Technical University of Crete
School of Production Engineering and Management

Specifying desired orientations for automated vehicles on complex lane-free roundabouts via Dynamic Programming

---

Καθορισμός επιθυμητού προσανατολισμού για αυτόματα οχήματα σε περίπλοκους κυκλικούς κόμβους χωρίς λωρίδες μέσω Δυναμικού Προγραμματισμού

Diploma Thesis

Maria - Konstantina Mavroeidi

Supervisor

Ioannis Papamichail

February 2023

Acknowledgements

I would like to thank my supervisors Prof. Ioannis Papamichail and Prof. Markos Papageorgiou for their guidance, insights and feedback throughout the course of writing my thesis.

In addition, I would like to offer my appreciation to Dr. Mehdi Naderi for his invaluable patience, motivation, and support.

Περίληψη

Η κυκλοφοριακή συμφόρηση αποτελεί ένα ολοένα και σοβαρότερο κοινωνικό πρόβλημα που συμβάλλει σε υπερβολικές καθυστερήσεις, σημαντική περιβαλλοντική ρύπανση και μειωμένη ασφάλεια της κυκλοφορίας. Η εισαγωγή αυτοματοποιημένων και διασυνδεδεμένων οχημάτων θα μπορούσε να αποτελέσει αποτελεσματική λύση για την αντιμετώπιση του προβλήματος αυτού, δεδομένου ότι μπορούν να ενθαρρυνθούν ώστε να λαμβάνουν καταλληλότερες αποφάσεις με βάση τις πληροφορίες που λαμβάνουν από την υποδομή, τα άλλα οχήματα και τους αισθητήρες τους.

Ο σχεδιασμός της διαδρομής που οφείλουν να ακολουθούν τα οχήματα σε μεγάλους, πολύπλοκους κυκλικούς κόμβους χωρίς λωρίδες κυκλοφορίας αποτελεί πρόκληση λόγω των γεωμετρικών χαρακτηριστικών τους και των συχνών συγκρούσεων μεταξύ των εισερχόμενων, πλοηγούμενων αλλά και εξερχόμενων οχημάτων. Μια θεμελιώδης δυσκολία είναι ο ορθός προσδιορισμός των επιθυμητών προσανατολισμών των οχημάτων στον κυκλικό κόμβο, έτσι ώστε τα οχήματα να εισέρχονται στον κυκλικό κόμβο και να κινούνται προς τις αντίστοιχες εξόδους τους ομαλά και με ασφάλεια. Ο καθορισμός των προσανατολισμών των οχημάτων θα πρέπει να λαμβάνει υπόψη την προκύπτουσα απόσταση διαδρομής, τη γωνιακή διαφορά από τα άλλα οχήματα και την εκμετάλλευση ολόκληρης της διαθέσιμης επιφάνειας του κυκλικού κόμβου για την αποτελεσματική ροή της κυκλοφορίας.

Η παρούσα διπλωματική εργασία προτείνει μια προσέγγιση βέλτιστου ελέγχου για τον προσδιορισμό των βέλτιστων προσανατολισμών οχημάτων σε κάθε σημείο του κυκλικού κόμβου, με βάση τον εκάστοτε επιλεγμένο κλάδο εξόδου, ώστε να ελαχιστοποιείται ένα σταθμισμένο άθροισμα της απόστασης διαδρομής και της απόκλισης από την κυκλική κίνηση. Παρουσιάζονται αναλυτικές λύσεις για τις δύο ακραίες περιπτώσεις, που αφορούν ξεχωριστά τη συντομότερη διαδρομή και την ελάχιστη απόκλιση από την κυκλική γωνία, αντίστοιχα. Ενώ, για τη γενική λύση του σταθμισμένου αθροίσματος των ανωτέρω, χρησιμοποιείται ένας αλγόριθμος βασισμένος στον δυναμικό προγραμματισμό (αναδρομικός Dijkstra) για να προσδιοριστούν οι βέλτιστοι προσανατολισμοί σε ένα δισδιάστατο χώρο-διακριτό πλέγμα της επιφάνειας του κυκλικού κόμβου. Υπό το πρίσμα της βέλτιστης λύσης, προτείνεται επίσης μια υπολογιστικά ελαφριά σχεδόν βέλτιστη προσέγγιση. Ως μια σύνθετη περίπτωση μελέτης, οι μέθοδοι εφαρμόζονται στον διάσημο κυκλικό κόμβο Place Charles de Gaulle στο Παρίσι, ο οποίος διαθέτει πλάτος 38 m και περιλαμβάνει 12 αμφίδρομους ακτινικούς δρόμους, άρα συνολικά 144 ζεύγη προέλευσης-προορισμού για τα οχήματα.

Abstract

Road traffic congestion is an increasingly grave social problem that contributes to excessive delays, substantial environmental pollution, and decreased traffic safety. The introduction of automated and connected vehicles could be an effective solution to tackle this problem since they can be promoted to make more appropriate decisions based on information received from the infrastructure, other vehicles, and their sensors.

Path planning for vehicles on large, complex, lane-free roundabouts is challenging due to the geometrical features and frequent conflicts among entering, navigating, and exiting vehicles. A fundamental difficulty is to properly determine the desired vehicle orientations on the roundabout so that vehicles enter the roundabout and move towards their corresponding exits smoothly and safely. The specification of vehicle orientations should consider the resulting trip distance, the angle difference from other vehicles, and the exploitation of the available roundabout surface for efficient traffic flow.

This diploma thesis proposes an optimal control approach to determine optimal vehicle orientations at each point on the roundabout, in dependence on the exit branch, to minimize a weighted summation of the trip distance and the deviation from the circular motion. Analytical solutions for two extreme cases, addressing only the shortest path or the minimum deviation from the circular angle, respectively, are derived. A Dynamic Programming-based (backward Dijkstra) algorithm is employed for the general weighted problem solution to deliver the optimal orientations in a 2-D space-discretized grid of the roundabout surface. In light of the optimal solution, a computationally light near-optimal approach is also proposed. As a challenging case study, the methods are applied to the famous roundabout of Place Charles de Gaulle in Paris, which features a road width of 38 m and comprises 12 bidirectional radial streets, hence a total of 144 origin-destination movements for the vehicles.

Table of Contents

Table of Figures

1 Introduction

In the last few decades, significant advancements in vehicle automation have been made and this might soon lead to full automation. To tackle traffic congestion and its consequences, like excessive delays, environmental pollution, and reduced traffic safety, traffic control of various kinds [1], [2] has been developed for decades. More recently, the development of a variety of Vehicle Automation and Communication Systems (VACS) that significantly improve vehicles' individual capabilities, have been considered in a new generation of traffic management tools [3], [4]. This trend continues with the development of high-automation or virtually driverless vehicles that are tested in real traffic conditions, see e.g., [5]. Recent studies have emphasized the advantages of employing linked vehicles, which could constitute the new normal in the not-too-far future. Soon, vehicles may communicate with each other and with the infrastructure; and drive automatically, based on their own sensors, communications, and appropriate movement control strategies. The majority of research involves the assumption that cars must conform to horizontal road lines that are designated by lanes. However, statistics show that switching lanes is a risky activity that causes a large portion of accidents nowadays.

Recently, the TrafficFluid concept was proposed [6], which is a novel paradigm for vehicular traffic, applicable at high levels of vehicle automation and communication. The TrafficFluid concept is based on two combined principles: (a) Lane-free traffic, whereby vehicles are not bound to fixed traffic lanes, as in conventional traffic, but may drive anywhere on the 2-D surface of the road; and (b) Vehicle nudging, whereby vehicles communicate their presence to other vehicles in front of them (or are sensed by them), and this may influence the movement of vehicles in front. Over the last couple of years, a number of movement strategies for automated vehicles on lane-free highways were developed, in accordance with the TrafficFluid paradigm, using different methodologies, such as: ad-hoc strategies [6], [7], optimal model predictive control [8], reinforcement learning [9], nonlinear feedback control [10]; and a generic simulation environment for lane-free traffic has also been developed [11]; see [12] for a brief review. Most of these strategies require availability of a desired vehicle orientation that determines the local vehicle movement direction if no collision-avoidance maneuver is required.

Roundabout is a key element in urban traffic which improves traffic efficiency in light traffic conditions [13]; but may become a bottleneck point in higher demands. Hence, efficient operation of roundabouts, which is indeed considered challenging because of the geometric complexities, can enhance traffic in its surrounding area. Several works in the literature focus on controlling automated vehicles on roundabouts, most of which addressing simple infrastructures. Specifically, a noticeable number of works consider single-lane roundabouts using various control approaches.

Some research [14]-[16] suggests priority management approaches, whereby a suitable policy, like "First-Come-First-Served", is utilized to assign the priorities to vehicles. If two vehicles have a conflict in the roundabout, the vehicle with a lower priority should stop or decrease its speed to let the higher-priority vehicle pass. Also, several presented methodologies [17]-[22] formulate the vehicles' movements, either for complete navigation or only for the merging part, on roundabouts as an optimal model predictive control problem aiming to minimize different criteria, including travel time, fuel consumption, and distance from the destination. Furthermore, a hierarchical structure is proposed in [23], to determine the optimum roundabout inflow and guarantee vehicles' safety.

Other works propose control approaches for two-lane roundabouts. In [24], two fuzzy controllers, designed based on real data, drivers' knowledge, and common reasoning, are employed to control steering angle and angular speed for a two-lane roundabout. Also, [25] - [27] combine optimal control and game theory to make decisions at the merging points or change lanes on the roundabout. An optimization embedded reinforcement learning method is suggested in [28] to make lane changing decisions at a four-lane roundabout.

A lane-free roundabout was first addressed in [29], where a comprehensive control strategy for vehicles on the basis of the bicycle model for vehicle dynamics was presented. A nonlinear controller, which had been developed in [10] and guarantees several features for straight lane-free roads, including collision and boundary-violation avoidance, desired speed tracking, and convergence of acceleration and orientation to zero, was modified to appropriately control vehicles in the roundabout. Since the modified controller requires a desired vehicle orientation, a heuristic approach to determine desired orientations proposed in [29] to be fed to the nonlinear controller. The approach was applied to the overly complex roundabout of Place Charles de Gaulle (Paris), which, due to its high complexity, is anyhow a lane-free road infrastructure even for today's conventional traffic.

This thesis aims to present a more transparent, systematic, and potentially more efficient way to determine the desired orientation of vehicles moving on lane-free roundabouts, based on the current vehicle location and its destination. For this goal, an optimal control problem is formulated and solved that minimizes a weighted summation of two criteria: (a) the trip distance to the destination, and (b) deviation from the circular angle. Regarding (b), it should be noted that, if all vehicle orientations are close to the circular angle, then they will be close to each other, something that mitigates the strength of any required collision-avoidance maneuvers. The defined problem is solved analytically for two extreme cases, namely the shortest path and the minimum deviation from the circular angle, respectively. For intermediate cases, it is difficult to find the solution analytically. Therefore, a backward Dijkstra algorithm is suggested to determine the optimal orientation in a discretized grid of the roundabout surface. The resulted desired orientations can be stored as an offline database, such that the vehicles can extract their current desired orientation based on their position and exit, while moving on the roundabout. The methodologies are applied to a specific, overly complex case study, the roundabout of Place Charles de Gaulle.

This thesis is composed of 5 chapters. The formulation of the optimum control problem, analytical solution of two extreme cases, the shortest path problem and the minimum deviation problem, are presented in Chapter 2. In Chapter 3, a Dynamic Programming-based approach is presented to solve the general problem in a discretized grid covering the surface of the roundabout. In Chapter 4, the case study is introduced and the results of applying the approaches are presented. Finally, conclusion remarks and future works are discussed in Chapter 5.

2 Optimal desired orientations

In this chapter, a systematic approach for determining the desired orientation at any point within a circular roundabout - individually for each exit branch - is provided. The problem of specifying desired orientations for the vehicle advancement is formulated as an optimal control problem that minimizes a cost function consisting of two weighted terms: (a) the trip distance to the destination; and (b) the deviation from circular angle. By connecting the desired orientations, a complete path from any point in the roundabout to any exit may be obtained.

It is important to highlight that the considered problem does not concern the actual vehicle's movement, but merely the desired orientations to be provided to the vehicle in order to generate its movement strategy. In other words, the specified orientations would coincide with the actual vehicle path, only if there are no other vehicles around that might cause it to deviate from the desired path to avoid collisions. For the same reason, the addressed problem does not involve vehicle acceleration or speed, as it aims at merely specifying optimal orientations and resulting paths, whereon the vehicle may drive at acceleration and speed specified by its movement strategy. Therefore, this approach does not need a vehicle model to be explicitly considered.

2.1 Optimal Control Problem

The optimal control problem may first be presented in a continuous framework. The vehicle's potential positions on the roundabout are represented by a radius r and an angle φ in polar coordinates, with the center of the roundabout being their origin. Since the ultimate goal is to determine the desired orientations without referring to the vehicle dynamics, φ (rather than the time t) is considered as the independent variable. Thus, the following state equation occurs:

$$\dot{r} := dr/d\varphi = u \tag{2-1}$$

where u is the radius change rate, which may be considered as the control signal. For example, if $u = 0$, then (2-1) states that the radius does not change, hence the vehicle orientation θ equals the circular angle, i.e., it coincides with the tangent of the circle with the current radius r; while the deviation s from the circular angle is zero. More generally, the formula $s = \theta - (\varphi + \pi/2)$ is used where $(\varphi + \pi/2)$ is the circular angle.

Furthermore, it is important to observe how the orientations of the vehicles alter as they move to an adjacent radius from their circular angle and how that affects collusions. More specifically, abrupt radius changes and large deviations are more likely to result in conflicts between vehicles. In order to avoid that, the deviations from the circular angle need to be restricted.

$$|s| < \bar{s} \tag{2-2}$$

where $\bar{s}$ is an upper limit for the deviation. Based on elementary geometric considerations, the relationship between the deviation s and the control signal u may also be derived as:

$$s = \tan^{-1}(-u / r) \tag{2-3}$$

The admissible state region for a circular roundabout is obviously $r \in [R_{\mathrm{in}}, R_{\mathrm{out}}]$, where R_{in} and R_{out} are the inner and outer roundabout radiuses, respectively. For every initial angle and admissible state (r_0, φ_0), the final angle φ_{e} is determined by the angle of the specific destination branch considered; while the final state, at the exit angle φ_{e} is, for all branches, $r(\varphi_{\mathrm{e}}) = R_{\mathrm{out}}$, since all branches are located at the outer radius of the roundabout. The control objective to be minimized is specified as:

$$J = \int_{\varphi_0}^{\varphi_{\mathrm{e}}} \left(\sqrt{u^2 + r^2} + w(u / r)^2 \right) d\varphi \tag{2-4}$$

where the first term reflects the trip distance from the origin to the exit point; the second term penalizes quadratically the deviation from the circular angle (see (2-2)) and w is the weighting coefficient that determines the relative importance of the two terms. Some control constraints may be added to the problem to suppress strong deviations from the circular angle. In conclusion, the optimal control problem reads as follows:

$$\begin{aligned} &\text{Minimize } J \\ &\text{subject to: } \dot{r} = u \\ &\qquad\qquad R_{\mathrm{in}} \le r \le R_{\mathrm{out}} \\ &\qquad\qquad r(\varphi_{\mathrm{e}}) = R_{\mathrm{out}} \\ &\qquad\qquad |\tan^{-1}(u / r)| < \bar{s} \end{aligned} \tag{2-5}$$

If the maximum-deviation constraint is disregarded, problem (4) can be analytically solved for two extreme cases: (a) the shortest path problem, i.e., for $w = 0$; and (b) the minimum deviation problem, i.e., for $w \to \infty$.

2.2 Extreme case 1: The shortest path problem

The shortest path has a clear physical meaning and can be readily derived. For better comprehension, two distinct cases are distinguished: visible and invisible destinations. The details for each destination type will be explained in the following parts.

2.2.1 Visible destination

In this case, where the origin and destination are relatively close to each other, the quickest and easiest way to get there is to take a direct path that is close to or on the outer roundabout boundary, without reaching the inner part of the roundabout. The geometry of the roundabout determines if a destination is visible from an origin or not.

If the straight line connecting a roundabout point, considered as the origin (r,φ), with the destination lies completely within the roundabout, then the destination is considered "visible"; and the shortest path obviously coincides with that straight line; while the slope of the line is the desired orientation at (r,φ). The visible area for an exit branch, grey-shaded in Figure 2-1, is described by:

$$V=\{(r,\varphi);R_{\mathrm{in}}\le r\le R_{\mathrm{out}},0\le\Delta\varphi\le\Delta\varphi_{\mathrm{vis}}(r)\} \tag{2-6}$$

where $\Delta\varphi\in[0,2\pi)$ is the vehicle angular distance from the exit point, and $\Delta\varphi_{\mathrm{vis}}(r)$ is a radius-dependent visibility threshold. The visible area is delineated upstream by the inner-circle tangent connected to the exit point, which is displayed with a light blue colour in Figure 2-1. Using trigonometric relationships, $\Delta\varphi_{\mathrm{vis}}(r)$ can be derived as below:

$$\Delta\varphi_{\mathrm{vis}}(r)=\cos^{-1}(R_{\mathrm{in}}/r)+\Delta\varphi_{\mathrm{vis}}(R_{\mathrm{in}}) \tag{2-7}$$

where:

$$\Delta\varphi_{\mathrm{vis}}(R_{\mathrm{in}})=\cos^{-1}(R_{\mathrm{in}}/R_{\mathrm{out}}) \tag{2-8}$$

2.2.2 Invisible destination

In this case, since the destination is not visible, the vehicle cannot go directly to the exit branch. That means that the vehicle will need to follow a circular motion around the roundabout until it reaches a position where the exit branch can be seen. Then, it will follow an almost direct path to the exit. The shortest path from a roundabout point (origin) (r,φ) to an invisible destination consists of three parts (Figure 2-2). The first part is the tangent of the inner circle that is connected to the origin, with touch point $(R_{\text{in}}, \varphi + \Delta\varphi_{\tan}(r))$ that can be determined through finding $\Delta\varphi_{\tan}(r)$ using the formula below:

$$\Delta\varphi_{\tan}(r) = \cos^{-1}(R_{\text{in}}/r) \tag{2-9}$$

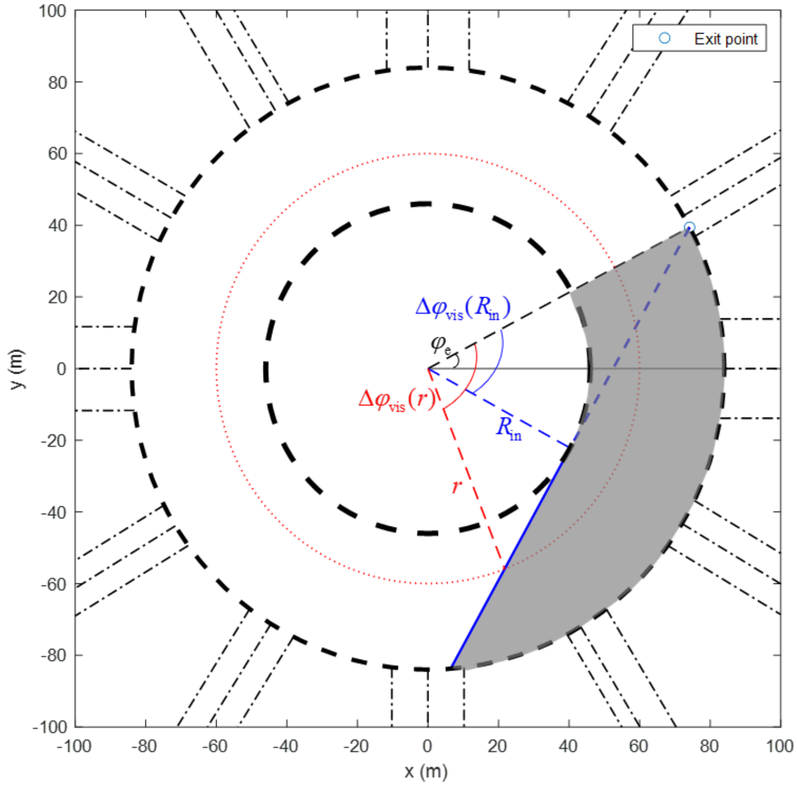

Figure 2-1- The visible area (grey-shaded) for a visible exit destination

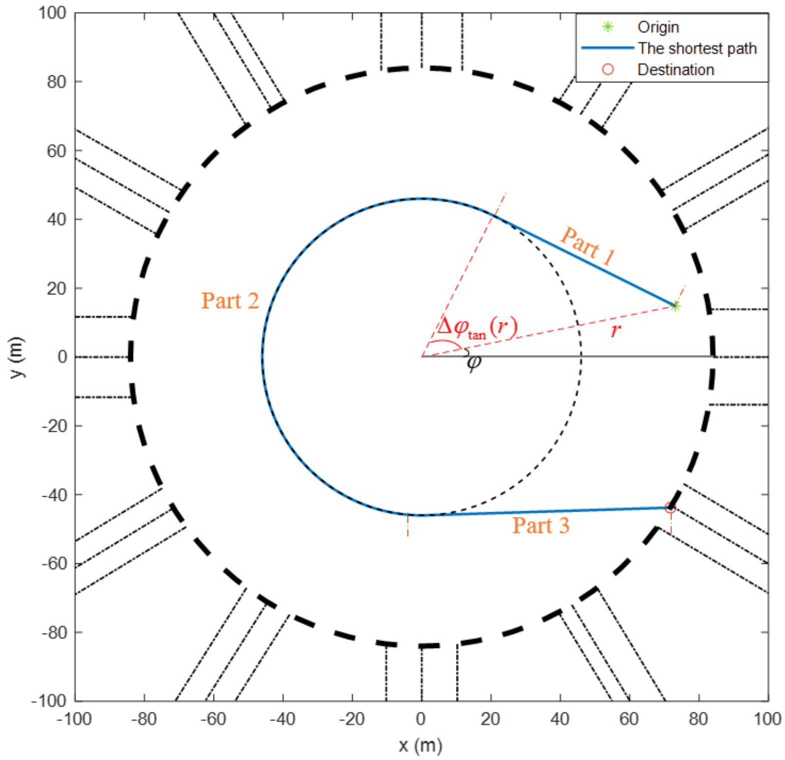

Figure 2-2 - The shortest path for an invisible destination

The desired orientation in this part is the slope of the tangent. In the second part, the path follows the inner boundary, i.e., the desired orientation is the circular angle, until the destination gets visible. After that, same as the case of visible destination, the desired orientation is the slope of a line connected to the exit point.

In conclusion, the desired orientation at every point on the roundabout (r,φ), with either visible or invisible destination $(R_{\text{out}},\varphi_{\text{e}})$, is:

$$\theta_{\text{d,SP}}(r,\varphi)=\begin{cases} \tan^{-1}\left(\dfrac{R_{\text{out}}\sin\varphi_{\text{e}}-r\sin\varphi}{R_{\text{out}}\cos\varphi_{\text{e}}-r\cos\varphi}\right) & 0\le\Delta\varphi\le\Delta\varphi_{\text{vis}}(r) \\ \varphi+\dfrac{\pi}{2} & \Delta\varphi>\Delta\varphi_{\text{vis}}(r) \ \& \ r=R_{\text{in}} \\ \tan^{-1}\left(\dfrac{R_{\text{in}}\sin(\varphi+\Delta\varphi_{\tan}(r))-r\sin\varphi}{R_{\text{in}}\cos(\varphi+\Delta\varphi_{\tan}(r))-r\cos\varphi}\right) & \text{otherwise} \end{cases} \tag{2-10}$$

where the first condition corresponds to the points in the visible area of an exit point, while the second and third conditions are applied when the destination is invisible. It should be noted that the desired orientation for the first and third conditions are calculated by transforming the points' positions to Cartesian coordinates and calculating the ratio between their difference in y coordinate (Δy) and their difference in x coordinate (Δx).

2.3 Extreme case 2: The Minimum deviation problem

By dropping the first term of the cost function, the solution minimizes the deviation from the circular angle. In this case, (2-5) can be written as below:

$$J = \frac{1}{2}\int_{\varphi_0}^{\varphi_e} z^2 d\varphi \qquad \text{(2-11)}$$
$$\text{subject to: } \dot{r} = -rz$$

where, for convenience, z is defined as $z := \tan(s)$ and treated as the control input. The Hamiltonian of this problem reads:

$$H = \frac{z^2}{2} - \lambda r z \qquad \text{(2-12)}$$

Considering Pontryagin's principle, the optimal solution should satisfy the following conditions:

$$\begin{cases} \dfrac{\partial H}{\partial z} = z - \lambda r = 0 \\ \dfrac{d\lambda}{d\varphi} = -\dfrac{\partial H}{\partial r} = \lambda^2 r \\ \dfrac{dr}{d\varphi} = \dfrac{\partial H}{\partial \lambda} = -\lambda r^2 \end{cases} \qquad \textit{(2-13)}$$

An intuitive solution is to have a constant deviation for the whole path, from origin to destination, i.e.:

$$z(\varphi) = c; \varphi_0 \le \varphi \le \varphi_{\mathrm{e}} \qquad \text{(2-14)}$$

This indeed satisfies the mentioned conditions in (2-13) and is the optimal solution of (2-11). To calculate the constant value, the following state equation is solved:

$$\dot{r} = -cr \Rightarrow r(\varphi) = r_0 \exp\left(-c(\varphi - \varphi_0)\right) \qquad \textit{(2-15)}$$

Then, by substituting the final condition, the constant value can be found as presented below:

$$\begin{aligned} &R_{\mathrm{out}} = r_0 \exp\left(-c(\varphi_e - \varphi_0)\right) \\ &\Rightarrow c = -\frac{\ln(R_{\mathrm{out}}) - \ln(r_0)}{\varphi_e - \varphi_0} \end{aligned} \qquad \textit{(2-16)}$$

Finally, the desired orientation for the minimum-deviation problem is:

$$\theta_{\mathrm{d,MD}}(r,\varphi) = \tan^{-1}\left(-\frac{\ln(R_{\mathrm{out}}) - \ln(r)}{\varphi_e - \varphi}\right) + \varphi + \frac{\pi}{2} \qquad \textit{(2-17)}$$

3 Determining the desired orientations using Backward Dijkstra algorithm

The solutions of the extreme cases, i.e., the shortest path and the minimum deviation problems which are presented in Chapter 2, may not be desirable due to sharp vehicle movements or uneven exploitation of the roundabout area. In fact, for the shortest path case, many vehicles may tend to move along the inner circle; while for the minimum deviation case, the outer boundary would be more crowded. Therefore, it is interesting to have a combination of these two cases by choosing a finite and non-zero weight w value in (2-4). However, finding the analytical solution in this general case is not straightforward. To overcome this issue, a Dynamic Programming-based approach, called backward Dijkstra algorithm, is proposed to compute a numerical closed-loop solution for the space-discretized problem. In a discretized grid for a given exit node, the algorithm finds the best orientation such that a cost function will be minimized.

3.1 Network Discretization

Firstly, a network of points, representing possible vehicle positions, will be created that covers the entire surface of the roundabout. One, or even more, of them may be considered as the exit point(s) that can be shown by:

$$(r,\varphi) = (R_{\mathrm{out}}, \varphi_{\mathrm{e}}) \tag{3-1}$$

To implement a Dynamic Programming-based method, the roundabout's surface has to be discretized to form a grid of nodes, with resolution defined by a selectable radius step size Δr and angle step size $\Delta\varphi$. The angle step size should be sufficiently small so that at least one node is located at each (entrance and exit) radial branch. Obviously, the higher the density, the higher computational effort required, which is not a considerable problem for an offline approach. For the edges (transitions) between the nodes of the grid, it is reasonable to assume that vehicle paths on the roundabout point only forward, in discrete $\Delta\varphi$ steps.

On the roundabout, vehicles move in a counter-clockwise direction and navigate through the roundabout in angle steps. A vehicle at the current position (r,φ) will select the optimal radius in the following angle to move towards, among a certain number of alternatives. The following radius must fall within the roundabout's inner and outer radii in order to qualify as the next radius; otherwise, it will not be considered. For each transition, from the current angle step to the next one, the following apply:

$$\begin{aligned}\varphi(k+1) &= \varphi(k) + \Delta\varphi \\ r(k+1) &= r(k) + q\Delta r\end{aligned} \tag{3-2}$$

where $q \in \{\underline{q}, \cdots, -1, 0, 1, \cdots, \overline{q}\}$ reflects corresponding edges (transitions) to next-step nodes in the grid, with corresponding orientations; and $\underline{q}$ and $\overline{q}$ are lower and upper limits, respectively, for the admissible orientations, similarly to $\overline{s}$ in (2-5) and k is the discrete angle step forward. It should be noted that k does not refer to a time step; it is angle step, instead.

Clearly, the allowable range of radiuses is $r \in [R_{\text{in}}, R_{\text{out}}]$, and any transitions leading out of the roundabout are suppressed while constructing the grid. For instance, for nodes on the outer boundary, it is not allowed to select a bigger radius, i.e., $q > 0$, otherwise outer-boundary violation would occur. In Figure 3-1, a very low-density grid, with $\Delta r = 9.5$ and $\Delta\varphi = \pi/8$, is portrayed. Since there are only 5 radiuses in the roundabout, changing radius results in a large deviation from the circular angle. Therefore, they can change their radius up to one step as shown in Figure 3-2.

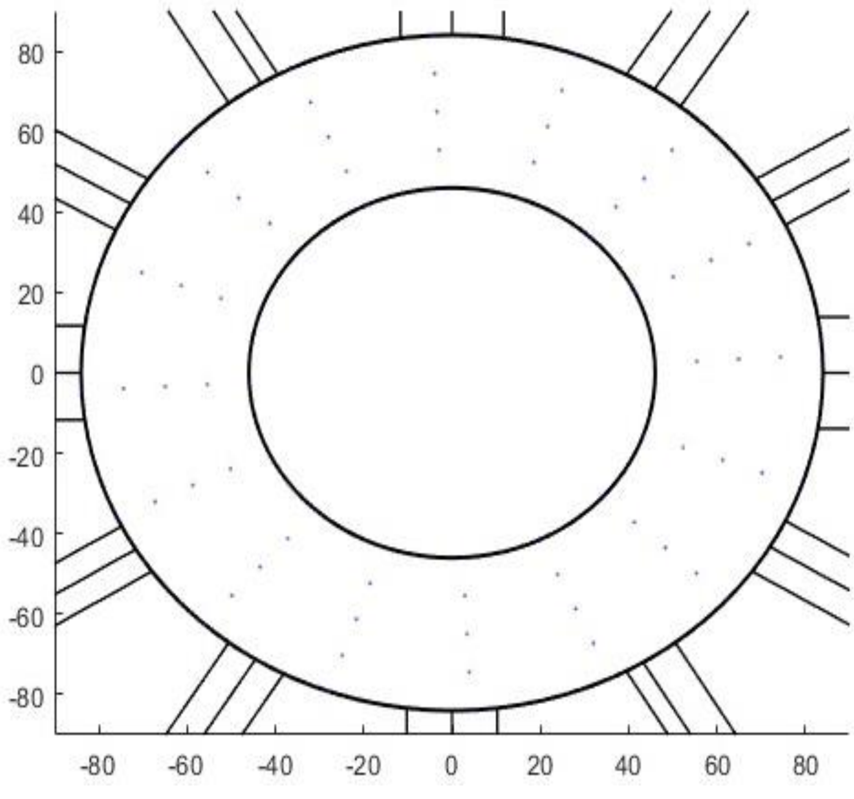

Figure 3-1 - A low-density grid

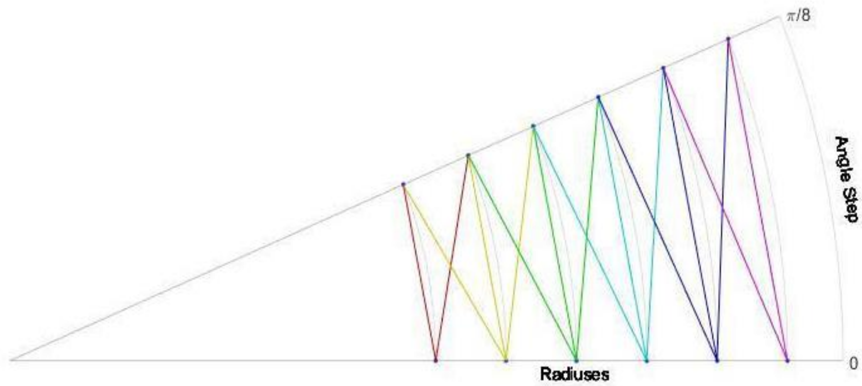

Figure 3-2 - Possible transitions graph- $\underline{q}=-1$ *and* $\overline{q}=1$

A denser network is shown in Figure 3-3, where $\Delta r = 3.8$ and $\Delta\varphi = \pi/16$. Furthermore, in this grid, a vehicle is allowed to change its radius up to 2 steps, as can be seen in Figure 3-4.

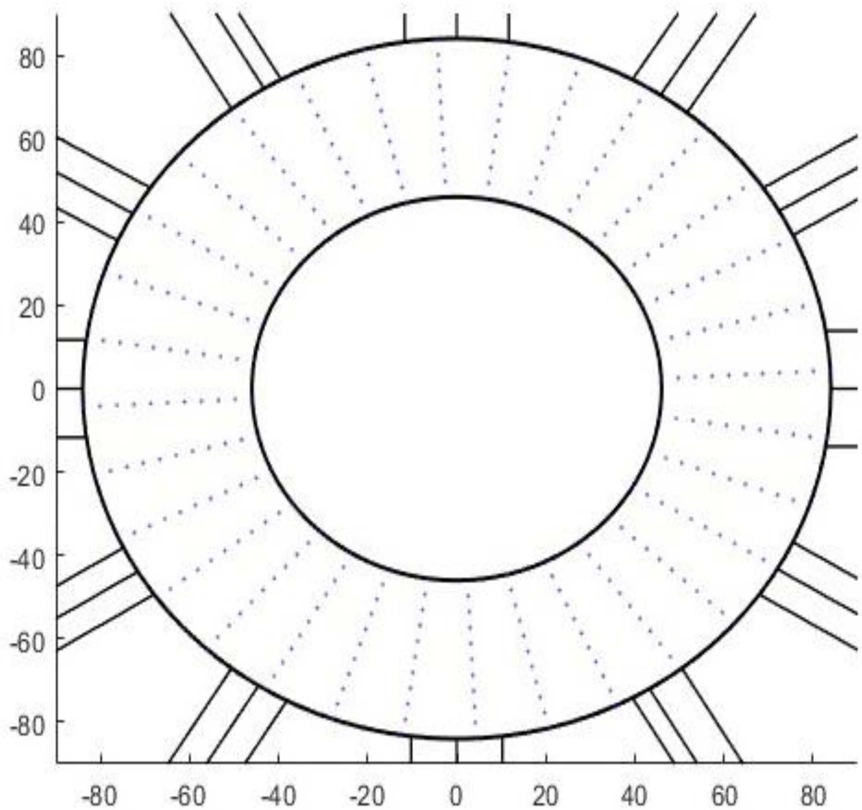

Figure 3-3 - A denser grid

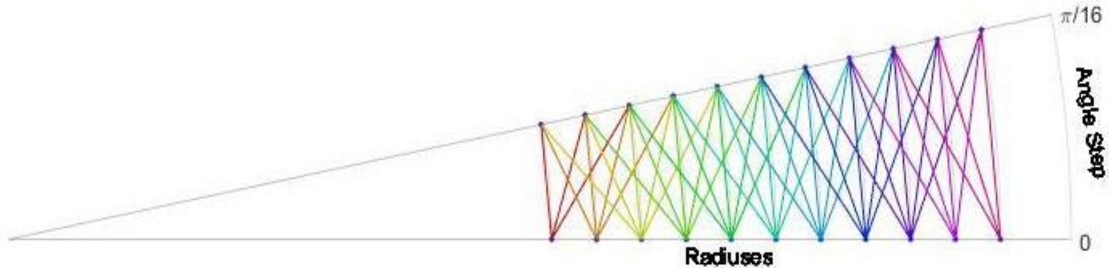

Figure 3-4 - Possible transitions graph- $\underline{q}=-2$ and $\overline{q}=2$

3.2 Defining the Criterion

Following the cost function introduced in Chapter 2, the employed criterion for transitions between two nodes in the discrete grid, where the radius changes by q steps, is defined as:

$$J(k) = d_{r,q}(k) + w\tan^2(s_{r,q}(k)) \quad (3\text{-}3)$$

where $d_{r,q}(k)$ and $s_{r,q}(k)$ are the transition distance and deviation from the circular angle, respectively, if the vehicle decides to change its radius by q steps. These terms are calculated as presented below.

3.2.1 Distance and deviation calculation

As mentioned, in order to make the optimal decision using the backward Dijkstra algorithm, it is needed to calculate the distance between the vehicle's current and next potential positions. The Euclidian distances for all one-step transitions are calculated using the following formula:

$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \quad (3\text{-}4)$$

where (x_1, y_1) and (x_2, y_2) represent the current and next position of a vehicle. To express the distance based on (r, φ), the following relations between Cartesian and polar coordinates variables are employed:

$$\begin{aligned} x &= r \cdot \cos(\varphi) \\ y &= r \cdot \sin(\varphi) \end{aligned} \quad (3\text{-}5)$$

Additionally, the following equations may be used to get polar coordinates based on Cartesian variables:

$$r = \sqrt{x^2 + y^2}$$

$$\varphi = \begin{cases} \tan^{-1}\left(\frac{y}{x}\right) & ,if \ \ x > 0 \\ \tan^{-1}\left(\frac{y}{x}\right) - \pi & ,if \ \ x < 0 \ \ and \ \ y < 0 \\ \tan^{-1}\left(\frac{y}{x}\right) + \pi & ,if \ \ x < 0 \ \ and \ \ y > 0 \\ \frac{\pi}{2} & ,if \ \ x = 0 \ \ and \ \ y > 0 \\ -\frac{\pi}{2} & ,if \ \ x = 0 \ \ and \ \ y < 0 \end{cases} \tag{3-6}$$

and $\varphi = 0$ corresponds to the positive axis of *x*-coordinate, and it increases counterclockwise. Using polar coordinates, the formula becomes:

$$d = \sqrt{r_1^{\,2} + r_2^{\,2} - 2r_1 \cdot r_2 \cdot \cos(\varphi_1 - \varphi_2)} \tag{3-7}$$

To prevent extra distance computations, the symmetry of the roundabout and the grid is now taken into consideration. More specifically, if a vehicle is located at (r, φ) and wishes to continue onto the next step, it will travel in accordance with this, at an angle $\varphi + \Delta\varphi$ and on a radius of $r + q\Delta r$. As a result, the following distance describes any vehicle with the current radius r and *q*-step transition:

$$\begin{aligned} d_{r,q} &= \sqrt{r^2 + (r + q\Delta r)^2 - 2r \cdot (r + q\Delta r) \cdot \cos((\varphi + \Delta\varphi) - \varphi)} \Rightarrow \\ d_{r,q} &= \sqrt{r^2 + (r + q\Delta r)^2 - 2r \cdot (r + q\Delta r) \cdot \cos(\Delta\varphi)} \Rightarrow \\ d_{r,q} &= \sqrt{2r^2 + 2r \cdot q\Delta r + q^2\Delta r^2 - 2r^2 \cdot \cos(\Delta\varphi) - 2r \cdot q\Delta r \cdot \cos(\Delta\varphi)} \Rightarrow \\ d_{r,q} &= \sqrt{2(r^2 + r \cdot q\Delta r) \cdot (1 - \cos(\Delta\varphi)) + q^2\Delta r^2} \end{aligned} \tag{3-8}$$

Additionally, using trigonometric relationships, the deviation from the circular angle can be obtained by (see Figure 3-5):

$$s_{r,q} = \tan^{-1}\left(\frac{-q\Delta r}{r\Delta\varphi}\right) \tag{3-9}$$

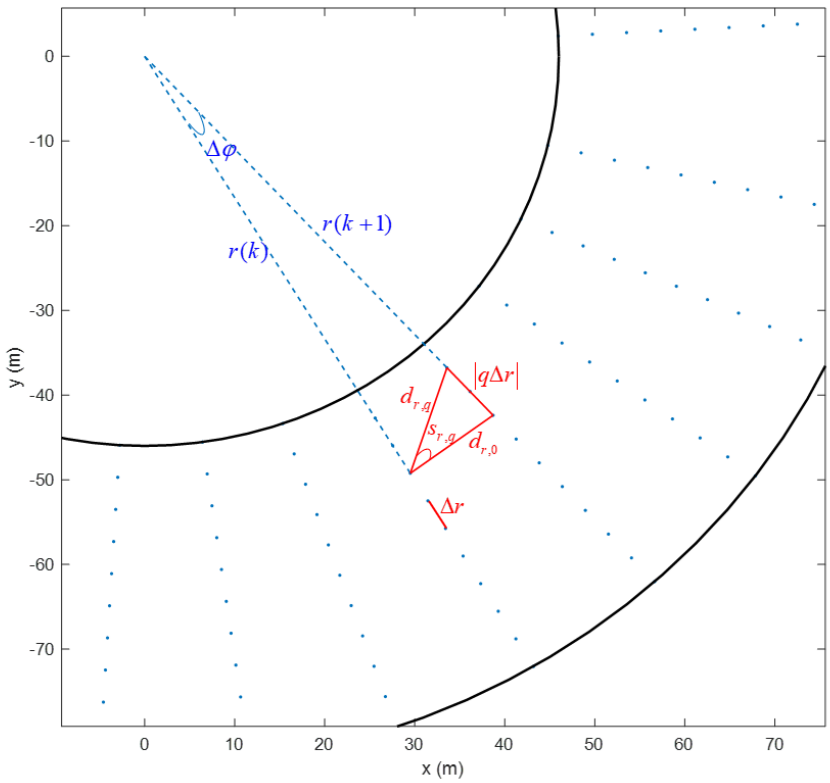

Figure 3-5 - Deviation calculation

3.2.2 Calculation of upper and lower limits for transition steps

Additionally, a vehicle at a certain point in the roundabout's surface, has a limited number of alternatives to change its radius. To determine the admissible steps that a vehicle can change, the maximum deviation may be replaced in (3-10). Then, $\overline{q} = -\underline{q}$ can be calculated as below:

$$\tan(\overline{s}) = \left(\frac{-q\Delta r}{d_{r,0}}\right)$$
$$\overline{q} = \left\lfloor \frac{d_{r,0}\tan(\overline{s})}{\Delta r} \right\rfloor = \left\lfloor \frac{2r^2(1-\cos\Delta\varphi)\tan(\overline{s})}{\Delta r} \right\rfloor \tag{3-10}$$

It should be noted that a vehicle on the inner boundary i.e., its radius equals R_{in}, can only choose its transition among $q \in \{0,1,\cdots,\overline{q}\}$. On the other hand, the options for a vehicle navigating along the outer boundary are $q \in \{\underline{q},\cdots,-1,0\}$.

3.3 Dijkstra's Algorithm

Dijkstra's algorithm [30] is an algorithm used for finding the shortest paths between a source node and every other node in a network for a given source node. The algorithm may also be used to identify the shortest routes from a single source node to a single destination node by terminating it once the shortest route has been identified. Network routing protocols are a prominent example of how shortest-path algorithms, in general, and in this case, are employed.

The algorithm exists in many variants. Dijkstra's original algorithm calculates the shortest path between two given nodes or given a single node as the "source" node, finds the shortest paths from the source to all other nodes in the graph, thus producing the shortest path tree. However, in this thesis, the goal is to find the optimal orientation (transition) at each discrete point (node) of the roundabout grid towards a specific destination point. To this end, a modified version of the Dijkstra algorithm in a Dynamic Programming-based way is introduced, whereby starting from the destination point and moving backwards iteratively, the optimal transition for all nodes of the roundabout grid is determined. The algorithm determines the optimal orientations at each node, such that the summation of the defined cost criterion from any origin to the exit point is minimized, i.e., the algorithm delivers a (discrete) closed-loop solution.

This algorithm doesn't attempt any direct "exploration" in the direction of the goal. Instead, the subsequent intersection's distance (in this case criterion) from the starting point is the only factor taken into account. As a consequence, until it reaches the destination, this algorithm advances outward from the starting point, interactively taking into account every node that is closer in terms of minimizing the criterion. When seen in this light, it becomes evident why the algorithm must always choose the minimum. It could also highlight one of the algorithm's flaws, namely that it operates quite slowly in some topologies.

3.3.1 Backwards Dijkstra's Algorithm (Single Exit Point)

In this case, a "destination" node will be specified, which represents an exit branch of the roundabout, and a backward search will be implemented. This modification is inspired by the backward DP method and it produces an optimal orientation towards the specific destination for every roundabout discrete point. There may be an initial node (with all its adjacent link arrows pointing out of it), however, it is not necessary. There is one single final (destination) node with all its adjacent link arrows pointing towards it. The algorithm steps are the following:

1. Mark all nodes unvisited. Create a set of all the unvisited nodes called the unvisited set.
2. Assign to every node a tentative cost function value: set it to zero for the final node and to infinity for all other nodes. The tentative cost function of a node v is the summation of the best path, corresponding to the minimum cost function, discovered so far between the node v and the final node. Since initially no path is known from any other vertex than the sink

itself (which is a path of cost zero), all other tentative distances are initially set to infinity. Set the final node as current.

3. For the current node, consider all of its unvisited neighbors and calculate their tentative cost function values through the current node. Compare the newly calculated tentative cost function to the one currently assigned to the neighbor and assign it the smaller one, along with the corresponding optimal direction. For example, if the current node *A* is marked with a cost function of 6, and the edge connecting it with a neighbor *B* has cost function 2, then the total cost function to *B* through *A* will be 6 + 2 = 8. If *B* was previously marked with a cost function greater than 8 then change it to 8; and change the optimal direction to point towards *A*. Otherwise, the current value and optimal direction will be kept.
4. When done considering all of the unvisited neighbors of the current node, mark the current node as visited and remove it from the unvisited set. A visited node will never be checked again.
5. If the initial node (if any) has been marked visited (when planning a route between two specific nodes) or if the smallest tentative cost function among the nodes in the unvisited set is infinity (when planning a complete traversal; occurs when there is no connection between the initial node and remaining unvisited nodes) or if all nodes have been marked visited, then stop. The algorithm has finished.
6. Otherwise, select the unvisited node that is marked with the smallest tentative distance, set it as the new current node, and go back to step 3.

3.3.3 Backwards Dijkstra's Algorithm (Multiple Exit Points)

If the branches are sufficiently wide, more than one exit point can be considered in a branch. This may lead to expediting transit times, preventing collisions, and better management of vehicle traffic at the exits, since it allows vehicles to fully utilize the exit branch width rather than being forced to exit from a single point. In this case, more than one points are considered as potential exit points with all their adjacent link arrows pointing towards them. There may be an initial node (with all its adjacent link arrows pointing out of it), however, it is not necessary. The algorithm steps are the following:

1. Mark all nodes unvisited. Create a set of all the unvisited nodes called the unvisited set.
2. Assign to every node a tentative cost value: set it to zero for the respective final cost for our final nodes and to infinity for all other nodes. The tentative cost of a node v is the cost of the shortest path discovered so far between the node v and any of the final nodes. Since initially no path is known from any vertex other than the final nodes (which are paths of cost zero), all other tentative costs are initially set to infinity. Set the final node with the smallest final cost as current.
3. For the current node, consider all of its unvisited neighbors and calculate their tentative costs through the current node. Compare the newly calculated tentative cost to the one currently assigned to the neighbor and assign it the smaller one, along with the

corresponding optimal direction. For example, if the current node A is marked with a cost of 6, and the edge connecting it with a neighbor B has a cost equal to 2, then the cost to B through A will be 6 + 2 = 8. If B was previously marked with a cost greater than 8 then change it to 8; and change the optimal direction to point towards A. Otherwise, the current value and optimal direction will be kept.

4. When all of the unvisited neighbors of the current node are considered, mark the current node as visited and remove it from the unvisited set. A visited node will never be checked again (this is valid and optimal in connection with the behavior in step 6.: that the next nodes to visit will always be in the order of 'smallest cost to final node first' so any visits after would have a greater cost).
5. If the initial node (if any) has been marked visited (when planning a route between the initial node and any of the final nodes) or if the smallest tentative cost among the nodes in the unvisited set is infinity (when planning a complete traversal; occurs when there is no connection between the remaining unvisited nodes and any final node) or if all nodes have been marked visited, then stop. The algorithm has finished.
6. Otherwise, select the unvisited node that is marked with the smallest tentative cost, set it as the new current node, and go back to step 3.

3.3.4 Real-time implementation

Due to the high computational effort of the numerical solution via the backward Dijkstra algorithm, specifically for a dense grid, it is not possible to run it online. Hence, a real-time scheme has to be implemented to determine the optimal orientation for each vehicle based on its current position and exit branch.

This may be achieved by storing the desired orientations for each roundabout location and exit branch, calculated offline by the backward Dijkstra algorithm, as a database which is accessible for the vehicles moving on the roundabout. At each location in the roundabout, a vehicle, depending on its exit branch, recalls the optimal desired orientation for its current location from the corresponding table.

3.4 Sub-Optimal Online Approach

In the light of the optimal results of the previous section, an alternative, sub-optimal method with negligible online computational effort may be proposed, which uses the optimal orientations related to the mentioned extreme cases that can be computed online very fast. Specifically, this can be achieved by finding the desired orientation at any location and calculating the weighted average of the orientations resulted from the shortest path and the minimum deviation cases as below:

$$\hat{\theta}_d(r,\varphi) = \alpha\theta_{\mathrm{d,SP}}(r,\varphi) + (1-\alpha)\theta_{\mathrm{d,MD}}(r,\varphi) \tag{3-11}$$

where $\theta_{\mathrm{d,SP}}(r,\varphi)$ and $\theta_{\mathrm{d,MD}}(r,\varphi)$ are the respective desired orientations corresponding to the shortest path (2-10) and the minimum deviation problems (2-17). Moreover, $0 \le \alpha \le 1$ is a selectable parameter. Note that the orientations derived from (3-11) combine the outcome of the two respective extreme problems, which is different than combining the two criteria as in the numerically solved general problem.

If desired, one may offline optimize α such that (3-11) yields an orientation close, as much as possible, to the result of the backward Dijkstra algorithm for a specific weight w. Specifically, α can be found by employing a Least Square (LS) approach, where (3-11) is employed to construct a regression equation as below:

$$\boldsymbol{\theta}_{\mathrm{d}} - \boldsymbol{\theta}_{\mathrm{d,MD}} = \alpha(\boldsymbol{\theta}_{\mathrm{d,SP}} - \boldsymbol{\theta}_{\mathrm{d,MD}}) \tag{3-12}$$

where $\boldsymbol{\theta}_{\mathrm{d}}$ is a vector containing the desired orientation, determined by the backward Dijkstra algorithm, at all points of the roundabout grid; while $\boldsymbol{\theta}_{\mathrm{d,SP}}$ and $\boldsymbol{\theta}_{\mathrm{d,MD}}$ contain the shortest path and the minimum deviation orientations, respectively, at those points. Then, α can be calculated by the LS solution:

$$\alpha = \left((\boldsymbol{\theta}_{\mathrm{d,SP}} - \boldsymbol{\theta}_{\mathrm{d,MD}})^T(\boldsymbol{\theta}_{\mathrm{d,SP}} - \boldsymbol{\theta}_{\mathrm{d,MD}})\right)^{-1}(\boldsymbol{\theta}_{\mathrm{d,SP}} - \boldsymbol{\theta}_{\mathrm{d,MD}})^T(\boldsymbol{\theta}_{\mathrm{d}} - \boldsymbol{\theta}_{\mathrm{d,MD}}) \tag{3-13}$$

The easiness of producing desired orientations with this approach offers an additional advantage, namely the possibility to modify in real time the value of α, and hence the produced orientations, in dependence of the current traffic conditions in the roundabout. Specifically, if the traffic density in the roundabout is low, vehicle conflicts are accordingly few, hence it may be preferrable to tend towards shortest paths (α small) to save trip time and fuel consumption. In contrast, if the traffic density in the roundabout is high, vehicle conflicts are accordingly frequent, hence it may be preferrable to tend towards minimum-deviation paths (α big) to mitigate the required vehicle maneuver intensity.

4 Results

In this section, the results of employing the suggested approaches in the previous chapters to determine the desired orientation in our case study, the roundabout of Place Charles de Gaulle, are presented. First, the structure of the roundabout is introduced, and then the results of analytical solutions, backward Dijkstra algorithm, and sub-optimal approach are presented.

4.1 The Place Charles de Gaulle roundabout

The Place Charles de Gaulle, traditionally known as the Place de l' Étoile, is a significant roundabout located in Paris, France, which is shown in Figure 4-1. It is the meeting place of twelve roadways (hence its original name, which translates to "Square of the Star"), including the Champs-Élysées, which is marked as Branch 1 in Figure 4-2. The Arc de Triomphe, which is located in the middle of Place Charles de Gaulle, is traversed by the Axe historique (or "historical axis") of Paris [31].

The Place Charles de Gaulle roundabout is considered an overly complex case due to the absence of lanes. It has an outer radius of 84 m and an inner of 46 m, respectively and comprises 12 bidirectional branches which results in 144 different Origin- Destination (OD) pairs. One may consider a different sub-problem for each OD or address them in groups considering one problem per destination which contains all results of the related (12 D problems).

Figure 4-1 - The Place Charles de Gaulle [32]

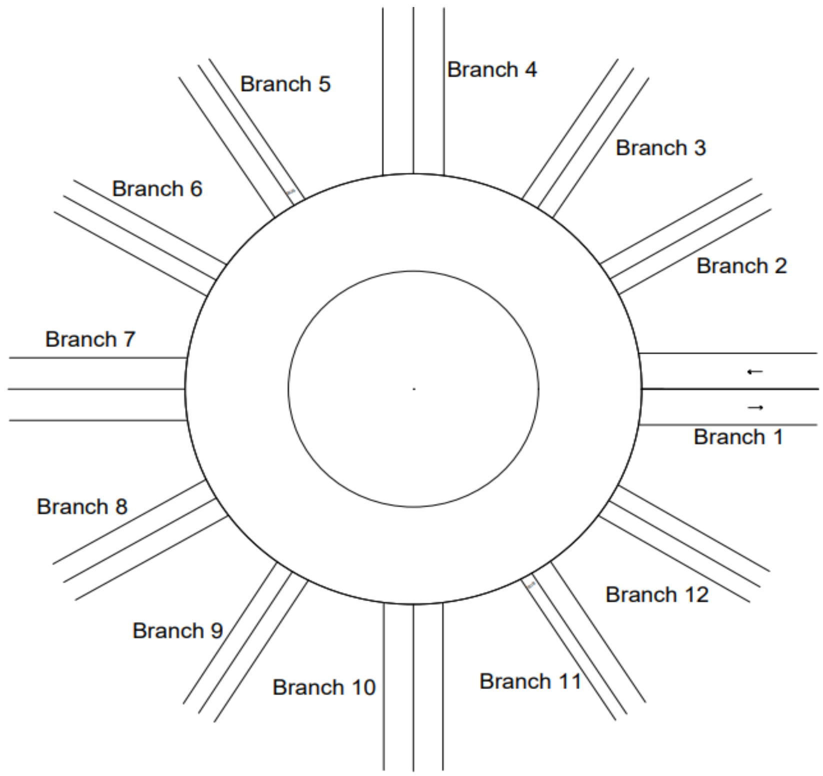

Figure 4-2 - Roundabout's Branches and Structure

4.2 Analytical solution for determining desired orientations in the extreme cases

As mentioned in Chapter 2, the defined problem is solved analytically for two extreme cases: the shortest path and the minimum deviation problems. Despite the infrastructure complexity, it is easy to generate the desired orientations for the two extreme cases in the continuous framework. However, to obtain clear and understandable outputs, the desired orientation is shown in a discretized grid, which is also used to display the result of the backward Dijkstra algorithm.

Figure 4-3 illustrates the results of the shortest path problem. Since the deviation limit is not taken into account to find the analytical shortest path, big deviations may be noticed. In particular, it can be observed that at points from which the destination is not visible, vehicles head directly to the inner circle via a tangent line that cause some sharp steering such that the deviation form circular angle may even reach 60°. However, one may saturate the orientations accordingly so that the desired threshold is satisfied. Vehicles continue to navigate on the inner circle until reaching the area where the destination is visible. At that point, they will attempt to follow a direct route to the destination. It is worth mentioning that the figure shows the orientations for one exit branch; if the orientations for all branches were plotted, many conflicts would be visible among vehicles with different destinations. Moreover, if many vehicles try to reach the inner boundary, the inner part of the roundabout would be much more crowded that the outer area.

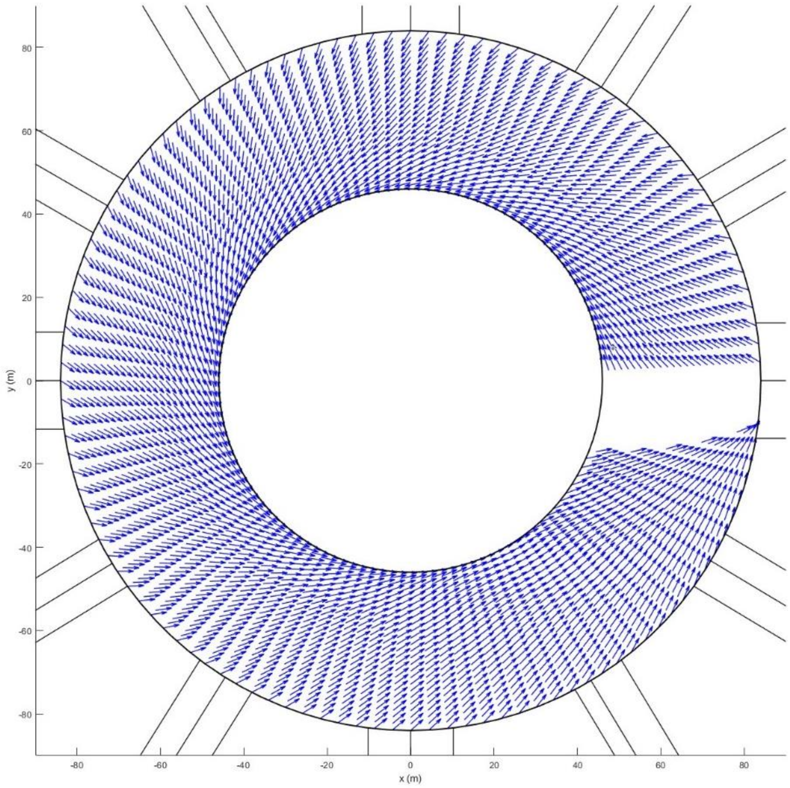

Figure 4-3 - Analytical solution of the shortest path problem

On the other hand, Figure 4-4 illustrates the desired orientations for the case of minimum deviation, where vehicles have a gradual deviation from their origin to the exit point. In particular, the desired orientation at points which are very far from the exit points have minor differences from the circular motions especially if they are close to the outer boundary, in which case their motion appears circular. It should be noted that only vehicles located on the outer boundary have completely circular movement. Obviously, smooth motions cause less conflicts among vehicles exiting from various branches. The weak point of employing these results is that all entering vehicles will move along the outer boundary while the inner part will not be taken into advantage.

4.3 Backward Dijkstra algorithm results for a single exit point

Finding the analytical solution in this general case is not straightforward. For intermediate scenarios, where a compromise between shortest path and minimum deviation from the circular angle is needed, it is difficult to find the solution analytically. Therefore, the backward Dijkstra algorithm is suggested to determine the optimal orientations in a discretized grid of the roundabout surface.

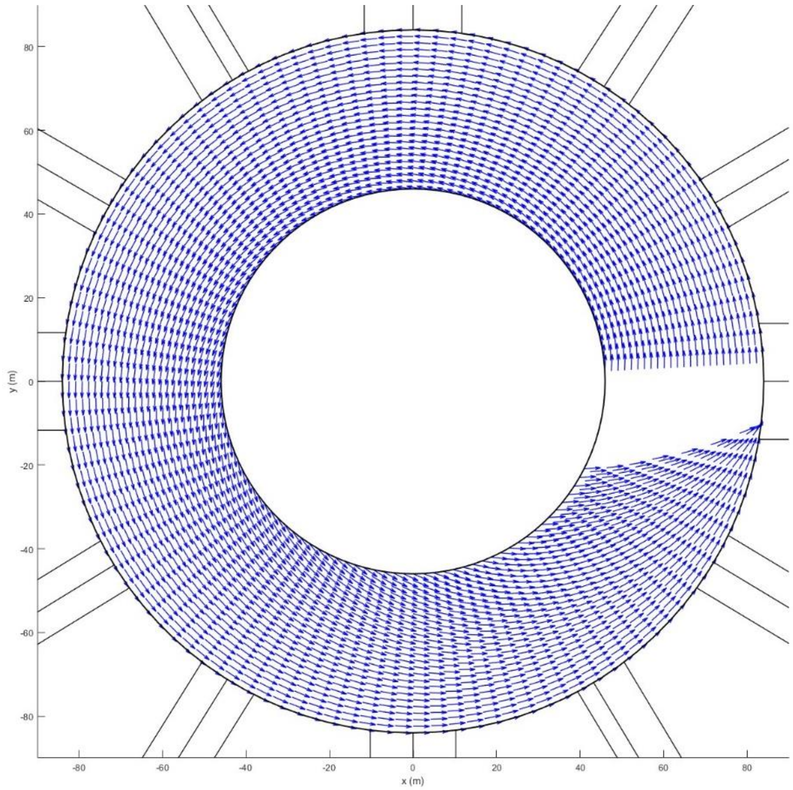

Figure 4-4 - Analytical solution of the minimum deviation problem

In this section, the roundabout surface is discretized by $\Delta r = 1.52m$ and $\Delta\varphi = 3^{\circ}$, which leads to a total of 2925 grid nodes. Also, the maximum admissible deviation is 15°, which leads to a maximum of 2 possible transitions at each node and 13923 possible segments between nodes. This setting is chosen to generate visually understandable results, while a higher resolution can be selected for the real-time implementation. The generated grid for this setting is shown in Figure 4-5, which is indeed a complicated graph.

All presented results concern one specific exit branch and the similar results can be obtained for other branches. The computation time to run the backward Dijkstra algorithm for this big roundabout (for one destination) amounts to approximately 78 seconds on an Intel(R) Core (TM) i7-5960X CPU @ 3.00GHz with 16.0 GB of installed RAM.

In the following figures, the optimal routes are portrayed for different weight values in the employed criterion, showing the different orientations a vehicle can base its movements on the roundabout, in order to exit from Branch 1. The optimal orientations for each node of the grid towards a single exit point can be seen in Figure 4-6 to Figure 4-9. The blank spaces in the figures indicate nodes that cannot be feasibly connected to the destination, as this would violate the maximum-deviation limit.

Figure 4-6 illustrates the results of the backward Dijkstra algorithm for the shortest path case, with $w = 0$. It can be observed that at points from which the destination is not visible, vehicles try to reach the inner circle as fast as possible, provided that their deviation from the circular angle does not violate the defied threshold. If a higher threshold for deviation from the circular angle is considered, the results will be more identical to the analytical. In the visible area, vehicles will attempt to navigate to the destination by taking the most direct route possible.

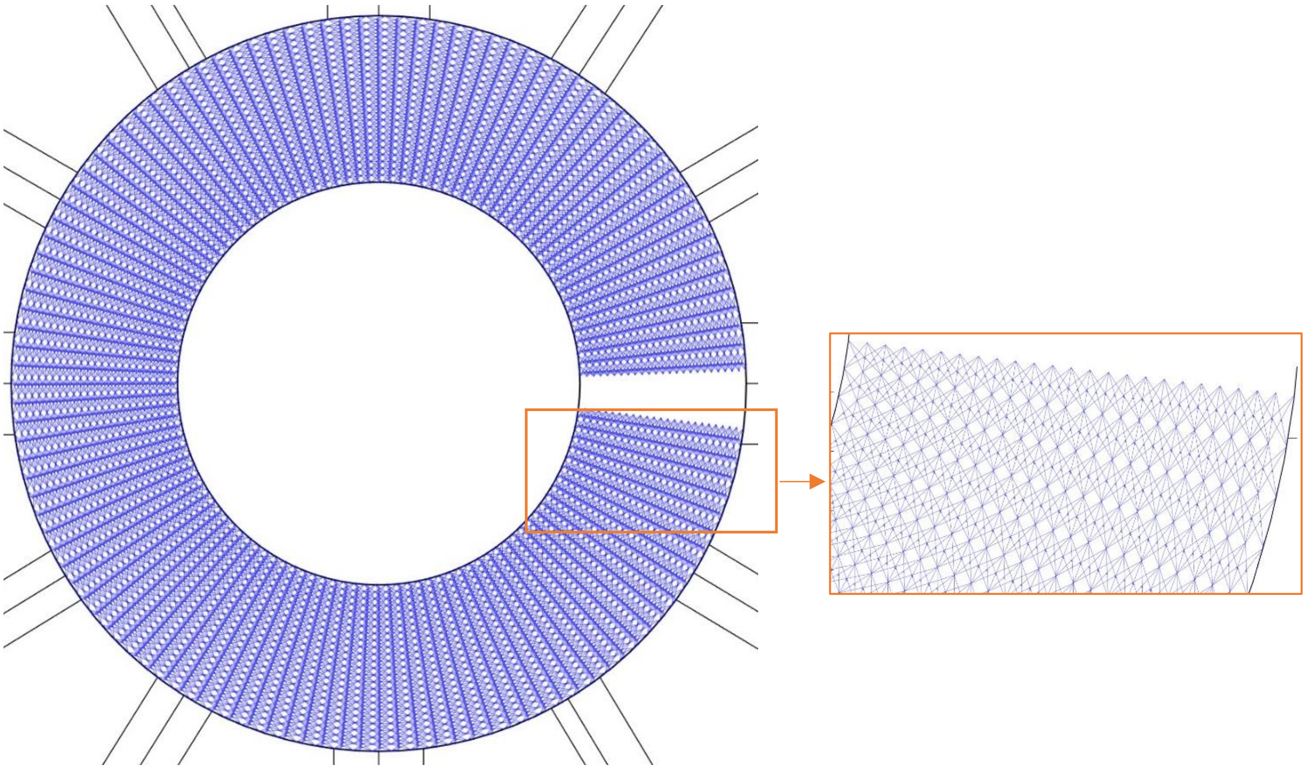

Figure 4-5 – The generated grid with the given setting

The result of the backward Dijkstra algorithm for $w = 10$ is depicted in Figure 4-7 where vehicles at very far origins have a tendency to reach the inner boundary. On the other hand, in closer origins, they do not get very close to the inner boundary. Consequently, the traffic will be more evenly distributed in the whole roundabout area.

Additionally, Figure 4-8 shows the results of the backward Dijkstra algorithm, with $w = 20$. As compared with Figure 4-7, less vehicles tend to reach inner boundary which led to less crowded inner parts while more vehicles will move in the outer parts.

Correspondingly, Figure 4-9 illustrates the desired orientations for the case of minimum deviation, i.e., $w \to \infty$, where vehicles mostly have circular motions and get close to the exit branch smoothly. If the origin is located on the outer boundary, e.g., when a vehicle enters the roundabout, the vehicle will move along the outer boundary. This case, axiomatically, leads to a highly crowded outer boundary while the inner parts are left unused.

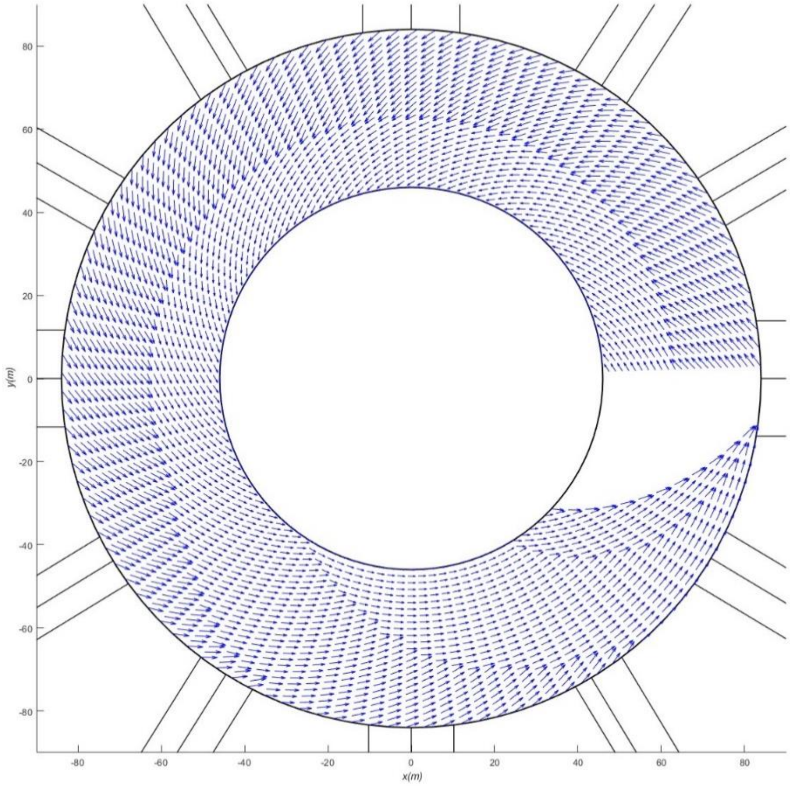

Figure 4-6 –Optimal Desired Orientations- w=0

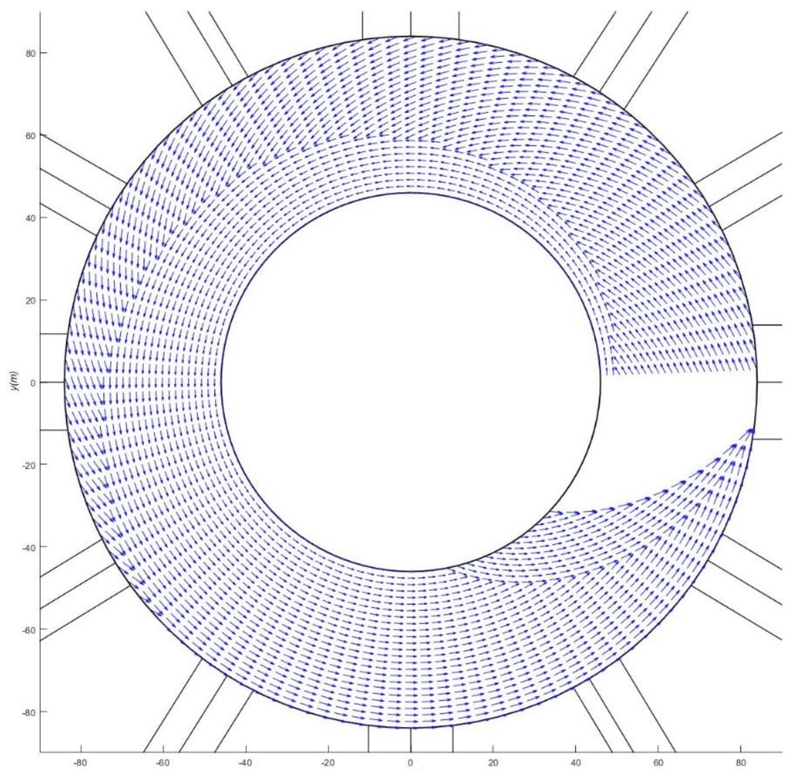

Figure 4-7 – Optimal Desired Orientations- w=10

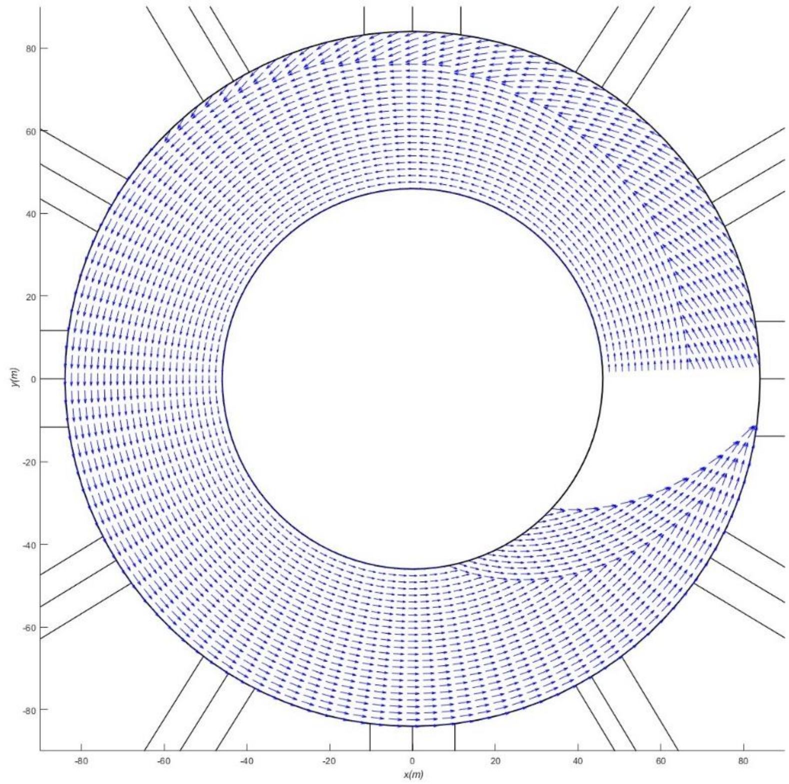

Figure 4-8 – Optimal Desired Orientations- w=20

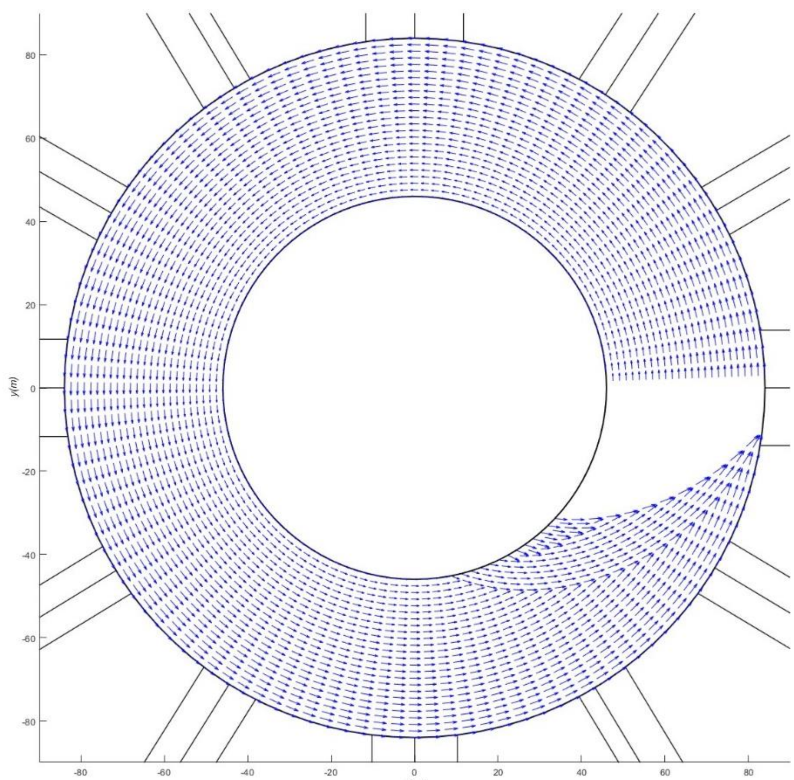

Figure 4-9 – Optimal Desired Orientations- $w \rightarrow inf$

Note that, in contract of the analytical solution, a vehicle has a limited number of choices for the next angle step that results in a few deviation angles. Therefore, the extremely smooth results seen in Figure 4-4 cannot be obtained using the backward Dijkstra algorithm unless the roundabout is discretized with a significantly high resolution. In fact, with smaller Δr and $\Delta\varphi$, the results become more identical to the analytical responses. That is why in the given network the orientation is completely circular, for most parts of the roundabout, and they deviate from the circular angle only when they are sufficiently close to the exit point.

4.4 Optimal paths

To achieve clearer results, different origin-destination pairs are chosen for which the optimal path determined by the backward Dijkstra algorithm are depicted. More specifically, considering Branches 4 and 10 as origins in the visible and invisible areas, respectively, and Branch 1 as the constant destination, the optimal paths for a vehicle starting from different points on the roundabout's surface are shown, in Figure 4-10 to Figure 4-15. However, the similar results can be drawn for any other destination. A higher resolution grid ($\Delta r = 0.38m$ and same $\Delta\varphi$ and allowable deviation) has been employed in this case compared to the previous results to achieve smoother paths.

For the visible destination case (Figure 4-10 - Figure 4-12), it can be seen that for $w = 0$, which depicts the shortest path, the created paths navigate quasi-linearly to the exit point. In contrast, when $w \to \infty$, which illustrates the minimum deviation from the circular angle, the paths gradually approach the outer boundary and reaches the exit point. It should be mentioned that, using higher resolution makes the results smoother and a vehicle can slightly deviate from the circular angle from a further point that leads to a more similar path to the analytical solution. If the starting point is on the outer radius (Figure 4-12), the vehicle's path follows the outer radius without deviating. When $w = 10$, the paths lie between those corresponding to the mentioned extreme cases. The larger value for the weight value, the closer path to the minimum deviation case while a smaller weight makes the path closer to the shortest one.

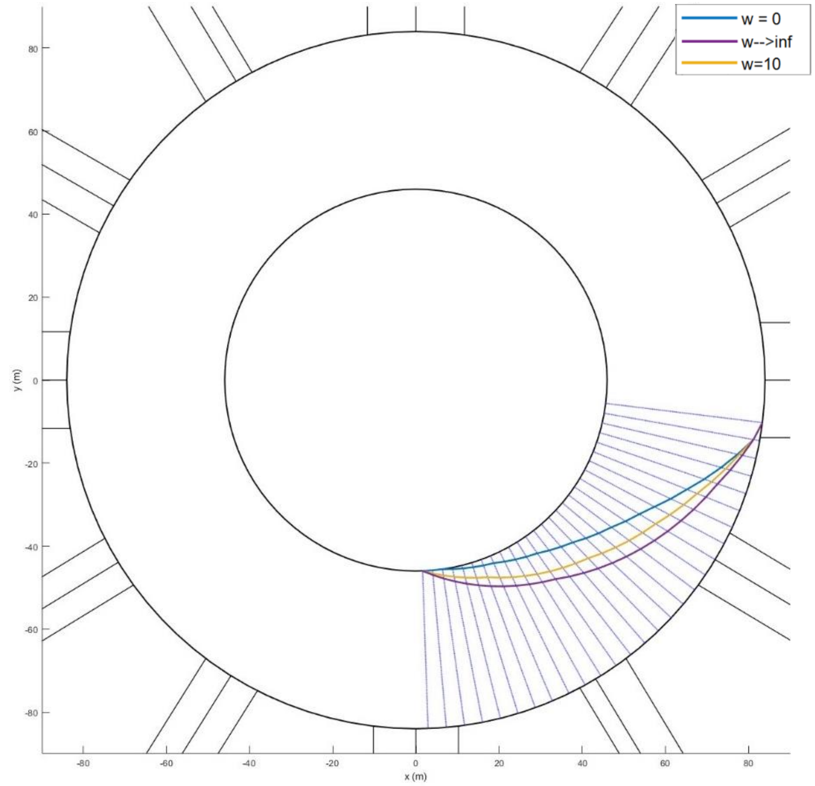

Figure 4-10 – Optimal paths for a visible OD with different weights - starting from the inner circle

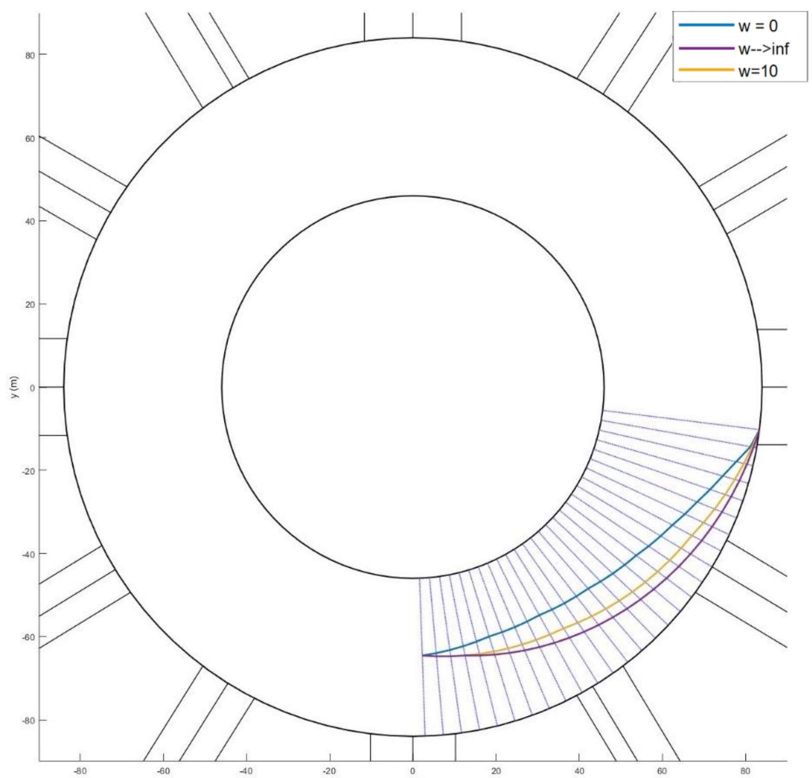

Figure 4-11 – Optimal paths for a visible OD with different weights - starting from the middle of the roundabout

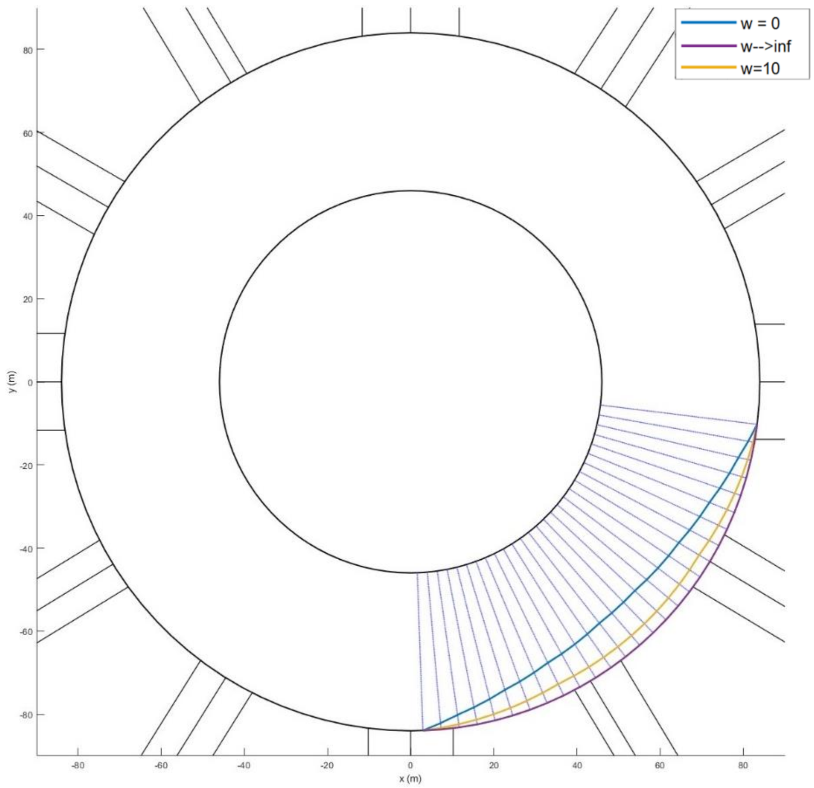

Figure 4-12 – Optimal paths for a visible OD with different weights - starting from the outer circle

The resulted paths for an invisible pair are shown in Figure 4-13 to Figure 4-15. In the shortest path case, $w = 0$, if the origin is on the inner boundary (Figure 4-13), it will move along that till the destination become visible. Otherwise, it will reach the inner circle via a quasi-tangent line keep moving along it until the destination becomes visible. After that, they navigate quasi-linearly to the exit. On the other hand, when $w \to \infty$, a vehicle gradually gets closer to the outer boundary, if it is not located on the outer radius, and reach it at the exit angle. If the starting point is on the outer radius (Figure 4-15), the vehicles follow the outer radius without deviating. When $w = 10$, the path lies between those corresponding to the mentioned extreme cases. More specifically, if the origin is on the inner circle, the path leaves the inner circle sooner compared to the shortest path. For vehicles starting from a point in the middle or outer parts of the roundabout, they may not reach the inner boundary. Finally, for the origins located on the outer boundary, vehicles leave the outer circle to follow a shorter path as opposed to the path which corresponds to the minimum deviation case; however, the distance followed, in this case, will be longer compared to the shortest path.

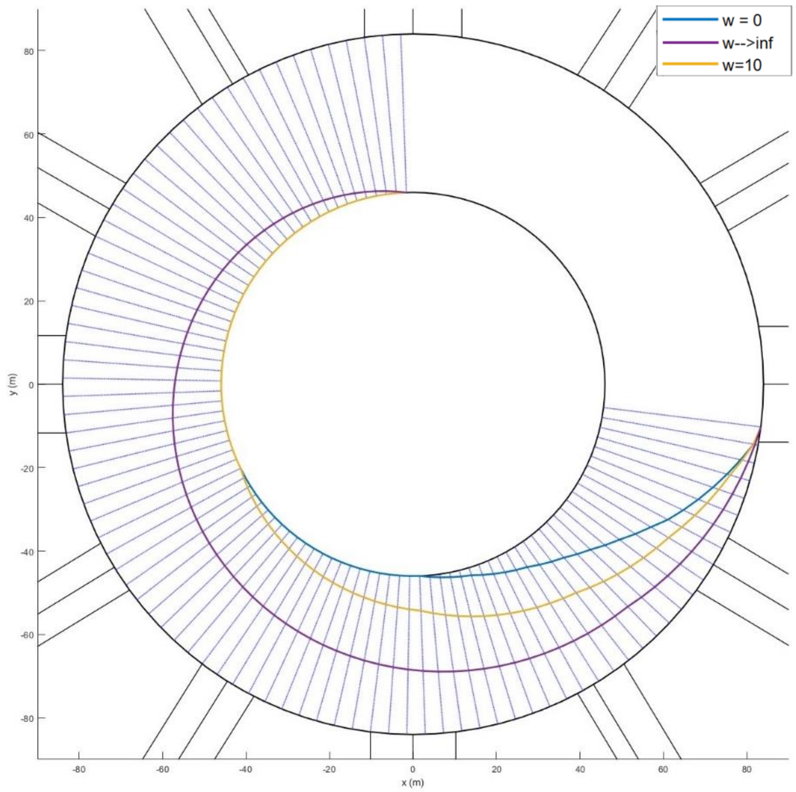

Figure 4-13 – Optimal paths for an invisible OD with different weights - starting from the inner circle

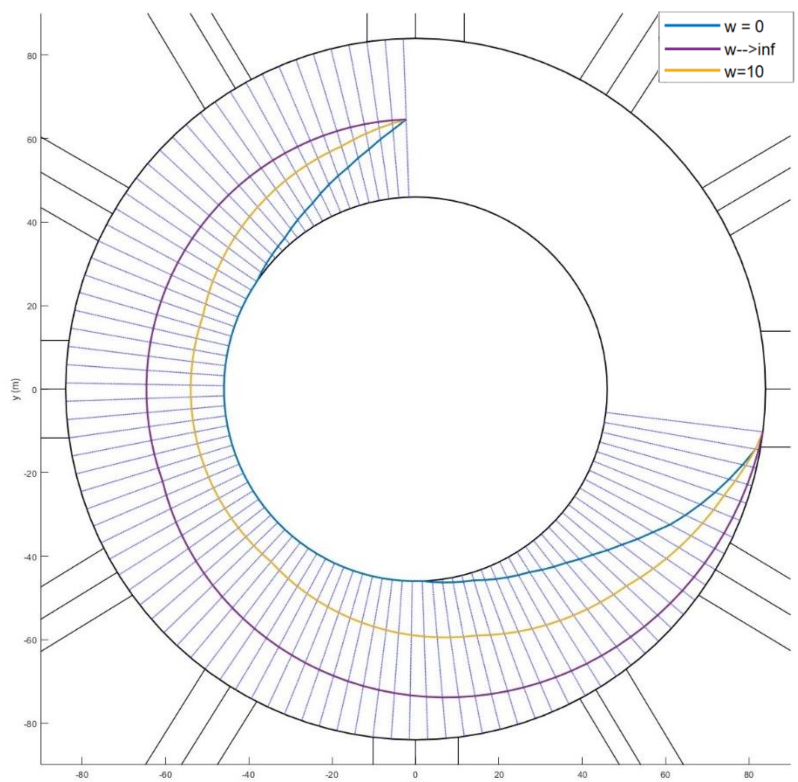

Figure 4-14 – Optimal paths for a visible OD with different weights - starting from the middle of the roundabout

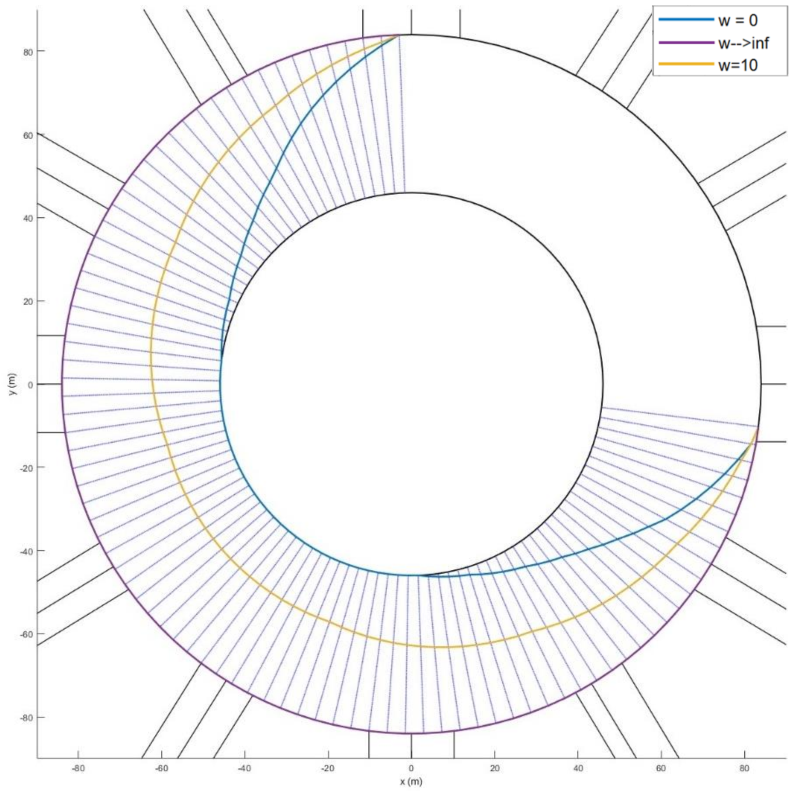

Figure 4-15 – Optimal paths for a visible OD with different weights - starting from the outer circle

4.5 Comparison between the results of the analytical solution and the backwards Dijkstra's algorithm

In this section, the results generated by the analytical solution and the Dynamic Programming based approach are compared for both visible and invisible OD pairs. Figure 4-16 demonstrates the shortest paths for an invisible OD pair. Marked in blue is the result of the analytical solution in which the path reaches the inner boundary through a tangent line, follows the inner circle, and exits through a tangent line. Whereas the orange one is the resulted path from the backward Dijkstra algorithm which is not very close to the analytically optimal path. This difference is due to considering the deviation threshold in the backward Dijkstra algorithm. Particularly, when the vehicle aims to reach the inner circle or when the destination gets visible, the analytical approach generates a big deviation from the circular angle while the DP-based approach produces a smoother angle to meet the deviation constraint. Considering a larger threshold for the deviation, e.g., 60°, the generated path would be closer to the analytical one. In addition, the deviation from the circular angle during the vehicle's movement is presented in Figure 4-17. It can be observed that the deviation of the analytical solution consists of three distinct parts. In the first part, deviation is positive which marks the vehicle's movement towards the inner circle. Then, in the second part, the vehicle moves along the inner boundary and consequently its deviation is equal to zero. Finally, the vehicle's deviation is negative when it moves towards the outer area. Although the resulted path of the backward Dijkstra follows the general concept of the analytical trajectories, some unsmooth changes can be noticed due to discretization since a continuous range of desired orientation and consequently deviation cannot be achieved.

Similarly, the optimal paths determined by the analytical solution and backward Dijkstra algorithm for the minimum deviation problem can be seen in Figure 4-18. Since the minimum deviation responses do not need extreme deviations, the results are very similar. Hence, the analytical solution does not violate the considered threshold. Furthermore, Figure 4-19 displays the deviation caused by these methods. The analytical approach results in a constant deviation value while moving from the origin to the destination, whilst the deviation related to the backward Dijkstra oscillates around the analytical value because of the limited alternatives it is provided with.

The analytical and DP- based shortest paths, correspondingly, for a visible OD are drawn in Figure 4-20. The optimal path determined by the algorithm is a curve compared to the analytical solution, which was a direct line, due to obeying to the deviation restriction. The resulted deviations are shown in Figure 4-21 where the analytical trajectory has bigger values, and the algorithm's result features some oscillations. The results related to the minimum deviation case of this visible pair can be seen in Figure 4-22 and Figure 4-23 where the two responses are very similar.

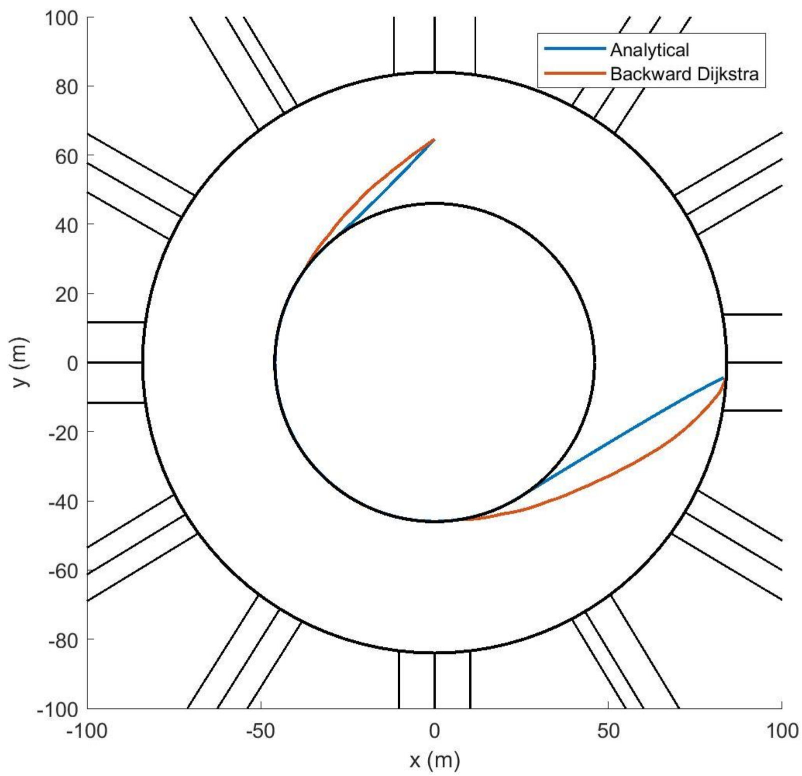

Figure 4-16 – Comparing analytical and backward Dijkstra paths- the shortest path, an invisible OD

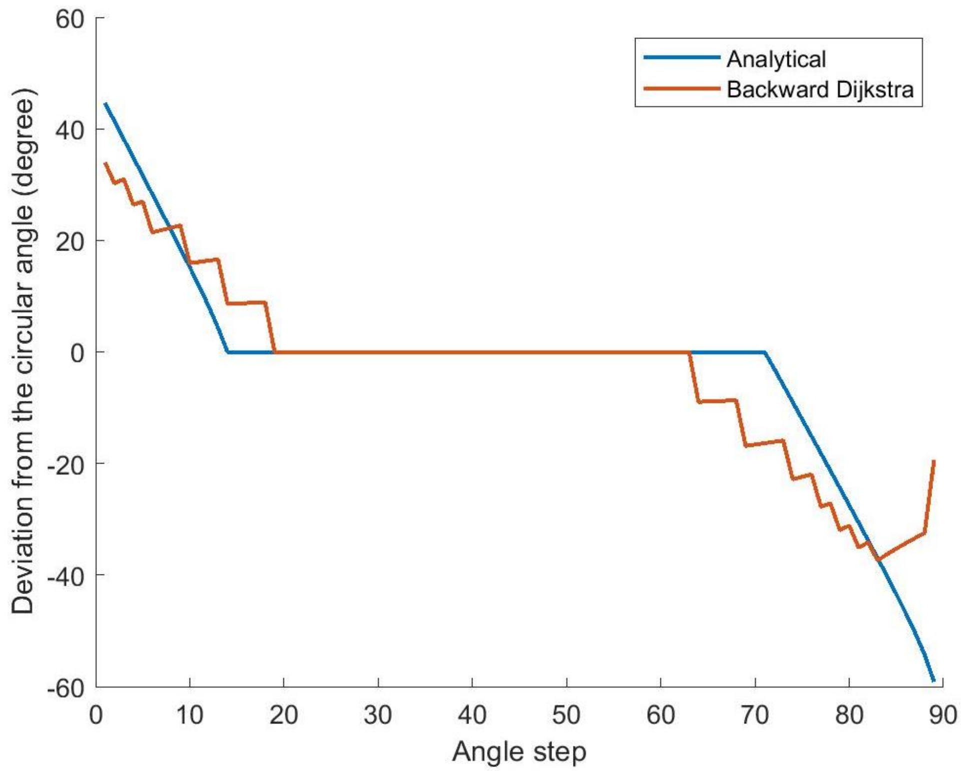

Figure 4-17 – Comparing analytical and backward Dijkstra deviations- the shortest path, an invisible OD

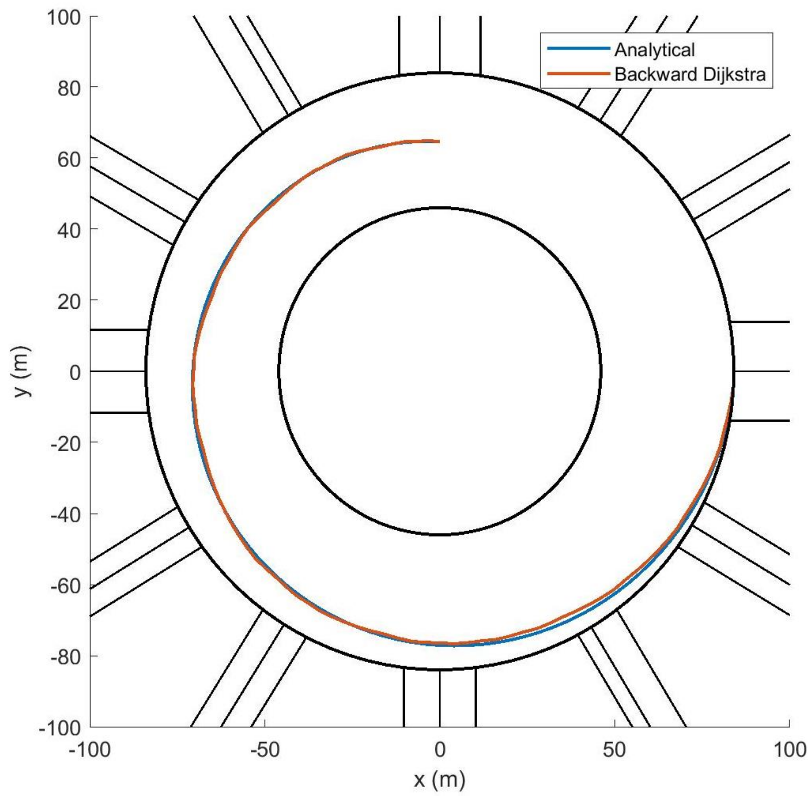

Figure 4-18 - Comparing analytical and backward Dijkstra paths- the minimum deviation, an invisible OD

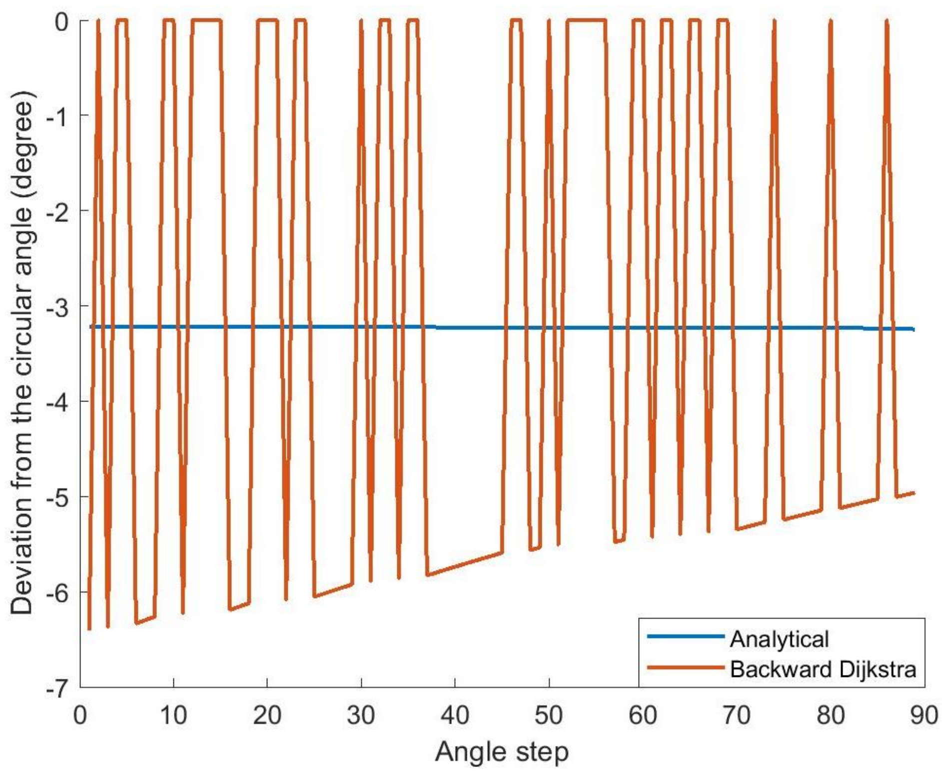

Figure 4-19 – Comparing analytical and backward Dijkstra deviations- the minimum deviation, an invisible OD

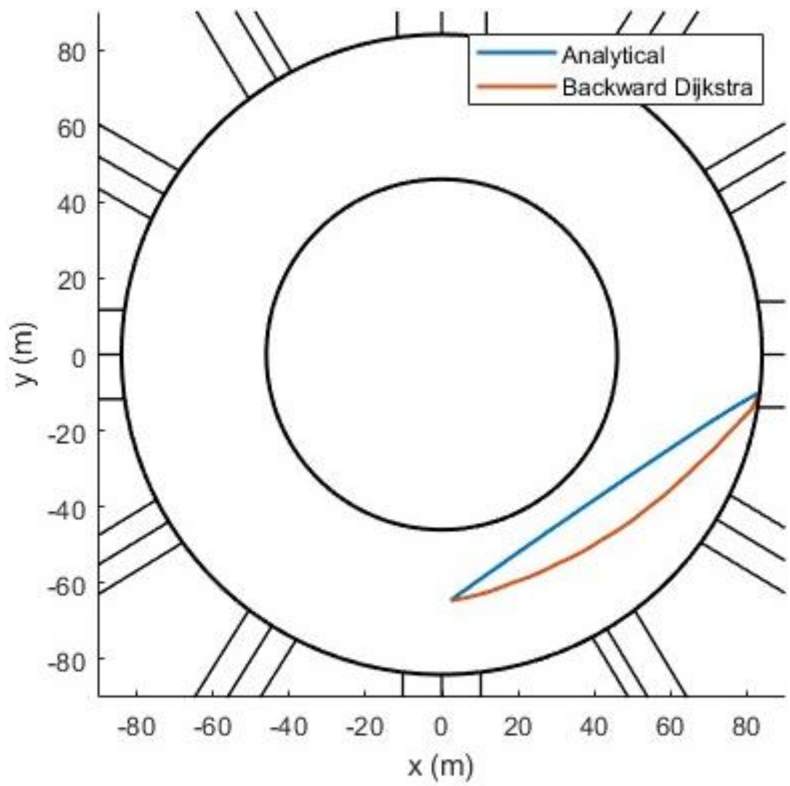

Figure 4-20 – Comparing analytical and backward Dijkstra paths- the shortest path, a visible OD

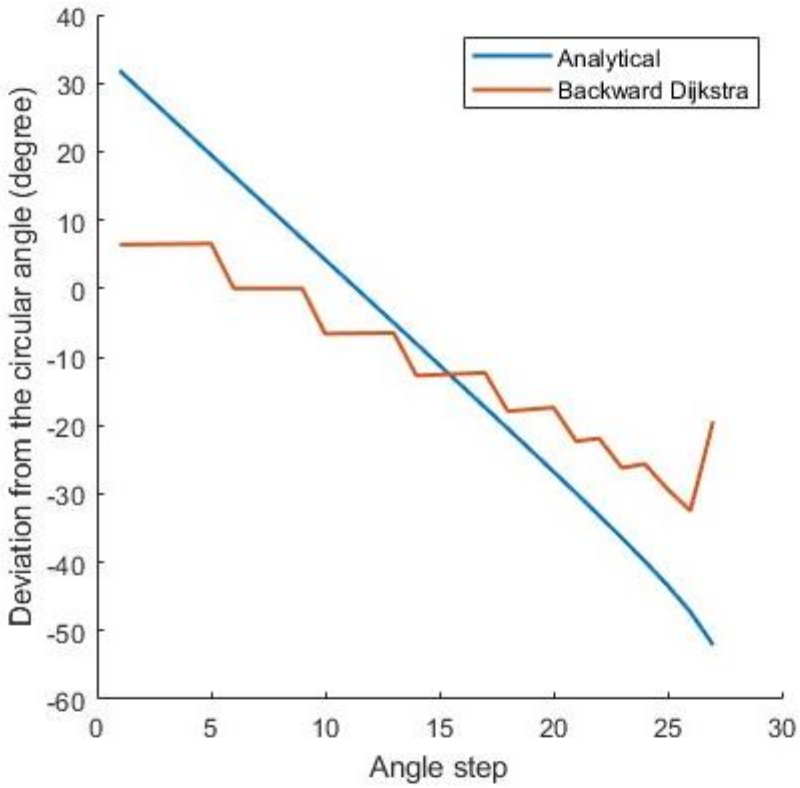

Figure 4-21 – Comparing analytical and backward Dijkstra deviations- the shortest path, a visible OD

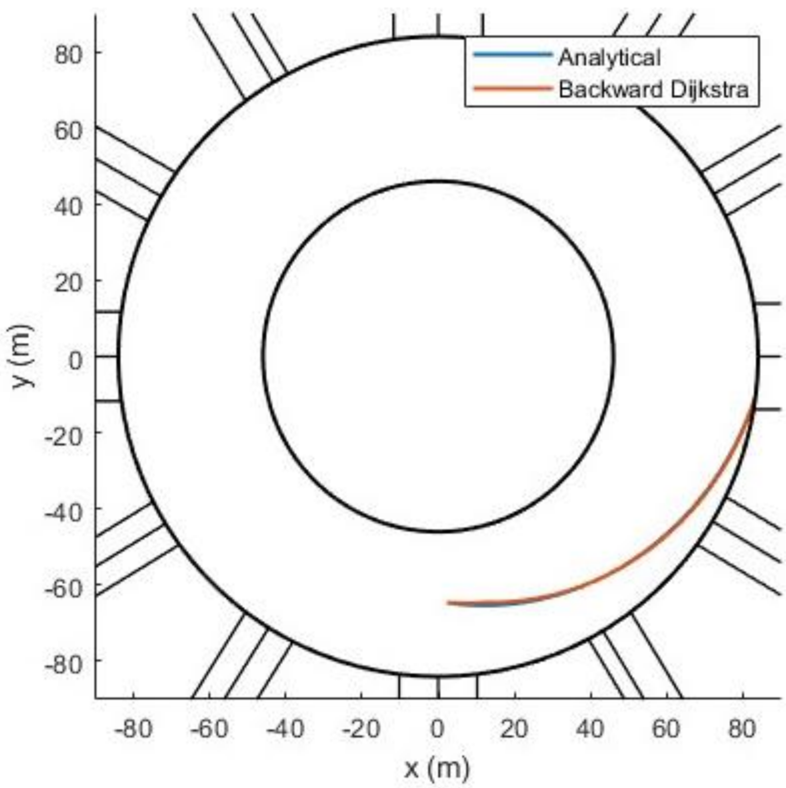

Figure 4-22 – Comparing analytical and backward Dijkstra paths- the minimum deviation, a visible OD

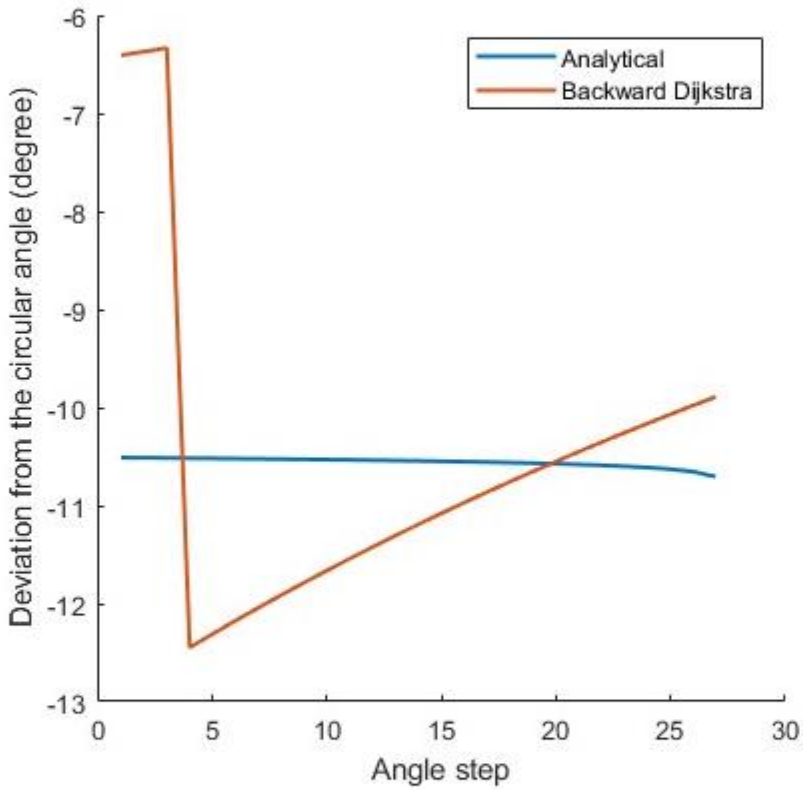

Figure 4-23 – Comparing analytical and backward Dijkstra deviations- the minimum deviation, a visible OD

4.6 Optimal Orientations considering multiple exit points

In this section, a similar procedure to section 4.3 is followed to determine the optimal network orientations, however, here vehicles are allowed to exit from multiple exit points rather than just one, if the exit branch is sufficiently wide. This may expedite transit times, prevent collisions, and better manage vehicle traffic at the exit. At the same time, the roundabout's surface is utilized more effectively.

As can be seen in Figure 4-24 to Figure 4-27, a greater proportion of the road network is accessible by vehicles. This occurs as a result of the vehicles' utilization of multiple exit points, which allows them to fully take advantage of the exit's width rather than being forced to exit from a single point. The density of the network, as well as the design of the roundabout, or more particularly the width of the exit in each case, are directly related to the existence of multiple exit points and the number of these exit points.

The density of the grid generated, remains the same as before, $\Delta r = 1.52m$ and $\Delta\varphi = 3^{\circ}$, which leads to a total of 2975 grid nodes. Also, the maximum admissible deviation is 15°, which allows a maximum of 2 possible radius transitions and 14161 possible segments between nodes. This setting is chosen to generate visually understandable results, while a higher resolution can be selected for the real-time implementation. All presented results concern one specific exit branch, in this case Branch 1. The computation time to run the backward Dijkstra algorithm for this roundabout (for one destination) and in this case 3 exit points, amounts to approximately 601 seconds. The time required for the stimulation is significantly higher than the previous case. That is due to the fact that the orientations in the roundabout are calculated for a vehicle to exit from all three exit points and then compared with one other in order for the best exit point to be selected for each discrete possible vehicle's position.

Figure 4-24 portrays the results of the backward Dijkstra algorithm for the shortest path case, i.e., $w = 0$. In contrast to the previous section, here vehicles are offered additional space in the roundabout in which they can navigate, especially near the exit branch. Similar to the above presented results, at points from which the destination is not visible, vehicles attempt to reach the inner circle as directly as possible, provided that their deviation from the circular angle does not violate the defined limit. Furthermore, where the destination is visible, orientations indicate an almost straight path towards it. Figure 4-25 depicts the results of the backward Dijkstra algorithm for $w = 10$. Vehicles at extremely distant origins have a propensity to reach the inner circle, whereas those at closer origins are unlikely to get very close to it. Traffic distribution across the roundabout area is notably more equitable. The results of the backward Dijkstra algorithm are also displayed in Figure 4-26, with $w = 20$.

Finally, Figure 4-27 shows the desired orientations in the case of minimum deviation. In this scenario, vehicles typically follow circular routes and approach the exit branch gradually. The orientations and induced paths follow the outside border for nodes that are close to the outer boundary, such as when a vehicle enters a roundabout.

As mentioned, the blank spaces in the figures indicate nodes that cannot be feasibly connected to the destination, as this would violate the maximum-deviation limit. In this case, however, they are significantly less.

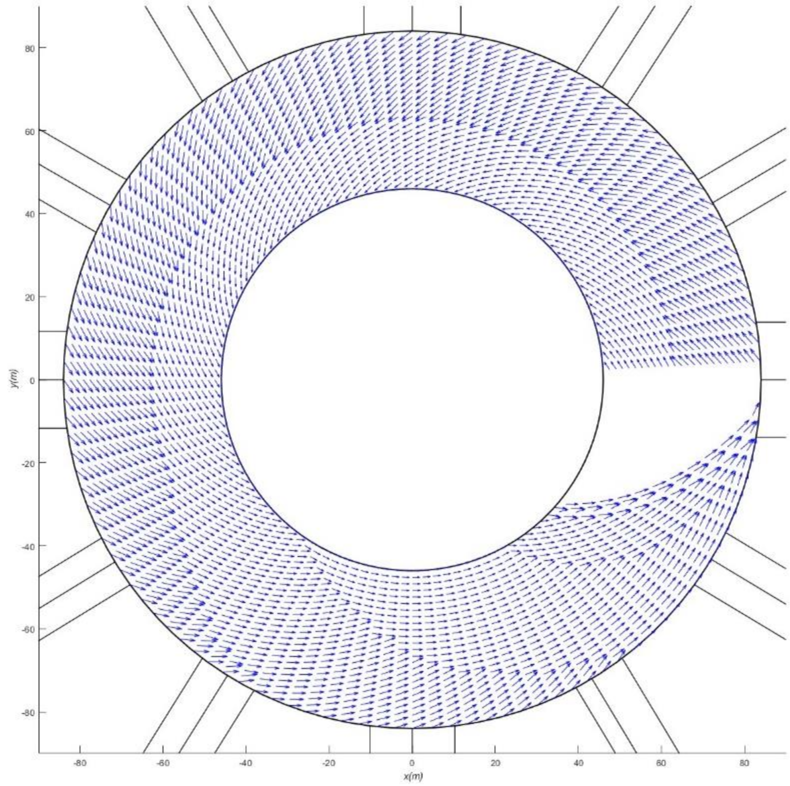

Figure 4-24 – Plot of Optimal Orientations for Destination Branch 1 (allowable deviation=15° and weight=0)

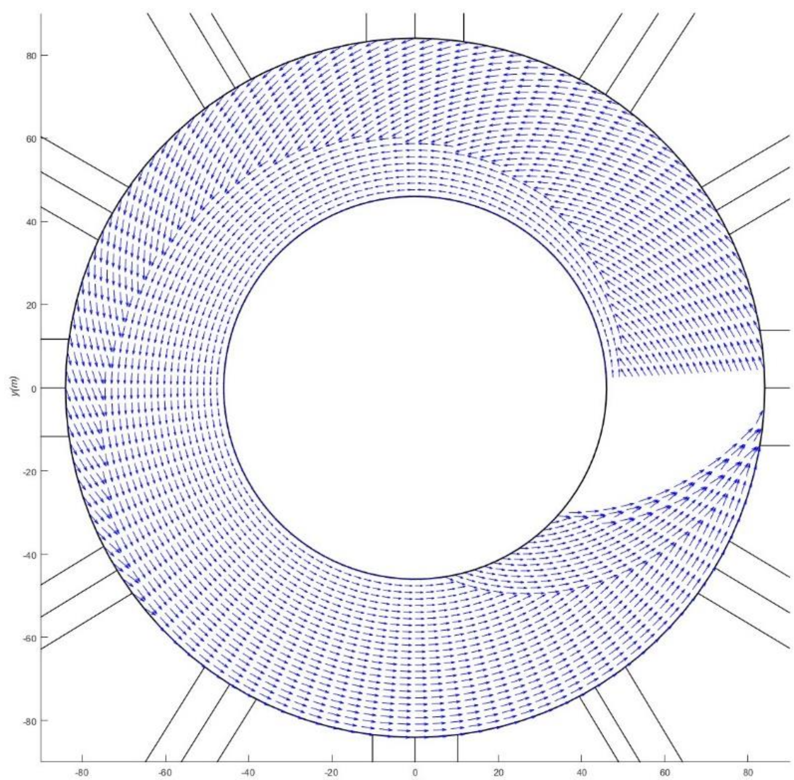

Figure 4-25 – Plot of Optimal Orientations for Destination Branch 1 (allowable deviation=15° and weight=10)

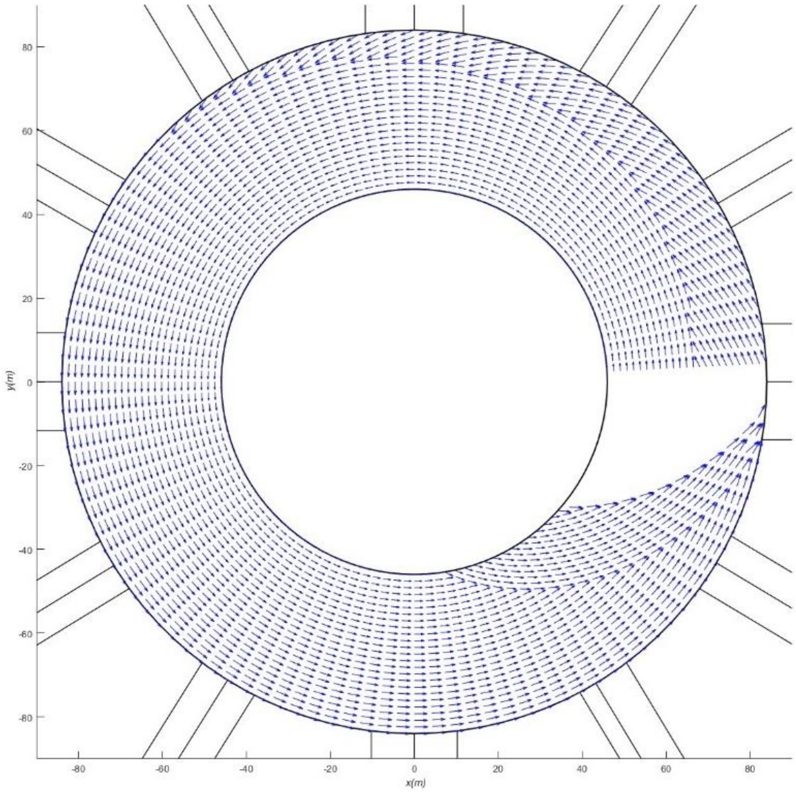

Figure 4-26 – Plot of Optimal Orientations for Destination Branch 1 (allowable deviation=15º and weight=20)

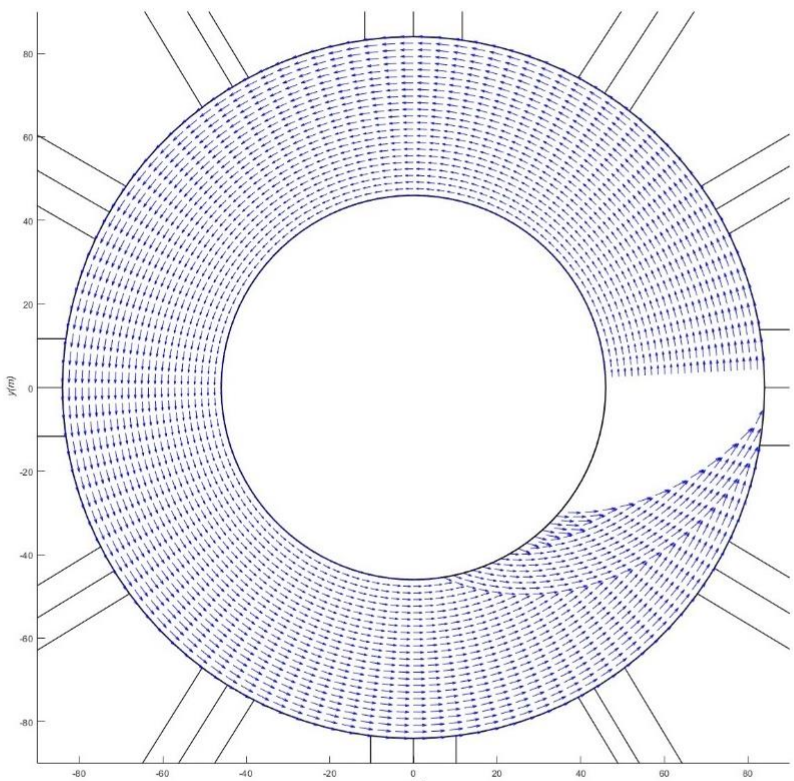

Figure 4-27 – Plot of Optimal Orientations for Destination Branch 1 (allowable deviation=15º and weight→ inf)

4.7 Sub-Optimal online approach

To investigate the accuracy of the presented sub-optimal approach for the real-time implementation, its results may be compared to the ones of backward Dijkstra algorithm. To obtain the optimal value of parameter α for a certain weight, results are collected from the backward Dijkstra algorithm, and it is calculated with the use of (4-15). Considering $w = 10$, the optimal value of the parameter is 0.25.

The optimal path is generated by the backward Dijkstra for a given OD and three different values, 0, 10 and infinite are illustrated in Figure 4-28. In addition, the sub-optimal path is generated by $\alpha = 0.25$ that is an alternative for the optimal path while the weight is 10. As can be seen, the sub-optimal path is sufficiently close to the related optimal path. Hence, one may use the instant combination of the shortest path and the minimum deviation analytical solutions instead of recalling the results of offline optimal orientations.

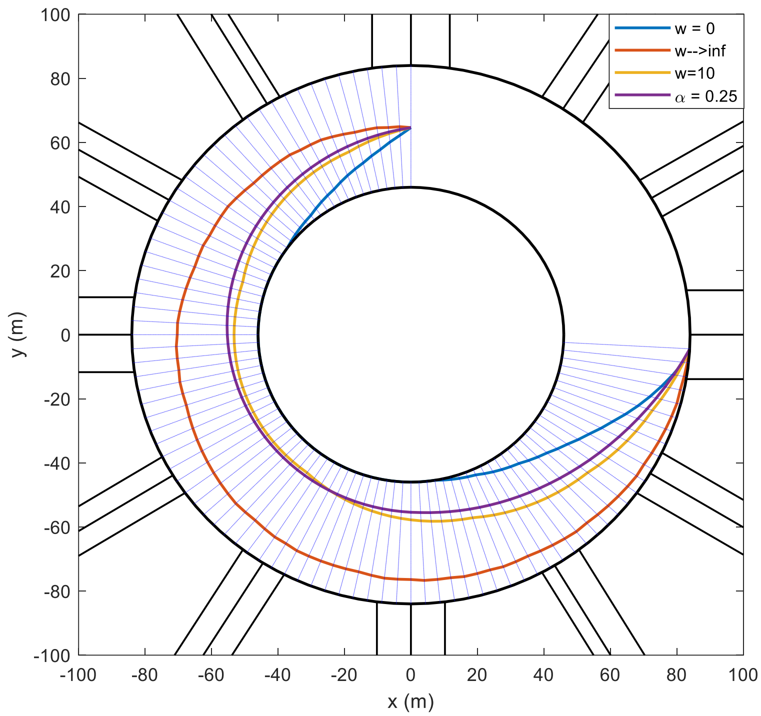

Figure 4-28 – Comparing the results of sub-optimal approach and backward Dijkstra

5 Conclusion and Future works

5.1 Concluding remarks

This thesis suggests a transparent and systematic approach to determine the optimal orientations of vehicles navigating on large lane-free roundabouts, which is considered to be a challenging task due to its intricate geometrical features and numerous potential vehicle conflicts. The suggested method determines the desired orientations by minimizing a weighted summation of the travel distance and deviation from the circular angle through solving an optimum control problem. The vehicle motion is firstly formulated as an optimal control problem in the continuous framework where the angle of polar coordinates is considered as the independent variable instead of time.

For two extreme cases, the shortest path and the minimum deviation from the circular angle, respectively, an analytical solution is obtained. The shortest path solution can be described using simple geometric concepts for two distinct cases. If the destination is visible from the origin, the shortest path would be a direct line connecting them and the desired orientation, as a result, will be the slope of that line. For invisible origin-destination pairs, the shortest path contains three parts. First, it connects the origin to the inner circle through a tangent. Then, it continues along the inner boundary till the destination gets visible, and lastly a direct line connects it to the exit point. Correspondingly, the obtained solution for the minimum deviation problem is to maintain a constant deviation from the circular angle from origin to destination. This translates to a vehicle smoothly moving closer to the outer boundary and reaching it at the exit point. If it is already located on the outer circle then it moves along the outer boundary.

For general weighted scenarios, it is not easy to obtain the analytical solution so, the backward Dijkstra algorithm is suggested to determine the optimal path, instead. For this goal, the roundabout surface is discretized to create a grid in which the algorithm considers a node as the destination and finds the best path from any possible origin to reach the destination such that the total criterion is minimized along this path. The algorithm is presented for two cases: a) considering only one exit point on the exit branch and b) considering multiple exit points. Obviously, employing the second strategy results in longer execution time; however, it facilitates exploiting a bigger area of the roundabout surface. Since running the backward Dijkstra is time-consuming, appropriate strategies are considered for real-time implementation. The first approach is storing the results in a database that is accessible for the vehicles navigating on the roundabout. They may recall the optimal desired orientation based on their instantaneous position on the roundabout and their intended exit branch. Alternatively, one can instantly combine the shortest path and the minimum deviation analytical solutions using a weighted average. The average parameter may be calculated through solving a single parameter optimization problem using Least Square approach.

The presented methodologies are applied to determine the desired orientation in an excessively complicated case study, the Charles de Gaulle roundabout in Paris, France, which comprises 12 bidirectional branches that results in 144 different origin-destination pairs. Moreover, its inner and outer radiuses are 46 m and 84 m, respectively, featuring a width of 38 m. Initially, the desired orientations are obtained at each point in the roundabout for a given destination. Then, based on the achieved orientations, some optimal paths are presented for both visible and invisible ODs, and a variety of starting points. Finally, the sub-optimal approach is compared with results of backward Dijkstra algorithm.

5.2 Future works

Considering the fact that most of the existing control schemes consider simple cases, it is substantial to focus on developing efficient strategies for vehicles moving on roundabouts. In particular, large and lane-free roundabouts have recently become the center of researchers' attention. Below a number of recommendations for future research are given:

1- Microscopic simulation of the presented approaches to check the effectiveness of employing the optimal desired orientation, especially in high density traffic.
2- Finding the optimal orientation for non-circular roundabouts like elliptic ones.

References

[1] M. Papageorgiou, C. Diakaki, V. Dinopoulou, A. Kotsialos, and Y. Wang, "Review of road traffic control strategies", Proceedings of the IEEE ,91, 2003, pp. 2043-2067.

[2] M. Papageorgiou, M. Ben-Akiva, J. Bottom, P.H.L. Bovy, S.P. Hoogendoorn, N.B. Hounsell, A. Kotsialos, and M. McDonald, "ITS and Traffic Management". Transportation (Handbooks in Operations Research and Management Science), Vol. 14), C. Barnhart and G. Laporte, Editors, North-Holland (Elsevier), 2007, pp. 715-774.

[3] C. Diakaki, M. Papageorgiou, I. Papamichail, and I. Nikolos, "Overview and analysis of vehicle automation and communication systems from a motorway traffic management perspective", Transportation Research Part A, 75, 2015, pp. 147–165.

[4] I. Papamichail, N. Bekiaris-Liberis, A.I. Delis, D. Manolis, K.-S.Mountakis, I.K. Nikolos, C. Roncoli, and M. Papageorgiou, "Motorway traffic flow modelling, estimation and control with vehicle automation and communication systems", Annual Reviews in Control, 48 ,2019, pp. 325-346.

[5] M. Aeberhard, S. Rauch, M. Bahram, G. Tanzmeister, J. Thomas, Y. Pilat, F. Homm, W. Huber, and N. Kaempchen, "Experience, results and lessons learned from automated driving on Germany's highways", IEEE Intelligent Transportation Systems Magazine 7, 2015, pp. 42-57.

[6] M. Papageorgiou, K.S. Mountakis, I. Karafyllis, I. Papamichail, and Y. Wang, "Lane-free artificial-fluid concept for vehicular traffic", Proceedings of the IEEE, 109, 2021, pp. 114-121.

[7] M. Rostami-Shahrbabaki, S. Weikl, M. Akbarzadeh, and K. Bogenberger, "A two-layer approach for vehicular flocking in lane-free environment". *11th Triennal Symposium on Transportation Analysis (TRISTAN)*, 2022.

[8] V.K Yanumula, P. Typaldos, D. Troullinos, M. Malekzadeh, I. Papamichail, and M. Papageorgiou, "Optimal path planning for connected and automated vehicles in lane-free traffic". *2021 IEEE Intelligent Transportation Systems Conference (ITSC)*, Indianapolis, USA. September 19-21, 2021, pp. 3542-3552.

[9] D. Troullinos, G. Chalkiadakis, I. Papamichail, and M. Papageorgiou, "Collaborative multiagent decision making for lane-free autonomous driving", *20th Intern. Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, online, 2021, pp. 1335-1343.

[10] I. Karafyllis, D. Theodosis, and M. Papageorgiou, "Lyapunov-Based Two-Dimensional Cruise Control of Autonomous Vehicles on Lane-Free Roads", *60th IEEE conference on Decision and Control (CDC)*, Austin, TX, USA, December 13-15, 2021, pp. 2683-2689.

[11] D. Troullinos, G. Chalkiadakis, D. Manolis, I. Papamichail, and M. Papageorgiou, "Lane-free microscopic simulation for connected and automated vehicles", *24th IEEE International Conference on Intelligent Transportation (ITSC)*, Indianapolis, IN, USA, 2021, pp. 3292-3299.

[12] M. Sekeran, M. Rostami-Shahrbabaki, A.A. Syed, M. Margreiter, and K. Bogenberger, "Lane-Free Traffic: History and State of the Art", *IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)*, Macao, China, 2022, pp. 1037-1042

[13] A. Flannery and T. Datta, Operational performance measures of American roundabouts, Transportation research record, 1572(1), 1997, pp. 68-75

[14] A. Danesh, W. Ma, C. Yu, R. Hao, and X. Ma, "Optimal roundabout control under fully connected and automated vehicle environment", *IET Intelligent Transportation Systems*, 2021, pp. 1-14.

[15] R. Azimi, G. Bhatia, R.R. Rajkumar, and P. Mudalige, STIP: Spatio-temporal intersection protocols for autonomous vehicles. *ACM/IEEE international conference on cyber-physical systems*, 2014, pp. 1-12.

[16] M. Martin-Gasulla, and L. Elefteriadou, "Traffic management with autonomous and connected vehicles at single-lane roundabouts", *Transportation Research Part C*, 125, 102964, 2021.

[17] K. Xu, C.G. Cassandras, W. Xiao, "Decentralized time and energy-optimal control of connected and automated vehicles in a roundabout with safety and comfort guarantees", *IEEE Transactions On Intelligent Transportation Systems,* 2022, DOI: 10.1109/TITS.2022.3216794.

[18] L. Zhao, A. Malikopoulos, and J. Rios-Torres, "Optimal Control of Connected and Automated Vehicles at Roundabouts: An Investigation in a Mixed-Traffic Environment", *IFAC PapersOnLine*, vol. 51, no. 9, 2018, pp. 73-78.

[19] M. Hafizulazwan, B.M. Nor, and T. Namerikawa, "Merging of Connected and Automated Vehicles at Roundabout using Model Predictive Control", *57th annual conference of Society of Instrument and Control Engineers of Japan*, 2018.

[20] R. Mohebifard, and A. Hajbabaie, "Connected automated vehicle control in single lane roundabouts", *Transportation Research Part C*, 131, 103308, 2021.

[21] I.H. Zohdi, and H.A. Rakha, "Enhancing Roundabout Operations via Vehicle Connectivity", *Journal of the Transportation Research Board*, 2381(1), 2013, pp.91-100.

[22] E. Debada, and D. Gillet, "Merging into Single-Lane Roundabouts in the Presence of Uncertainty", *IEEE 21st International Conference on Intelligent Transportation Systems (ITSC)*, 2019, pp. 3168–3175.

[23] C.Wang, Y. Wang, S. Peeta, "Cooperative roundabout control strategy for connected and autonomous vehicles", *Applied Sciences*, 12(24), 2022, p. 12678

[24] J.P. Rastelli, and M.S. Penas, "Fuzzy logic steering control of autonomous vehicles inside roundabouts", *Applied Soft Computing*, 35, 2015, pp. 662-669.

[25] N. Ding, X. Meng, W. Xia, D. Wu, L. Xu, and B. Chen, "Multi-vehicle Coordinated Lane Change Strategy in the Roundabout under Internet of Vehicles based on Game Theory and Cognitive Computing", *IEEE Transactions on Industrial Informatics*, vol. 14, no. 8, 2015.

[26] P. Hang, C. Huang, Z. Hu, Y. Xing, and C. Lv, "Decision Making of Connected Automated Vehicles at an Unsignalized Roundabout Considering Personalized Driving Behaviours", *IEEE Transactions on Vehicular Technology*, vol. 70, no. 5, 2021.

[27] R. Tian, S. Li, N. Li, I. Kolmanovsky, A. Girard, and Y. Yildiz, "Adaptive Game-Theoretic Decision Making for Autonomous Vehicle Control at Roundabouts", *IEEE Conference on Decision and Control (CDC),* 2018, pp. 321-326.

[28] Y. Zhang, B. Gao, L. Guo, H. Guo, and H. Chen, "Adaptive Decision-Making for Automated Vehicles Under Roundabout Scenarios Using Optimization Embedded Reinforcement Learning", *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 12, 2020, pp.5526-5538.

[29] M. Naderi, M. Papageorgiou, I. Karafyllis, I. Papamichail, "Automated vehicle driving on large lane-free roundabouts", *25th IEEE International Conference on Intelligent Transportation Systems (ITSC),* Macao, China, 2022, pp. 1528-1535.

[30] E. Dijkstra, "A note on two problems in connexion with graphs." *Numerische Mathematik 1* (1959): pp. 269-271.

[31] https://en.wikipedia.org/wiki/Place_Charles_de_Gaulle

[32] https://www.reddit.com/r/europe/comments/azcnpg/arc_de_triomphe_and_place_charles_de_gaulle_paris/