

Control for Cooperative Autonomous Driving Inspired by Bird Flocking Behavior

Daniël Lijcklama à Nijeholt
University of Twente
P.O. Box 217, 7500AE Enschede
The Netherlands

d.j.lijcklamaanijeholt@student.utwente.nl

ABSTRACT

Nowadays, it is impossible to imagine a world without cars. However, the widespread use of cars has several negative implications, such as accidents and greenhouse gas emissions. Recent scientific advances have introduced cooperative autonomous driving (CAD), which could alleviate these issues through danger-anticipation and fuel-efficient driving. However, there is a variety of challenges left before this can be fully implemented. One of these challenges is to develop a set of rules and algorithms to control how cars move. The research in this paper aims to explore the possibility of basing the control process in CAD on bird flocking behavior. A simulation was constructed based on the physical limitations of cars, incorporating the rules governing bird flocking to determine how these cars act. Experiments were conducted using this simulation to analyze the behavior and performance of cars controlled by bird flocking rules. In situations where the three rules are in balance, collision-free behavior is achievable. Furthermore, if appended with a fourth, goal-driven rule, flocks can travel collision-free to specified locations. However, further research is required to determine how this balance can be established without collisions.

Keywords

Cooperative Autonomous Driving, Autonomous Vehicles, Self-Driving Cars, Boid, Bird Flocking, Control Theory

1. INTRODUCTION

It is hard to imagine a world without cars. Regardless, the common occurrence of cars comes with a variety of negative consequences. The primary issue in traffic is safety. A study in 2018 by the National Highway Traffic Safety Administration has shown there have been almost 2.2 million car crashes in America over a span of 2.5 years. Moreover, in 94% of these cases, the driver was the critical reason for the crash [7]. Cooperative Autonomous Driving (CAD) removes the human from the equation, making it a potential solution to this issue.

Furthermore, a study by the Environmental Protection Agency has shown that in 2017 the transportation sector was responsible for 29% of all greenhouse gas emissions.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

32nd Twente Student Conference on IT Jan. 31st, 2020, Enschede, The Netherlands.

Copyright 2020, University of Twente, Faculty of Electrical Engineering, Mathematics and Computer Science.

Cars make up 59% of this sector and are therefore the primary source of greenhouse gas emissions in traffic [5]. Studies have already shown that the implementation of CAD can contribute to the reduction of these emissions [4].

Moreover, roads are not always able to handle the high number of traffic, resulting in congestion. Time spent in traffic jams is wasted time and therefore costs a lot of money [8]. While traffic jams cannot be prevented completely, less abrupt actions and better anticipation on actions could reduce the probability of congestion. This can be achieved by the implementation of CAD.

Research into CAD has been ongoing since the 1980s [8]. However, there are still a lot of challenges left before CAD can be fully deployed. The remaining challenges can be broken into four categories: technical, non-technical, social and policy [8]. One of the technical challenges is control, which involves specifying algorithms and rule-sets to determine the actions of vehicles independent of and relative to other vehicles. Each vehicle gathers information about its surroundings via sensors and wireless communication with other vehicles. Based on this information, the vehicle must decide how to act in any given situation. Constructing procedures that can handle unpredictable environments and perform optimally is the goal within control challenges. To ensure that CAD is collision-free and delivers the benefits it promises, these procedures have grown to be large sets of rules to deal with any traffic situation that might arise.

Looking at natural phenomena, guaranteeing collision-free behavior does not necessarily require a complicated set of rules. A primary example of this is the flocking behavior of birds. Each bird in a flock determines its actions based on three rules and on the actions of its direct neighbors [13]. Not only are such bird flocks a magnificent sight to see, rarely does a single bird fall down after colliding with another. If this control process is so simplistic for birds, why must it be so complicated for traffic? This is the question that drives the research proposed in this paper.

1.1 Research Questions

Formally, the primary research question to be answered is:

- What is the performance of cooperative autonomous driving when controlled by rules governing bird flocking behavior?

This question will be answered by addressing the following sub-questions:

1. What are the requirements for collision-free flocking behavior on a two-dimensional plane using car characteristics?

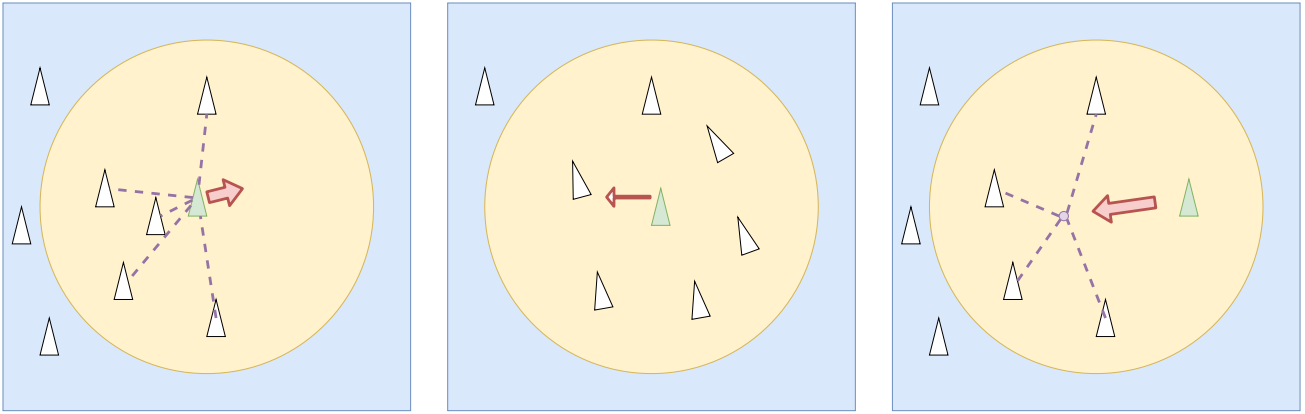


Figure 1. Boid rules visualization (from left to right: *Separation, Alignment & Cohesion*)

2. How does car flocking perform when goal-driven behavior is incorporated?

2. BACKGROUND

Scientists have been fascinated by the flocking behavior of birds since the dawn of time. However, the first empirical studies date back to the 1930s [2]. Most early studies are credited to biologists, but other fields started to show interest from the 1970s. In 1986, Reynolds laid the foundation for simulations of this flocking behavior [13]. Reynolds referred to such a simulation as a *Boid*, which is the common term to date. In this model, control is distributed over each individual bird in the flock. Each of these birds bases its actions on the actions of the birds in its direct neighborhood. The resulting behavior is based on three rules: *Separation, Alignment* and *Cohesion* [14]. Figure 1 visualizes these rules.

Separation dictates that birds should steer away from its neighboring birds, the weight of which increases as the distance between birds decreases. Because every bird in the flock abides this separation rule, every bird will generate space around itself. More space around a bird means that it has more leeway, which decreases the chance of collisions occurring. In a balanced flock, separation forces from all sides sum to zero, indicating that the bird is sufficiently separated from all its neighbors.

The second rule, alignment, dictates that any bird should adapt its direction and velocity based on its neighbors. This prevents birds from having to make sudden moves to prevent collisions. Moreover, without alignment, collisions might be unavoidable, since turning speed and deceleration are limited. If all birds maintain the same direction and velocity, the separation and cohesion forces will not change either, resulting in a balanced flock.

The primary purpose of the third and final rule, cohesion, is to ensure that the flock stays together. It dictates that any bird should steer towards the center of its neighboring birds. Therefore, if at any point some bird moves away from the flock significantly, it will be steered back towards the flock by this rule. In a balanced flock, each bird will be either exactly in the center of its neighboring birds, or the cohesion force is balanced out by the separation forces.

2.1 Related Research

Bird flocking behavior has inspired several research directions. Most notably into optimization algorithms, such as Particle Swarm Optimization [9]. Reza Olfati-Saber has provided a theoretical framework for the use of bird

flocking in multi-agent dynamic systems [12]. This framework has been the foundation for most research inspired by bird flocking. The only previous research found that applies these concepts to CAD was performed by Yasuhiro Hayashi and Toru Namerikawa [6], which focused primarily on basic flocking behavior.

3. METHOD

To obtain results to substantiate answers to the research questions posed, a simulation environment will be constructed in steps. Experiments will be performed on this environment to obtain these results. Each subsequent research questions will require an extension of this simulation. The rest of this section describes the capabilities of the simulation environment and the corresponding experiments for each research question.

3.1 Car Flocking

To answer the first sub-question, a simulation is required that closely resembles a two-dimensional Boid. There are two elements to this simulation: the cars and the two-dimensional plane they reside on. The cars will be modelled as circles to simplify the calculations necessary to determine collisions. Moreover, values will be obtained from literature for acceleration, deceleration and turning speed. The properties of the cars will be determined by both the physical limitations of cars as well as the limitations imposed by human comfort. The actions of the cars will be decided based on the three rules explained earlier. Since the cars will not have a goal at this stage, the velocity will be made constant to ensure that the cars stay in motion.

At this point, experiments will be run on the simulation to answer the first sub-question. The amount of collisions will be measured for several variations of the weights of each flocking rule. Moreover, the performance will be assessed visually, comparing the simulation to flocking behavior as observed in birds. Throughout this iterative process, observations might shed light on additional requirements or limitations to ensure collision-free behavior. Based on the results of these experiments, the first sub-question can be answered.

3.2 Goal-Driven behavior

For the second sub-question, the simulation needs to be extended to include goal-driven behavior. For a car, the goal is to travel from point A to point B as quickly as possible. Naturally, in order for this to happen, the car cannot collide in the process. Therefore, the simulation

will be extended with a fourth rule, which steers the car towards the goal. Like the previous rules, this rule will have weight attached to it. Moreover, this rule can be implemented in multiple ways. First, each car abides by this fourth rule and thus steers towards the goal individually. Second, some but not all cars abide by the fourth rule and the rest follow due to the cohesion rule. Third, a single leader will steer towards the goal, while the rest follow.

The extended simulation can then be used for experiments that allow sub-question two to be answered. Starting from the flocking situation of the previous sub-question, a destination will be provided, upon which the flock will start steering towards it. The weight distributions of the rules will again be varied. If collisions can be prevented sufficiently, other metrics will be used to compare the performance of each implementation, such as the time required for the flock to reach its destination and the maximum flock size for collision-free behavior.

4. SIMULATION DESIGN

The simulation consists of a world containing a variable number of cars, which is updated a variable number of times per second for a variable simulation time. Note that a higher number of updates per second increases the accuracy compared to reality, but also increases the necessary computation time. The world is modelled as an infinite plane but is initialized as a rectangle with specified height and width. These dimensions limit the initialization space for the cars and the visual representation of the simulation.

Collision detection is the most computationally heavy aspect of the simulation. Therefore, the implementation was simplified as much as possible. Two cars are considered collided if the distance between their mid-points are less than the length of the car. This will underestimate collisions around the corners of the car, while overestimating collisions to the sides of the car.

4.1 Car

There are two important aspects to consider for the model of the car: appearance and movement. The dimensions and limitations of the car are based on the specifications of a 2019 Toyota Camry LE [1], which is among the most popular cars in America [11]. The following constants describe the model of the car:

- Length: 4.9 m
- Width: 1.8 m
- Wheelbase: 2.8 m
- Maximum Steering Angle: 37°
- Steering Angle Change: 37° s^{-1}

The distance between the front of the car and the front wheels are modelled to be equal to the distance between the back of the car and the back wheels, which means that the center of the wheelbase is equal to the center of the car length. The maximum steering angle that the wheels can make is based on the turning radius of the Camry, but not limited by the velocity of the car. Therefore, at higher velocities, the model could steer sharper than physically possible. The steering angle change ensures that the steering angle does not change instantaneously, which would crash the car. For simplicity purposes, the steering angle change is modelled as a constant, but can be negative for reverse rotation.

The movement is based on the simple car model [10]. This kinematics model represents car movement in terms of position, direction and velocity, without considering forces. It consists of three formula that determine the change in x-direction, change in y-direction and change in car angle respectively. During every simulation step, these changes are applied for a given time step, after which the velocity and steering angle are updated according to the acceleration and steering angle change respectively.

Variable attributes of the car model include maximum velocity and acceleration. While the possible values of these attributes are limited by the specifications of the Camry, these limitations are not enforced in the simulation. Therefore, when performing experiments, reasonable values for these variables must be provided. Other variable attributes are the initial position, direction, velocity, acceleration, steering angle and steering angle change of a car.

4.2 Flocking

As described in section 2, flocking behavior can be achieved by combining three rules. In the simulation, the effect of each of these rules is modelled as a vector. The vectors resulting from the three flocking rules are based on the positions and directions of the six nearest neighboring cars. Research has shown that birds use the topological distance over metric distance to determine their neighbors [3]. Therefore, in the simulation, the same is done. The vectors used are always determined with respect to the car. Therefore, the point of origin is always the position of the car.

The separation vector is the sum of the separation forces that a car experiences from its neighbors. The direction of a separation force is equal to the direction from the neighbor to the car. The magnitude of this force is the inverse of the distance between the neighbor and the car. Therefore, shorter distances result in a larger separation force. Every force has equal weight in the summation, because the inverse function already ensures that shorter distances carry exponentially higher weight than longer distances.

The alignment vector is the sum of the directions of the neighbors. The direction of a car is modelled as a vector with length one. Therefore, when summing these directions, each neighbor carries equal weight. Moreover, if all neighbors face the same direction, the magnitude of the resulting vector will be maximal, increasing its weight relative to the other flocking forces.

The cohesion vector originates at the car in question, ending in the average position of its neighbors. Each neighbor has equal weight in this average position. Note that the magnitude of this vector by itself is high relative to the other two flocking vectors.

The weighted sum of these three vectors is calculated, which results in a vector signifying the direction a car should move. The smallest angle between this target direction and the current direction the car is steering towards is calculated, which corresponds to either a clockwise or counterclockwise rotation of the steering angle to reach the target direction. Note that the direction of the wheels is used instead of the direction of the car in the calculation of the smallest angle. This not only ensures that the target direction is reached as soon as possible, but also damps how much a car will oversteer.

4.3 Goal

The goal is modelled as a single point in space. There

can only be at most one goal at any given, which all cars will travel towards. Since it is inconvenient and unnecessary for a car to occupy this point exactly, a car will be considered to have reached the goal when it is within one car length of that point. Furthermore, since cars will arrive at the goal in a flock, it is infeasible for every car to come within one car length of this point. Therefore, a car will also have reached its goal if any of its neighbors has reached its goal. As a result, only one car of a flock must come within one car length of the goal, after which the entire flock will have reached the goal.

Goal-driven behavior is modelled analogously to the implementation of the flocking rules: as a vector. For a specific car, the goal vector originates at the position of this car and ends at the goal. Therefore, goal-driven behavior is stronger as the distance to the goal increases. This vector is incorporated into the weighted sum of the basic flocking vectors.

5. EXPERIMENT SETUP

This section describes the experiments conducted on the simulation. Four experiments were conducted on a scenario with basic flocking behavior only, which were repeated for a different scenario including a goal. The purpose of the first experiment was to optimize the weight distribution of the flocking rules. The second experiment analyses the performance over time. The third experiment analyses the performance for different maximum velocities. The fourth experiment analyses the performance for different car counts.

Two performance measures are used for each experiment. The first performance measure is the amount of collisions. The second performance measure denotes how dense the flock is. This is expressed as the mean squared error of a car to the center of its flock. Note that this can never be zero, since this would indicate that all cars occupy the same position. The squared error is used because a coherent flock is considered to take as little space as possible. Therefore, cars that extend this space at the edge of the flock are punished extra hard. Furthermore, this measure should only be used to analyze performance relative to other measurements, since the mean squared distance has no relevance as an absolute value. Performance is measured at the end of each time step.

Because performance is dependent on the initial positions and directions of cars, these are randomized. Therefore, experiments consist of a number of simulations and performance measures are obtained as statistics over these simulations. This random distribution of cars allows cars to overlap initially. Therefore, collisions can be recorded before the start of a simulation, which must be considered when analyzing the results.

5.1 Basic Flocking

In all experiments on basic flocking behavior, cars are initially placed in an area of 100 by 100 meters. The positions and directions of these cars are uniformly distributed. The initial velocity, steering angle and steering angle change are set to zero. Each simulation in this experiment will run for 30 seconds.

The first experiment was performed to determine a good weight distribution for the basic flocking rules. The car count was set to 25, each having a maximum velocity of 5.0 m s^{-1} and a constant acceleration of 1.0 m s^{-2} . Every possible combination of weights for the basic flocking rules in the range [1, 3, 9, ..., 729] was used in 20 simulations with 50 simulation steps per second (343 sets of 20 simula-

tions). This range was established through visual analysis. The total amount of collisions at the end of a simulation was used as the performance measure. From the results, the weight combination with the least amount of collisions on average was chosen, which was [243, 9, 3] for [separation, alignment, cohesion] respectively. The rest of the experiments use these weights for the corresponding rules.

The goal of the second experiment is to visualize the performance change over time. It uses the same scenario as the first experiment, with the weight distribution now constant. This scenario was simulated 100 times with 100 steps per second to increase the accuracy of the experiment relative to the first one.

In the third experiment, the performance was measured for different maximum velocities in the range [3, 6, 9, ..., 30]. The constant acceleration was increased from 1.0 m s^{-2} to 2.0 m s^{-2} . For each possible maximum velocity, the scenario was simulated 50 times with 50 steps per second.

The fourth experiment looks into the performance of basic flocking behavior for various amounts of cars in the range [10, 20, 30 ..., 100]. The rest of the scenario is equal to the scenario described for the first experiment. The scenario was simulated 50 times with 50 steps per second for each possible car count.

5.2 Goal-Driven Behavior

In all experiments on goal-driven behavior, the scenario will span an initial area of 300 by 100 meters. This area can be split into three square sub-areas of 100 by 100 meters. The leftmost of these sub-areas will contain the cars initially, uniformly distributed, analogous to the experiments on basic flocking behavior. The initial velocity, steering angle and steering angle change are set to zero. The goal will be placed in the center of the rightmost of these three sub-areas. Therefore, a minimum of 150 meters must be travelled before the goal can be reached by any car. Each simulation runs until the goal has been reached, after which it continues for 30 more second. Therefore, the total simulation time varies per simulation.

The goal of the first experiment was to determine a good weight for the newly incorporated goal-driven rule, relative to the weights for basic flocking. The car count was set to 25, each having a maximum velocity of 5.0 m s^{-1} and a constant acceleration of 1.0 m s^{-2} . Every possible weight for the new rule in the range [0.1, 0.2, 0.4, 0.8, 1. 2, 4, 8] was used in a simulation 50 times with 50 simulation steps per second. This range was determined through visual analysis. The total amount of collisions at the end of a simulation was used as the performance measure. From the results, the weight with the least amount of collisions on average was chosen. This was 0.4, which is the value used throughout the rest of the experiments.

The other three experiments conducted for goal-driven behavior are exactly the same as the three experiments performed for the basic flocking behavior, replacing only the scenario with the scenario described for goal-driven behavior.

6. RESULTS

This section describes the results obtained from performing the experiments described in the previous section.

6.1 Basic Flocking

Figure 2 shows how the total amount of collisions progresses over time. As explained in section 5, the random distribution of cars enables collisions before the simulation

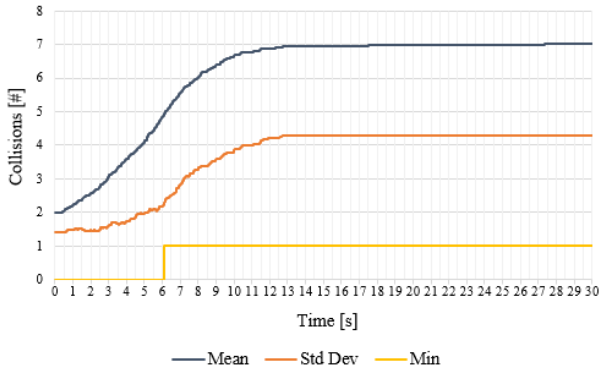


Figure 2. Cumulative collision count over time with basic flocking behavior

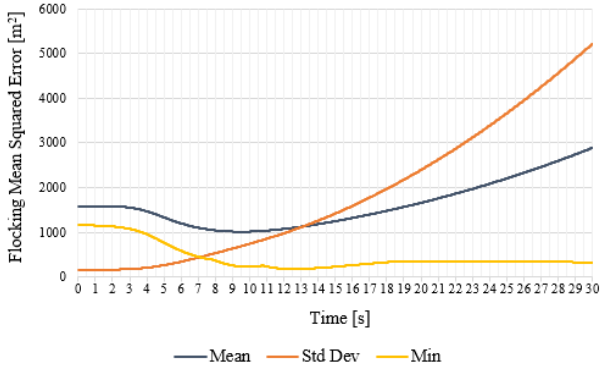


Figure 3. Flocking effectiveness over time with basic flocking behavior

starts. This is visible in beginning of the graph, where on average 1.98 cars have collided before simulation start. As time progresses, the amount of collisions increases. This is another result of the random car distribution. This distribution limits the possible trajectories of cars. Therefore, in some situations, it is inevitable for cars to collide.

The graph shows a clear difference in collision count progression before and after the halfway point. This is because the cars need time to form a balanced flock, which requires the separation and cohesion rules to be balanced out, allowing alignment to take over and maintain that balance. This search process also contributes to the collision count. However, once this balance has been found, collisions rarely occur. After 15 seconds, an average of 6.95 collisions have occurred, increasing to 7.01 collisions at the end. This means that 6 collisions have occurred in 15 seconds, spread out over 100 simulations.

Figure 3 shows how the density of the flock changes over time. Because the cars have not yet formed an effective flock in the first 15 seconds, the measurements in the graph do not correctly reflect how effective the flock is during that time period. The minimum plot shows how dense a flock can be and that a flock is able to maintain this density over time.

However, on average, the flocks drift farther apart over time. Moreover, the spread of data increases alongside the mean. This can be explained by the absence of a common goal. If cars are distributed such that there is a large enough gap between two sets of cars, these two sets will not share neighbors. Therefore, the sets of cars will

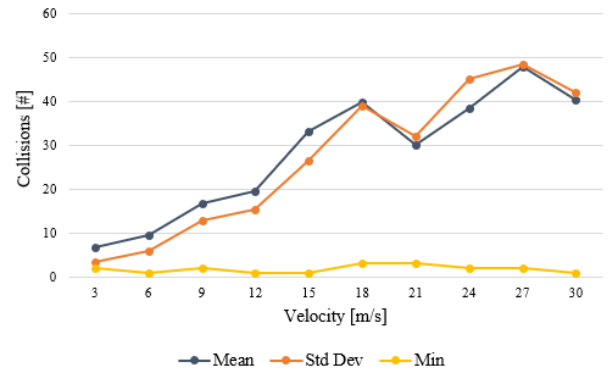


Figure 4. Cumulative collision count for different velocities with basic flocking behavior

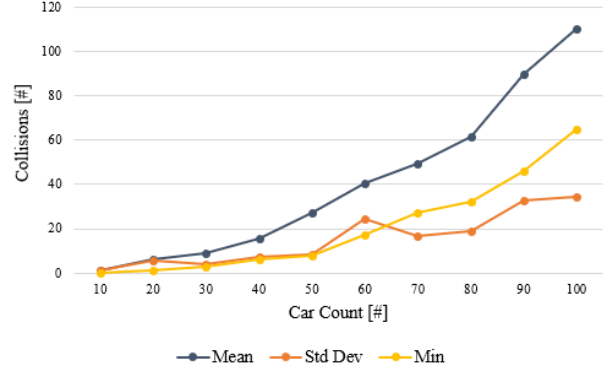


Figure 5. Cumulative collision count for different car counts with basic flocking behavior

act independently. Without a common goal, these sets of cars can steer away from each other. Unless the sets move closer, the sets will never share neighbors and will continue to steer away. However, since these cars are considered to be one flock, the mean squared distance to the center will increase continuously, which is what is visible in the graph.

Figure 4 shows the total amount of collisions at the end of a simulation for different maximum velocities. Overall, the amount of collisions increases for higher velocities. This has to do again with the formation of a balanced flock. At higher velocities, cars have less time to anticipate trajectories of other cars, increasing the amount of collisions that occur. Note that this is the only aspect impacted by higher velocities. After the 15 second mark, 0.04 collisions occur on average at 15 m s^{-1} and 30 m s^{-1} , 0.02 collisions at 21 m s^{-1} , 24 m s^{-1} and 27 m s^{-1} , while no collisions occur at the remaining velocities.

Figure 5 shows the total amount of collisions at the end of a simulation for different car counts. Additional cars increase the average amount of collisions, because it both increases the probability of two cars spawning on top of each other as well as the probability of unavoidable collisions due to trajectory limitation. However, an increase in car count does not impact the amount of collisions occurring after 15 seconds. Only at car counts 40 and 60 was a single collision detected out of 50 simulations, while other car counts were collision-free after 15 seconds.

Figure 6 shows the flocking density for different velocities. It shows that the cars drift further apart on average as the maximum velocity increases. This is explained by

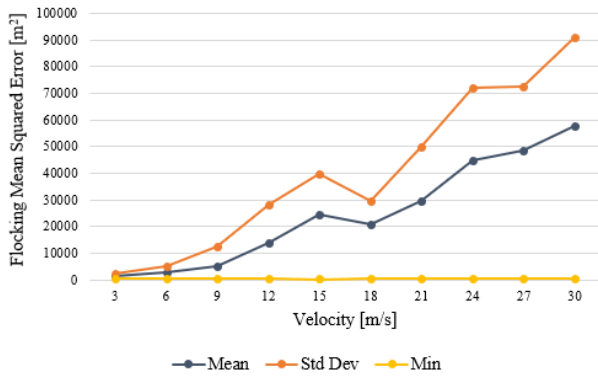


Figure 6. Flocking effectiveness for different velocities with basic flocking behavior

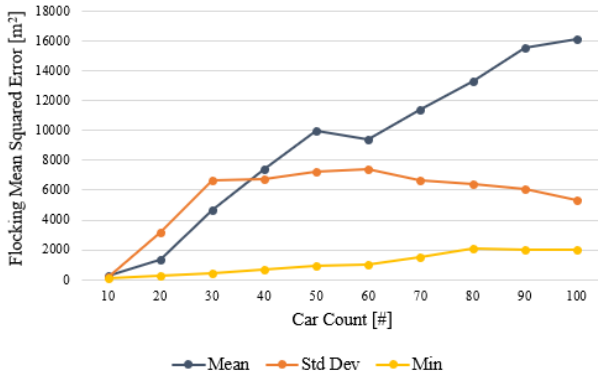


Figure 7. Flocking effectiveness for different car counts with basic flocking behavior

the occurrence of separate sets of cars, as mentioned before. Higher velocities increase the distance cars can travel within the simulation time. Therefore, there will be more distance between these separate sets of cars, increasing the flocking error.

Whereas an increase in velocity results in worse flocking density, an increase in car count enables better flocking density, which is visible in figure 7. Note that more cars require more space regardless of flocking, resulting in relatively higher errors for higher car counts on that basis alone. This is expressed by the minimum error, which increases with the car count. Moreover, this explains why the average error increases as well. However, note how the standard deviation stops increasing significantly with more than 30 cars. This is due to the fact that larger car counts decrease the chance that separate sets of cars arise.

6.2 Goal-Driven Behavior

Figure 8 shows the progression of collision counts over time when goal-driven behavior is incorporated. Note that the total duration of each simulation is different, depending on the time required to reach the goal. On average, this took 36.92 seconds. The fastest simulation took 32.43 seconds to reach the goal, whereas the slowest took 50.06 seconds. Therefore, the fastest simulation was done after 62.43 seconds. For this reason, all data after this time was excluded from the plot.

The plot has roughly the same shape as the corresponding plot for basic flocking behavior. The main difference is found at the end of the graph. Towards the end, the average collision count starts to increase again. This is be-

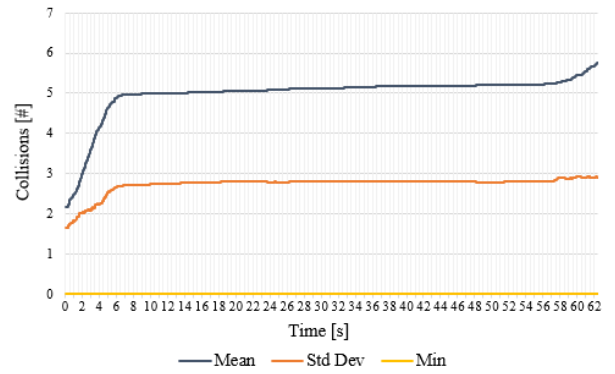


Figure 8. Cumulative collision count over time with goal-driven behavior

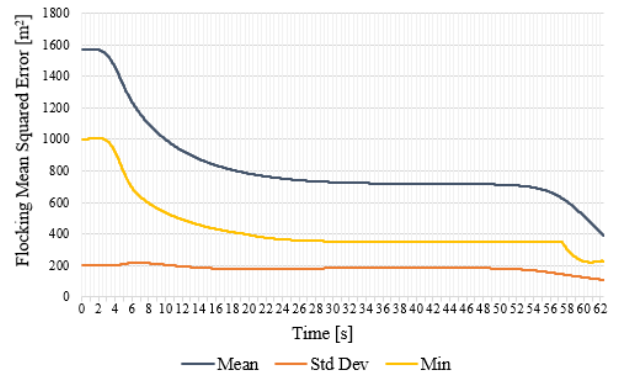


Figure 9. Flocking effectiveness over time with goal-driven behavior

cause the goal-driven force persists after the goal has been reached, resulting in flocking behavior around the goal. However, since a new equilibrium must be found, much like in the initial phase, the collision count goes up. Note that there is a delay between this change occurring and the time the goal is reached. When close to the goal, the goal-driven force is at its smallest. Therefore, cars passing the goal will not immediately turn around, but will rather stay aligned with their neighbors. However, once a car has travelled far enough past the goal, the goal-driven force will start forcing it to turn around. The time required for this to happen is the delay noticed earlier.

When a goal is included, a simulation can be separated in three phases. The first two phases coincide with the two phases in basic flocking behavior. However, note that the average collision count is around five now, which was around seven before. Between 15 and 30 seconds, the average collision count increases from 5.02 to 5.11, which is slightly more than before. However, excluding flocking behavior in the third phase, goal-driven behavior decreases the amount of collisions that occur.

Figure 9 shows the flocking density over time when a goal is included. This data was obtained from the same experiment as the previous plot. Therefore, the same holds true for the variable simulation time, as explained. Compared to the corresponding graph for basic flocking behavior, it is clear that goal-driven behavior completely removes the possibility for separate sets of cars to arise. The difference in errors is caused entirely by the random distribution of cars initially. Cars far away from the goal will never be able to catch up to cars close to the goal, since velocity is

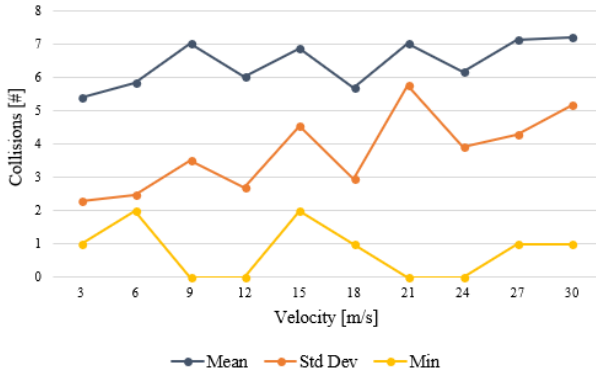


Figure 10. Cumulative collision count for different velocities before goal reached

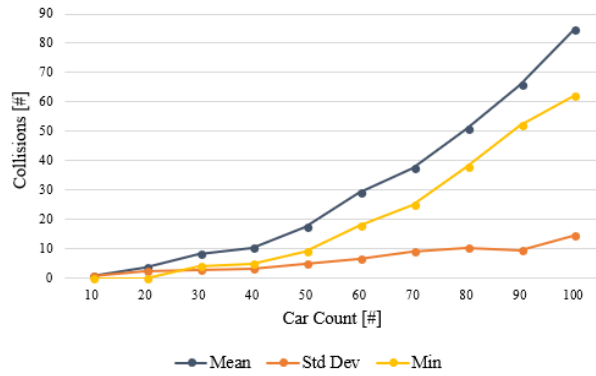


Figure 11. Cumulative collision count for different car counts before goal reached

constant, resulting in a stretched-out flock, increasing the error.

Near the end of the simulation, when the flock starts circling around the goal, the flocking density becomes even better. This is due to the fact that the cars lagging behind are allowed to catch up, since the front of the flock will turn back around towards the goal. However, since the collision count also increases at this stage, the error measurement will not be accurate beyond this point. Moreover, since cars are not required to move once the goal has been reached in reality, behavior in this third phase will not be further analyzed in this section.

Figure 10 shows the collision counts for different maximum velocities measured at the point in time when the goal is reached. With the inclusion of a goal, changes in velocity appear to have no significant impact on the amount of collisions that occur. This is explained by a side-effect of the goal force. Generally, a flock stabilizes once the balance between separation and cohesion has been found, allowing alignment to be the dominating force. Since all cars in the flock have the same goal and have to travel roughly the same direction, the goal force will have the same effect as the alignment force. Therefore, the flock takes a shortcut to alignment. However, the initial placement and trajectories of cars still make collisions inevitable.

Figure 11 shows the collision counts for different car counts measured at the point in time when the goal is reached. The shape of the plot is comparable to the shape of the plot of the basic flocking counterpart. For each car count, the amount of collisions is reduced slightly in comparison. This is because the goal force produces relatively more

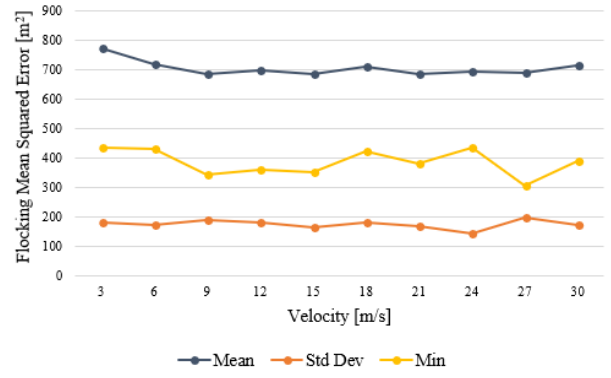


Figure 12. Flocking effectiveness for different velocities when goal reached

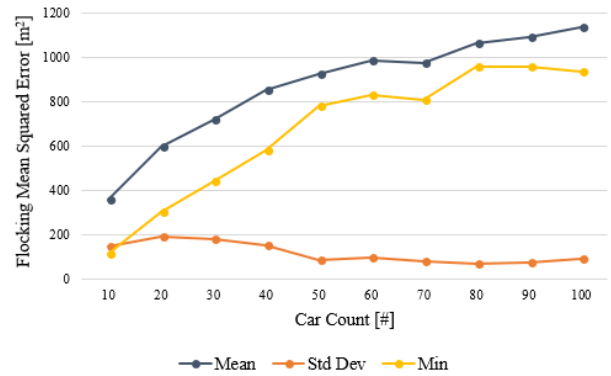


Figure 13. Flocking effectiveness for different car counts when goal reached

favorable trajectories for the cars, preventing some of the collisions that would otherwise have occurred. However, as explained, the initial placement of cars makes it near impossible to prevent all collisions from occurring.

Figure 12 shows the flocking density for different velocities measured at the point in time when the goal is reached. This plot further proves that goal-driven behavior completely prevents a flock from partitioning. It also shows that the maximum velocity has no significant impact on the density of the flock. The spread in the measurements is caused solely by the initial positions and directions of the cars, because cars are unable to accelerate or decelerate to increase flock density.

Figure 13 shows the flocking density for different car counts measured at the point in time when the goal is reached. As before, the initial placement of cars determines fully how dense the flock is. The increase in error as the car count increases is due to the fact that more cars take up more space. For lower car counts, the flocking error is small, because there is a lot of space available for each car, decreasing the chance of collisions. However, for higher car counts, the flocking error does not change significantly, while the total available space does not change either. Therefore, the probability for collisions increasing drastically.

7. DISCUSSION

A major flaw of the simulation used in this research is the lack of a proper controller for the car. During each time step, a car determines which target direction it should flock to. The control goal is to align the direction of its

wheels with this target. However, in the current simulation, the only control parameter is the steering angle change. Moreover, this steering angle change can only be set to a constant value, or its negative counterpart for the opposite rotation. Furthermore, the car does not anticipate the change over time. Therefore, the car will oversteer heavily and then slowly oscillate towards the target. Oversteering is likely to contribute to the collision count and should therefore be prevented. This requires an optimized controller to be implemented.

For simplicity purposes, acceleration was kept constant in the simulation used, since acceleration as a control parameter is less important and more complicated to implement relative to steering. However, with an optimized controller, acceleration and deceleration could be controlled to prevent collisions. Moreover, this would allow flocks to increase their density, because trailing cars could catch up to the rest, while leading cars could allow this by decelerating.

The goal-driven force also requires a revision. Throughout the experiments conducted, the distance to the goal was kept the same. However, if this distance were to increase, the goal force would increase with it. If this distance grows large enough, the resulting goal force could completely overtake the basic flocking forces, causing cars to ignore each other and focus solely on the goal. Therefore, the relative weight of the goal force cannot be linear.

Another limitation of this simulation was previously mentioned; the maximum steering angle. A constant value is used, based on the turning radius specified for the Camry. However, in reality, the maximum steering angle decreases as velocity increases to prevent the car from crashing. The same holds true for the maximum steering angle change, which could also crash a car if too high. It is infeasible to incorporate these aspects into a kinematics model as used in this research. Therefore, a dynamics model is required over a kinematics model to obtain a more accurate representation of reality.

8. CONCLUSIONS

The three basic flocking rules suffice as a control scheme to prevent cars from colliding. However, this only holds true if a balance is established between the separation and cohesion forces, while the cars are all aligned. The process of establishing this balance is only collision-free if the initial positions and directions of the cars are favorable. Cars facing the same direction are likely to collide using the basic flocking rules alone. Moreover, cars positioned too close to each other can experience infeasible flocking forces, putting them on dangerous trajectories. Furthermore, without an optimized controller, oversteering will occur and serve to amplify these effects. Therefore, the weighted sum of flocking vectors suffices once a flock is balanced but is too simplistic to handle complex starting positions.

The inclusion of goal-driven behavior makes it easier for flocks to find a balance between the flocking rules. The performance of car flocking improves as a result. The presence of a goal decreases the amount of collisions while searching for this balance. However, since it does not mitigate them entirely, further research is required to address this issue.

Based on this research, there are several recommendations for future research. First, the simulation environment itself could be improved by, for example, replacing the kinematics model by a dynamics model. This would increase

the reliability of the data obtained from the simulations. Second, the control process of the cars can be improved, as mentioned in the discussion, to further increase the performance. As a result, balanced flocks will likely never collide. Third, the transition from stationary cars to a balanced flock and the behavior once the goal is reached needs to be investigated further. Fourth, virtual walls should be incorporated, which cars steer away from. These can be used to model roads, serving as a basis for the simulation of realistic scenarios. Some functionality was already implemented, but progress was discontinued due to time limitations.

9. REFERENCES

- [1] Auto123. 2019 Toyota Camry | Specifications - Car Specs | Auto123.
- [2] I. L. Bajec and F. H. Heppner. Organized flight in birds, 10 2009.
- [3] M. Ballerini, N. Cabibbo, R. Candelier, A. Cavagna, E. Cisbani, I. Giardina, V. Lecomte, A. Orlandi, G. Parisi, A. Procaccini, M. Viale, and V. Zdravkovic. Interaction ruling animal collective behavior depends on topological rather than metric distance: Evidence from a field study. *Proceedings of the National Academy of Sciences of the United States of America*, 105(4):1232–1237, 1 2008.
- [4] J. Conlon and J. Lin. Greenhouse Gas Emission Impact of Autonomous Vehicle Introduction in an Urban Network. *Transportation Research Record*, 2673(5):142–152, 5 2019.
- [5] U. EPA. Fast Facts on Transportation Greenhouse Gas Emissions | Green Vehicle Guide | US EPA, 2019.
- [6] Y. Hayashi and T. Namerikawa. Flocking algorithm for multiple nonholonomic cars. In *2016 55th Annual Conference of the Society of Instrument and Control Engineers of Japan, SICE 2016*, pages 1660–1665. Institute of Electrical and Electronics Engineers Inc., 11 2016.
- [7] N. Highway Traffic Safety Administration and U. Department of Transportation. TRAFFIC SAFETY FACTS Crash • Stats Critical Reasons for Crashes Investigated in the National Motor Vehicle Crash Causation Survey. Technical Report March, U.S. Department of Transportation, 2018.
- [8] R. Hussain and S. Zeadally. Autonomous Cars: Research Results, Issues, and Future Challenges, 4 2019.
- [9] J. Kennedy and R. Eberhart. Particle swarm optimization. In *Proceedings of ICNN'95 - International Conference on Neural Networks*, volume 4, pages 1942–1948, 1995.
- [10] S. LaValle. 13.1.2.1 A simple car, 2006.
- [11] M. Macesich. Most-popular SUVs, trucks, cars in America right now - Santander Consumer USA.
- [12] R. Olfati-Saber. Flocking for multi-agent dynamic systems: Algorithms and theory. *IEEE Transactions on Automatic Control*, 51(3):401–420, 3 2006.
- [13] C. W. Reynolds. Flocks, herds, and schools: A distributed behavioral model. In *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1987*, pages 25–34. Association for Computing Machinery, Inc, 8 1987.
- [14] C. W. Reynolds. Boids (Flocks, Herds, and Schools: a Distributed Behavioral Model), 2001.