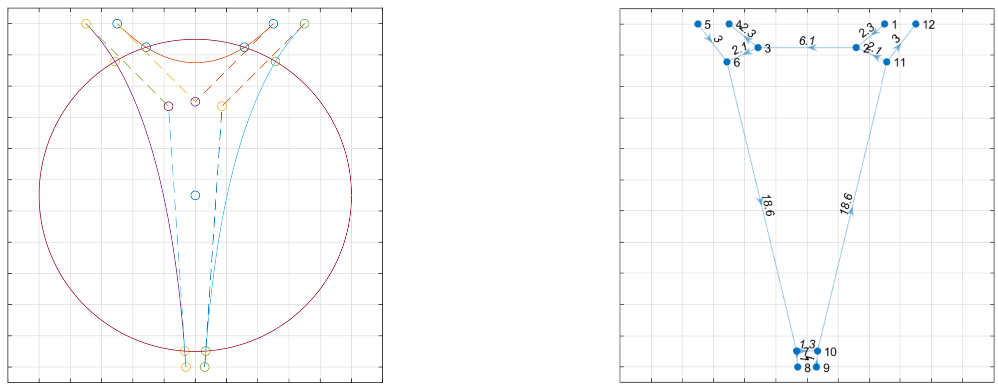
(b) Step 2: Draw virtual line around the roundabout(light green)



(c) Step 3: Find 2 intersection points of the two virtual lanes

Figure 2.15: Identifying event points at roundabout

The same procedure is used for path construction, and the mathematical derivation and other details are explained in chapter 3. Once all event points on the circular virtual lane are found, the angles of the event points relative to the center of the roundabout are calculated and sorted in ascending order. Connectivity is defined by connecting the event points in that order. The used function and control points, and the resulting graphical representation of an example roundabout are as shown in Figure 2.16.



(a) Virtual lanes to find the event points

(b) Graphical representation

Figure 2.16: 8 entry/exit lanes (4-leg) roundabout

The event point detection method proposed above relies on the intersection of a virtual lane from the start point to the end point and a circular virtual lane, which depends on the size of the circular radius and the location of the entry and exit lines where the virtual lane is drawn. The drawback is that if the radius of the roundabout is smaller than a certain value, the two virtual lanes may not intersect and the event point may not be identified. In this case, the proposed method is invalid. We conclude that this drawback is not much of a problem. More details and investigations are explained in Appendix A.

2.4 Road Network Visualization

The objective is to generate a graphical representation of the road network for path planning based on the definition of connectivity explained in the previous section and a realistic visualization. Realistic visualization takes into account the number of lanes, boundary lines, width, parking lots, center lines, etc.. Graphical representation of road network expresses the given road network with nodes (event points) and their edges (connectivity). Using a MATLAB function `digraph()`, graphical representations can be easily created by defining edges using a traffic matrix for intersections, merging and diverging points for roundabouts, and event points for accessing parking lots and start and end points for road segments introduced in the previous section.

2.4.1 Road Network Information

In order to generate a graphical representation of a road network, information about the road network is first required in advance. The road network elements and the necessary information about them are defined as follows;

- Road segment:

$$R(n_l, w_l, p_l, pa)$$

  - n_l: Number of lanes
  - w_l : Lane width
  - p_l: Road segment position data
  - pa: Parking lot information

where number of lanes n_l is 1×2 vector that has a first component indicating the number of lanes in one direction and a second component indicating the number of lanes in another direction. The road segment position data p_l is a set of x and y representing location of the road segment. The other parameters are scalars.

- Parking lot (Sub-class of a road segment):

$$pa(n_{pa}, l_{pa}, w_{pa}, s_{pa})$$

  - n_{pa}: Number of a parking lot
  - l_{pa}: Length of a parking lot
  - w_{pa}: Width of a parking lot
  - s_{pa}: Start point of a parking lot with respect to the associated road segment length

where $0 \leq s_{pa} < 1$, the location where the parking lot begins, expressed as a ratio of the length of the road to that location and the total length of the road as shown in Figure 2.17. All parameters are scalars.

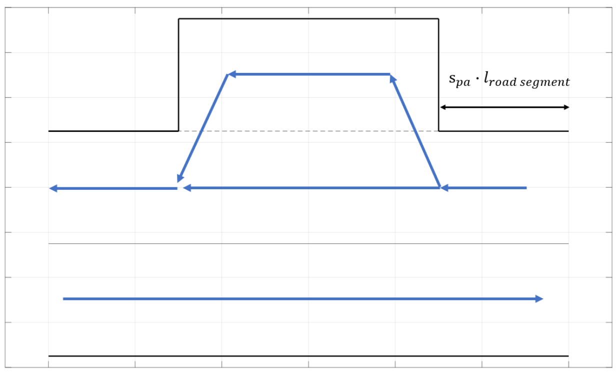


Figure 2.17: Visualization of how s_{pa} is used

- Intersection:

$$C(p_c, T, I_r)$$

  - p_c: Position of its center
  - T: Traffic matrix
  - I_r: Set of road segment IDs to be connected

- Roundabout:

$$D(r, w, p_c, I_r)$$

  - r: Radius
  - w_r: Circular virtual lane width
  - p_c: Position of its center
  - I_r: Set of road segment IDs to be connected

The main objective is to generate a graphical representation and realistic visualization of a simple, basic road network. Several assumptions are made to simplify the complex physical road network as listed below;

- All lanes have the same width of $w = 3.5m$ [7]
- The inscribed diameter of roundabout is [12];

$$D_{roundabout} = \begin{cases} 40m(\text{virtual lane with 18.25m radius}) & \text{Single-Lane} \\ 60m(\text{virtual lanes with 28.25m and 24.75m radius}) & \text{Double-Lane} \end{cases} \tag{2.1}$$

- Road segments connected to a roundabout are multi-way
- Each direction of the road segment has one lane
- Lanes in both directions have the same distance from the center line of the road segment
- Road segments are straight

The first assumption is to make the lane widths independent and reduce complexity. The second assumption is also to reduce the complexity when investigating the limitation of the event point detection method, and double-lane roundabout is omitted in the model. The third assumption is added in order to ensure that event points around a roundabout will always be in the same repetitive pattern(merging-diverging-....). Since road segments connected to roundabouts are often multi-directional, this assumption often holds true, but exceptions are expected to exist. Lastly, since lane approximations are computationally expensive, road segments are assumed to be straight.

2.4.2 Boundary Lines in Road Segments

Road segments include lanes, center lines, boundary lines, and stop lines. Boundary lines are the side edges of the road where vehicles cannot cross. All the stated lines are calculated using simple trigonometry. In the part of the road segment where there is a parking lot, the boundary line shifts by the width of the parking lot. Equation 2.2 shows how boundary lines are calculated;

$$\text{boundary}_1(t) = \begin{cases} \left[x_{\text{road}}(t) - w_l \cdot cos(\pi/2 - \theta(t)) \quad , \quad y_{\text{road}}(t) + w_l \cdot sin(\pi/2 - \theta(t))\right] & l_l(t) < s_{pa} l_{l\text{total}}, l_l(t) \geq n_{pa} l_{pa} + s_{pa} l_{l\text{total}} \\ \left[x_{\text{road}}(t) - (w_{pa} + w_l) \cdot cos(\pi/2 - \theta(t)) \quad , \quad y_{\text{road}}(t) + (w_{pa} + w_l) \cdot sin(\pi/2 - \theta(t))\right] & s_{pa} l_{l\text{total}} \leq l_l(t) < n_{pa} l_{pa} + s_{pa} l_{l\text{total}} \end{cases}$$
$$\text{boundary}_2(t) = \begin{cases} \left[x_{\text{road}}(t) + w_l \cdot cos(\pi/2 - \theta(t)) \quad , \quad y_{\text{road}}(t) - w_l \cdot sin(\pi/2 - \theta(t))\right] & l_l(t) < s_{pa} l_{l\text{total}}, l_l(t) \geq n_{pa} l_{pa} + s_{pa} l_{l\text{total}} \\ \left[x_{\text{road}}(t) + (w_{pa} + w_l) \cdot cos(\pi/2 - \theta(t)) \quad , \quad y_{\text{road}}(t) - (w_{pa} + w_l) \cdot sin(\pi/2 - \theta(t))\right] & s_{pa} l_{l\text{total}} \leq l_l(t) < n_{pa} l_{pa} + s_{pa} l_{l\text{total}} \end{cases} \tag{2.2}$$

where $l_l(t)$ denotes the length from the start point of the lane to a certain point along the lane.

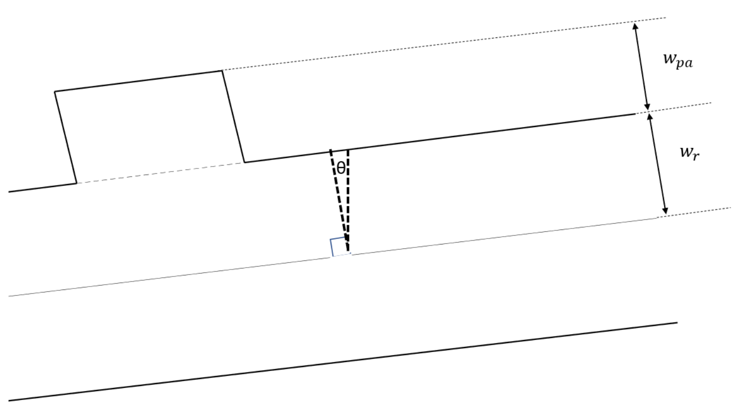


Figure 2.18: Road segment

2.4.3 Boundary lines at intersections

Road segment boundary lines have the same start and end points as the lanes within the road segment. Therefore, within intersections, the boundary lines of the associated road segments need to be connected in order to fully represent the physical road network.Second- or third-order Bezier curves are used to connect boundary lines. Although second-order Bezier curves are mainly used, there are cases where second-order Bezier curves are not suitable, such as when the curves to be drawn are unrealistic. For example, the following cases are possible, in which case third-order Bezier curves are used.

- The intersection point of the boundary lines to be connected is located outside the intersection area

- The gradients of the boundary lines to be connected are equal

When using a quadratic Bezier curve, the intersection of the line segment drawn from the start point of the boundary line considering the direction drawn and the line segment drawn from the end point is used as the control point. When using a cubic Bezier curve, the two intersection points of the line segment drawn considering the start (end) point and direction of the boundary line and the line segment with a slope perpendicular to the average slope of the two boundary lines are defined as the intermediate control points. The same procedure is used for path construction, and the mathematical derivation and other details are explained in chapter 3.

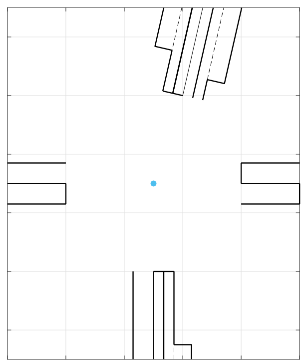
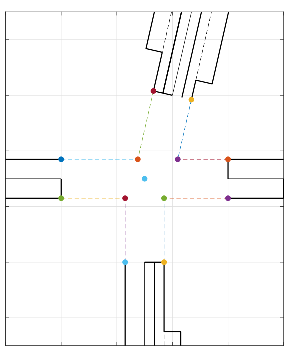
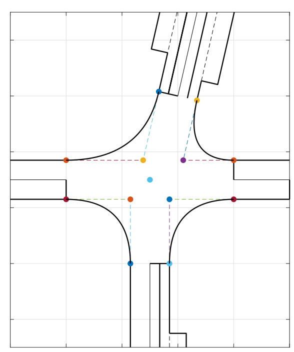


Figure 2.19: Intersection without connecting the boundary lines



(a) Step 1: Find intersection points of the boundary lines



(b) Step 2: Connect the boundary lines using found control points

Figure 2.20: Boundary line connection in an intersection

2.4.4 Boundary lines at roundabouts

A roundabout has an inner and outer boundary, the inner boundary line is inner perimeter of the roundabout. The outer boundary is a outer parameter, each radius determined by the user-defined runabout radius value and the width of the lane. The inner boundary can be drawn in all ranges from 0 to 2π. However, the outer boundary must be open within a certain angular range to allow connection to the road segment entering the roundabout.

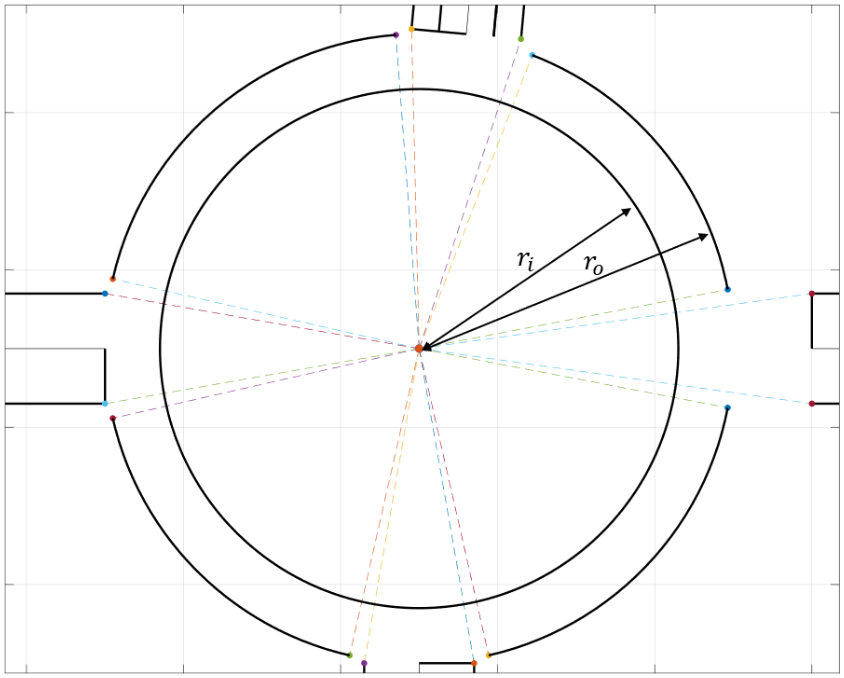
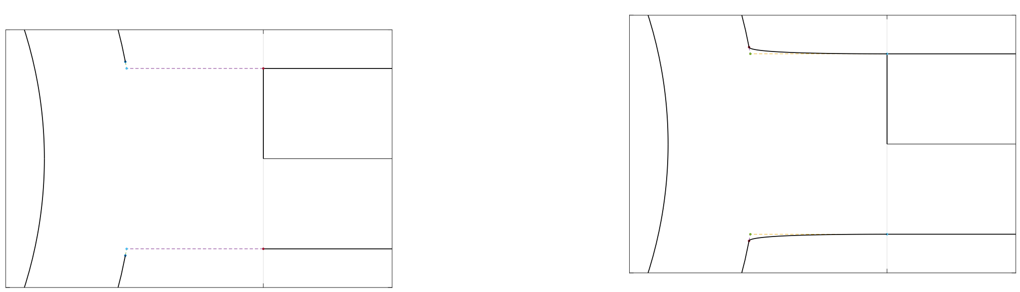


Figure 2.21: Roundabout without connecting the boundary lines

Figure 2.21 shows realistic visualization of a roundabout without connecting the boundary lines. The outer boundary line is drawn only within a certain angular range, which is determined based on the angles to the center of the roundabout of the start and end points of the boundary lines of the road segments entering the roundabout, and 0.05 radians are added or subtracted from that angle so that a Bezier curve is properly drawn.

$$\begin{aligned} \text{boundary}_\text{i}(\theta_b) =& r_\text{i}\left[cos(\theta), \quad sin(\theta)\right] \qquad 0 < \theta \leq 2\pi \\ \text{boundary}_\text{o}(\theta) =& r_\text{o}\left[cos(\theta), \quad sin(\theta)\right] \qquad \theta_b[i] + 0.05 < \theta \leq \theta_b[i+1] - 0.05, \qquad i = 0, 2, 4, ... \end{aligned} \tag{2.3}$$

Same procedure as for the intersections is taken to connect the boundary lines as shown in Figure 2.22.



(a) Step 1: Find intersection points of the boundary lines

(b) Step 2: Connect the boundary lines using found control points

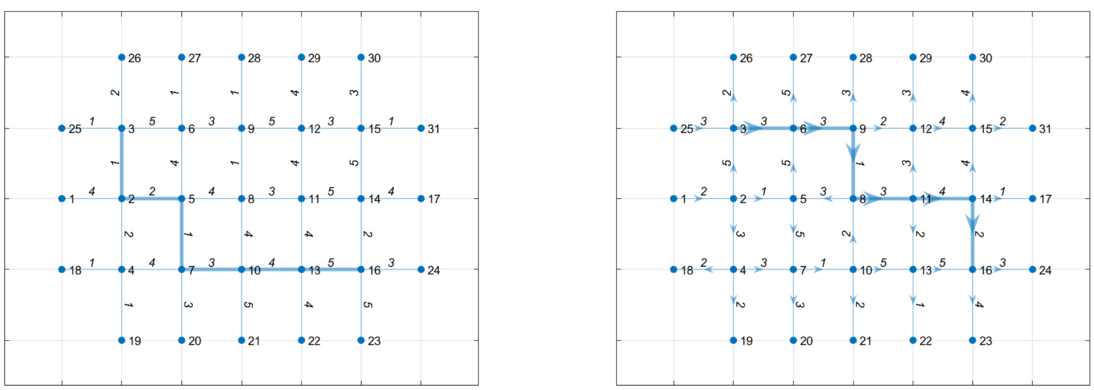
Figure 2.22: Boundary line connection in a roundabout

Chapter 3

Path Planning

3.1 Search-based Path Planning over Road Networks

Dijkstra algorithm is a common method used to find the shortest path to a certain destination [10]. Matlab function `shortestpath()` can be used in the model to find it in the graphical representation of the road network.



(a) Undirected roads (b) One-way roads

Figure 3.1: Result of `shortestpath()`

Figure 3.1 shows the result of running `shortestpath()` function. Depending on the cost(travel time and distance), the shortest path can be chosen. The direction/s of the road segments can also be implemented by using `digraph()` function as shown in Figure 3.1b. In this way, vehicles entering the road from the wrong direction can be easily constrained. However, such road network graphs generally have the disadvantage that start and end points can only be placed at nodes. Therefore, when the `shortestpath()` function is executed in the model, the start and end points are selected by clicking on the realistic visualization created, and the nodes closest to the clicked position become the actual start and end points.

Travel cost, defined as the physical length of an edge, is important for finding the shortest path using a graphical representation of the road network. This is implemented as 'weights' in the model. A mathematical derivation of the physical length is as follows;

$$
\begin{aligned}
\text{norm}_p(t) &= \sqrt{\frac{dx}{dt}^2 + \frac{dy}{dt}^2} \\
\text{l}_l = \int_s^e \text{norm}_p(t)dt &= \int_s^e \sqrt{\frac{dx}{dt}^2 + \frac{dy}{dt}^2}\, dt
\end{aligned}
\tag{3.1}
$$

where, s and e are the limits of integration, which depend on the function used for path construction. When Bezier curve is used, the limit is $[0, 1]$, and when circle function (runabout circular virtual lanes) is used, the limit is the range of angles to be drawn.

3.2 Path Conversion from Road Network Graphs to Physical Road Networks

A path in a road network graph only contains information about which edges and nodes a vehicle needs to pass through to get to a certain destination. However, it does not contain information about how a vehicle should behave within that edge (lane) or node (intersection, roundabout, etc.). Therefore, it is necessary to convert the path in the road network graph into physical path to obtain the location information where the vehicle actually passes. There are several existing road modelling techniques to express the lanes in the road segments and intersections.

- Polyline
  - Combination of linear segments
- Clothoid
  - Curves whose curvature varies linearly with curve length [2]

$$
\begin{aligned}
x(l) &= x_0 + \int (cos(\tau + \kappa_0 l + \frac{cl^2}{2}))dl, \quad 0 < l < L \\
y(l) &= y_0 + \int (sin(\tau + \kappa_0 l + \frac{cl^2}{2}))dl, \quad 0 < l < L
\end{aligned}
\tag{3.2}
$$

  where

  - x_0, y_0 :initial coordinates of the curve;
  - τ_0 : initial bearing of the curve;
  - κ_0: initial curvature;
  - c: curvature rate with the curvilinear abscissa (dκ/dl), with $\kappa = \kappa_0 + cl$;
  - l: curvilinear abscissa;
  - L: total length of the curvature;
- Cubic Hermite Spline
  - CHS is C1 and C2 continuous curves made of cubic polynomial that connects two control points [14]

$$
\begin{aligned}
f_{CHS}(x) &= \begin{bmatrix} 2x^3 - 3x^2 + 1 \\ 2x^3 + 3x^2 \\ x^3 - 2x^2 + x \\ x^3 - x^2 \end{bmatrix} \begin{bmatrix} p_i \\ p_{i+1} \\ v_i \\ v_{i+1} \end{bmatrix} \\
x &= \frac{u - u_i}{u_{i+1} - u_i} \quad u \in [u_i, u_{i+1}]
\end{aligned}
\tag{3.3}
$$

  where

– p_i, p_{i+1}: coordinate of control points

– v_i, v_{i+1}: tangent vectors of control points

– u: knot of control points

- B-spline

  – A B-spline is a spline composed of control points, node vectors, and primary functions. The shape of a B-spline curve can be changed by modifying one or more of these control parameters. [16]

$$B(t) = \sum_{j=0}^{n} p_j N_{j,k}(t) \quad (t_{k-1} < t < t_{n+1})$$
$$N_{j,k}(t) = \frac{(t-t_j)N_{j,k-1}(t)}{t_{j+k}-t_j} + \frac{(t_{j+k+1}-t)N_{j+1,k-1}(t)}{t_{j+k+1}-t_{j+1}} \quad (t_{k-1} < t < t_{n+1}) \tag{3.4}$$
$$N_{j,0}(t) = \begin{cases} 1 & (\text{if} \quad t_j < t < t_{j+1}) \\ 0 & (\text{others}) \end{cases}$$

where

– p_j: coordinate of control points

– $N_{j,k}$: Primary function

– t: Node vectors(polynomial with highest degree k)

- Bézier curve

  – a Bézier curve is contained in the convex hull of its control points (i.e., parameters), and it smoothly interpolates between the first and last control point. [1]

second-order Bezier curve

$$\begin{bmatrix} x_b & y_b \end{bmatrix} = (1-t)^2 \cdot p_1 + 2(1-t)t \cdot p_c + t^2 \cdot p_2 \quad 0 \le t \le 1 \tag{3.5}$$

third-order Bezier curve

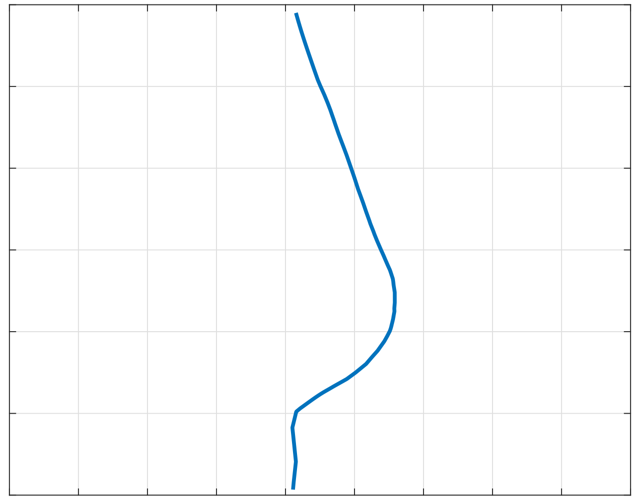
$$\begin{bmatrix} x_b & y_b \end{bmatrix} = (1-t)^3 \cdot p_1 + 3(1-t)^2 t \cdot p_{c1} + 3(1-t)^2 t \cdot p_{c2} + t^3 \cdot p_2 \quad 0 \le t \le 1 \tag{3.6}$$

where

– $p_1, p_c, p_{c1}, p_{c2}, p_2$: control points

Each has its advantages and disadvantages in terms of computational cost, accuracy, and storage efficiency. Storage efficiency is defined as the number of functions and control points required to represent a road of a given shape. In addition, in road modeling techniques, it is very important to choose a universal method that can represent any type or shape of road segment or intersection. Global continuity is also important to prevent jerky vehicle movements and ensure comfort. Therefore, a road modeling method that is computationally inexpensive, accurate, storage efficient, and universally applicable to all types of roads is optimal. Polylines do not preserve global continuity, so the polyline method is not going to be evaluated.
Since accuracy is an essential element in road modeling, we calculated the number of control points required to achieve a certain level of accuracy for each road modeling method and the number of functions to represent the shape of road segments, and compared the storage efficiency and searched for the optimal method.
To test the road modeling techniques, two real setups (intersection and lane) extracted from OpenStreetMap were used; the OpenStreetMap data (OSM files) were imported into MATLAB and used as a row data for road modeling.

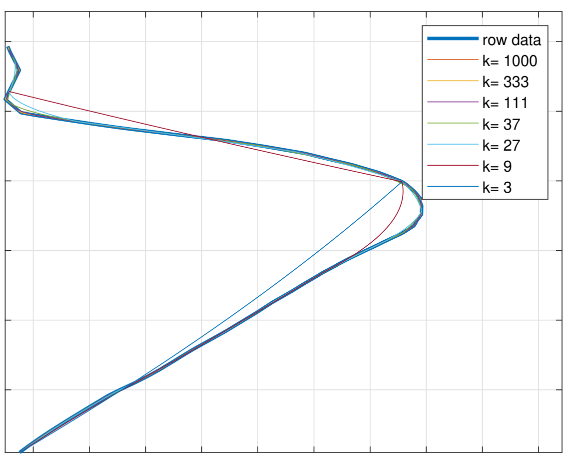
(a) in OpenStreetMap



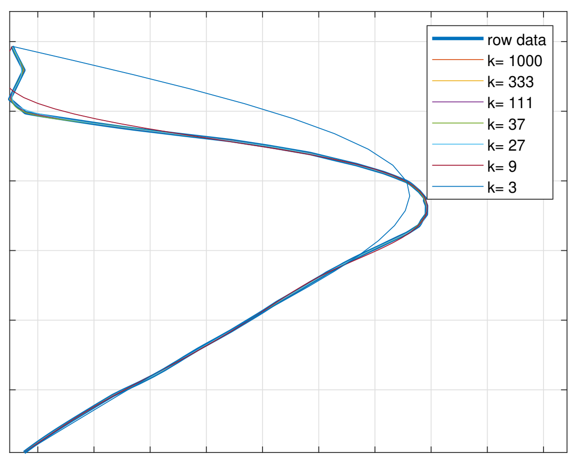
(b) Imported in Matlab

Figure 3.2: A road 'Vestdijk' in Eindhoven

This road was chosen because it contains both small and large changes in angle and is suitable for looking at the different behaviors of road modelling techniques and how they react to them.The row data has the coordinate of the road in GPS data(log, lat), so it is converted to meters. The number of the data points of the row data is $n = 1000$ and the curves will be drawn using data points $k = [3, 9, 27, 37, 111, 333]$ as control points. The error between curve and row data at each data point for a given number of control points is calculated and used to compare the performance of road modeling techniques.

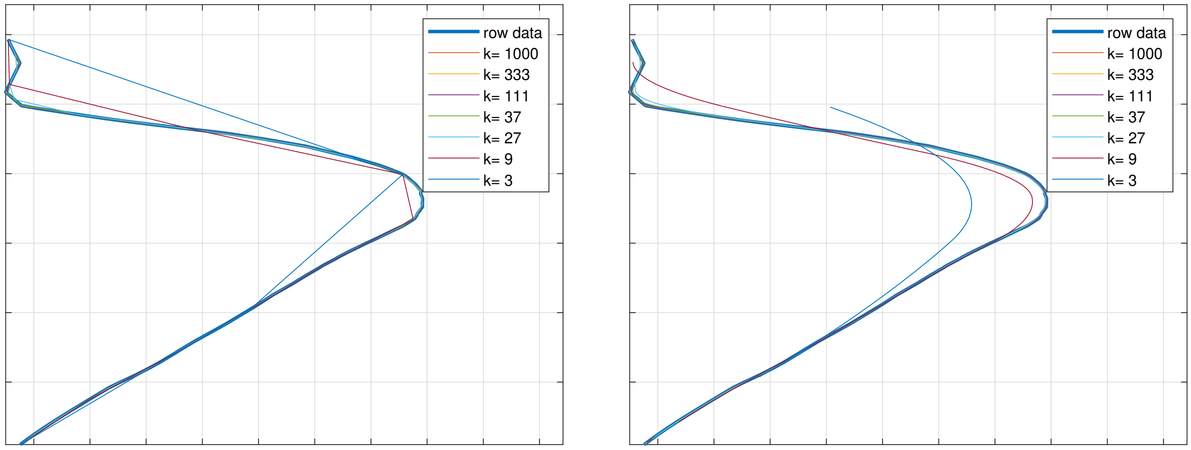


(a) Second order



(b) Third order

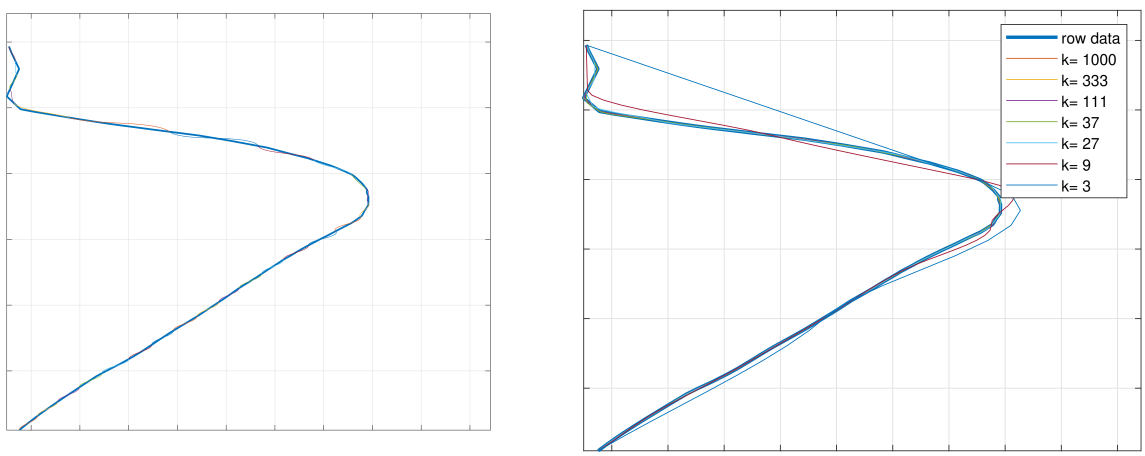
Figure 3.3: Bezier curves with different number of control points



(a) 2nd order

(b) Third order

Figure 3.4: B-spline curves with different number of control points

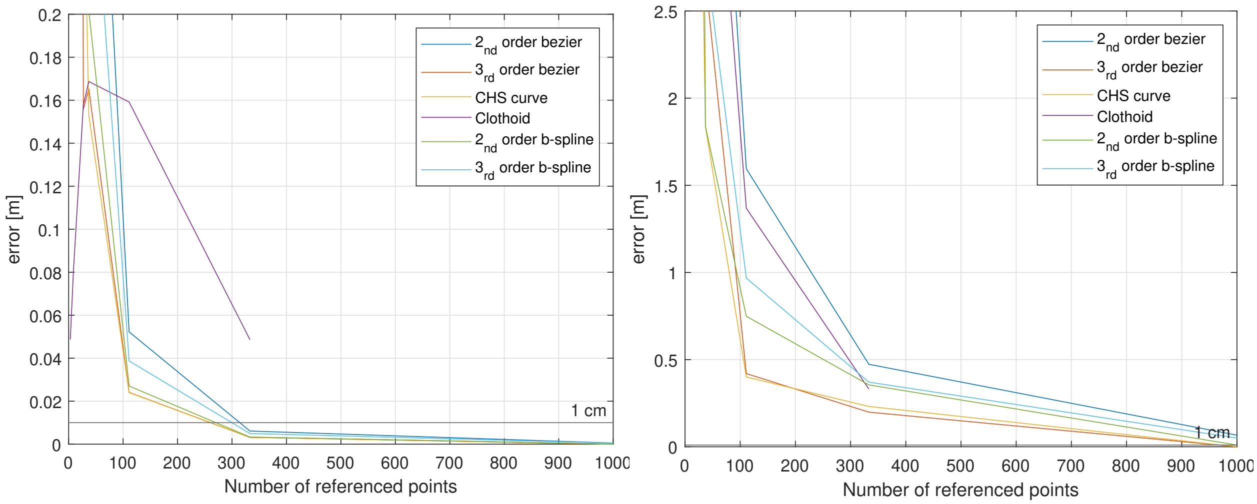


(a) Clothoid curves with number of control points $k = 27$

(b) CHS curve

Figure 3.5: Other curves

The method of reducing the number of control points described above is to take data at certain intervals. This method results in poor curve response to relatively abrupt changes in road geometry. The solution is to optimize the method in a way that adds more points where this occurs. This is beyond the scope of this project and will not be discussed in detail. Roads modeled with clothoids have been found to exhibit jiggly behavior and are not well suited for modeling such roads. Choosing a different function for different situations will lead to the best road modeling.



(a) Mean error

(b) Maximum error

Figure 3.6: Errors with different road modelling methods

Figure 3.6a shows the mean error and Figure 3.6b shows the max error between the curve and the row data. These two graphs show that while the average error converges to a centimeter level, the maximum error can be in decimeters, indicating that it is important to look at both errors when evaluating road modeling methods. According to Figure 3.6, the third order bezier curves best models the road shape with minimum control points and the CHS curve is the next best choice. The average error for all the techniques except for clothoid curve is found to converge to 1 cm or less at about 320 control points out of 1000 points. The clothoid curves shows a strange behavior with no clear trend visible, which may be a problem with how it is implemented. The problem may be that there is noise in the row data and the scripts or the curves are not built to be robust against noise. For modeling road network, cubic Bezier curves were chosen for ease of use and accuracy, and quadratic Bezier curves when accuracy is not critical.

3.2.1 Path construction

In order to optimize path construction, a distinction is made between the use of quadratic curves, cubic Bezier curves, and other functions for each road network element. This section describes what functions are used for each road network element.

At the intersection, virtual lanes need to be constructed using Bezier curves and the boundary lines of road segments entering/exiting an intersection need to be connected. As for the virtual lanes construction, both second and third order Bezier curve are used depending on cases. For example, left-turn and right-turn lanes are constructed with second order Bezier curves, while straight ahead and U-turn lanes are constructed with third order Bezier curves. This is because in the case of straight ahead and U-turns, the entry and exit lanes often have approximately the same slope, and the intersections of the lanes are likely to be far from the intersection or never intersect, which will result in improperly drawn quadratic Bezier curves. Virtual lanes with third-order Bezier curves are constructed in the same way as those for entering and exiting a parking lot.

A virtual lane for turning right or left at an intersection is constructed with a second order Bezier curve when the entry/exit lane is within the intersection area. Three control points used to construct it are the start point, the intersection point of the two lanes and the end point. In case when the gradient of the lanes are almost equal or the intersection points lie outside the intersection area, third order Bezier curve is used. The mathematical expression of second-order ant third-order Bezier curve and the procedure when they are used is as follows;

Second-order Bezier Curve

1. Find the intersection of the line segment drawn from the start point of the boundary line considering the direction drawn and the line segment drawn from the end point

2. Construct 2rd order Bezier curve using the obtained intersection and start and end points as control points

Third-order Bezier Curve

1. Compute mid-point of start/end points
2. Compute the gradient of a line perpendicular to the mean gradient of the two points
3. Calculate the intersection of a perpendicular line drawn from the point and two straight lines.
4. Construct third order Bezier curve using the obtained intersection and points as control points

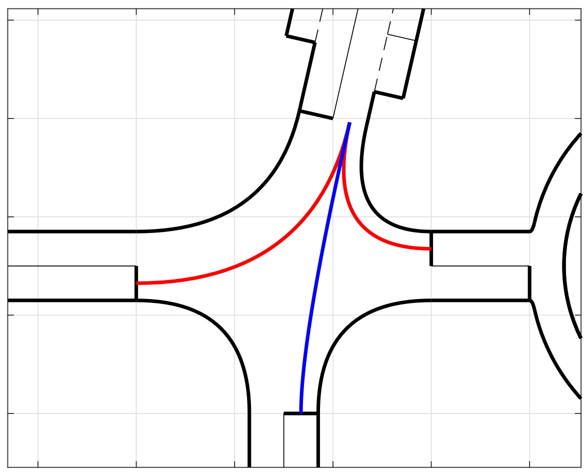


Figure 3.7: Virtual lanes at an intersection: third order Bezier curve(blue), second order Bezier curve(red)

Figure 3.7 shows virtual lanes constructed using Bezier curves based on the conditions described. Red is the virtual lane constructed using a second Bezier curve and blue is the virtual lane constructed using a third Bezier curve, since the intersection of the entry and exit lanes is outside the intersection area.

A U-turn is defined as entering an intersection from one road and exiting from a lane belonging to the same road. Based on one of the assumption, the both entry and exit lanes have exactly the same gradient. Since the intersection of the two lanes does not exist, a third-order Bezier curve is used to construct the virtual lane. The virtual lanes are constructed in the similar way as those for entering and exiting a parking lot. In addition, a coefficient is used to move control points along a gradient of the lanes in order to adjust the path as shown in Figure 3.8.

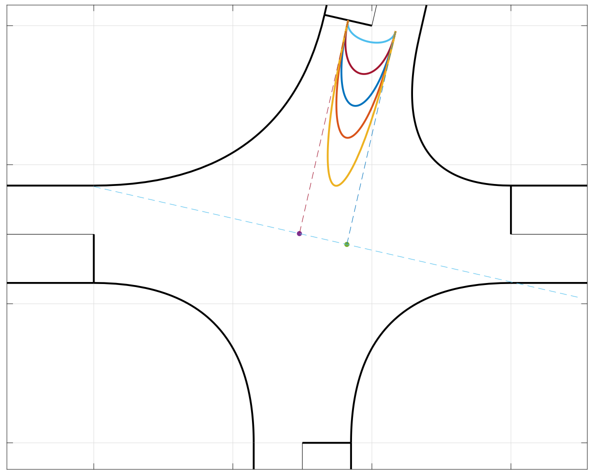


Figure 3.8: Virtual lanes for U-turn with coefficient for shifting the control points; [0.1, 0.3, 0.5, 0.7, 1](from nearest to farthest)

The reason for using the coefficient is that without the coefficient, the path to make a U-turn would be unnecessarily long, requiring drivers to go closer to the center of the intersection to make the U-turn, unnecessarily confusing other drivers. The y-intercept of the line segment orthogonal to the entrance/exit lanes is computed with respect to the intersection center. This line segment is used to find the two control points, i.e., the intersection of the control point and the two lanes. Shifting the control point shortens the path length and prevents a path from getting too close to the center of the intersection.
The coefficient u_{shift} to shift the control points along the gradient of the lane is implemented as follows;

$$\begin{aligned} p_{2shift} &= u_{shift}(p_2 - p_1) + p_1 \\ p_{3shift} &= u_{shift}(p_3 - p_4) + p_4 \end{aligned} \tag{3.7}$$

The coefficient u_{shift} is determined by shifting the control point along the slope of the lane so that the distance between the control point and the start or end point of the lane is $4m$. Its implementation is as follows;

$$\begin{aligned} p_{2shift} =& u_{shift}(p_2 - p_1) + p_1 \\ p_{2shift} - p_1 =& u_{shift}(p_2 - p_1) \\ ||p_{2shift} - p_1|| =& ||u_{shift}(p_2 - p_1)|| = 4 \\ u_{shift} =& \frac{4}{||p_2 - p_1||} \end{aligned} \tag{3.8}$$

Figure 3.9 visualizes how the control points are shifted. All the constructed paths in an intersection can be seen in Figure 3.10.

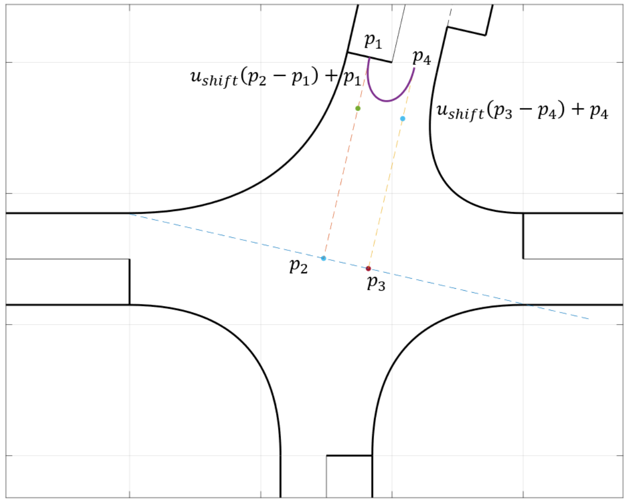


Figure 3.9: A virtual lane for U-turn and its control points shifted with the coefficient u_{shift}

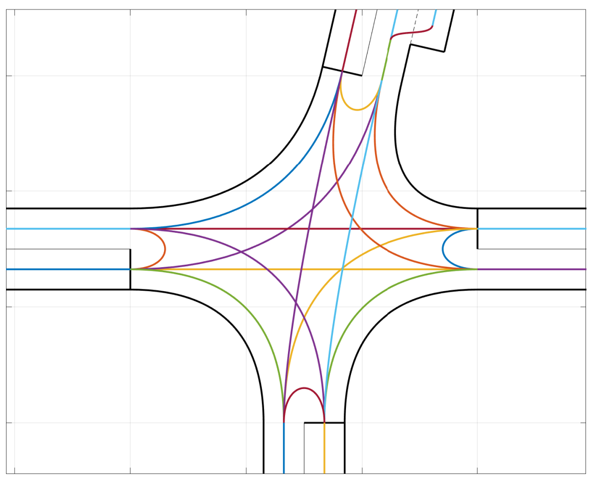


Figure 3.10: Intersection with all the virtual lanes

At a roundabout, event points are first found by applying the previously proposed event point detection method, and virtual lanes connecting them are constructed using a second order Bezier curve and a simple circle function.

Specifically, a second Bezier curve is used to connect the starting point to the merging point found, and the diverging point to the end point. A circle function, i.e., a circular virtual lane centered on the roundabout, is used to connect the merging and diverging points.

The circular virtual lane around the roundabout can be simply defined as;

$$x^2 + y^2 = r^2 \tag{3.9}$$

where r is the radius of the circle drawn by the virtual lane. By substituting Equation 3.5 into Equation 3.9,

t corresponding the merging and diverging point(intersections of the two functions) can be found;

$$x_b^2 + y_b^2 = r^2 \tag{3.10}$$

$$((1-t)^2 \cdot x_1 + 2(1-t)t \cdot x_c + t^2 \cdot x_2)^2 + ((1-t)^2 \cdot y_1 + 2(1-t)t \cdot y_c + t^2 \cdot y_2)^2 = r^2 \tag{3.11}$$

Once a merging or divergence point ($p_{\text{merge}}, p_{\text{diverge}}$) is found, the gradient of the circular virtual lane at that point is calculated, the intersection with the entry/exit lane is obtained, and used as a control point for creating the Bézier curve. Equation 3.12 shows sets of control points used;

$$\begin{aligned} P_{\text{set merge}} &= \begin{bmatrix} p_1, & p_{c1}, & p_{\text{merge}} \end{bmatrix} \\ P_{\text{set diverge}} &= \begin{bmatrix} p_{\text{diverge}}, & p_{c2}, & p_2 \end{bmatrix} \end{aligned} \tag{3.12}$$

Where;

- p_1, p_2; Start/end point respectively
- p_{c1}, p_{c2}; Intersection point of the line segment with the slope at the starting point and the line segment with the slope of the circular virtual lane at $p_{\text{merge}}, p_{\text{diverge}}$.

Figure 3.11 shows all resultant paths constructed based on the above derivation.

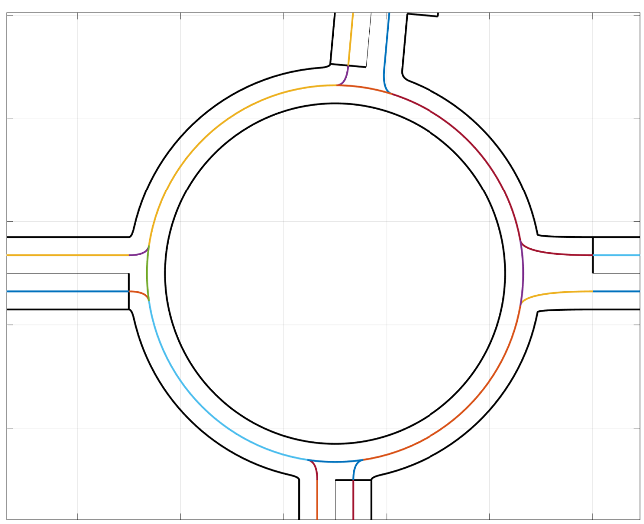


Figure 3.11: Roundabout with virtual lanes

In the Netherlands there is usually a parking lot next to a road segment. Third-order Bezier curves, which are more flexible than second-order Bezier curves, are used to construct virtual lanes to access parking lots. This is because if the intersection of the start and end points cannot be found, a quadratic Bezier curve cannot maintain global continuity.

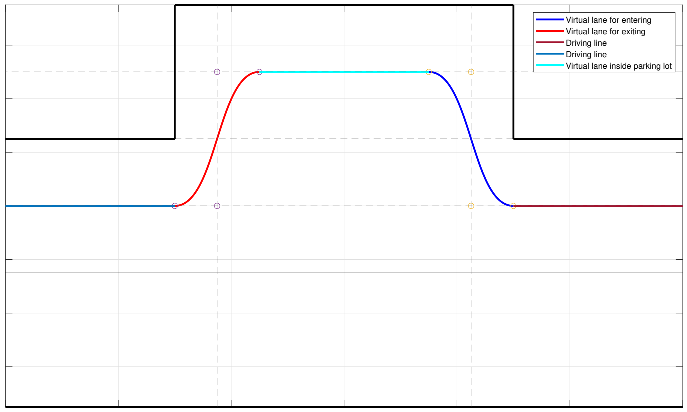
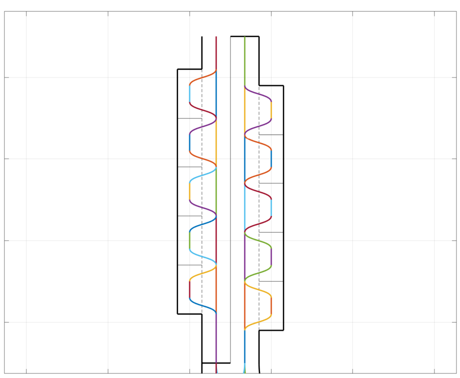
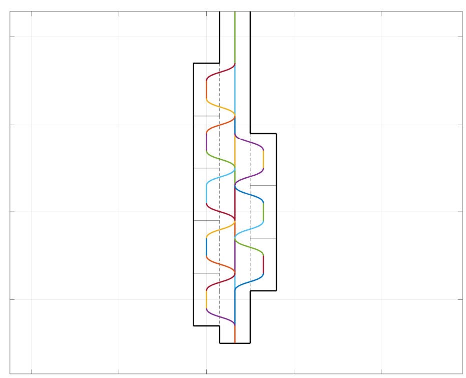


Figure 3.12: Virtual lane construction to access/exit parking lots

Figure 3.12 shows how the control point is found by constructing a virtual lane connecting the event point in the lane and the event point in the parking lot. The procedure as for intersections is used for constructing a virtual lane. The constructed virtual lane never crosses the boundary line. Figure 3.13a shows how the virtual lanes look like in the physical road network.



(a) Multi-way road segment



(b) One-way road segment

Figure 3.13: Virtual lanes to enter/exit parking lots in the visualization

In order to allow parking to be added on both sides of a one-way road segment, the event points associated with parking lots on one side are distinguished from those associated with parking on the opposite side to make it easier to find out which event points on the road segment need to be connected to which event points in the parking lot. The results are shown in Figure 3.13b. Table 3.1 shows the summary of which functions are used to construct paths in each road network element. The found physical paths are assigned to the corresponding edges in the graphical representation in order to make it easy to convert paths in road network graph to physical road network. In this conversion, after finding a combination of edges and nodes for the shortest path, the preconstructed physical paths corresponding to each of the edges found are called, and the set of physical paths are joined together to create a global physical path to a certain destination.

Table 3.1: Road network elements and function used to construct paths

Road Network Element	Functions
Road Segment	• third-order Bezier curve
Parking Lot	• third-order Bezier curve
Intersection	• third-order Bezier curve • second-order Bezier curve
Roundabout	• third-order Bezier curve • second-order Bezier curve • Circle function

Chapter 4

Numerical Simulation

A realistic visualization and graphical representation of the road network is created based on the method of modelling the road network elements and constructing paths defined in the previous section. The created model is tested on different road networks and elements to observe its behavior and validity, including how it handles complex situations. Figure 4.1 shows example of positional information defined by a user, and the visualizations are created based on this type of input data. The corresponding realistic visualization and graphical representation are shown in Figure 4.2.

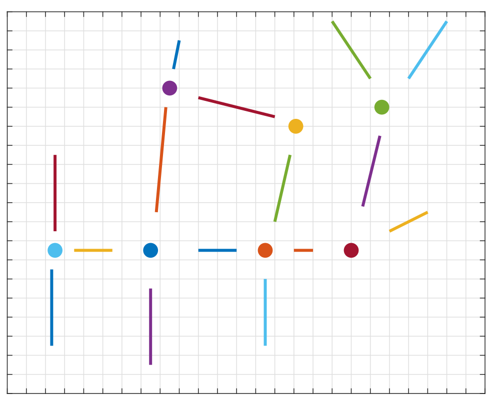
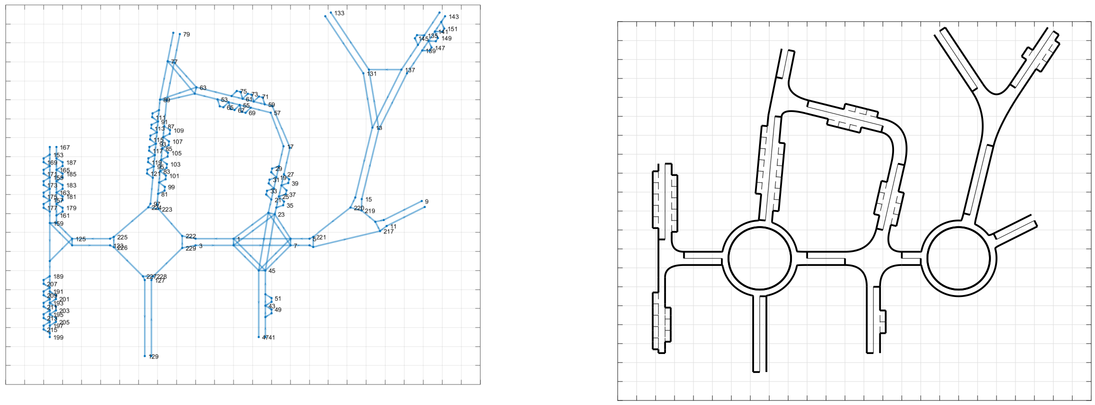


Figure 4.1: User-defined positional information (road location, intersection center, roundabout center).



(a) Graphical representation(For visibility, not all node IDs are shown)

(b) Realistic visualization

Figure 4.2: Road network visualization

Figure 4.2b shows the realistic visualization with several road network elements with different configurations. Thus, the model can handle one-way and multi-directional road segments and can handle either or both parking lots at any location along a road segment. In order to make it clear how many vehicles can fit in a parking space, a line segment dividing the parking space for each vehicle is added. In addition, a dotted line separating the lane and the parking lot from the road segment is also drawn. For intersections, different 3-leg, 4-leg, T-leg, and Y-leg (and L-leg) geometries are modeled appropriately, and roundabouts with different associated lane locations can also be handled. A graphical representation of the road network ensures that all nodes are correctly positioned, all edges show correct connectivity, and are oriented in the correct direction.

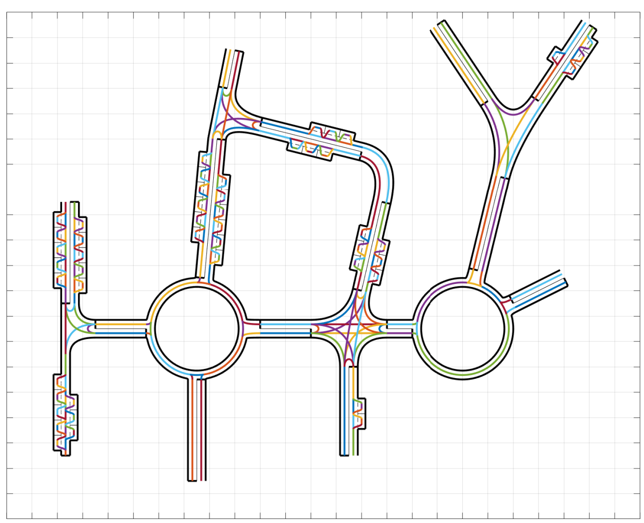
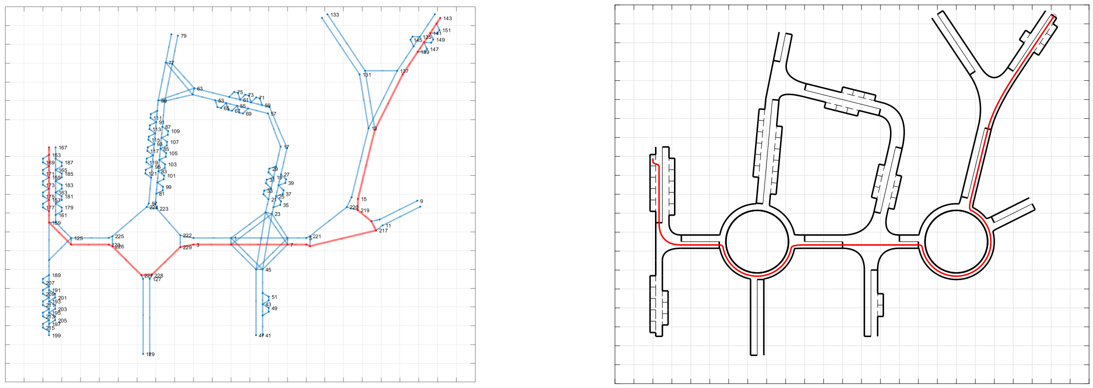


Figure 4.3: Realistic visualization of the road network with all paths

Figure 4.3 shows all the constructed paths plotted on the realistic visualizations. These preconstructed physical paths are used for path planning.

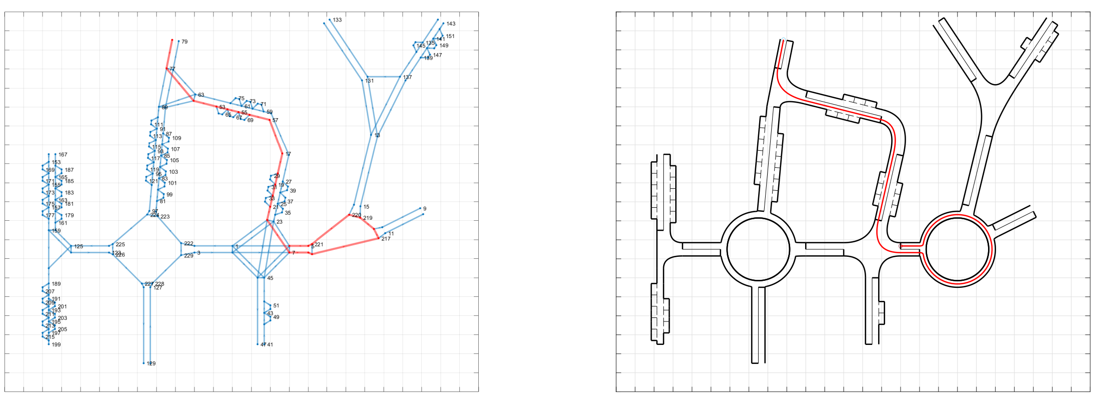


(a) Graphical representation

(b) Realistic visualization

Figure 4.4: Path Planning Over Road Networks

Figure 4.4 shows the found shortest path in graphical representation and the realistic visualization. It can be seen that the road network graph Figure 4.4a is properly converted to the physical path Figure 4.4b.

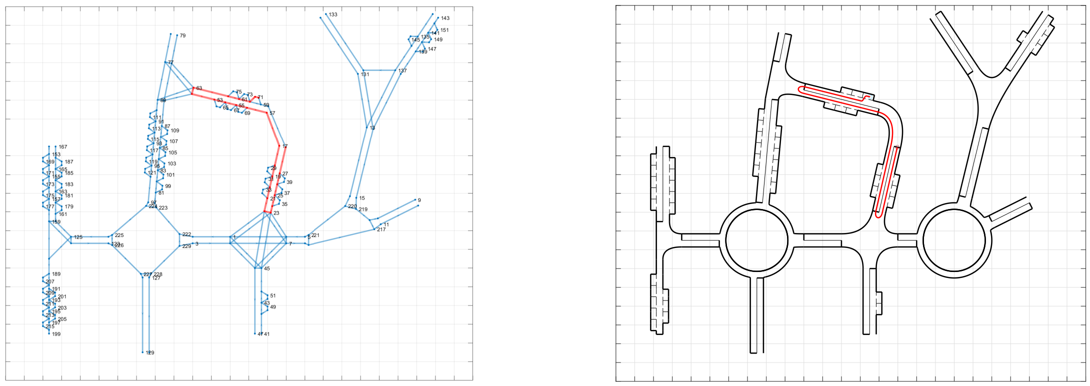


(a) Graphical representation

(b) Realistic visualization

Figure 4.5: Path Planning Over Road Networks that prominently represents connectivity at the intersection and a directional constraint at the roundabout

Figure 4.5 shows the path that is clearly influenced by directional constraints. At the second intersection through which the route passes, the destination is located at one of the starting points associated with the intersection. However, the vehicle cannot go directly to that point because the destination and the node entering the intersection are not connected. Therefore, it moves to another node and enters a roundabout. The roundabout has directional constraint and the vehicle can only travel counterclockwise, so it goes around the roundabout to reach its final destination.

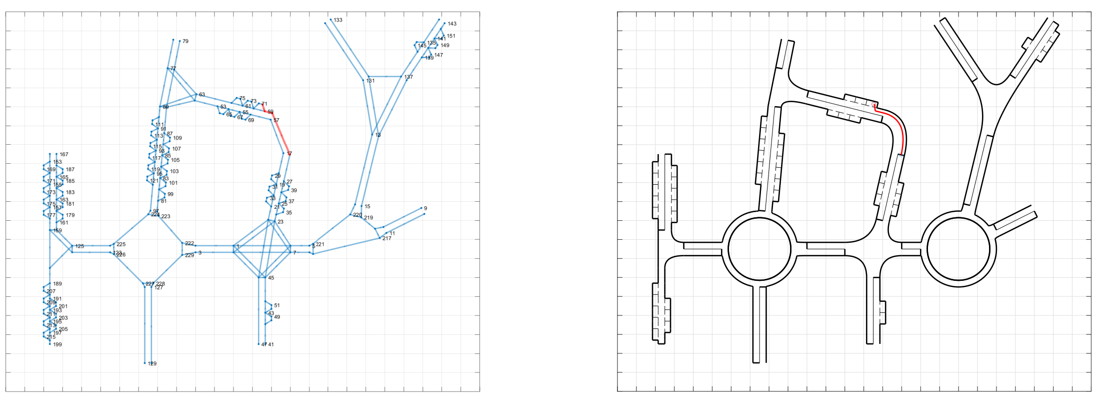


(a) Graphical representation

(b) Realistic visualization

Figure 4.6: Path Planning Over Road Networks including U-turn

Figure 4.6 shows an example path that includes U-turn. Due to the directional constraints, the vehicle cannot go directly to its destination and instead makes two U-turns to detour. Furthermore, directional constraints prevent vehicles from going in the opposite direction of the lane direction, so the path becomes different if the start and end points are interchanged as shown in Figure 4.7.



(a) Graphical representation

(b) Realistic visualization

Figure 4.7: Path Planning Over Road Networks with start and end points of the path shown in Figure 4.6 interchanged

Chapter 5

Conclusion

The goal of this project is to design a graphical and physical representation of the road network for path planning for autonomous vehicles. Several road network elements are discussed, including road segments, parking lots, intersections, and roundabouts, and how they are implemented in the representation. To create a graph of the road network, it is important to consider where to place the nodes, how to connect them at the edges, and effectively store the data in a way that maintains accuracy, storage efficiency, and ease of use. Information to include in the road network elements such as connectivity and directional constraints, and the assumptions that must be made when modeling them are also discussed. A method for detecting event points at roundabouts using merging and diverging points is presented, and its limitations are simulated using existing roundabout configurations. In parallel, realistic visualizations and their creation based on given data were discussed.

With regard to path planning, several road modeling techniques for physical path construction are presented and their performance, especially their accuracy, is tested. Second and third order Bezier curves are finally selected as the road modeling techniques for the current project, and which function to use in which context for each road network element is discussed.

Finally, numerical simulations are performed on the created graphs and paths to observe and validate their behavior for different scenarios.

5.1 Improvements and Future Steps

The proposed approach has several issues related to computational cost. One of the most computationally expensive procedures is where the event point detection method is applied. This solves a fourth-order polynomial, which is relatively computationally expensive and consequently time-consuming. In order to reduce the computation time, it is necessary to shorten the computation time with a simpler method. Another cause is the use of the Matlab function `fplot()` for realistic visualization. The realistic visualization created has hundreds of functions to plot, which significantly increases the computation time. Also, since each edge of the graph is plotted by a function, the arrangement of nodes and edges must be modified to reduce the number of functions.

As explained in chapter 2, there are several prerequisites for generating a road network graph. Some of them are critical when modeling road networks involving complex configurations. For example, road segments are currently assumed to be straight, which is not necessarily true. The lanes of the road segment were approximated using a cubic Bezier curve and four control points, which did not work correctly when the lanes were not straight. Attempts were made to fit lanes of all shapes with sufficient accuracy, but the computational cost was high because the accuracy had to be calculated by measuring the offset between the data points and the curve. It was found that the more data points each Bezier function approximated, the higher the computational cost became, and since it was not possible to come up with an alternative that would reduce the computational cost, and since this was not a problem for straight roads, it was decided to omit it.

The basic road configuration, multi-lane, has not yet been implemented. This was omitted first to simplify the model, but time constraints prevented this from being implemented in the proposed approach and this assumption still remains.
There are several forms of roundabouts that are different from those implemented in the model, as described in section 6.2. These are those with multiple lanes and have not yet been incorporated into the model. By modifying the proposed approach and removing as many of these approximations as possible, the proposed approach will eventually have the flexibility to handle a variety of complex situations. The following is a list of the necessary modifications known at this time.

Future Steps

- Reduce computational cost

  Modify event points detection method

  Reduce number of functions to proposed approach the road network

- Find an alternative method to approximate the lanes using Bezier curves
- Implement multi-lanes
- Implement different roundabout and intersection forms

Chapter 6

Appendix A: Limitation of Event Point Detection Method at Roundabouts

The method proposed to detect event points relies on the intersection of a virtual lane from the start point to the end point and a circular virtual lane, which depends on the size of the radius of the roundabout and the location of the incoming and outgoing lanes where the virtual lane is drawn. If the radius of the roundabout is lower than a certain value, the two virtual lanes may not intersect and the event points cannot be identified. In this case, the proposed method is invalid. In this section, the radius limits are discussed. The Euclidean distance of the constructed Bezier curve to the center of intersection can be computed by rewriting Equation 3.5;

$$d_{ic} = ||P_b(t) - P_{ic}|| \tag{6.1}$$

The minimum of Equation 6.1 is the minimum radius of the roundabout required to for the virtual lanes to intersect.

$$t_{min} = argmin(||P_b(t) - P_{ic}||) \tag{6.2}$$

Specifically, the derivative of Equation 6.1 was determined and the value of t such that the derivative is zero was found to be the t corresponding to the minimum d_{ic}. To investigate the effect of the angle between lanes, the end point P_1 was moved along the perimeter of the roundabout and the minimum required radius was calculated as shown in Figure 6.1.

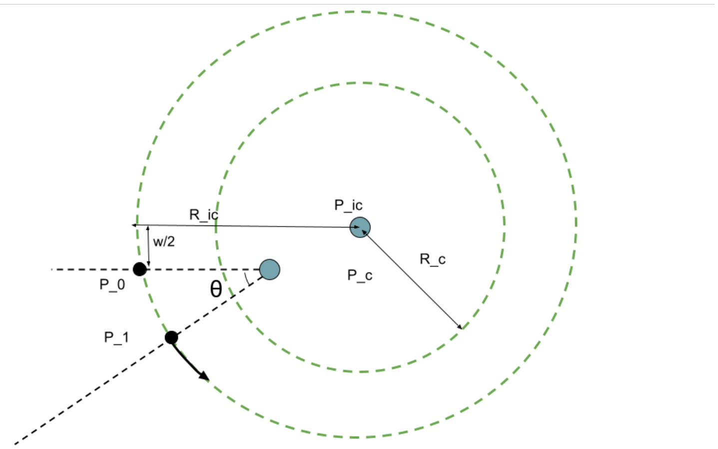


Figure 6.1: How the influence of the angle is investigated

The intersection point P_c of the virtual lane shown in Figure 6.1 is calculated such that the distance from that point to the start and end points, respectively, is equal.

$$\begin{aligned} P_c &= \begin{bmatrix} x_c \\ y_c \end{bmatrix} = \begin{bmatrix} -\frac{w}{2tan(\theta)} \\ -\frac{w}{2} \end{bmatrix} \\ P_1 &= \begin{bmatrix} x_c + cos(\frac{\theta}{2})(x_c - x_0)\sqrt{2(1 - cos(\theta))} \\ y_c - sin(\frac{\theta}{2})(x_c - x_0)\sqrt{2(1 - cos(\theta))} \end{bmatrix} \end{aligned} \tag{6.3}$$

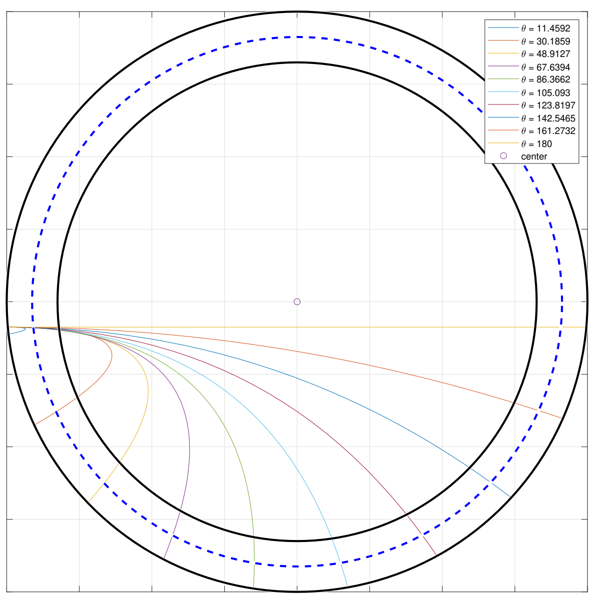
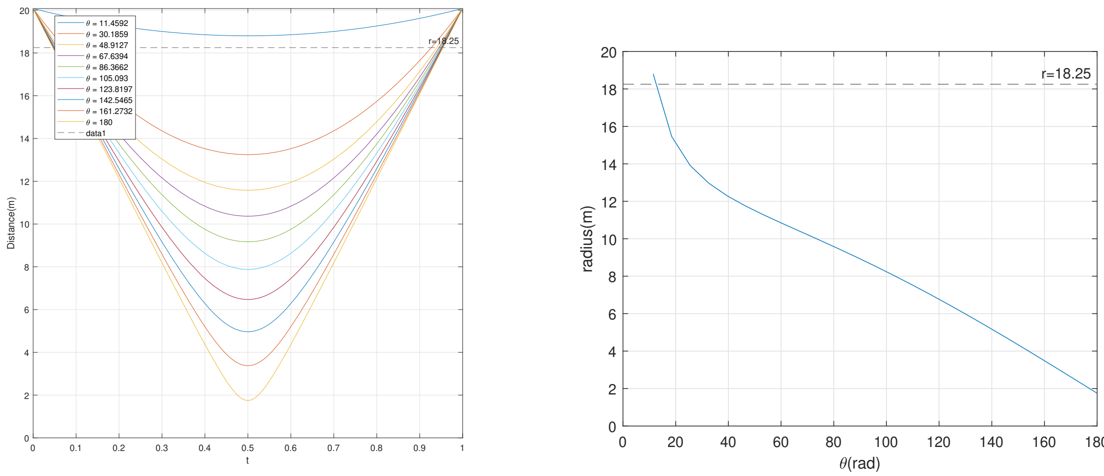


Figure 6.2: Example roundabout with virtual lanes for event points detection with different angles

Figure 6.2 shows the roundabout with virtual lanes drawn to detect event points. The blue dotted line shows the center of the circular lane around the roundabout and the intersection points of that and the virtual lanes with different position of end points are defined as event points. Virtual lanes run outside of the boundary, but do not necessarily have to be inside. This is because these lanes are used to detect event points, and the appropriate virtual lanes will be redrawn using the detected event points.



(a) Euclidean distance from the center of roundabout with different angles

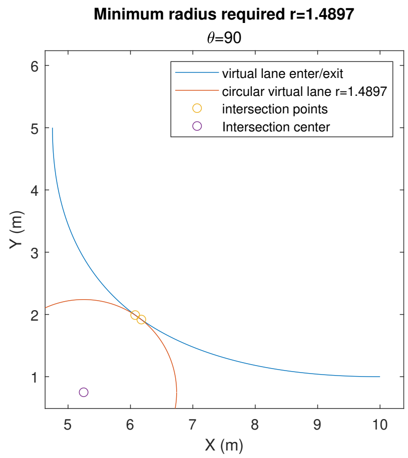
(b) Minimum radius as a function of angleθ

Figure 6.3: Minimum radius of roundabout required

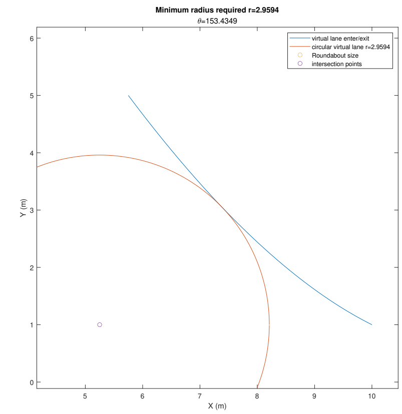
As shown in Figure 6.3, the minimum radius of roundabouts required decreases as the angle increases. In other words, this limitation of the proposed method becomes more critical when the lanes intersect with smaller angle. For example, when $\theta = 11.4592°$, the virtual lanes do not intersect. This is considered negligible because the roads would interfere with each other when intersecting at such small angles, which is not practical.

6.1 Investigating the limitation of the proposed event point identification method by applying it to the existing setup

The minimum required radius was investigated using an existing roundabout in Eindhoven. The intersection points of two virtual lanes are determined numerically using the Bisection method to find the radius r when there is only one intersection in different setups. The Bezier curves are drawn using the start/end points and the intersection point of the entering/exiting lane.

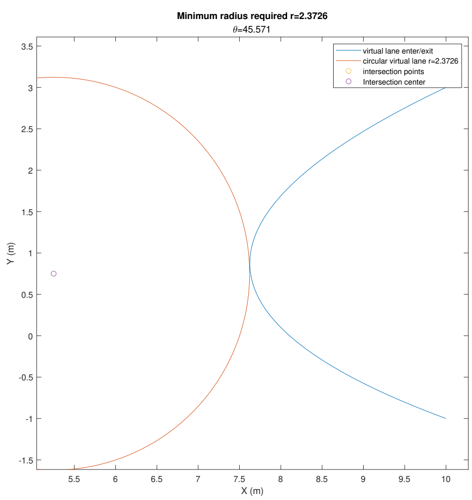


(a) Entrance:[10, 1] Exit:[5.75, 5]

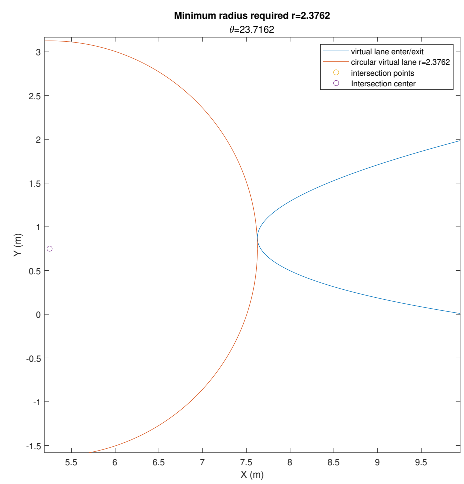


(b) Entrance:[10, 1] Exit:[5.75, 5]

Figure 6.4: Minimum required radius when the intersection is located at [5.25, 0.75] and angle $\theta = 90, 153.4°$



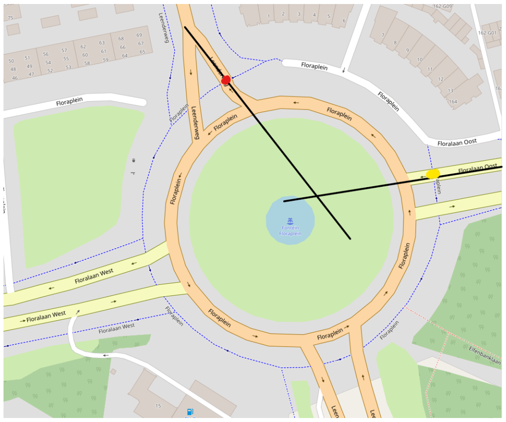
(a) Entrance:[10, -1] Exit:[10, 3]



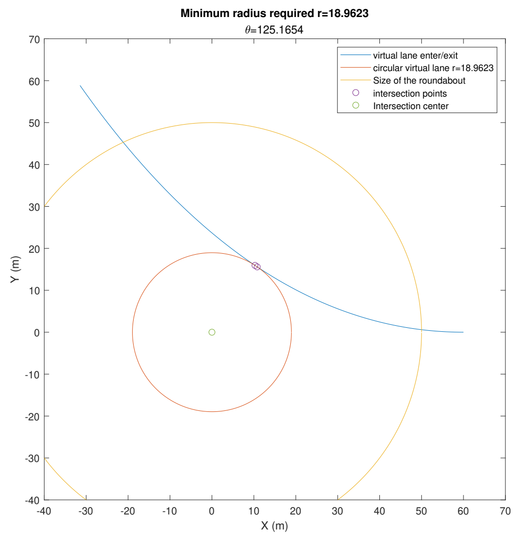
(b) Entrance:[10, 0] Exit:[10, 2]

Figure 6.5: Minimum required radius when the intersection is located at [5.25, 0.75] and angle $\theta = 45.6, 23.7°$

From the above figures, it can be seen that the angle between the lanes affects the required minimum radius, and the further away from 90 degrees, the larger the minimum radius. Now the method is applied to the existing roundabout to investigate whether the limitation is fatal or not.

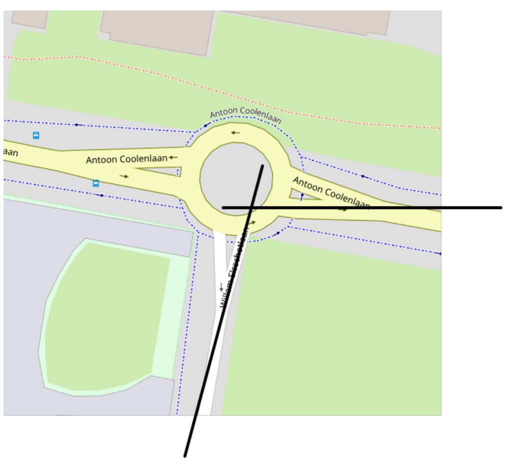


(a) Example 1, r=50m

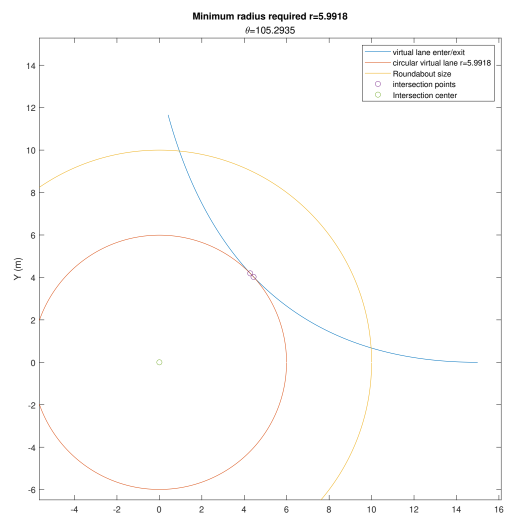


(b) Implemented in matlab

Figure 6.6: Minimum required radius(r=19.0m) and the actual radius of the roundabout(r=50m)

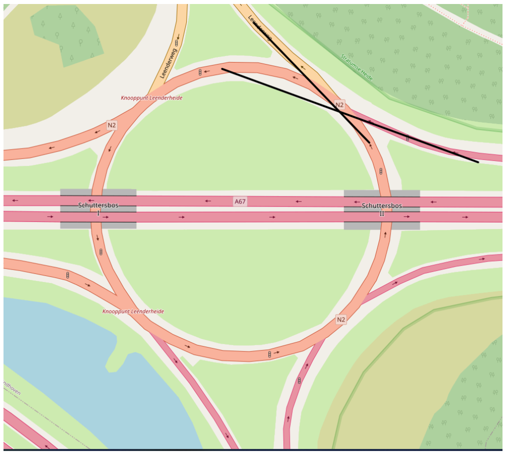


(a) Example 2, r=10m

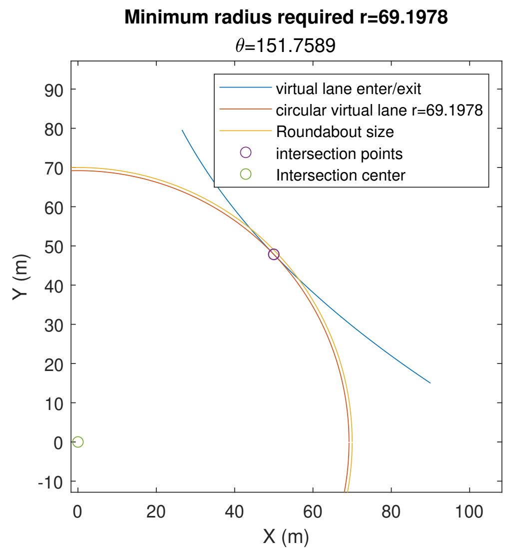


(b) Implemented in Matlab

Figure 6.7: Minimum required radius(r=6.0m) and the actual radius of the roundabout(r=10m)



(a) Example 3, r=70m



(b) Implemented in Matlab

Figure 6.8: Minimum required radius(r=69.2m) and the actual radius of the roundabout(r=70m)

The first two cases shown in Figure 6.6 and Figure 6.7 can safely modelled using the proposed method because the difference between the minimum required radius and the actual roundabout radius is considered large enough. However, the last case, Figure 6.8, may pose a problem because the difference between the radius of the roundabout and the minimum radius is small, about 0.8 m, and the merging and diverging points are close together. This problem is due to the fact that the angle between the entering and exiting lanes is close to $180°(\theta = 152°)$ and the entry/exit lanes are located close together, so the intersection of the two lanes is close to the edge of the roundabout. This indicates that the risk of the proposed method is that two intersections may not occur if the entry and exit lanes are close to each other and the angle between them is large, close to $180^{c}irc$.

Figure 6.9: Satellite image of the roundabout shown in Figure 6.8

Also, the roundabout in Figure 6.8 has multiple lanes and the entry and exit lanes are separated from the circular lanes aorund the roundabout as shown in Figure 6.9, making it similar to a channelized roundabout shown in Figure 2.5. Therefore, treating such a roundabout as a channeling roundabout rather than a regular roundabout can prevent risk. This problem may also be solved as a byproduct of implementing multilanes in the model.

6.2 Various Roundabout Forms

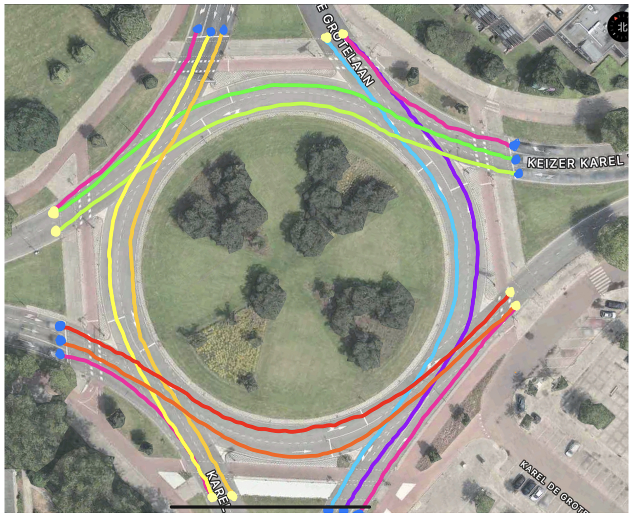


Figure 6.10: Symmetric multi-lane roundabout with constant radii; 3 entry lanes(2 out of 3 actually enter) and 2 exiting lanes each

Figure 6.10 shows the existing multi-lane roundabout in Eindhoven. This roundabout contains two lanes inside. The virtual lanes drawn in the figures follows the first step introduced in the previous section(take the nearest (possible) exit). However, the virtual lanes shown in pink in the figure does not actually enter the roundabout, but use additional lanes that are not part of the roundabout. Therefore, it is necessary to improve the method to detect such lanes and treat them differently from other lanes. Alternatively, a solution could be to allow for the existence of lanes that do not intersect, although there is a risk that lanes that should intersect will not intersect. Additionally, knowing which lanes should intersect which circular lanes is important for proper placement of event points.

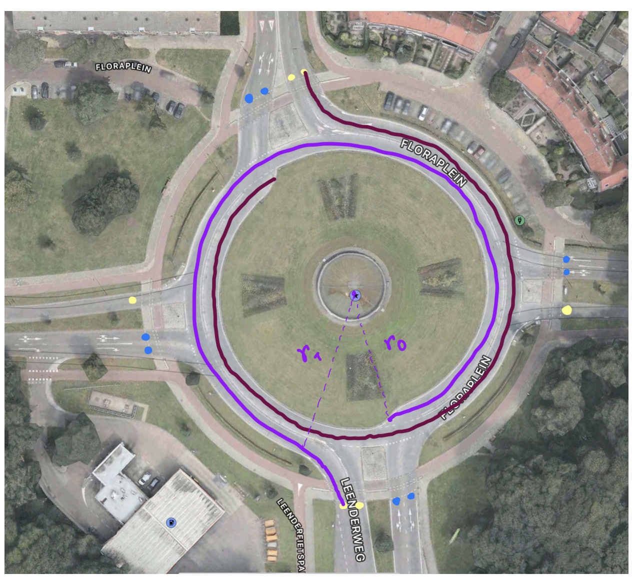


Figure 6.11: Symmetric multi-lane roundabout with inconstant radii

Figure 6.11 shows another existing roundabout in Eindhoven. This roundabout does not have a constant radius for the circular lanes. Since the current method assumes that the radii of the circular lanes are constant, it is necessary to rewrite the function that constructs the lanes to account for changes in radius. A solution can be to just use the Bezier curve. Furthermore, in this case, since the circular lane is not a perfect circle, it is also necessary to define the start and end points of it.

Chapter 7

Appendix B: Matlab Implementation

This section describes how the graphical representation and realistic visualization of the road network, the construction of paths and the execution of path planning are implemented in Matlab.

7.1 Input Data

Firstly, the code below shows an example of how the road network information described in subsection 2.4.1 are input to Matlab.

```
R(5).nl=[1, 1]; %number of lanes in each direction
R(5).w=3.5;%Width of the lane
R(5).pl=[linspace(0, 20, 100)', linspace(0, 0, 100)'];%lane position set
R(5).pa(1).n=3;%number of parking lots
R(5).pa(1).l=6;%length of a parking lot
R(5).pa(1).w=3;%width of a parking lot
R(5).pa(1).s=0.3;%starting point of a parking lot. e.g. 0.3*total length of a road
R(5).pa(2).n=3;%Parking lot on the other side
R(5).pa(2).l=6;
R(5).pa(2).w=3;
R(5).pa(2).s=0.1;

C.i=1;%intersectoin id
C.w=3.5;%width of virtual lanes
C.p_ic=[35, 0];%center of an intersection
C.T=[0, 1, 0, 1, 0, 1, 0, 1;%traffic matrix
    0, 0, 0, 0, 0, 0, 0, 0;
    0, 1, 0, 1, 0, 1, 0, 1;
    0, 0, 0, 0, 0, 0, 0, 0;
    0, 1, 0, 1, 0, 1, 0, 1;
    0, 0, 0, 0, 0, 0, 0, 0;
    0, 1, 0, 1, 0, 1, 0, 1;
    0, 0, 0, 0, 0, 0, 0, 0;
    ];
C.Ic=[1, 2, 5, 6];%Road segments connected at the intersection

D(1).nl=1;%Number of circular virtual lanes of a roundabout
D(1).r=18.25;%Radius of a circular virtual lane of a roundabout
D(1).w=3.5;%Width of the virtual lanes
D(1).p_ic=[80, 0];%Center of the roundabout
D(1).Ic=[2, 3, 4];%Road segments connected at the roundabout
```

7.2 Road Segments

The code below computes the position of the lane centers in case of multi-way road segments. $R(i).l(1).p$ contains function `flip()` to match the direction of travel.

```
R(i).l(1).p=flip([R(i).pl(:, 1)-R(i).w/2*cos(pi/2-R(i).angle), R(i).pl(:, 2)+R(i).w/2*sin(pi/2-R(
    i).angle)]);%position of a lane
```

```
R(i).l(2).p=[R(i).pl(:, 1)+R(i).w/2*cos(pi/2-R(i).angle), R(i).pl(:, 2)-R(i).w/2*sin(pi/2-R(i).angle)];
```

The code below computes the position of the event points inside the parking lots and along the lane to access the parking lots. $R(i).dpl\_cumsum$ is the cumulative sum of the road length and it is used together with $R(i).pa(1).s$ to compute the where the parking lot starts. $dist\_s\_point\_relative$ indicates position of the event points inside the parking lots and it is the relative distance from the start point of the parking lot. After the parking location expressed as the length of the road segment from the start point to the location of the parking event point is found, the row data is interpolated to have exactly equal data point(s) to their coordinates as written in line 9-10. Same procedure is taken for boundary line computation.

```
1  R(i).dpl_cumsum=[0; cumsum(sqrt(R(i).dpl(:, 1).^2+R(i).dpl(:, 2).^2))];%Cummulative sum of the road length
2  R(i).pa(1).se_id=round(length(R(i).pl)*R(i).pa(1).s);%index in row data of position data where parking spot starts and ends
3  R(i).pa(1).dist_s=R(i).dpl_cumsum-max(R(i).dpl_cumsum)*R(i).pa(1).s;%distance between cummulative sum at starting position of parking lot and road segment
4  dist_s_point_relative=[2, 4];%Distance between the 2 points in a parking lot and a point to access it
5  dist_s_point=[];
6  for j=0:R(i).pa(1).n-1
7      dist_s_point=[dist_s_point, dist_s_point_relative+(j*6)];%Location of points in parking lots in terms of length of road segment.
8  end
9   R(i).pa(1).p=interp1(R(i).pa(1).dist_s, R(i).pl, dist_s_point);%coordinates of points on a road segment used to find the positions of parking lots
10  R(i).pa(1).angle=interp1(R(i).pa(1).dist_s, R(i).angle, dist_s_point)';%Angle of the coordinates of points on a road segment used to find the positions of parking lots
11
12 R(i).pa(1).p=[R(i).pa(1).p(:, 1)-(R(i).w.*(sum(R(i).nl)/2)+R(i).pa(1).w/2).*cos(pi/2-R(i).pa(1).angle), R(i).pa(1).p(:, 2)+((R(i).w.*(sum(R(i).nl)/2)+R(i).pa(1).w/2)).*sin(pi/2-R(i).pa(1).angle)];%Coordinates of parking lots
13 R(i).l(1).angle=interp1(R(i).dpl_cumsum', R(i).angle, [0, linspace( R(i).pa(1).s*R(i).dpl_cumsum(end), R(i).pa(1).s*R(i).dpl_cumsum(end)+R(i).pa(1).l*R(i).pa(1).n, R(i).pa(1).n+1), max(R(i).dpl_cumsum)]); %angle of the road segment(lane) at the point on lane to access parking lot
14
15 points_pa= interp1(R(i).pa(1).dist_s, R(i).pl, 6*linspace(0, R(i).pa(1).n, R(i).pa(1).n+1))+[-((R(i).w/2.*(sum(R(i).nl)-1))*cos(pi/2-R(i).l(1).angle(2:end-1)))', ((R(i).w/2.*(sum(R(i).nl)-1))*sin(pi/2-R(i).l(1).angle(2:end-1)))']; %positions of the parking lots
16 R(i).l(1).P_set=[R(i).l(1).P_set(1, :); points_pa ; R(i).l(1).P_set(2, :)];%Event points on the road segment
```

7.3 Intersection

The following code calculates the distance from each data point on the boundary and lane to the center of the intersection. The minimum value of this distance is the start and end point.

```
norm_l1=sqrt((R(C(i).Ic(j)).l(1).p(:, 1)-C(i).p_ic(1)).^2+(R(C(i).Ic(j)).l(1).p(:, 2)-C(i).p_ic(2)).^2);%Distance of lhe lane center to the center of intersection
norm_l2=sqrt((R(C(i).Ic(j)).l(2).p(:, 1)-C(i).p_ic(1)).^2+(R(C(i).Ic(j)).l(2).p(:, 2)-C(i).p_ic(2)).^2);
norm_b1=sqrt((R(C(i).Ic(j)).boundary1(:, 1)-C(i).p_ic(1)).^2+(R(C(i).Ic(j)).boundary1(:, 2)-C(i).p_ic(2)).^2);%Distance of boundary line to the center of intersection
norm_b2=sqrt((R(C(i).Ic(j)).boundary2(:, 1)-C(i).p_ic(1)).^2+(R(C(i).Ic(j)).boundary2(:, 2)-C(i).p_ic(2)).^2);
```

The following code construct a Bezier curve to connect associated boundary lines in an intersection in case 3rd-order Bezier curve is used. Lines 2 and 3 ensure that the gradient $m\_b\_perpendicular$ is always integer. p_b_mid is the mid point of two points to be connected and this is used to compute two intermediate control points $p2, p3$.

```
1 m_b_perpendicular=-2/(C(i).m_b(2*k+1)+C(i).m_b(2*k+2));
2 m_b_perpendicular(isinf(m_b_perpendicular))=1e13;
3 m_b_perpendicular(isnan(m_b_perpendicular))=1e-13;
4
5 p_b_mid=mean(C(i).p_b((2*k+1):(2*k+2), :));
6 y_int_b_perpendicular=p_b_mid(2)-m_b_perpendicular*p_b_mid(1);
7 x2=(y_int_b_perpendicular-C(i).y_int_b(2*k+1))./(C(i).m_b(2*k+1)-m_b_perpendicular);
8 y2=C(i).m_b(2*k+1)*x2+C(i).y_int_b(2*k+1);
```

```
y_int_b_perpendicular=p_b_mid(2)-m_b_perpendicular*p_b_mid(1); %y-interept of a line perpendicular
    to the one with the mean of a set of entry/exit lane
p2=[x2, y2];
x3=(y_int_b_perpendicular-C(i).y_int_b(2*k+2))/(C(i).m_b(2*k+2)-m_b_perpendicular);
y3=C(i).m_b(2*k+2)*x3+C(i).y_int_b(2*k+2);
p3=[x3, y3];

P_bez=[C(i).p_b(2*k+1, :);[x2, y2];[x3, y3];C(i).p_b(2*k+2, :)];%Control points to draw bezier
    curve to connect the boundary lines
C(i).boundary_connect(k+1, :)= simplify(B_3*P_bez);
```

7.4 Roundabout

Same procedure as one for intersection is taken for finding start and end point is taken at roundabout. $D(i).dpol$ is the gradient of the contour line at the start and end points and is used to compute the coordinates of the control points connecting the boundary lines.

```
D(i).ol=[(D(i).r+D(i).w/2)*cos(t)+D(i).p_ic(1), (D(i).r+D(i).w/2)*sin(t)+D(i).p_ic(2)];%Outer
    radius of a roundabout
D(i).dol=diff(D(i).ol);
D(i).pol=[(D(i).r+D(i).w/2)*cos(D(i).d_angle)+D(i).p_ic(1), (D(i).r+D(i).w/2)*sin(D(i).d_angle)+D(
    i).p_ic(2)]; %Start/end points where the circule should be drawn
D(i).dpol=vpa(subs(D(i).dol, D(i).d_angle));
D(i).m_pol=D(i).dpol(:, 2)./D(i).dpol(:, 1);
D(i).y_int_pol=D(i).pol(:, 2)-D(i).m_pol.*D(i).pol(:, 1);

D(i).p_b_c=[(D(i).y_int_pol-D(i).y_int_b)./(D(i).m_b-D(i).m_pol), D(i).m_b.*(D(i).y_int_pol-D(i).
    y_int_b)./(D(i).m_b-D(i).m_pol)+D(i).y_int_b];%Intersection point of the boundary lines
```

7.5 Construction of physical paths and assignment to corresponding edges

To identify event points, the index number of p, an array of all event points, is used as the node ID. The following code shows how to assign edges to lanes of a road segment and construct a physical path using the Matlab function `ismember()` to find the index number of p of event points along the road segment and assign the start point of an edge to the edge's assign to s, the set of all start points, and to u, the set of all end points of the edge. At the same time, the physical path corresponding to the edge is computed (lines 5-14) and assigned to '$edge_p$'.' s', 'u' and '$edge_p$' are later combined to create a dataset with the full set of start and end points of the edge and the corresponding physical paths. bez_lane_fit is a function that is created to approximate lanes of road segments.

```
1  index_s=ismember(p, R(i).l(j).P_set(k, :), 'rows');
2  index_u=ismember(p,  R(i).l(j).P_set(k+1, :), 'rows');
3  s=[s, find(index_s==1)];
4  u=[u, find(index_u==1)];
5  if size(R(i).l(j).m_func, 1)==0 %Fit the lane position
6      R(i).l(j).l_func=[R(i).l(j).l_func; bez_lane_fit(R(i).l(j).p(R(i).l(j).p_id(k):R(i).l(j).p_id(
          k+1), :), [])];
7  else
8      R(i).l(j).l_func=[R(i).l(j).l_func; bez_lane_fit(R(i).l(j).p(R(i).l(j).p_id(k):R(i).l(j).p_id(
          k+1), :), R_l_m_func(1))]; %To maintain continuity, use the slope of the lane at the
          starting point of the data
9  end
10 R(i).l(j).m_func=[R(i).l(j).m_func; diff(R(i).l(j).l_func(end, 2))./diff(R(i).l(j).l_func(end, 1))
       ];
11 R(i).l(j).angle_func=[R(i).l(j).angle_func; atan2(diff(R(i).l(j).l_func(end, 2)), diff(R(i).l(j).
       l_func(end, 1)))];
12 R_l_m_func=matlabFunction(R(i).l(j).m_func(end)+1e-13*t); %matlabFunction to avoid getting errors
       due to division by 0
13 edge_p(end+1).l_func=R(i).l(j).l_func(end, :);
14 edge_p(end).range=[0, 1];
```

The entire set of physical routes, edges, and nodes have been prepared now. Based on this information, a graphical representation of the road network and a path conversion from the road network graph to the physical road network are performed.

Bibliography

[1] Planning¨ Planning¨omür Arslan and Aron Tiemessen. Adaptive Bézier Degree Reduction and Splitting for Computationally Efficient Motion Planning¨Omür. (January):1–23, 2022.

[2] David Betaille and Rafael Toledo-Moreo. Creating enhanced maps for lane-level vehicle navigation. *IEEE Transactions on Intelligent Transportation Systems*, 11(4):786–798, 2010.

[3] S Carozza, F Jalayer, A Miano, and G Manfredi. Probabilistic Connectivity Analysis for a Road Network Due To Seismically-Induced Disruptions. In *16th European Conference on Earthquake Engineering*, number January, 2017.

[4] Krzysztof Czarnecki. WISE Requirements Analysis Framework for Automated Driving Systems Operational World Model Ontology for Automated Driving Systems Part 1: Road Structureca/waterloo--intelligent--systems--engineering--lab/projects/wise--requirements--analysis--frame. Technical Report July, 2018.

[5] Chunzhao Guo, Kiyosumi Kidono, Junichi Meguro, Yoshiko Kojima, Masaru Ogawa, and Takashi Naito. A low-cost solution for automatic lane-level map generation using conventional in-car sensors. *IEEE Transactions on Intelligent Transportation Systems*, 17(8):2355–2366, 2016.

[6] Gi Poong Gwon, Woo Sol Hur, Seong Woo Kim, and Seung Woo Seo. Generation of a Precise and Efficient Lane-Level Road Map for Intelligent Vehicle Systems. *IEEE Transactions on Vehicular Technology*, 66(6):4517–4533, 2017.

[7] LE Hall and R.D. Powers. Overview of cross section design elements, 1995.

[8] Lingfei Ma, Ying Li, Jonathan Li, Zilong Zhong, and Michael A. Chapman. Generation of horizontally curved driving lines in HD maps using mobile laser scanning point clouds. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 12(5):1572–1586, 2019.

[9] Kamaldeep Singh Oberoi, Geraldine Del Mondo, Yohan Dupuis, and Pascal Vasseur. Towards a qualitative spatial model for road traffic in urban environment. In *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, volume 2018-March, pages 1–6, 2018.

[10] Brian Paden, Michal Čáp, Sze Zheng Yong, Dmitry Yershov, and Emilio Frazzoli. A survey of motion planning and control techniques for self-driving urban vehicles. *IEEE Transactions on Intelligent Vehicles*, 1(1):33–55, 2016.

[11] Jianglin Qiao, Dongmo Zhang, and Dave de Jonge. Graph Representation of Road and Traffic for Autonomous Driving. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 11672 LNAI, pages 377–384, 2019.

[12] Bruce W. Robinson. Roundabouts: an informational guide. Technical report, U.S. Department of Transportation, Federal Highway Administration, 6 2000.

[13] TomÅ Tollazzi. First developments of different roundabout types. In *Springer Tracts on Transportation and Traffic*, volume 6, pages 11–55. Springer, Cham, 2015.

[14] Tao Zhang, Stefano Arrigoni, Marco Garozzo, Dian ge Yang, and Federico Cheli. A lane-level road network model with global continuity. *Transportation Research Part C: Emerging Technologies*, 71:32–50, 2016.

[15] Yunfei Zhang, Zexu Zhang, Jincai Huang, Tingting She, Min Deng, Hongchao Fan, Peng Xu, and Xingshen Deng. A hybrid method to incrementally extract road networks using spatio-temporal trajectory data. *ISPRS International Journal of Geo-Information*, 9(4), 2020.

[16] Ling Zheng, Bijun Li, Bo Yang, Huashan Song, and Zhi Lu. Lane-level road network generation techniques for lane-level maps of autonomous vehicles: A survey. *Sustainability (Switzerland)*, 11(16):1–19, 2019.

[17] Ling Zheng, Bijun Li, Hongjuan Zhang, Yunxiao Shan, and Jian Zhou. A high-definition road-network model for self-driving vehicles. *ISPRS International Journal of Geo-Information*, 7(11):1–14, 2018.

[18] Yiyang Zhou, Yuichi Takeda, Masayoshi Tomizuka, and Wei Zhan. Automatic Construction of Lane-level HD Maps for Urban Scenes. In *IEEE International Conference on Intelligent Robots and Systems*, number C, pages 6649–6656, 2021.