Control for Cooperative Autonomous Driving Vehicles at Roundabouts Inspired by Bird Flocking Behaviour

Jesper de Haan
University of Twente
Faculty EEMCS
Enschede, the Netherlands

***Abstract*—With the high increase in number of motorized vehicles over the past years, higher road capacity is required to prevent traffic jams and increase road safety. With the rise of cooperative autonomous driving algorithms, the congestion can be lowered resulting in better road throughput. This research dives into the feasibility of a cooperative autonomous driving algorithm based on bird-flocking behaviour regarding roundabouts. The vehicle's behaviour is determined by three forces: cohesion, alignment and separation. Combined with some merging rules a successive algorithm is designed to manage the traffic on a roundabout and plaza. The algorithm is put to the test in a simulation showing capacities of 6000 and 5500 vehicles per hour with an average travel time on the roundabout or plaza of 6.54 and 4.71 seconds. In order to achieve safe operation, sending rates of 8 and 15 packets per second are required.**

***Index Terms*—Cooperative Autonomous Driving, Bird Flocking Behaviour, Roundabouts, Plaza, OpenGL**

I. Introduction

In present-day traffic, jams and crashes are commonly seen. With the increase of motorized vehicles over the past years, problems as such are becoming bigger and bigger [1]. The origin of these problems can be branched into two factors. Factor one is the infrastructure which cannot manage the demand for road capacity. Here, the present-day intersection cannot handle the high throughput of the connecting roads. Factor two is the behaviour of vehicles on the road, which is controlled by the drivers. In case of a collision, roads need to be closed which lowers the amount of vehicles passing by.

A simple solution to the first problem is increasing the capacity of roads and intersections by making them bigger. The question that arises is whether one has to solve the problem by occupying more space, if only available, for that one peak moment of the day. Another solution suggests an increase in traffic flow such that more vehicles can pass by within a given time interval. However, this calls for safety issues which relates to the second problem: the driver. According to [2], 15% of the sold cars have the option for autonomous driving by 2030. The remainder has a person behind the steering wheel causing a major problem: human error. The human perception and reaction time creates an inefficient usage of the road. Increasing speed limits results in higher crash rates which will eventually result in a lower throughput on the road [3].

A better solution, linked to the problem of human error, is taking away the control from the human driver. Using computer systems to replace the human driver and enable communications between vehicles, results in a situation where jams and crashes are not only avoided right on the spot. But potential situations in the future can also be seen and prevented. This idea is the basis behind a so-called Cooperative Autonomous Driving (CAD) system. The research done in this paper dives into the creation of an algorithm to control cooperative autonomous driving vehicles. In a CAD system, vehicles communicate parameters such as their position, speed, acceleration and heading to surrounding vehicles over a wireless link. A vehicle within the system uses this received information from its neighbours to determine its behaviour.

This algorithm has to be based on a set of rules by which vehicles handle their behaviour. This solution may sound high-tech but the footing of this algorithm is found within something that already existed tens of millions of years before the creation of humans, namely birds. Everyone has seen a group of birds flying through the sky making turns without noticeable communication and most importantly collisions. Research on this has shown that this complex-looking behaviour of a flock of birds, named boids, is based on three relatively simple forces [4]. The paper stated these forces as flock centering, velocity matching and collision avoidance which represent staying together, copying each other's velocity and avoiding running into each other respectively. This generated interest in the investigation of how feasible such an implementation would be on modern traffic.

The first step in this investigation process is finding the analogies of this bird behaviour and the moving principles of vehicles. The outcome stated that the three forces suffice for a control algorithm for vehicles but a carefully designed weighted sum of the forces is needed for proper behaviour [5]. The next step in the process studies the design of the weighted sum to control the vehicles within spatial constraints such as roads [6]. As stated in [5], a new simulation environment was needed to get enough freedom in the control of vehicles. Therefore this environment was created in [6], using OpenGL such that lots of calculations can be done in parallel. Using this approach, a correctly weighted sum was designed to achieve proper control of the vehicles. This allowed the vehicles to drive straight and make turns within the boundaries of the road. As a result, the vehicles moved as a flock without collisions.

To represent realistic traffic situations, the next step is adding intersections such that different destinations can be reached. The intersections investigated in [7] are a Y-junction and a highway exit. Here, vehicles have to cross each other to get to their correct path. This path is mapped out using way-points over which the vehicles reach their different destination.

With this research done, this paper investigates the next step, a control algorithm that can be built on top of the existing algorithm to manage the vehicle's behaviour on a roundabout. Therefore the research question is stated as follows:

- *How feasible is a control algorithm based on the bird flocking behaviour for cooperative autonomous driving vehicles on roundabouts?*

To answer this research question, the following sub-questions will be evaluated:

- *What is the impact of implementing the algorithm on computational and network load?*
- *To what extent can the throughput of the roundabout be increased by the algorithm?*

In the remainder of this paper some background information is given in Sec. II whereafter the designed system is explained in Sec. III. Next, the results are shown and discussed in Sec. IV whereafter a conclusion is made in Sec. V.

II. Preliminaries

For a better understanding of the remainder of the paper, this section gives the needed background information gathered from earlier research. Interpretations and assumptions are stated here.

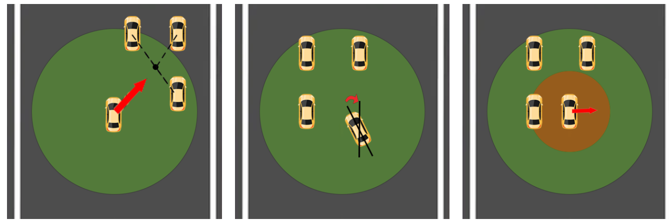
A. Forces

Starting with the findings of Reynolds in [4] about the used forces in a boid, i.e. a flock of birds. He stated that the group of the birds use the following three forces to determine their movements:

- Flock centering: Moving a bird into the center of a group of the other birds. The strength of this force is bigger for higher distances to this center.
- Velocity matching: Copy the velocity of the neighbouring birds.
- Collision avoidance: Move away from birds that are too close. The strength of this force is inversely proportional to the distance between the two birds.

To apply these rules to the vehicles the kinematics of such have to be discussed. Where birds have more freedom regarding moving in a certain direction, vehicles are bound by their maximum steering angle. For birds, the position and velocity of their center of mass are enough for proper simulation. For vehicles however one needs to convert this velocity of its center to a steering angle of its front axis. The computation of this conversion can be found in [6].

Now the vehicle's kinematics are known, the used forces need to be brought to this domain as well. For vehicles, these



(a) Cohesion (b) Alignment (c) Separation

Fig. 1: Visualization of the forces of flocking behaviour for vehicles

forces will act on their x and y directions since the z-direction will follow from the ground. The flock-centering force can be described as cohesion where a vehicle moves into the center of the neighbouring vehicles. The velocity matching force is interpreted as alignment where a vehicle copies the x and y velocities of its neighbours such that their absolute angle in space will be the same. The collision avoidance force will be transformed into separation such that a vehicle will generate a force in the adverse direction of the close vehicles. A visual interpretation of these forces can be found in Fig. 1.

In [6], the forces described above are mathematically calculated using the vehicle's positions and speeds as follows. For the cohesion force $\boldsymbol{f_{C,i}}$, the center point of all vehicles within a certain range needs to be found. For two vehicles, a force pointing towards this point is calculated using Eq. (1). Here, $\boldsymbol{pos_i}$ and $\boldsymbol{pos_j}$ are the x and y coordinates of vehicle *i* and *j*. Calculating this force for all the vehicles within the cohesion range yields up in a force pointing to the center of all of them.

$$\boldsymbol{f_{C,i}} = \boldsymbol{pos_i} - \boldsymbol{pos_j} \tag{1}$$

The alignment force $\boldsymbol{f_{A,i}}$ of vehicle *j* acting on vehicle *i* is calculated using the velocity of vehicle *j*. To make the force more stable an average is taken of vehicle *j*'s current velocity $(\boldsymbol{vel_j})$ and its desired velocity $(\boldsymbol{vel'_j})$ for the next iterative step. Eq. (2) shows this relation.

$$\boldsymbol{f_{A,i}} = \frac{\boldsymbol{vel_j} + \boldsymbol{vel'_j}}{2} \tag{2}$$

The separation force $\boldsymbol{f_{S,i}}$ is determined by the positions and distances of both vehicles. The direction of vehicle *i* to vehicle *j* is determined by the positions of both vehicles, a minus sign is added to rotate this force such that it will point away from the neighbouring vehicle. The distance between vehicle *i* and *j*, d_i is determined by the absolute value of the difference of their positions. When d_i is smaller than the minimum distance r_S, the separation range, the exponent will create a strong force away of the other vehicle *j*. The result can be found in Eq. (3).

$$\boldsymbol{f_{S,i}} = -e^{r_S - d_i} \cdot \frac{\boldsymbol{pos_i} - \boldsymbol{pos_j}}{|\boldsymbol{pos_i} - \boldsymbol{pos_j}|} \tag{3}$$

For control of the vehicle a sum of the three forces is needed. Between each two vehicles, the forces are calculated based on their position and velocity. A fully worked-out computation of this can be found in [5], [6]. As [6] stated a weighted sum is needed. As a result of this, one can determine the needed acceleration of and vehicle based on the presence of another vehicle by Eq. (4). Here, W_C, W_A, W_S are the weights for the different forces. Since the total acceleration of a vehicle is determined by all the surrounding vehicles a sum of Eq. (4) is needed for all vehicles. A force only comes into play when being inside a certain distance, as can be seen by the green and red circles in Fig. 1, some constraints are needed for the sum. These constraints are set by the different ranges for the forces denoted by r_C, r_A, r_S. The resulting acceleration a vehicle experiences from its neighbouring vehicles within the ranges is given in Eq. (5).

$$\boldsymbol{a_{single}} = W_C \cdot \boldsymbol{f_C} + W_A \cdot \boldsymbol{f_A} + W_S \cdot \boldsymbol{f_S} \tag{4}$$

$$\boldsymbol{a_{vehicles}} = \sum_{d_i \leq r_C} W_C \cdot \boldsymbol{f_{C,i}} + \sum_{d_i \leq r_A} W_A \cdot \boldsymbol{f_{A,i}} + \sum_{d_i \leq r_S} W_S \cdot \boldsymbol{f_{S,i}} \tag{5}$$

Although vehicles can group, align and avoid collisions with each other, there is no avoidance from other objects such as the side of the roads. To restrict the space to only a road, some marking of the sides is needed. Additional forces from these sides can be added to the algorithm to avoid collisions. For simplicity, a virtual vehicle can be projected on these side walls aligned with the rotation of the road such that the same equations can be used to calculate the forces. Simply adding these forces to Eq. (5) will result in vehicles staying on the road.

In Eq. (5) one finds the weights and the forces. The forces are found in Eqs. (1) to (3), the weights are still needed to be determined. This is done in [6] by some trial and error. Testing was done by placing a flock of vehicles on a straight road and sweeping the weights. This testing was done in a simulation environment specifically created for the validation of this bird flocking behaviour on vehicles. More information about the simulation will be given later. The result of the simulation showed optimal values of the weights:
$W_C = 0.26, W_A = 1.50$ and $W_S = 1.00$.

B. Destination

With the validation of the controlling principle, the new aspect of destination tracking is researched in [7]. The research focuses on road junctions such as Y-junction and highway exits. To create different paths on the roads for different directions, way-points are added which will be followed by vehicles based on their destination. Because some vehicles spawning on the left side of the flock have to take the right side of the Y-junction control is added to cross other vehicle's paths without collision. As the paper stated, the weights of the sum in Eq. (5) had to be changed for proper control. For the system with destination, the optimal weights turned out to be: $W_C = 0.1, W_A = 6.0$ and $W_S = 1.6$

C. Simulation environment

The studies described, used a simulation environment to test and validate their algorithms. This simulation environment was created for the sake of testing the birds flocking behaviour along vehicles. The creation of this simulator started during the research of [6] and was expanded with destination tracking during [7]. The simulation is written in Python using the OpenGL shaders. With this, the simulation was able to run on the GPU such that all the calculations of different vehicles could be done in parallel. Resulting in a simulation that can run quite fast and makes it easy to extract data.

In this simulation the following assumptions were made: For determining the vehicle kinematics a Toyota Corolla LE from 2019 was used. Its length is 4.9 meters and has a width of 1.8 meters. To calculate the forces, statistics of neighbouring vehicles, such as positions and speeds, were needed. These statistics are saved in global buffers so therefore this can be seen as if there is a perfect instantaneous communication link between the vehicles i.e. the most up-to-date data of all vehicles are known by all the other vehicles. Regarding the first sub-question about the load to the network for this algorithm, a standard is made in order to determine the sending rate of packets to neighbouring vehicles. The ETSI EN 302 637-2 standard [8] describes the generation and distribution of data packets used in cooperative autonomous driving. For this, an algorithm is designed which calculates the rate based on vehicle's parameters such as speed and acceleration. Here, the packet rate lies between 1 and 10 packets per second.

III. Design

This section discusses the used intersection designs and the implementation of it in order to validate the algorithm with the simulator.

A. Roundabout

As stated in the research question the goal of this paper is to check how feasible a control algorithm, based on bird flocking behaviour, is on roundabouts. To answer this question, the first step is designing a roundabout onto which the algorithm is tested. In [9] lots of designs of roundabouts, from simple present-day to really innovative, are discussed. The turbo-roundabout is widely adopted in present-day traffic due to its high throughput compared to other designs. Therefore, the turbo-roundabout design will form the basis of the newly designed roundabout for testing the algorithm.

Regarding the second sub-question about the throughput, one has to take a look into how this throughput is determined. As stated in [10] current roundabout capacities are either measured using gap-acceptance-based models or by empirical regression.

The gap-acceptance-based models base their results on the needed time gap between two vehicles on the roundabout to make it possible to merge another vehicle between them. Also, the follow-up headway, the time needed for two vehicles entering from the same lane using the same gap, has a big influence. Due to the previously discussed human-error this time gap needs to be quite big. Besides that, the models account for things such as not using blinkers or not paying attention due to distractions to the human driver. To make proper models and thus good approximations lots of intersection characteristics are needed. The promising side of this method is the fact that automated vehicles can take away human errors, lower the needed gap time and follow-up headway due to the better ability of anticipation. This will thus eventually higher the theoretical throughput.

Looking at the empirical regression method, the throughput is determined by the amount of vehicles passing the intersection within a certain time interval. This method shows a more practical capacity. The downside here is that for precise measurement, a consistent flow of entering vehicles is needed. The promising side is the computer taking away human error and the existence of the specifically designed simulation which can be used for the measurement of data with consistent entry flow.

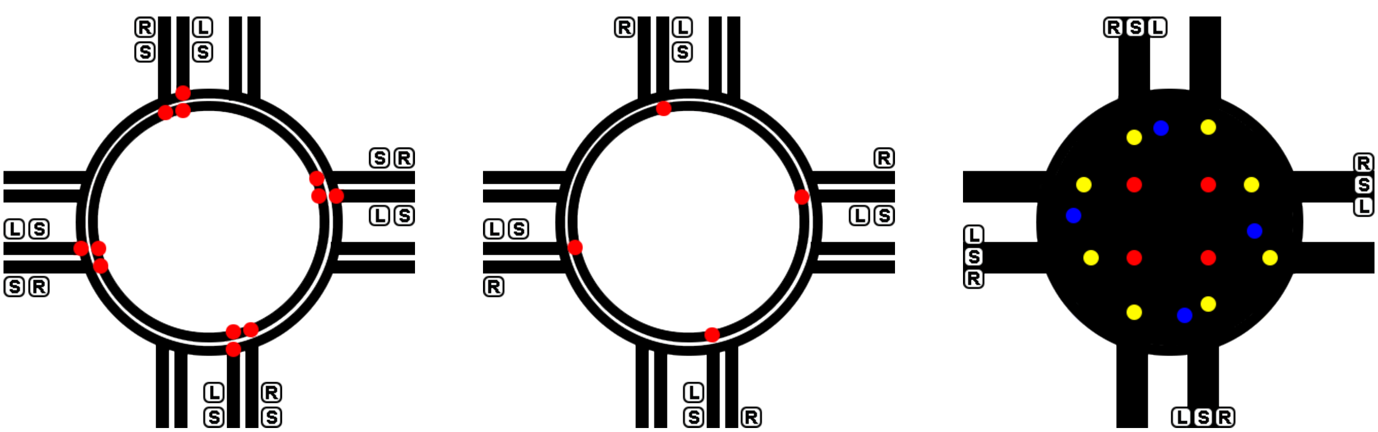
Another capacity evaluation method that is currently under investigation is the conflict technique [11]. It calculates the throughput based on the number of conflict points, which are the points where two different streams of vehicles have to be merged into one stream. Taking a look at a turbo-roundabout from this point of view the conflict points are shown in Fig. 2a. Here, for each entry direction, the right lane can be used for going right or straight and the left lane can be used for either going straight or left. This results in three conflict points for each entry side giving a total of twelve. Taking the option to go straight, away from the right lane, results in only one conflict point as can be seen in Fig. 2b. From a conflict point of view, this will result in a higher capacity. However, the total number of vehicles using the inner lane will be increased, therefore the capacity of the total roundabout will probably be determined by the capacity of this inner lane. Another result of the four conflict points is the ease of the merging process. Furthermore, existing roundabouts could be more easily transformed since the outer lane of the roundabout can be replaced with shortcuts for right turns, leading to no change in the existing single-lane roundabout.

The state-of-the-art turbo roundabouts are either evaluated using the gap-acceptance method or the empirical regression method. They show capacities of around 3500 vehicles/hour [12]. To find the capacity of the designed roundabout in this paper the empirical regression is used. The gap-acceptance method requires a precise model to be made which is quite time-consuming and therefore out of interest. The conflict point technique lacks good qualitative comparison results because it is still under investigation.

B. Plaza

Tests and results of the roundabout, which will be described later, showed delay differences due to the forced driving direction of a roundabout. Therefore another approach, in controlling the traffic on the junction is designed and evaluated. This idea takes the basis of the roundabout, but the inner circle is removed such that an open plaza is formed within the outer bounds. This opens the possibility for straight-going vehicles to literally go straight over the roundabout. Left-turning vehicles can cut corners now as well.

Using this design, a lot more conflict points are created. Since vehicles have more freedom in where to drive, the whole plaza can be seen as little conflict point. This is because the sum of all forces is different for each vehicle due to its unique surrounding pattern of vehicles and walls. This opens the possibilities for multiple paths from a certain entry to an exit. Although the conflict points are shifted for each vehicle, the main conflict points, caused by the different direction streams, are shown in Fig. 2c. Here the red dots are the conflict points for vehicles going straight. The blue dots are for taking a turn to the left and the purple dots are the conflict points where vehicles going straight and left meet each other. The vehicles are now sorted, based on their action, on the right, middle or left side of the entry lane.



(a) Turbo-roundabout design　(b) Used roundabout design　(c) Used plaza design

Fig. 2: Conflict points for the different designs

C. Algorithm

To manage the behaviour of the vehicles on the junctions, an algorithm is needed which calculates all the forces acting on a vehicle. The designed algorithm of [6] is used as a starting point. This algorithm consists of the computation of forces between vehicles and side walls of the road. In [7] a new force was added for the destination tracking. It made use of a force pointing to the next coming way-point of its desired path. At a roundabout, different paths can be taken to get to the same destination. Therefore a way-point approach is not very suitable for this use case.

The correct path finding of the roundabout is based on cohesion or separation on the different side road walls of the roundabout. Based on its destination, the vehicle is either attracted to an exit or repelled. When a vehicle is close to its exit, a strong force is applied to make the vehicle move from the lanes of the roundabout to the correct exit lane. Virtual walls are used to make vehicles enter and leave the roundabout based on their spawn point and destinations. For vehicles with matching spawns and destinations, they don't exist, for other vehicles, they are seen as normal walls. At the exit, the vehicles will form flocks again with their surrounding vehicles of equal destination. The mathematical representation of the forces from the exits on the roundabout $\boldsymbol{a_{R-exits}}$ is given in Eq. (6). Here *L* labels the exits and *D* is the destination of the vehicle. The weights of the forces have the same value as before with the vehicles. The force only has effect when the vehicle is within a certain distance r_E from the exit. For the walls, a virtual vehicle is used as discussed in Sec. II-A. Now, $\boldsymbol{a_{walls}}$ can be calculated the same as $\boldsymbol{a_{vehicles}}$. The behaviour of a vehicle on the roundabout is determined by the forces of the surrounding vehicles, walls and exits and is denoted in Eq. (7).

$$\boldsymbol{a_{R-exits}} = \sum_{d_i \leq r_E} \begin{cases} W_C \cdot f_C + W_A \cdot f_A & L = D \\ W_S \cdot f_S & L \neq D \end{cases} \tag{6}$$

$$\boldsymbol{a_{roundabout}} = \boldsymbol{a_{vehicles}} + \boldsymbol{a_{walls}} + \boldsymbol{a_{R-exits}} \tag{7}$$

For the plaza design, the pathfinding is based on the cohesion to the point in the middle of the exit. Since there are no walls in the middle, a vehicle will directly drive towards the correct exit instead of following the rotation of the roundabout. Furthermore, alignment and separation from walls and other vehicles are added. Again, when arrived at the exit, the vehicle formed a flock with its surrounding vehicles of the same exit. The mathematical representation of the force from the exits of the plaza $\boldsymbol{a_{P-exits}}$ is shown in Eq. (8). Since these forces effects the vehicles behaviour immediately when the vehicle is on the roundabout, no constraints for the sum is needed. The vehicle also doesn't drive along the other exits, so no separation is needed from these. For the plaza, the overall behaviour is determined as shown in Eq. (9).

$$\boldsymbol{a_{P-exits}} = \sum \begin{cases} W_C \cdot f_C + W_A \cdot f_A & L = D \\ 0 & L \neq D \end{cases} \tag{8}$$

$$\boldsymbol{a_{plaza}} = \boldsymbol{a_{vehicles}} + \boldsymbol{a_{walls}} + \boldsymbol{a_{P-exits}} \tag{9}$$

As shown in Fig. 2 the number of conflict points for the roundabout is reduced such that there are as few as possible. Whereas for the plaza design, there are a lot more. The next step is diving into the control of these points. This is the point where the bird flocking algorithm falls short. All the individual birds inside a flock, have the same destination. Besides this, they can use a somewhat unlimited 3D space to prevent collisions. Therefore, they don't experience some kind of braking along their flight. Thus, the bird flocking algorithm doesn't solve this kind of situation. Vehicles, differently, do have different destinations and their two-dimensional space is bounded by the size of the roads. To solve the situations where one vehicle has to cross another, some additional rules are needed for correct control.

To merge the vehicles it is checked whether the surrounding vehicle is within a certain angle in front of the vehicle within a certain distance. If so it is checked, between the two vehicles, which vehicle has the smallest angle to its neighbour, i.e. this vehicle blocks the path of the other vehicle the most. The other vehicle will brake, such that the blocking vehicle can drive further whereafter the other vehicle will resume its path. Using this, no forced rules such as "right side goes first" are needed. Each situation will be evaluated and solved differently.

For safe operation of the algorithm the weights, as used in Eq. (5), needed to be determined. For the roundabout design the original weights, $W_C = 0.1, W_A = 6.0$ and $W_S = 1.6$, did already a good job in managing the traffic. For the plaza design, the separation force needed to have some more effect on the vehicle's decisions leading to the following weights for good control: $W_C = 0.1, W_A = 4.5$ and $W_S = 2.2$.

D. Simulation

To create designs from Figs. 2b and 2c in the simulation environment, the side "walls" were described in the .wls file. The size of the road is chosen such that a flock with a width of three vehicles, as designed in the previous stage of the simulation, can enter the junction. This opens also the possibility to split the entering vehicles into the separate lanes as designed in Sec. III. With this, a suitable radius is chosen for the roundabout resulting in the designs which can be seen in Fig. 3. The entry and exit lanes are labeled based on their connection to the roundabout or plaza. The virtual walls, as described earlier, are shown in yellow. The exit walls (EW) are used in the roundabout to attract or repel the vehicles according to their labeling and the vehicle's destination.

The vehicles are spawned at the beginning of the entry lanes. As can be seen in Fig. 2, the entry lane is split into going right or going straight/left for the roundabout. For the plaza, the directions are split into three separate lanes. Therefore the spawned vehicles are sorted to the left, middle or right side

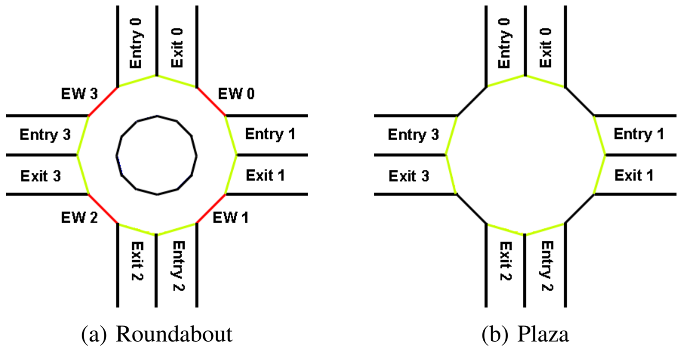


Fig. 3: Designs of the intersections used in the simulation

of the entry road according to their destination. This spawn is done periodically, some random offset is added to make the arrival times irregular. The destination of the vehicles is chosen randomly with equal division as well.

In a real world situation, vehicles share their parameters periodically over a wireless connection link. Based on the standard described in Sec. II-C the sending rate is determined. The surrounding vehicles stores this received data and uses it to determine its behaviour. However, the simulation assumes a perfect instantaneous communication link. Therefore, a structure has to be added to the simulator such that a sending rate can be simulated. This is done by reading other vehicle's data from a copy of the global buffer. This copy will be updated periodically such that the vehicles will determine their behaviour based on the last copied data in stead of the last calculated data of each vehicle by the simulation. The sending rate can be changed by altering the period between the updates of the copied buffer.

In the simulation, there are 50 steps each second, i.e. all the variables such as positions and speeds are calculated and updated 50 times per second. Using the global buffer to determine a vehicle's behaviour represents a sending rate of 50 Hz. To verify the algorithm, this rate was kept at 50 Hz. Later on, tests were conducted to find the minimum sending rate. This was done by changing the number of copies per second between the global buffer and the copy used to determine a vehicle's behavior. For steps where the copied data is outdated, it was tried to make predictions of the data based on the integration of the last known velocity. This however, resulted in worse results compared to using the outdated data. For the implementation of predictions, more precise calculations using acceleration and more history data are needed.

IV. Results and Discussion

As the research question stated, the main point of interest is the investigation of the concepts of bird flocking behaviour as a basis for the control algorithm. This idea was put to the test using the designed roundabout and plaza in the simulation described in Sec. III. The data shown in this section was gathered from the simulation environment. Every situation was simulated for 2 minutes starting with an empty intersection. The first vehicles reach the intersection 10 seconds after the start. Each situation was repeated 5 times such that there is 10 minutes of total simulation time per case. The confidence intervals are added in the figures for a confidence level of 95%.

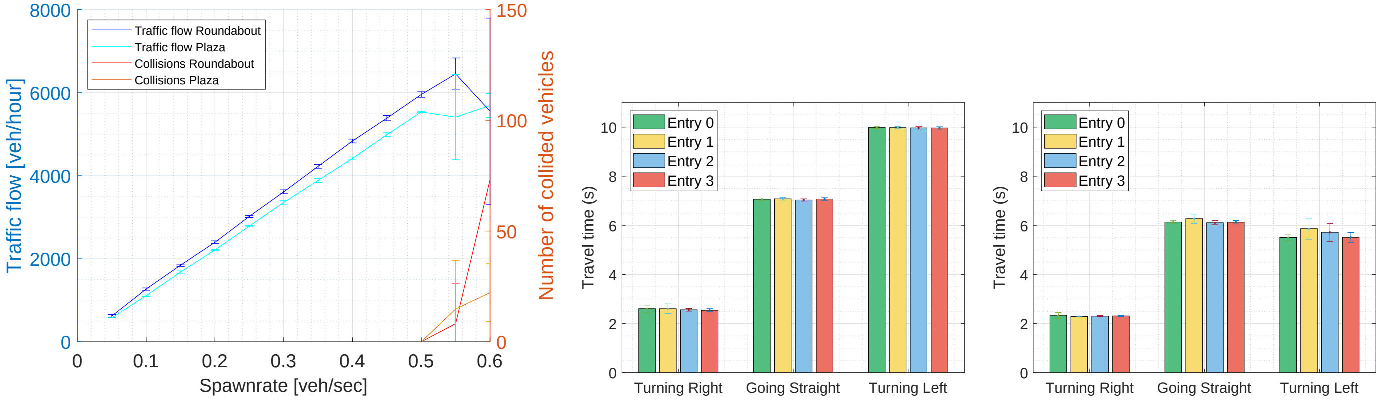
As the design stated, the bird flocking algorithm does not include braking. Therefore, it turned out that a control algorithm based on only the three forces of the bird flocking does not suffice enough for fully correct management. The first problem occurred when merging two lanes at the roundabout. A vehicle driving on the circular lane of the roundabout needs to be merged with an entering vehicle. Using only the forces these vehicles succeed in forming a flock where they are driving next to each other. However, when the vehicle on the left side of the flock needs to take an exit and the vehicle on the right side doesn't, the force attracting the leaving vehicle is stronger compared to the separation force of the other vehicle. Now, the leaving vehicle will steer into the right vehicle to move towards its exit. The situation is sketched in Fig. 9 in Appendix B. Lowering the attracting force of the exit doesn't solve the problem because then the force is too weak to attract vehicles without the merging problem. Adding more separation between the vehicles resulted in a situation where the right vehicle does indeed rotates away from the leaving vehicle. But due to the separation from the exit, which is not of its destination, the vehicle rotates such, that it is fully against the driving direction conflicting with the other vehicles. The second problem occurred at the plaza where the vehicle's paths to their exit crossed each other. Adding more separation results in both vehicle diverging from their path to the exit such that they will have to make longer routes to their exit. The problems are solved using the following rule. When both vehicles are close to the conflict point it is checked which vehicle blocks the other vehicle's path the least at that given time. This vehicle will brake and wait until the other vehicle passes the conflict point. If the other vehicle has passed, the waiting vehicle resumes its path.

Due to the first vehicle of a flock of vehicles needing to brake, a possible vehicle behind him also has to brake. Using the separation force of the algorithm will result in the vehicle passing the braking vehicle. This maneuver will lead to a collision with the vehicle in front of the braking vehicle. Therefore, a braking rule has been added for vehicles that have a braking vehicle close in front of them.

With these rules, vehicles from all directions were able to pre-sort themselves on the entry lanes and drive to their correct exit, which is either turning right, going straight or turning left. A picture of how the simulation looks like can be found in Appendices C and D. Using this simulation the necessary data was gathered which will be discussed in this section.

A. Roundabout

Different aspects such as safety, capacity and fairness, are important when evaluating the traffic junction. At first, the capacity of the design is determined. As explained in Sec. III, the empirical regression method is used. For this method, data is needed on traffic entering and leaving the roundabout. This



(a) Traffic flow and number of collisions of roundabout and plaza

(b) Travel time of vehicles on the roundabout per entry per destination

(c) Travel time of vehicles on the plaza per entry per destination

Fig. 4: Results of the simulations

data was easily gathered from the designed simulation. With logging the spawn point, destination, entry time and exit time the capacity and fairness of the roundabout can be computed.

The graph blue line in Fig. 4a shows the traffic flow of the roundabout vs. the spawn rates on its entrances. For the traffic flow calculation the spawn rates were set equal for all entry directions. With the data collected, the mean and confidence interval are plotted. At a spawnrate of 0.55 this confidence interval becomes quite big due to some of the simulated runs having collisions. Here, the roundabout wasn't able to manage the requested throughput resulting in queues which degrades the roundabout's traffic flow. The roundabout is too crowded such that vehicles cannot pass each other such that they have to wait or cause collisions with other vehicles. This can also be seen in Fig. 4a where the red line shows the number of collisions during the simulation. Therefore, for a reliable result and safe control on the roundabout, a spawn rate of 0.5 vehicles/second per entry can be seen as the maximum. This results in a capacity of 6000 vehicles/hour of the roundabout. Comparing this to the state of the art, where roundabouts have an average throughput of 3500 vehicles/hour [12], the simulation showed a significant increase which gives high potential in solving the current traffic congestion problems.

In Fig. 4b, the travel time on the roundabout is shown per entry and destination. All the data gathered for the traffic flow calculations was sorted by entry and their action, i.e. turning right, going straight or turning left. An average was taken for each entry and action which is plotted. The confidence intervals are shown but are really small for most of the cases. Here one can see that there is not much difference in travel time between entries with the same action. From this, one can conclude that, with equal spawn rates, the throughput over all entries is equally divided. The difference in travel time per action is visible. This is because, due to the roundabout orientation, vehicles turning left also have to drive by the exit to turn right and going straight. According to the data used for determining fairness, the average travel time for all vehicles, independent of their entry or destination, is 6.54 seconds. This means that an average vehicle will occupy some space on the circular lane of the roundabout for 6.54 seconds.

B. *Plaza*

Another approach to the traffic junction is the open plaza where vehicles don't have to follow a specific driving direction. For this we will also look at the safety, capacity and fairness. These parameters are determined in the same way as done for the roundabout using the designed simulation. A picture of the simulation during execution can be found in Appendix D. The traffic flow vs spawn rate graph of the plaza is shown in the cyan line in Fig. 4a. Here, again a drop can be seen after a spawn rate of 0.5 due to vehicles having to wait to go to their exit or occurring collisions. Again the confidence interval becomes quite big at this spawnrate. However, in this case, it seems that for higher spawn rates the traffic flow rises more. But, when looking into the number of collisions per spawn rate, shown in Fig. 4a with the orange line, one sees that for these spawn rates, safety comes into play. Therefore, with a spawn rate of 0.5 vehicles/second per entry directions the capacity of the plaza can be set at 5500 vehicles/hour. With this, the capacity of the roundabout is higher compared to the capacity of the plaza.

Now looking at the fairness of the plaza depicted in Fig. 4c we see a difference in the travel time for taking a left turn. This is because vehicles going left don't have to drive by the other exit as was the case in the roundabout design. Again, one can observe not much difference in travel time for the different entries so there are no priority lanes. It turned out to be that for making a left turn, some vehicles have to brake more often then the other actions. This braking results in difference in travel time within the same action which can be seen by the size of the confidence intervals.

An interesting insight when looking at Figs. 2c and 4c is the similarity in the number of conflict points to cross and the duration of the travel time. For turning right there are no conflict points to cross so the travel time is short. For going straight and turning left the number of conflict points is equal which comes back in almost equal travel time duration. These results give hope to the newly investigated throughput calculation method using the conflict points. Although there is a slight decrease in the capacity of the roundabout, the average travel time per vehicle is reduced from 6.54 seconds to 4.71 seconds. This is mostly caused by the drop in travel time when turning left.

All the previous results shown were obtained while testing the algorithm with the same spawn rates for each entry. In real life, however, roundabouts are created to connect the main road for ongoing traffic with a perpendicular road for local traffic. Here the throughput of the main road was set to 0.75 vehicles/second, much higher compared to the throughput of the perpendicular road set at 0.1 vehicles/second. Besides the higher throughput, the destination for most of the vehicles is going straight on the main road. The different throughput is set up in the simulation and the division of destination is set to 20% going right, 60% going straight and 20% going left. For the side road, the division of destination remains the same for each action.

The result of the test described can be seen in Figs. 5 and 6. In this figure, the entries of the main road are entry 1 (yellow) and entry 3 (red). One can see that in both designs, these entries have a bit faster or equal travel time compared to the other entries. Also the confidence intervals of the other entries are smaller then the main road entries. Due to the busy stream of vehicles going straight from the main road entries, the vehicles from the local traffic entries have are more likely to brake/wait. Therefore there is more deviation in their travel time, and thus a bigger confidence interval, and a slight increase in the average travel time. With this, we can conclude that, although the travel time of the local entries is be a bit higher and differ more, no hard priority paths are formed by the algorithm.

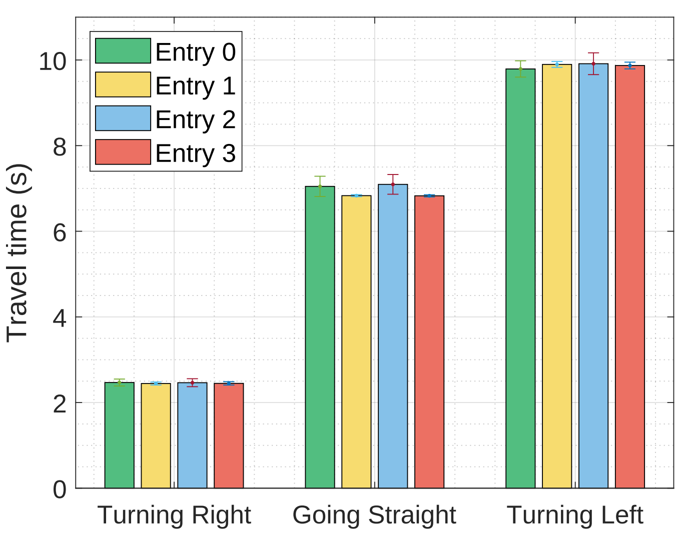


Fig. 5: Fairness of the roundabout with main and side roads

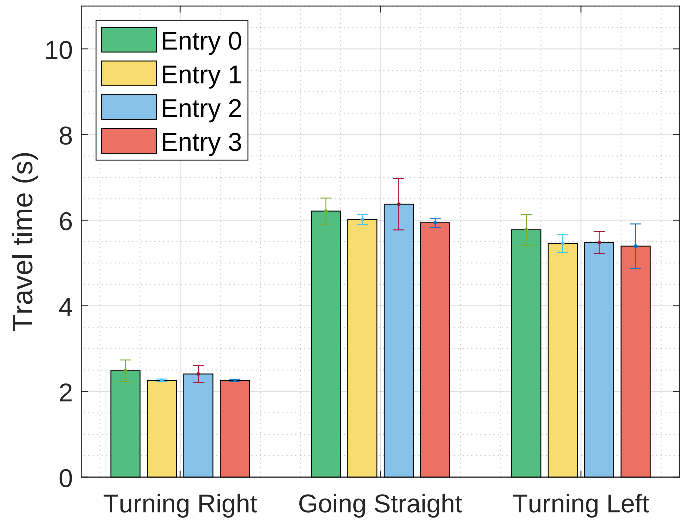


Fig. 6: Fairness of the plaza with main and side roads

With the algorithm proven, the next step is to look into the computational and network load. As stated in Sec. III-D a copy will be made of the global buffer with a certain period to simulate a certain sending rate. For this experiment the sending rate was altered until correct behaviour on the intersection occurred. Again the results were gathered by simulating five times a two minute simulation. At the beginning of the simulation the roundabout is empty. In Figs. 7 and 8 average of the number of collisions are shown over the duration of the simulation. This is shown such that one can see the influence of the number of vehicles on the roundabout on the required package rate. The higher the sending rates, the later in time the collisions occur to happen. This is due to the number of vehicles on the roundabout. In the beginning of the simulation, vehicles are relatively far away from each other so slightly outdated data of the neighbours is good enough to avoid collisions. When more vehicles enter the roundabout and the vehicles are coming closer to each other, more recent data is needed to avoid collisions. If the sending rate cannot provide enough up-to-date data, it will result in a collision. For the roundabout in Fig. 7, one can see that from a rate of 6 packets/second no collisions occur. For the plaza, Fig. 8, this rate lies at 13 packets/second. This is mostly due to the fact that in the plaza design, vehicles are approaching other vehicles from all different sides since there is no restricted driving direction. This increases the chance of collisions resulting in a higher needed sending rate to avoid these. Based on running the simulation environment five times and some safety margin included, on can conclude the following sending rates. For the roundabout design a sending rate of 8 Hz is enough for safe operation. For the plaza, this rate lies at 15 Hz. Looking at the standard in [8], the required packet rate for the roundabout lies within the boundaries. For the plaza however, the required rate lies above the maximum of the standard.

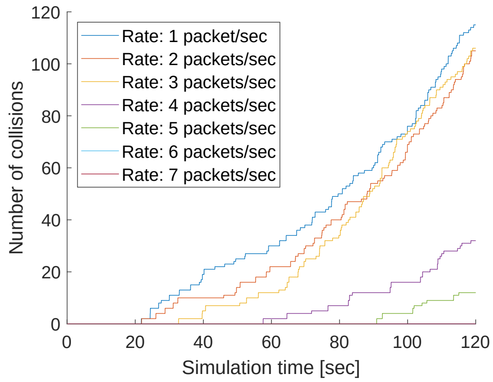


Fig. 7: Collisions due to outdated data on the roundabout

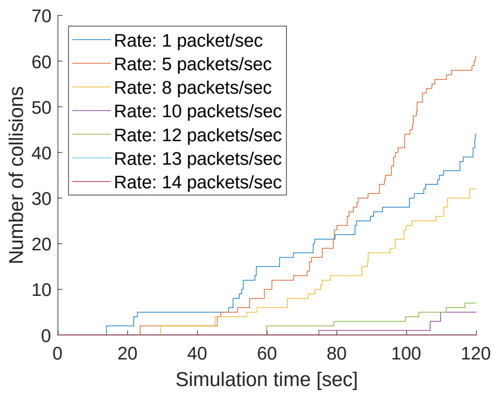


Fig. 8: Collisions due to outdated data on the plaza

V. Conclusion

With the need for an increased road capacity, smart autonomous driving algorithm can play a big role in solving this problem. With quite some research done in this field, gathered knowledge and tools can be used for more in-depth analysis. With prior research on implementing bird flocking behaviour, where movement is based on three forces: cohesion, alignment and separation, on vehicles, a basis was laid down for specific research such as road junctions. Using the knowledge and simulation environment from earlier research, a roundabout design was created based on the turbo-roundabout implementation of the present day. Using the correct combination of the three forces and some merging rules a successful algorithm was created for managing the vehicles. Here, a roundabout was achieved with a throughput of 6000 vehicles/hour where vehicles have an average travel time on the roundabout of 6.54 seconds.

Based on the results of this simulation a new design was made which removed the inner circle of the roundabout resulting in a plaza. With some slight changes in the algorithm and weights of the forces, the simulation succeeded in the tests. The results of this gave a capacity of 5500 vehicles/hour and an average travel time on the plaza of only 4.71 seconds.

Answering the research question: *How feasible is a control algorithm based on the bird locking behaviour for cooperative autonomous driving vehicles on roundabouts?*

This paper explains and verifies an algorithm based on the three forces of the bird flocking behaviour. For correct behaviour a merging rule was needed to be included in the algorithm. Therefore we can conclude that a control algorithm for roundabouts based on only the flocking behaviour of birds does not suffice for fully correct behaviour on the roundabout. However, it lays a really good basis in order to determine the directions vehicles have to move. With the addition of speed control by some additional rules a successful program is constructed and verified in a simulated environment.

For further investigation, research needs to be done in the field of the speed control of the vehicles on the intersections. Now vehicles brake last moment to merge and avoid collisions. With the implementation of a planning algorithm, vehicles can change their speed based on a planned time slot on the roundabout. With this, vehicles can better time their arrival such that merge problems are fixed in advance and no hard braking is needed. Another improvement can be made by better investigating the required sending rates. With a better implementation of predicting current data based on previous data, it is possible that a lower packet rate suffices for correct control. Another addition is loss rate of packets resulting in some missing data. With these additions, a more realistic simulation can be made which brings a step closer to the implementation of algorithms as such in real life.

References

[1] Eurostat, Passenger cars, by type of motor energy. [Online]. Available: https://ec.europa.eu/eurostat/databrowser/view/road_eqs_carpda/default/table?lang=en

[2] Wifitalents.com, Self Driving Cars Statistics: Latest Data & Summary [Online]. Available: https://wifitalents.com/statistic/self-driving-cars/

[3] SWOV (2009) The relation between speed and crashes. [Online] Available: https://fietsberaad.nl/CROWFietsberaad/media/Kennis/Bestanden/FS_Speed.pdf?ext=.pdf

[4] C. W. Reynolds, "Flocks, Herds and Schools: A Distributed behavioural Model," SIGGRAPH Comput. Graph., vol. 21, no. 4, pp. 25–34, 8 1987. [Online]. Available: https://doi.org/10.1145/37402.37406

[5] Lijcklama à Nijeholt, D.J. (2020) Control for Cooperative Autonomous Driving Inspired by Bird Flocking behaviour. [Online] Available: https://essay.utwente.nl/80590/

[6] Blondel, Joppe (2021) Distributed Control Algorithm for Cooperative Autonomous Driving Vehicles Inspired by Flocking Behaviour.[Online] Available: https://essay.utwente.nl/85648/

[7] Limbeek, T.F. (2023) Design and implementation of destination tracking and velocity control as an addition to a boids-inspired traffic simulation. [Online] Available: https://essay.utwente.nl/97664/

[8] EUROPEAN Standard, "Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 2: Specification of Cooperative Awareness Basic Service." Accessed: Jun. 17, 2024. [Online]. Available: https://www.etsi.org/deliver/etsi_en/302600_302699/30263702/01.04.01_30/en_30263702v010401v.pdf

[9] Tollazzi, T(2015) Alternative types of roundabouts: An informational guide. Heidelberg: Springer.

[10] Šarić A and Lovrić I (2017) Multi-lane Roundabout Capacity Evaluation. Front. Built Environ. 3:42. doi: 10.3389/fbuil.2017.00042

[11] N. Wu and W. Brilon, "Total Capacity of Roundabouts Analyzed by a Conflict Technique", Transportation Research Record: Journal of the Transportation Research Board, vol. 2672, no. 15, pp. 9–22, Aug. 2018, doi: 10.1177/0361198118788171.

[12] Overkamp, D. P., & van der Wijk, W. (2009). Roundabouts-Application and design, A practical manual, Royal Haskoning DHV. Ministry of Transport. Public Works and Water management, Partners for Roads.

APPENDIX

A. Declarations

During the preparation of this work the author used Grammarly and the overleaf built-in spell-checker in order to spell- and grammar-check the written text. During the preparation of this work the author used chatbot ChatGPT based on GPT-3.5 in order to write the Matlab code needed for generating the plots without adding any non-proven information. After using these tools/services, the author reviewed and edited the content as needed and takes full responsibility for the content of the work.

B. Merge problem

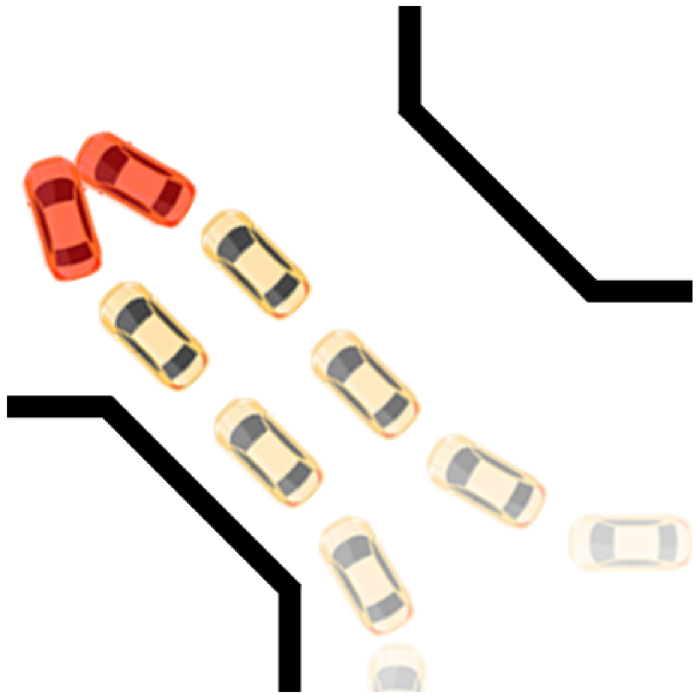


Fig. 9: Picture of the merge problem when only using the three forces

C. Picture of the roundabout simulation

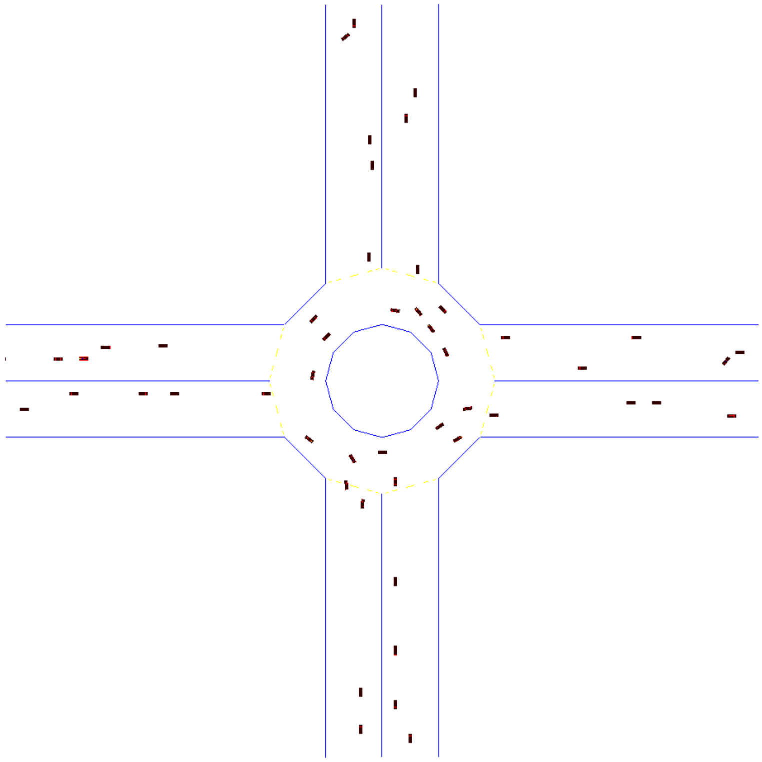


Fig. 10: Picture of working algorithm in roundabout simulation

D. Picture of the plaza simulation

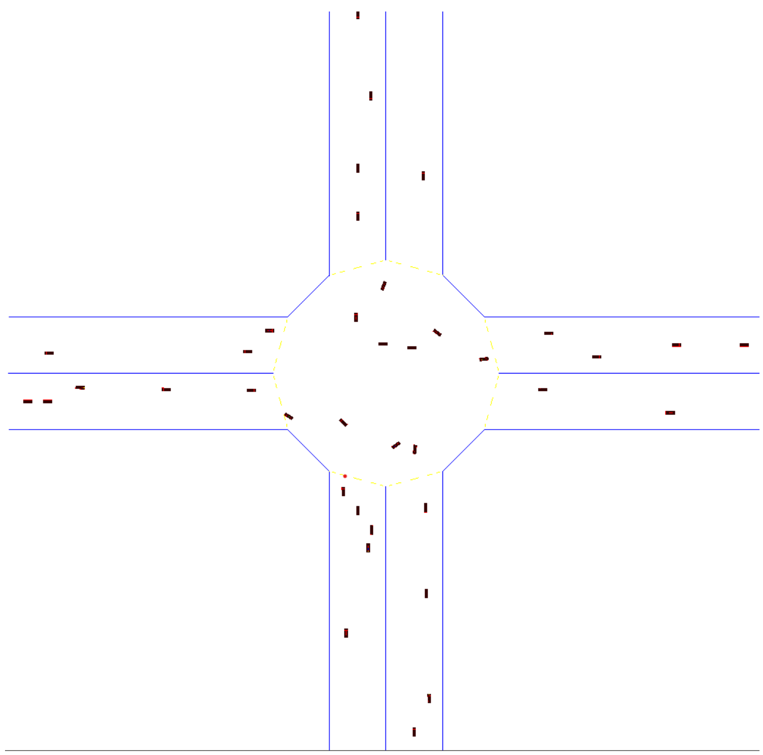


Fig. 11: Picture of working algorithm plaza in simulation