NTNU
Norwegian University of
Science and Technology

# A Reciprocal Collision Avoidance Algorithm for Nonholonomic Vehicles with Constant Forward Speed

Erlend Hårstad

Master of Science in Cybernetics and Robotics
Submission date: June 2018
Supervisor: Kristin Ytterstad Pettersen, ITK
Co-supervisor: Martin Syre Wiig, FFI

Norwegian University of Science and Technology
Department of Engineering Cybernetics

# Problem Description

Autonomous vehicles are increasingly used in both scientific and commercial applications. During autonomous or semi-autonomous operations, the capability to avoid other vehicles without human intervention is crucial for mission success and vehicle safety. In complex environments shared by multiple reactive vehicles, the vehicles must react quickly while also considering the reactive nature of other vehicles. Hence, there is a need for a reciprocal collision avoidance algorithm. Many vehicles, such as cars, can be modeled as a vehicle with nonholonomic constraints, i.e. they can move forwards and turn, but not move sideways. Such a model can also be used as a simplified model of a ship or a fixed wing aircraft, however these vehicle have a limited speed envelope and may have significant constraints on the forward acceleration due to high mass.

This master assignment builds on a project where a modification to the distance criterion of the constant avoidance angle algorithm was developed and tested in simulations. The goal of the master assignment is to further develop the algorithm to cope with the reactive nature of other vehicles i.e. extend the use of the algorithm to the case of reciprocal collision avoidance. The following subtasks are proposed:

- Further modify the algorithm to cope with the added challenge of other reactive vehicles
- Ensure safe and predictable collision avoidance by designing the algorithm such that the vehicles respects the International regulations for preventing collisions at sea (COLREGS)

- Verify the performance of the algorithm though testing in simulated scenarios
- Provide an analysis of the vehicle's behavior and the performance of the algorithm

# Abstract

There have been an increased interest in autonomous vehicles in recent years following advancements in technology. Such advanced systems requires robust and reliable Collision Avoidance (CA) systems. Reactive Collision Avoidance (RCA) algorithms assume that there is no central coordinator or direct communication between vehicles and only requires local sensor data which can be obtained from on-board sensors. Many vehicles such as cars and ships are limited by nonholonomic constraints i.e. they can move forward and turn, but not move sideways. Large ships also have a limited speed envelope and significant constraints on their forward speed due to high mass. This thesis presents an reciprocal CA algorithm for nonholonomic vehicles with constant speed. The algorithm uses the concept of Collision Cone (CC) to detect possible collisions. The focus remains on vessels at sea and the algorithm is designed such that the CA maneuver is carries out in a predictable and safe manner by respecting the international regulations for preventing collisions at sea, also known as Collision Regulations (COLREGS). The algorithm considers the reactive nature of other vessels and is designed so that vehicles share the responsibility of avoiding a collision between them in a fair way. Issues such as reciprocal dances and deadlocks are common for reciprocal CA algorithms. The proposed algorithm is designed to minimize the occurrence of such effects. Simulations illustrating how the algorithm respect the COLREGS are given and the performance of the algorithm have been extensively tested and verified through Monte Carlo simulations. Strengths and weaknesses of the algorithm are identified and suggestions for improvements are given.

# Sammendrag

Som følge av teknologiske framskitt i nyere tid har interessen for autonome kjøretøy økt betraktelig. Slike avanserte systemer krever robuste og pålitelig systemer for kollisjons-unngåelse (CA). Reaktive algoritmer for kollisjons-unngåelse (RCA) antar at det ikke finnes en sentral koordinator eller direkte kommunikasjon mellom ulike fartøy. Kun lokale sensor-data som er trivielt å hente fra sensorer på kjøretøyet er nødvendig.
Mange fartøy som biler og skip er begrenset av ikke-holonomiske egenskaper. De kan bevege seg framover og svinge, men kan ikke bevege seg sidelengs. Store skip har i tillegg betydelige aksjelerasjonsbegrensinger grunnet høy masse.
I denne masteroppgaven presenteres en algoritme for gjensidig CA for ikke-holonomiske kjøretøy med konstant fremdriftshastighet. Algoritmen bruker konseptet kollisjons-kjegler (CC) for å identifisere potensielle kollisjoner. Fokuset for oppgaven er større sjøgående fartøy og algoritmen er designet slik at fartøyene følger de Internasjonale reguleringene for å unngå kollisjoner på sjøen, kort kalt COLREGS. Algoritmen tar hensyn til den reaktive naturen til andre kjøretøy og er designet slik at ansvaret for å unngå en kollisjon er delt mellom kjøretøyene på en rettferdig måte. Gjensidige danser og vranglås mellom kjøretøy er vanlige utfordringer for slike CA algoritmer. Den foreslåtte algoritmen er designet for å minimere slike effekter. Simuleringer som viser hvordan kjøretøy respekterer reglene i COLREGS er gitt sammen med Monte Carlo simuleringer for å verifisere algoritmen. Styrker og svakheter er blitt identifisert og forslag til forbedringer er foreslått.

# Preface

This master's thesis is written as a compulsory part of a two-year Master's program in Cybernetics at the Department of Engineering Cybernetics at the Norwegian University of Science and Technology (NTNU) in collaboration with Forsvarets Forskningsinstitutt (FFI).

This thesis is an extensions of a project thesis carried out in the fall of 2017 and is based on the same Reactive Collision Avoidance (RCA) algorithm. The project presented alterations to the RCA algorithm to improve performance in cluttered environments. However, these alterations made some assumptions on the behavior of the obstacle which is not trivial to make when the obstacles is reactive, as is the case in this thesis. Thus, the main results of the project is not included in this thesis. The project also included a literature review of the field of RCA and a presentation of the RCA algorithm. These parts are included in this thesis. The simulator used to verify the performance of the proposed algorithm in this thesis is originally created by Martin Syre Wiig. A full review of the background material is given in Section 1.2

I would like to thank Martin Syre Wiig (FFI) for all the great help and guidance throughout this thesis. I would also like to thank Kristin Y. Pettersen (NTNU) for supervising this thesis.

Trondheim, June 06, 2018
Erlend Hårstad

# Contents

# List of Tables

# List of Figures

# Abbreviations

**APF** Artificial Potential Fields. 14, 15, 25

**B-ORCA** Bicycle Reciprocal Collision Avoidance. 22, 25, 46

**CA** Collision Avoidance. iii, v, 1–3, 5, 6, 16, 17, 19, 20, 22, 23, 25, 35, 36, 39, 40, 43–47, 49, 52, 56–62, 65, 66, 68–72, 75, 76, 83, 86, 87, 89, 90, 92

**CC** Collision Cone. iii, v, 6, 8, 32, 37, 39, 40, 43, 44, 46, 49, 52, 62, 66, 67, 69–72, 92

**COLREGS** Collision Regulations. iii, v, xiii, xv, 2, 25, 27, 40–45, 47, 49–55, 57, 59, 66–69, 71, 75, 76, 90

**CPA** Closest Point of Approach. 28, 44, 45

**DNF** Did Not Finish. 58–61, 63, 67, 68, 71, 85

**DW** Dynamic window. 15, 16

**HRVO** Hybrid Reciprocal Velocity Obstacle. 19, 20, 46

**IMO** International Maritime Organization. 2, 25, 38

**ORCA** Optimal Reciprocal Collision Avoidance. 19–23, 40, 46, 66

**RCA** Reactive Collision Avoidance. iii, v, vii, 1–3, 5, 16, 32, 37, 44

**RR** Reserved Region. 23–25

**RVO** Reciprocal Velocity Obstacle. 17–20, 40, 46, 66, 67, 71

**VO** Velocity Obstacle. 9–13, 17–20, 22, 39, 44, 46, 66, 70, 71

# Nomenclature

$\alpha_o$ Constant avoidance angle added to the CC. 32, 46

$\boldsymbol{p}(t)$ Current position. 23, 28, 29, 31, 58

$\boldsymbol{p}_0(0)$ Starting position. 57

$\boldsymbol{p}_A$ Current position of vehcile A. 92

$\boldsymbol{p}_B$ Current position of vehcile B. 92

$\boldsymbol{p}_o(t)$ Obstacle position. 30

$\boldsymbol{p}_t$ Target position. 31, 32, 57, 58

$\boldsymbol{v}(t)$ Current velocity of vehicle. 28

$\boldsymbol{v}_A$ Current velocity of vehicle A. 6, 7, 10–13, 17–22, 39, 66, 70, 72, 92

$\boldsymbol{v}_B$ Current velocity of vehicle B. 6, 7, 10, 13, 17, 18, 20, 39, 66, 70, 92

$\boldsymbol{v}_o(t)$ Current velocity of obstacle. 40

$\psi(t)$ Current heading. 29, 32, 40

$\psi_d(t)$ Desired heading from guidance law. 31, 32, 35, 40, 66

$\psi_o(t)$ Current heading. 30, 34, 36

$\psi_{dca}$ Desired heading from collision avoidance law. 40, 43

$d_{\mathbf{cpa}}$ CPA parameter; minimum distance between vehicles. 28, 45

$d_{\mathbf{min}}$ Minimum allowed distance to any vehcile or obstacle. 31, 35, 45, 46, 49, 58–61, 67, 69, 70

$d_{\mathbf{switch}}$ Distance to other vehicle that activated CA mode. 72, 92

$r(t)$ Current yaw rate. 29, 30

$r_o(t)$ Current obstacle yaw rate. 30

$r_{max}$ Bound on yaw rate. 30–32, 35, 49, 72, 85

$r_{o,max}$ Bound on obstacle yaw rate. 30, 31

$t_{\mathbf{cpa}}$ CPA parameter; time until collision. 28, 45

$u(t)$ Time dependent forward speed of vehicle. 29, 84

$u_o(t)$ Time dependent forward speed of obstacles. 30, 35, 37, 38, 68, 69

$u_{o,max}$ Bound on obstacle forward speed. 30, 35

$u$ Time independent forward speed of vehicle. 29, 30, 33–35, 37, 38, 49, 57, 60, 68, 69, 84, 85

$x$ Cartesian coordinate x. 23, 29–32, 41

$y$ Cartesian coordinate y. 23, 29–32, 41

# Chapter 1

# Introduction

## 1.1 Motivation

Following advancement in technology in recent years, the interest for fully autonomous vehicles have spiked, both for scientific and commercial applications. From an economic point of view as well as considering safety, autonomous vehicles have a huge potential. Autonomous or semi-autonomous operations can greatly reduce the need for manpower and increase the safety of passengers and cargo. A study show that 50% of accidents sea is initiated by human error and another 30% of accidents occur due to human failure to avoid an accident [1]. To ensure vehicle safety and mission success it is crucial for an autonomous vehicle to have an robust and reliable Collision Avoidance (CA) system. Reactive Collision Avoidance (RCA) algorithms supply fast computations of new control inputs while only relying on local sensor data. By assuming that there exist no general coordinator or direct communication link with the surrounding obstacles, RCA methods are considered to be very robust. RCA methods can guarantee both safety and liveness of the vehicle in the present of static and dynamic obstacles. However, in real-life applications it is also important to be able to avoid collisions with other reactive vehicles, both manually driven vehicles and other autonomous vehicle. Hence, there is a need for reciprocal collision avoidance algorithms.

## 1.2 Background and contributions

This thesis builds on a project thesis carried out in the fall of 2017. The project presented a dynamic constraint to the switching rule defined in [2] to improve the overall performance of the algorithm. However, the constraint made assumptions on the behavior of the obstacles which is not trivial to make when the obstacles are reactive. Hence, the main results from the project is not included in this thesis. The project also included a literature review covering multiple RCA algorithms. The RCA review in Section 2.1 is partially copied from the project and restructured to fit the context of this thesis. Chapter 3, which includes a system description, required measurements, control system and a formal description of the RCA algorithm was initially written as a part of the project.

The reciprocal CA algorithm presented in this thesis is based on an RCA algorithm designed for vehicles with nonholomic constraints and constant forward speeds [2]. The algorithm ensure safety of the vehicle in the presence of static and dynamic vehicles. However, in real-life applications it is also important to ensure vehicle safety in the presence other reactive vehicles. The main contribution of this thesis is to extend this algorithm to the multi vehicle case, where the obstacles are other reactive vehicles which is assumed to make a similar CA reasoning when a conflict is detected. As the main focus is vehicles at sea e.g. large ships with a limited speed envelope, the algorithm is designed such that the vehicles make decisions based on the Collision Regulations (COLREGS) defined by the International Maritime Organization (IMO). Hence, the algorithm makes predictable CA maneuvers similar to whats expected by manual control vehicles at sea.

The CA algorithms have been extensively tested through a series of Monte Carlo simulations as well as simulations showing how the vehicles makes CA maneuvers which comply with the defined COLREGS. Strengths and weakness of the algorithm have been identified and analyzed. Suggestions for improvements and future work is proposed. An comparison between the proposed algorithm and an similar algorithm developed by Andreas L. Aarvold as part of his master thesis are given in Appendix A.

The algorithm is simulated in a `MATLAB` simulator originally created by Martin

Syre Wiig to simulate the RCA algorithm in [2]. This simulator have been modified to simulate the proposed algorithm and new functionally is added. A full review of the simulator and the added functionality is given in Appendix B.

## 1.3 Outline

Chapter 2 presents relevant theory that are used in this thesis as well as a review some reactive and reciprocal CA algorithms to give some additional context to the reader. Chapter 3 presents the RCA algorithm that is the basis for this thesis along with a system description and assumptions that are made by the algorithm. Chapter 4 presents the changes to the algorithm that is proposed to enable it to tackle the reciprocal nature of other reactive vehicles, along with challenges associated with reciprocal behavior. Chapter 5 presents the results from various simulations of the algorithm. A discussion of the results are presented in Chapter 6. A final conclusion and suggestions for future works is given in chapter 7. The performance of the algorithm is compared with a similar algorithm in Appendix A. A review of the simulator and the changes made to it throughout this thesis is presented in Appendix B.

## 1.4 Notation

For consistency, mathematical symbols throwout this thesis use the following typesetting: Scalar values are written in a non-bold font, vectors are written in lowercase bold font and matrices are written in uppercase bold font. Sets are written in uppercase calligraphic style. Set subscripts such as $CC_{A|B}$ should be read as $CC$ induced on A by B and vector subscript such as $\boldsymbol{v}_{A,B}$ should be read as the relative vector between $\boldsymbol{v}_A$ and $\boldsymbol{v}_B$. All regularly used symbols is listed in the nomenclature.

# Chapter 2

# Literature review

This chapter presents a review some well-known Reactive Collision Avoidance (RCA) concepts followed by a review of the field of reciprocal Collision Avoidance (CA). Section 2.1 is partially copied from the project thesis leading up to this master thesis. Other theory and concepts that are used in this thesis is reviewed in this chapter as well.

## 2.1 Reactive collision avoidance

While a global path planner can plan a path or trajectory *a-priori*, it may not be able to handle unexpected obstacle as they appear along the path. Thus, global path planners are often combined with RCA. RCA algorithms are characterized by using no planning and only local sensor information. It is generally assumed that there does not exist any communication link between the vehicles and the obstacles. Also, there exists no central coordinator that calculates safe passage for all vehicles. This makes RCA robust in the sense that they can deal with a wide range of obstacles, both static and dynamic. Below follows a review of some well-known RCA concepts.

### 2.1.1 Collision Cone approaches

With roots in aerospace literature, the collision cone approach [3] describes an efficient way of determining if a collision between a vehicle and multiple arbitrary shaped objects with unknown trajectories is imminent. These methods are motivated by the conviction that collision avoidance and collision achievement are, in principle, the same problem. Hence, a framework for predicting collisions between moving objects is developed and used in a CA scheme. The collision cone approach formalizes the concept of a collision cone (Collision Cone (CC)) and presents analytic results to obtain the CC between arbitrary shaped objects in a dynamic environment.

#### 2.1.1.1 Collision cone

The authors of [3] first formalize the condition for a collision between two point objects and then extends this into a collision between a point object and a circle, resulting in the definition of the CC. A brief recap of the result is presented below. Assuming that two point objects are moving at constant velocities, $\boldsymbol{v}_A$ and $\boldsymbol{v}_B$, the relative velocity components $V_r$ and $V_\theta$ , are characterized by the kinematic equations

$$(V_r)_{AB} = \dot{r} = \boldsymbol{v}_B \cos(\beta - \theta) - \boldsymbol{v}_A \cos(\alpha - \theta) \tag{2.1}$$

$$(V_\theta)_{AB} = r\dot{\theta} = \boldsymbol{v}_B \sin(\beta - \theta) - \boldsymbol{v}_A \sin(\alpha - \theta) \tag{2.2}$$

where the geometry behind the the points $A$ and $B$ is shown in (2.1)

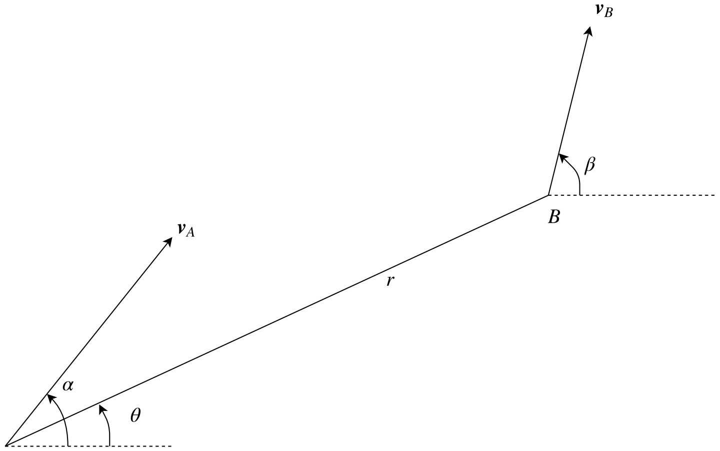

Figure 2.1: Collision geometry between two point objects

These equations can be extended to characterize the relative components between a point, $A$, and a arbitrary point on a circle, $C$

$$(V_r)_{AC} = \dot{r} = \boldsymbol{v}_B \cos[\beta - (\theta + \phi)] - \boldsymbol{v}_A \cos[\alpha - (\theta + \phi)] \tag{2.3}$$

$$(V_\theta)_{AC} = r\dot{\theta} = \boldsymbol{v}_B \sin[\beta - (\theta + \phi)] - \boldsymbol{v}_A \sin[\alpha - (\theta + \phi)] \tag{2.4}$$

Again, the geometry is illustrated in figure (2.2)

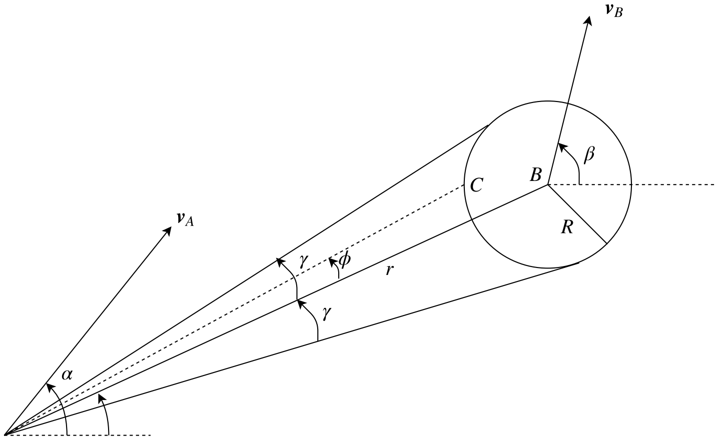

Figure 2.2: Collision geometry between a point and a circle

It is shown by [3] that if any point and a circle with radius $R$ is moving at constant velocities such that they satisfy (2.5), they will continue to satisfy (2.5) for all future time.

$$r^2(V_\theta)^2_{AB} \leq R^2\{(V_r)^2_{AB} + (V_\theta)^2_{AB}\} \tag{2.5}$$

If the vehicle is able to measure the relative velocity components with respect to the moving obstacle, then (2.5) can be directly used to predict a collision.

#### 2.1.1.2 Collision avoidance

Assuming that the vehicle can measure the relative velocity components given by (2.3)-(2.4), a collision can then by directly determined by (2.5). If the vehicle is headed for a collision with an obstacle, several approaches for avoiding the collision is proposed. One is by keeping the heading constant while altering the velocity vector such that it lies outside of the CC. Alternatively, the forward speed can be kept constant while altering the heading such to the same effect. A third possibility is to combine the two, by controlling both the heading and the speed of the vehicle.

#### 2.1.1.3 Related work

The collision cone approach [3] does not consider vehicles with nonholomomic constraints. In [4] a nonlinear time scaling which allows vehicles to reactively accelerate/de-accelerate without altering their geometric path is introduced. This method is completely independent of the vehicles kinematics and dynamics. It allows for navigation among static obstacles as well as other reactive vehicles.

Another approach which extends the collision cone concept to consider vehicles with nonholonomic constraints is presented in [2]. This is done by keeping a constant avoidance angle to the obstacles. In other words, the collision cone is extended by a constant angle $\alpha_o$ on each side. Unlike [4] where the heading is decoupled from the collision avoidance law, [2] decouples the acceleration of the vehicle and keeps the vehicle at a constant speed throughout the avoidance maneuver. This makes it suitable for large ships with high mass and thus a limited speed envelope.

In [5] a time scaled collision cone is used together with estimation of obstacle trajectories. A framework for predicting possible interceptions between the vehicle and a large number of estimated trajectories is shown to be computational efficient. This method allows vehicles to traverse cluttered environments where obstacles with unknown and time-varying velocities are present.

### 2.1.2 Velocity obstacle approaches

The Velocity Obstacle (VO) approach [6] utilizes the concept of VO [7] to determine safe control inputs. The VO defines the a set of vehicle velocities that would result in a collision with a given obstacle at some future time. The Velocity obstacle approach also considers the actuator constraints of the vehicle by mapping these into velocity constraints using forward dynamics. The avoidance maneuvers can then be determined by selecting vehicle velocities outside of the VO. By computing new avoidance maneuvers at regular time intervals the algorithm has shown good performance in presence of moving obstacles.

#### 2.1.2.1 The velocity obstacle

The analysis of the VO presented in [6] is restricted to circular objects. This is a common assumption that limits the complexity of the analysis. This is not, however, generally considered to be a severe limitation, as it has been shown that polygons can be represented by a number of circles [8]. Also, the obstacles trajectories are assumed to be arbitrary, but the position and velocity vector has to be known. This information can generally be obtained from sensor data. To further simplify the analysis, the the obstacle is mapped into the configuration space of the vehicle by reducing the vehicle to a point, $\hat{A}$, and enlarging the obstacle to a circle, $\hat{B}$, by adding the radius of A. Then a cone $\mathcal{VO}_B^A$ is defined as the set of relative velocities resulting in a collision between $\hat{A}$ and $\hat{B}$:

$$\mathcal{VO}_B^A = \{\boldsymbol{v}_{A,B} \mid \lambda_{A,B} \ \cap \ \hat{B} \neq \emptyset\} \tag{2.6}$$

where $\boldsymbol{v}_{A,B}$ is the relative velocity of $\hat{A}$ with respect $\hat{B}$, given by $\boldsymbol{v}_{A,B} = \boldsymbol{v}_B - \boldsymbol{v}_A$ and $\lambda_{A,B}$ is the line of $\boldsymbol{v}_{A,B}$. This cone is bounded by the two tangents from $\hat{A}$ to $\hat{B}$. Then any relative velocity that lies inside of these tangent lines will cause a collision between the vehicle and the obstacle. Provided that the obstacle maintains it current shape and speed, choosing a relative velocity outside of the cone, $\mathcal{VO}_B^A$, is guaranteed to be collision free. The velocity obstacle is then defined by translating the $\mathcal{VO}_B^A$ by the obstacles velocity, $\boldsymbol{v}_B$, which is shown in figure 2.3.

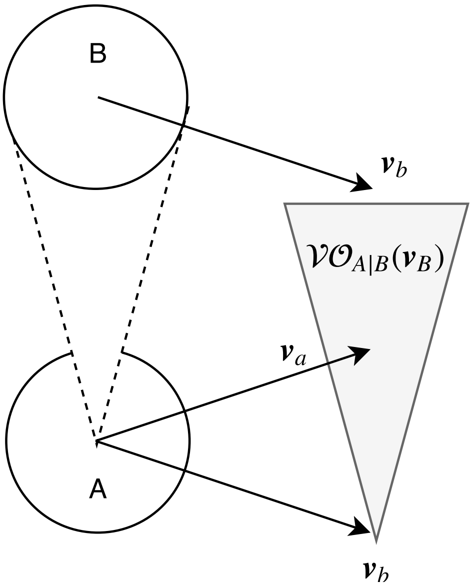

Figure 2.3: Velocity obstacle

Note that this VO is based on a linear approximation of the obstacle's trajectory. Hence collision prediction of obstacles that do not move on a straight line may be inaccurate. In a cluttered environment with several moving obstacles there is a need to prioritize collision avoidance of the obstacle closest to the vehicle. This is accounted for by defining a subset of VO and then subtracting this set from the VO, yielding a modified VO. The subset is given by

$$\mathcal{VO}_h = \{\boldsymbol{v}_A | \boldsymbol{v}_A \in VO_{A|B}, ||\boldsymbol{v}_{A,B}|| \leq \frac{d_m}{T_h}\} \tag{2.7}$$

where $d_m$ is the shortest distance between the vehicle and the obstacle and $T_h$ is a time horizon. If a collision occur within the time horizon $T_h$, it is considered an *imminent* collision. $T_h$ is a design parameter, which could be based on the system dynamics and the trajectory of the obstacle. The modified VO is shown in Figure 2.4:

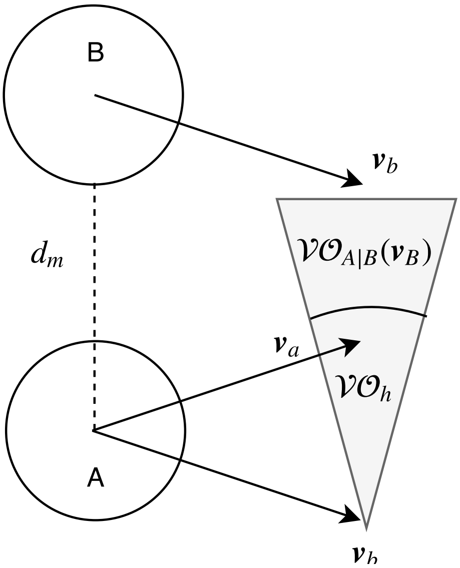

Figure 2.4: Modified velocity obstacle

#### 2.1.2.2 The avoidance maneuver

Having defined VO as the set of vehicle velocities that result in a collision, all velocities not in this set can be chosen to navigate the vehicle safely. However, not all of these velocities are reachable for the vehicle. Hence, a new set of reachable velocities that accounts for vehicle dynamics and actuator constraints is defined [6]. This set is constructed by mapping the actuator constraints to acceleration constraints and is given by

$$\mathcal{RV}(t+\Delta t) = \{\mathbf{v} \mid \mathbf{v} = \boldsymbol{v}_A(t) \oplus \Delta t \cdot \mathcal{FA}(t)\} \tag{2.8}$$

where the set of feasible acceleration, $\mathcal{FA}(t)$, is given by

$$\mathcal{FA}(t) = \{\ddot{\mathbf{x}} \mid \ddot{\mathbf{x}} = f(\mathbf{x}, \dot{\mathbf{x}}, \mathbf{u}), \mathbf{u} \in U\} \tag{2.9}$$

Note that this set may be hard to find for a nonlinear system. Subtracting the VO from the *reachable velocities*, the set of *reachable avoidance velocities* [6] is defined as

$$\mathcal{RAV}(t+\Delta t) = \mathcal{RV}(t+\Delta t) \ominus \mathcal{VO}(t) \tag{2.10}$$

A avoidance maneuver is obtained by choosing any of the *reachable avoidance velocities* in RAV. It is important to note that nonholonomic constraint is not accounted for in this analysis, but at high speeds the dynamic constraints will be more restrictive, making the nonholonomic constraints only relevant at low speeds. A method for dividing the set RAV into the three non-overlapping subset, $S_f$, $S_r$ and $S_d$ representing velocities corresponding to moving in front of, behind the obstacle or diverging it is proposed [6]. This may help to choose the best *reachable avoidance velocity.*

#### 2.1.2.3 Related work

For many vehicles with nonholonomic constraints such as car-like vehicles, the set of *reachable velocities* at an instant is a single velocity, the velocity in the direction of the rear wheels. This can be accounted for by considering velocities that are reachable within a given time horizon. However, no guarantee for collision free navigation can be given as the kinematically constrained vehicle moves on an arc rather than on a straight line. The generalized velocity obstacle [9] solves this problem by deriving an expression for the position of a car-like vehicle at some future time, given a constant control input. This position can then be evaluated against the VO to determine if a collision will happen. At each time step a range of control inputs is evaluated and a optimal control input is found by choosing the one that is closed to the desired control input.

In [6] collision avoidance is guaranteed by choosing velocities outside of the $VO$ under the assumption that the velocities $\boldsymbol{v}_A$ and $\boldsymbol{v}_B$ is constant within a time horizon, $\tau$. This implies that the vehicle will have to adapt to the chosen velocity, $\boldsymbol{v}_A$ instantaneously. However, the vehicles acceleration constraints may prohibit this behavior and thus, collision free navigation is no longer guaranteed. Such acceleration constraints are accounted for in the *acceleration velocity obstacle* [10]. By using proportional control of the acceleration, the vehicle will smoothly arrive at its new velocity. The acceleration applied at time $t$, is proportional to the difference between the current velocity and the new velocity.

### 2.1.3 Other reactive collision avoidance concepts

Artificial Potential Fields (APF) methods are based on the idea that obstacles exert repulsive forces onto the vehicle while the target applies an attractive force onto the vehicle. The sum of all forces determines the direction and the speed of the vehicle. These artificial forces are designed to drive the vehicle towards its goal. The simplicity and elegance to these methods are among the reasons for their popularity. APF methods can be implemented quickly and initially provide acceptable performance without requiring to much adaptations to its specific application. Using APF for reactive collision avoidance is widely used both for vehicle manipulators and mobile vehicles [11].

The APF methods suffers from some well known issues like getting stuck in local minimas of the potential fields, not being able to find passage between closely spaced obstacles and oscillations in narrow passages and in presence of obstacles [12]. A lot of work has been done trying to overcome these issues, although this continues to be an active field of study. However, little research has been done on APF methods in dynamic environments. Some of the work that has been done to APF is briefly reviewed below.

By designing the shape of the potential field to flow around obstacles concavities, harmonic potential field methods such as [13, 14] has proven better performance with avoiding local minimas than traditional APF methods. Although these are impossible to fully solve deterministically using reactive algorithms.

A systematic approach to handle the inherent oscillation problems in the presence of obstacles and in narrow passages is presented in [15]. This paper identifies that the gradient descent approach, which is traditionally used in APF methods to determining a velocity vector that point toward the target, produce oscillatory trajectories even when the discrete system is stable. Hence, the use of the modified Newton method is suggested as an alternative way of determining the velocity vector. The same method is applied to nonholonomic vehicles in [16]. This has shown to improve the performance and reduce the oscillatory behavior. However, trajectories generated by this APF would not be optimal.

Other approaches focuses on adapting APF method to vehicles with nonholonomic constraints. By moving the vehicle's reference point away from the center of the vehicle the performance of nonholonomic vehicles is improved [16, 17].

APF do not consider moving obstacles and very little research has been done on using the APF in a dynamic environment. One way of adapting the APF to consider moving obstacles could be to extend the magnitude of the repulsive force induced by the obstacle in the direction of the obstacles velocity vector. However this is a field that needs more research [18].

In contrast to the APF, the Dynamic window (DW) approach [19], is especially designed to deal with the vehicle's kinematic constraints, such as limitations to velocities and accelerations. The authors have derived the DW approach directly from the motion dynamics of the vehicle. A 2-dimensional search space of linear and angular velocities is constructed by approximating possible trajectories within the time interval based on the dynamic motions of the vehicle. To further limit the search space, only velocities that is reachable in the next timestep are considered. This set of velocities is called the DW. A combination of linear and angular velocities within the DW is chosen by minimizing a cost function.

The DW approach does consider nonholonomic constraints of the vehicle, however this only applies for first order nonholonomic constraints. Some vehicles are subject to second order nonholonomic constraints, which makes DW an unsuited solution. To account for these constraints a modified version of the DW is proposed by [20]. This algorithm modifies the way the vehicle trajectories are predicted as well as the search space, reducing the prediction error to about one percent of the original DW. A case study that compares the two models shows significant improvement in performance when applied to an autonomous unmanned vehicle.

By reducing the DW to a holonomic dynamic window approach, the authors of

[21] have overcome the local minima limitation to of DW. The proposed algorithm incorporates information about the connectivity of the free space into the DW. Hence, the method integrate global goal behavior with local obstacle avoidance without any prior knowledge about the environment, making it well suited for unknown environments. Another method for combining the DW with global path planning is proposed in [22].

DW has shown excellent performance in theoretical setups, but it is possible to construct examples where the algorithm fail to converge toward the goal configuration. A theoretical treatment of the convergence properties of the DW is given by [23]. The DW is viewed as a model predictive control (MPC) and by using a control Lyapunov function the authors propose a version of DW which is proved to be trackable and convergent.

Similar methods to the DW which also formulate the local obstacle avoidance problem as a constrained optimization problem in the velocity space is the Curvature Velocity Method (CVM) [24], The Lane Curvature Method (LCM) [25] and the Beam-Curvature Method (BCM) [26].

## 2.2 Reciprocal collision avoidance

In most reactive CA algorithms the main focus is on ensuring CA with static and/or moving obstacles that comes into the vehicles path, however other reactive vehicles is often not considered. Reciprocal CA algorithms aims to solve the challenge of safe traversal for multiple vehicles where each vehicle is implemented with the same CA algorithm but without any centralized coordinator or any form of communications between the vehicles. The only information known by a vehicle is what the vehicle can observe with its on-board sensors, like in the case for RCA. The reactive nature of other vehicles raises a new set of challenges, like reciprocal dances, where vehicles end up driving in a circle around each other or oscillations. Thus, special considerations is needed to handle the reactive nature of other vehicles.

### 2.2.1 Reciprocal velocity obstacle

By regarding other vehicles as moving obstacles, the VO can be directly implemented to work as a reciprocal CA algorithm. However, as shown by [27] this causes unwanted oscillation in the vehicles trajectories. Lets say that vehicle A and vehicle B is headed for a collision, i.e $\boldsymbol{v}_A \in \mathcal{VO}_{A|B}$ and $\boldsymbol{v}_B \in \mathcal{VO}_{B|A}$. Both vehicles will then choose a new velocity ${\boldsymbol{v}_A}^{new}$ and ${\boldsymbol{v}_B}^{new}$ outside of their respective VO's. In the next time step, both VO's have changed due to the new velocity vector of the other vehicle. This can in turn cause the new velocity vectors to be inside the VO while the old velocities is now safe and can be selected again. Hence the trajectories of the vehicles will oscillate.

Reciprocal Velocity Obstacle (RVO) seeks to resolve this issue under the assumption that the other vehicle will also try to avoid a potential collision in a similar fashion. The concept of RVO is simple and intuitive; rather than choosing a new velocity outside of the VO for each vehicle, the new velocity is chosen as the average of the current velocity and the new velocity that lies outside of the vehicles VO. Thus, implicitly assuming that the other vehicle will take half the responsibility of avoiding the collision. More formally, the RVO can be formulated as:

$$\mathcal{RVO}_B^A(\boldsymbol{v}_A, \boldsymbol{v}_B) = \{{\boldsymbol{v}_A}^{new} | 2{\boldsymbol{v}_A}^{new} - \boldsymbol{v}_A \in \mathcal{VO}_B^A(\boldsymbol{v}_B)\} \tag{2.11}$$

where ${\boldsymbol{v}_A}^{new}$ is the new velocity and $\mathcal{VO}_B^A(\boldsymbol{v}_B)$ is the VO of vehicle B to vehicle A. Geometrically it can be interpreted as translating the $VO_B^A$ such that the apex lies at $\frac{\boldsymbol{v}_A + \boldsymbol{v}_B}{2}$. This is shown in figure (2.5)

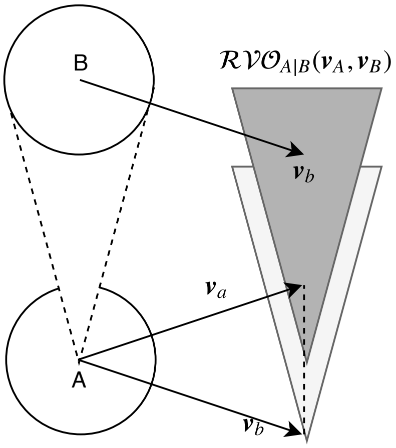

Figure 2.5: Reciprocal velocity obstacle (RVO): The apex of the RVO is moved to the average of $\boldsymbol{v}_A$ and $\boldsymbol{v}_B$

The assumptions that the other vehicle takes half the responsibility of avoiding a collision is based on some important properties of the VO. Due to symmetry, it can be stated that if vehicle A is headed for a collision with vehicle B, vehicle B is also headed for a collision with vehicle A:

$$\boldsymbol{v}_A \in \mathcal{VO}_{A|B}(\boldsymbol{v}_B) \Leftrightarrow \boldsymbol{v}_B \in \mathcal{VO}_{B|A}(\boldsymbol{v}_A) \tag{2.12}$$

Secondly, there is a translational invariance property

$$\boldsymbol{v}_A \in \mathcal{VO}_{A|B}(\boldsymbol{v}_B) \Leftrightarrow \boldsymbol{v}_A + u \in \mathcal{VO}_{A|B}(\boldsymbol{v}_B + u) \tag{2.13}$$

From these properties it can be shown that if $\boldsymbol{v}_A$ is on the left side of the centerline of $\mathcal{VO}_{A|B}$ then $\boldsymbol{v}_B$ is on the left side of the centerline of $\mathcal{VO}_{B|A}$. Thus, it can be guaranteed that the vehicles will choose to pass each other on the same side if each vehicle choose the new velocity outside of the RVO that is closest to the current velocity. This behavior ensures oscillation free movement for each vehicle.

Although simulations shows how RVO is able to guide several hundreds of vehicles in dense environments, it has been shown that introducing a third vehicle can cause vehicles to choose to pass each other on different sides, hence causing oscillatory behavior known as reciprocal dances. To encounter this, the authors propose an extension to the RVO named Hybrid Reciprocal Velocity Obstacle (HRVO). The concept is equally intuitive as the RVO and is based on the VO centerline symmetry previously described. This method uses one edge from the RVO and one edge from the VO, hence the name hybrid RVO. Let $\boldsymbol{v}_A$ be to the left of the centerline of the VO, then the right side of the RVO is considered to be the side that we do not want the vehicle to choose. To encourage this, the right edge of the RVO is replaced with the right egde of the VO, thus enlarging the VO on this side. The apex of the new VO is where the left side of the RVO intersect the left edge of the VO. More formally, the HRVO for vehicle $A_i$ induced by all other vehicles and obstacles is defined by:

$$\mathcal{HRVO}_{A_i} = \bigcup_{\substack{A_j \in \mathcal{A} \\ j \neq i}} \mathcal{HRVO}_{A_i|A_j} \cup \bigcup_{O_j \in O} \mathcal{VO}_{A_i|O_j} \tag{2.14}$$

where $\mathcal{A}$ and $O$ are sets including all vehicles and obstacles, respectivly. The vehicle can then choose its new velocity outside the HRVO that is closest to its preferred velocity. This method implicitly states that if the vehicle choose to pass on the wrong side, it will consider the other vehicle as an dynamic obstacle and will take full responsibility of avoiding the collision. Should the vehicle chose the correct side to pass the other vehicle, it can still assume that the other vehicle will take half the responsibility.

### 2.2.2 Optimal reciprocal velocity obstacle

Although RVO and HRVO have shown its ability to resolve conflicts between a large number of reactive vehicles, the formulation of the RVO only guarantees collision avoidance under specific conditions and do not give a *sufficient* condition for reciprocal CA in general. In that sense the Optimal Reciprocal Collision Avoidance (ORCA) [28] is a more robust CA algorithm as it do provide a *sufficient* condition for multiple vehicles to avoid collision amongst each other. Hence, it can guarantee collision-free navigations for all vehicles. The authors also point out that it is possible to give a *necessary* condition, but only under the assumption that there is a central coordination among the vehicles. However, cooperative CA is a different field of study which is not a part of the scope in this thesis.

Like RVO and HRVO, ORCA is based on the VO. It follows from the definition of VO that there exist infinity many pairs of $\boldsymbol{v}_A$ and $\boldsymbol{v}_B$ that will guarantee that vehicle A and B do not collide. In ORCA, the pair of $\boldsymbol{v}_A$ and $\boldsymbol{v}_B$ that maximizes the amount of permitted velocities close to the desired velocity is selected. Hence the name, Optimal Reciprocal Collision Avoidance. In a pair-wise collision, the velocity space is divided in a half plane of admissible velocities, named $\mathcal{ORCA}^{\tau}_{A|B}$. The velocity space is divided in fair manner, such that vehicle A and B have an equal amount of velocities close to their desired velocity. More formally, this set is defined as:

$$\mathcal{ORCA}^{\tau}_{A|B} = \{\boldsymbol{v} | (\boldsymbol{v} - ({\boldsymbol{v}_A}^{opt} + \frac{1}{2}\boldsymbol{u})) \cdot \boldsymbol{n} \geq 0\} \tag{2.15}$$

where ${\boldsymbol{v}_A}^{opt}$ is the optimal velocity of the vehicle, normally it would be the current velocity, $\boldsymbol{v}_A$. $\boldsymbol{u}$ is the vector from the relative velocity vector $\boldsymbol{v}_A$ - $\boldsymbol{v}_B$ to the closest point on the boundary of the VO:

$$\boldsymbol{u} = (\underset{\boldsymbol{v} \in \partial VO^{\tau}_{A|B}}{\arg\min} ||\boldsymbol{v} - ({\boldsymbol{v}_A}^{opt} - {\boldsymbol{v}_B}^{opt})||) - ({\boldsymbol{v}_A}^{opt} - {\boldsymbol{v}_B}^{opt}) \tag{2.16}$$

and let $\boldsymbol{n}$ be the outward normal from the VO at the point $\boldsymbol{v}_A - \boldsymbol{v}_B + \boldsymbol{u}$. This is shown in figure 2.6

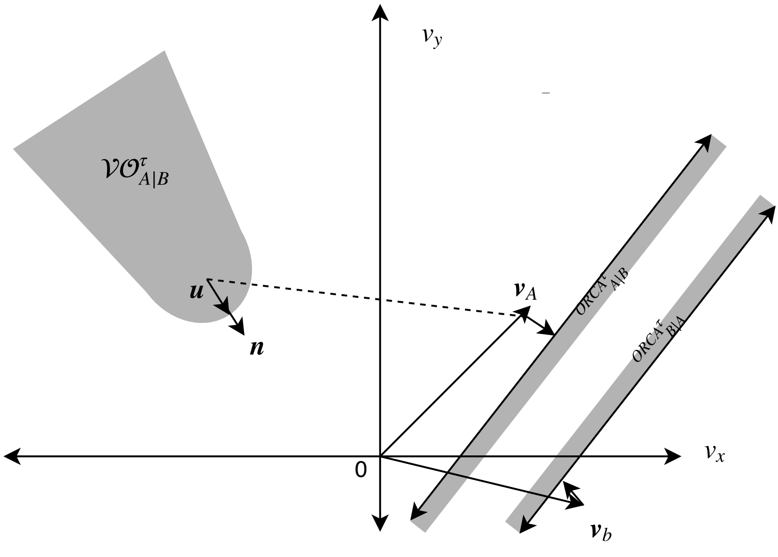

Figure 2.6: ORCA

Note that in the definition of ORCA (2.15), $\frac{1}{2}\boldsymbol{u}$ implicitly ensures that each vehicle take at least half the responsibility in avoiding each other.

The concept of ORCA is easily extended to the n-body collision avoidance case by defining the set of admissible velocities as the intersection of all half-planes of admissible velocities induced by each other vehicle

$$\mathcal{ORCA}^{\tau}_{A} = D(\mathbf{0}, \boldsymbol{v}_A{}^{max}) \cap \bigcap_{B \neq A} \mathcal{ORCA}^{\tau}_{A|B} \tag{2.17}$$

where $D(\mathbf{0}, \boldsymbol{v}_A{}^{max})$ is a circle with its center at origo and radius $\boldsymbol{v}_A{}^{max}$ representing a maximum speed bound. This is shown geometrically in figure (2.7)

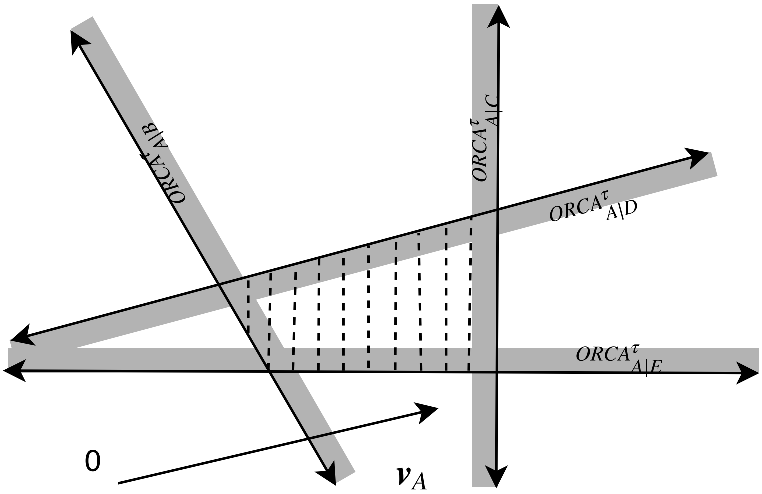

Figure 2.7: Half planes of admissible velocities induced onto vehicle A by vehicle B-E. The resulting convex set (dashed lines) defines all admissible velocities for vehicle A

Do to the linear nature of $\mathcal{ORCA}^{\tau}_{A|i}$ and the convexity of the maximum speed bound, $\boldsymbol{v}_A$, $\mathcal{ORCA}^{\tau}_{A}$ will always be a convex set. Thus, the new velocity can be found by linear programing where the objective function is

$$\boldsymbol{v}_A{}^{new} = \underset{\boldsymbol{v} \in \mathcal{ORCA}^{\tau}_{A}}{\arg\min} \; ||\boldsymbol{v} - \boldsymbol{v}_A{}^{pref}|| \tag{2.18}$$

and $\mathcal{ORCA}^{\tau}_{A}$ is the constraints.

Although ORCA has shown great performance and is considered to be the benchmark for reciprocal CA, just like VO it assumes that the vehicles are holonomic, which is a severe limitation when applying the algorithm to real applications. In Bicycle Reciprocal Collision Avoidance (B-ORCA)[29], ORCA is further extended to handle the kinematic constraints of vehicles. The concept of B-ORCA is based on a kinematic constrained vehicle's ability to track a holonomic trajectory within a maximum error bound. In other words; the vehicle is able to stay within a radius equal the maximum error bound. Under that assumption, the radius of the vehicle can be enlarged by the error bound. ORCA can then be implemented as if the vehicle was holonomic. One major drawback of this approach is that it is not trivial for other vehicles to obtain the maximum error bound of a vehicle without any form of communication between vehicles.

### 2.2.3 Reserved regions

The authors of [30] base their reciprocal CA algorithm on Reserved Region (RR). By letting each vehicle claim exclusive ownership of a RR, collision-free navigation is guaranteed as long as the RR stay disjoint and all vehicles stay inside their RR. One major advantage with RR over other decentralized cooperative CA algorithms, is the ability to guarantee safety for nonholonomic vehicles with constant speeds. In contrast to the motion of nonholonomic vehicles, which is limited by dynamic and kinematic constraints, the RR corresponding to the vehicle can be regarded as holonomic. This is supported by the definition of the RR, which enables the nonholonomic vehicle to roll along the interior border of the RR. Thus, its possible to stop a RR without stopping the vehicle itself. The control policy proposed in [30] uses a set of discrete states, all corresponding to a constant control input, to control the movement of the RR and thus the vehicle.

From the definition of the RR given by [30], it follows that the center of the RR is defines as

$$(x^c, y^c) = c(x, y, \theta) = (x + \sin(\theta), y - \cos(\theta)) \tag{2.19}$$

Let $R_s$ be the minimum safety distance from one vehicle to another and $R_c$ be the curvature radius of the vehicle, then the RR for the $i$th vehicle is defined as a disc centered at $c(\boldsymbol{p}(t))$ with radius $R_c + R_s$:

$$\mathcal{R}_i = \{(x, y) \in \mathbb{R}^2 : ||(x, y) - c(\boldsymbol{p}(t))||_2 \leq R_c + R_s\} \tag{2.20}$$

The reserved region is illustrated in figure (2.8).

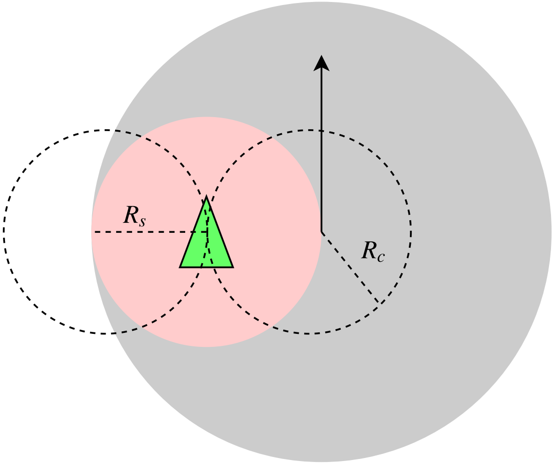

Figure 2.8: Reserved region: The red circle centered at the vehicle corresponds to the minimum safety distance from the vehicle. The dotted circles illustrates the curvature radius of the vehicle, while the gray circle is the RR.

The RR is controlled by a set of rules that is defined *a priori*. This set consist of four discrete states; `hold`, `straight`, `roll`, `roll2`. In the `hold` state, the RR is stopped while the vehicle roll along the inner border of the region. In the `straight`-state both the RR and the vehicle is moving on a straight line toward the vehicles target position. The `roll`-state is activated if the path towards the target position is blocked by a stationary RR. Then the course of action is to roll on the edge of the blocking region in a counter-clockwise fashion until the path towards goal is clear. However, if the blocking region is non-stationary, the contact between to regions that are both trying to roll on each other may be lost. Thus a second roll state, `roll2`, is defined to try to recover contact between the regions. These states are combined in what the authors have called the Generalized Roundabout Policy.

Although this method can guarantee both the safety and the liveness of all vehicles, do to the Right-Turn-Only Steering policy i.e. the counter-clockwise rolling and the `hold` state, the trajectories of the vehicles is not optimal in a minimal-time or minimal-length sense. By assuming that other vehicles respect the RR, this method is restricted to the case where all vehicles is implemented with the same policy. Also, there is no trivial way to measure the turning rate of other vehicles. To overcome this, the authors have made an assumption that all vehicles are homogeneous i.e. they have the same turning rates and forward speed. This assumption will however limit the use of this policy in real-life applications.

In [31], RR are used on combination with APF to create a method for trajectory tracking and CA for teams of nonholonomic vehicles with constant speed. An auxiliary system tracks the desired trajectory while maintaining a safe distance to other vehicles. Also, a local heading controller is design such that the vehicle follows the auxiliary system within a predefined, bounded error, similar to B-ORCA. This method holds an advantage over [30] in that it allow the vehicles to be heterogeneous i.e. vehicles can have different turning rates and constant speeds, which makes the method more flexible.

## 2.3 COLREGS

The Convention on the International Regulation for Preventing Collisions at Sea was formalized by the International Maritime Organization (IMO) in 1972 and became effective in 1977. The rules, commonly refereed to as Collision Regulations (COLREGS) is a part of this convention. The convention is divided into 5 parts (A-E), covering different areas. Part B - Steering and sailing rules, contains the most relevant rules for this thesis and the relevant rules are given in full below, collected from [32]

### 2.3.1 Rule 13 - Overtaking

**(a)** *Notwithstanding anything contained in the Rules of part B, sections I and II, any vessel overtaking any other shall keep out of the way of the vessel being overtaken.*
**(b)** *A vessel shall be deemed to be overtaking when coming up with another vessel from a direction more than 22.5 degrees abaft her beam, that is, in such a position with reference to the vessel she is overtaking, that at night she would be able to see only the sternlight of that vessel but neither of her sidelights.*

**(c)** *When a vessel is in any doubt as to whether she is overtaking another, she shall assume that this is the case and act accordingly.*
**(d)** *Any subsequent alteration of the bearing between the two vessels shall not make the overtaking vessel a crossing vessel within the meaning of these Rules or relieve her of the duty of keeping clear of the overtaken vessel until she is finally past and clear.*

### 2.3.2 Rule 14 - Head-on situation

**(a)** *When two power-driven vessels are meeting on reciprocal or nearly reciprocal courses so as to involve risk of collision each shall alter her course to starboard so that each shall pass on the port side of the other.*
**(b)** *Such a situation shall be deemed to exist when a vessel sees the other ahead or nearly ahead and by night she could see the masthead lights of the other in a line or nearly in a line and/or both sidelights and by day she observes the corresponding aspect of the other vessel.*
**(c)** *When a vessel is in any doubt as to whether such a situation exists she shall assume that it does exist and act accordingly.*

### 2.3.3 Rule 15 - Crossing situation

*When two power-driven vessels are crossing so as to involve risk of collision, the vessel which has the other on her own starboard side shall keep out of the way and shall, if the circumstances of the case admit, avoid crossing ahead of the other vessel.*

### 2.3.4 Rule 16 - Action by give-away vessel

*Every vesse which is directed to keep out of the way of another vessel shall, so far as possible, take early and substantial action to keep well clear.*

### 2.3.5 Rule 17 - Action by stand-on vessel

**(a)**
**(i)** *Where one of two vessels is to keep out of the way the other shall keep her course and*

*speed.*

**(ii)** *The latter vessel may however take action to avoid collision by her manoeuvre alone, as soon as it becomes apparent to her that the vessel required to keep out of the way is not taking appropriate action in compliance with these Rules*

**(b)** *When, from any cause, the vessel required to keep her course and speed finds herself so close that collision cannot be avoided by the action of the give-way vessel alone, she shall take such action as will best aid to avoid collision.*

**(c)** *A power-driven vessel which takes action in a crossing situation in accordance with subparagraph (a)(ii) of this Rule to avoid collision with another power-driven vessel shall, if the circumstances of the case admit, not alter course to port for a vessel on her own port side.*

**(d)** *This Rule does not relieve the give-way vessel of her obligation to keep out of the way.*

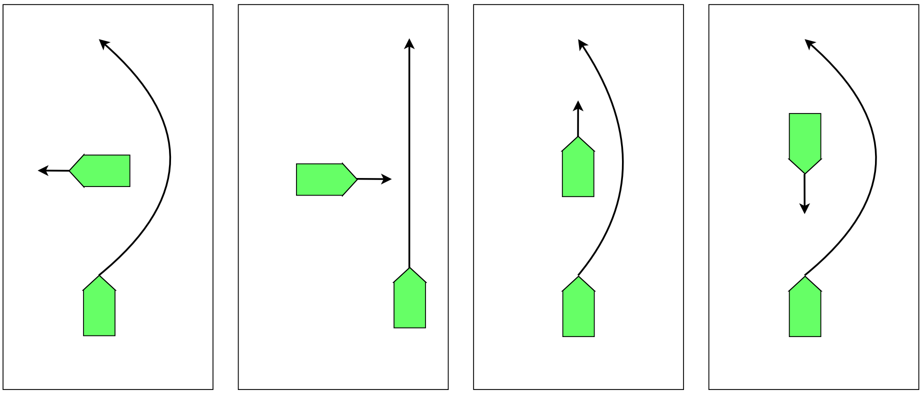

Figure 2.9: COLREGS situations from the left: Crossing from right, crossing from left, overtaking and headon

## 2.4 Closest point of approach

Closest Point of Approach (CPA) is a method to determine at which future time, $t_{\text{cpa}}$, the distance between two vehicles is smallest and the distance, $d_{\text{cpa}}$, between the vehicles at this time. The CPA only requires the current position, $\boldsymbol{p}(t)$ and velocity, $\boldsymbol{v}(t)$, of the two vehicles aswell as making the assumtions that both vehicles have a rounded shape and travels along linear trajectories keeping constant speed, i.e the velocity vectors $\boldsymbol{v}(t)$ is constant from $t = 0$ to $t = t_{\text{cpa}}$. The time until CPA and the distance between the vehicles at this time is calculated as follows [33]:

$$t_{\text{cpa}} = \begin{cases} 0, & \text{if} \quad ||\mathbf{v}_A - \mathbf{v}_B|| \leq \epsilon \\ \frac{(\mathbf{p}_A - \mathbf{p}_B)\cdot(\mathbf{v}_A - \mathbf{v}_B)}{||\mathbf{v}_A - \mathbf{v}_B||^2}, & \text{othervise} \end{cases} \tag{2.21}$$

$$d_{\text{cpa}} = ||(\mathbf{p}_A + \mathbf{v}_A t_{\text{cpa}}) - (\mathbf{p}_B + \mathbf{v}_B t_{\text{cpa}})|| \tag{2.22}$$

An illustration of CPA is given in figure (2.10). The red dots indicated the vehicles positions at given time instances. The dashed lines is the distance between the vehicles at that time instance.

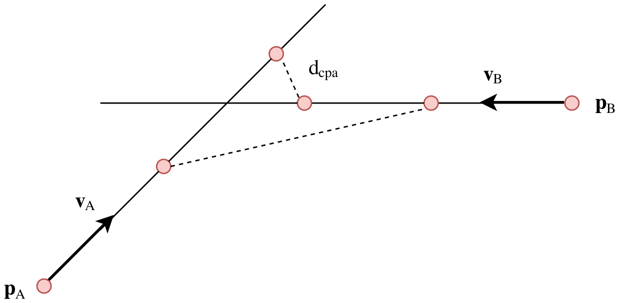

Figure 2.10: Clostest point of approach. The red dots along the vehicle trajectories corresponds to a position at a spesific time. The dashed lines represent the distance between the two vehicles A and B

# Chapter 3

# Reactive collision avoidance of nonholonomic vehicles

## 3.1 System description

### 3.1.1 Vehicle model

The vehicle to be controlled is modeled as a unicycle-type vehicle

$$\dot{x}(t) = u(t)\cos(\psi(t)), \qquad x(0) = x_0, \tag{3.1}$$

$$\dot{y}(t) = u(t)\cos(\psi(t)), \qquad y(0) = y_0, \tag{3.2}$$

$$\dot{\psi}(t) = r(t), \qquad \psi(0) = \psi_0, \tag{3.3}$$

where $x(t)$ and $y(t)$ are the Cartesian coordinates of the vehicle, $u(t)$ is the forward speed and $\psi(t)$ and $r(t)$ is the heading and turning rate, respectively. The vehicle position is denoted $\boldsymbol{p}(t) \triangleq [x(t), y(t)]^T$. Furthermore, two assumptions on the behavior of the vehicle is made

*Assumption 1:* The forward speed $u > 0$ is constant

*Assumption 2:* The heading rate is bounded by

$$r(t) \in [-r_{max}, r_{max}] \tag{3.4}$$

where $r_{max}$ is a constant parameter.

### 3.1.2 Obstacle model

The obstacles is also modeled as a nonholonomic vehicles

$$\begin{aligned}
\dot{x}_o(t) &= u_o(t)\cos(\psi_o(t)), & x_o(0) &= x_{o,0}, \qquad (3.5)\\
\dot{y}_o(t) &= u_o(t)\cos(\psi_o(t)), & y_o(0) &= y_{o,0}, \qquad (3.6)\\
\dot{\psi}_o(t) &= r_o(t), & \psi_o(0) &= \psi_{o,0}, \qquad (3.7)\\
\dot{u}_o(t) &= a_o(t), & u_o(0) &= u_{o,0} \qquad (3.8)
\end{aligned}$$

where $x_o(t)$ and $y_o(t)$ are the Cartesian coordinates of the obstacle, $u_o(t)$ and $a_o(t)$ is the obstacles forward speed and acceleration, and $\psi_o(t)$ and $r_o(t)$ is the heading and heading rate of the obstacle. The position is denoted $\boldsymbol{p}_o(t) \triangleq [x_o(t), y_o(t)]^T$ and the velocity vector is denoted $\boldsymbol{v}_o(t) \triangleq [\dot{x}_o(t), \dot{y}_o(t)]^T$. Further assumptions about the obstacle are

*Assumption 3:* The forward speed is bounded by

$$u_o(t) < u_{o,max} < u \tag{3.9}$$

*Assumption 4:* The forward acceleration is bounded by

$$a_o(t) \in [-a_{o,max}, a_{o,max}] \tag{3.10}$$

where $a_{o,max} > 0$ is a constant parameter.

*Assumption 5:* The heading rate is bounded by

$$r_o(t) \in [-r_{o,max}, r_{o,max}] \tag{3.11}$$

where $r_{o,max} > 0$ is a constant parameter.

*Assumption 6:* The obstacle is modeled as a moving circular domain, $D_o(t)$ with radius $R_o$ which is centered around $\boldsymbol{p}_o(t)$.

## 3.2 Required measurements

It is assumed that the vehicle is able to measure the minimum distance between the vehicle and the obstacle $d_o(t)$ and its time derivative $\dot{d}_o(t)$ within a sensing range $d_{sense} > 0$. This minimum distance is defined as

$$d_o(t) = \min_{\boldsymbol{p}_D \in \mathcal{D}_o(t)} ||\boldsymbol{p}_D - \boldsymbol{p}(t)|| \tag{3.12}$$

where $|| \cdot ||$ is the Euclidean norm. Furthermore, the vehicle is able to measure the obstacle velocity $\boldsymbol{v}_o(t)$ and the angles $a^{(1)}$ and $a^{(2)}$ between the $x - axis$ and the edges of the vision cone $\mathcal{V}_o(t)$ from the vehicle to the obstacle.

## 3.3 Control system

### 3.3.1 Control objective

The control objective is to guide the vehicle from its current position, $\boldsymbol{p}(t)$, to a target position, $\boldsymbol{p}_t = [x_t, y_t]$ while keeping atleast a minimum safety distance, $d_{\min}$ to the obstacle at all time. That is

$$d_o(t) \geq d_{\min} > 0 \quad \forall \quad t \tag{3.13}$$

### 3.3.2 Heading controller

The heading controller will always turn the vehicles towards the desired heading, $\psi_d(t)$, at maximum turning rate, $r_{max}$. This behavior is defined by

$$r(\psi_d(t)) \triangleq \begin{cases} 0, & \tilde{\psi} = 0 \\ r_{max}, & \tilde{\psi} \in (-\pi, 0) \\ -r_{max}, & \tilde{\psi} \in (0, -\pi] \end{cases} \tag{3.14}$$

where $\tilde{\psi} \triangleq \psi(t) - \psi_d(t) \in (-\pi, \pi]$ is the heading error state. The interval ensures that the vehicle always makes the shortest turn towards the desired heading, $\psi_d(t)$.

### 3.3.3 Guidance law

When the control system is in guidance mode, the desired heading, $\psi_d(t)$, is obtained by a pure pursuit guidance law [34]

$$\psi_d(t) \triangleq atan2(y_t - y(t), x_t - x(t)) \tag{3.15}$$

This guidance law will direct the heading of the vehicle straight towards the target position, $\boldsymbol{p}_t$.

## 3.4 Collision avoidance

The Reactive Collision Avoidance (RCA) algorithm [2] build on the concept of the Collision Cone (CC). To account for the vehicles nonholonomic constraints, a extended vision cone is defined. The RCA algorithm keeps a constant avoidance angle $\alpha_o \in (0, \frac{\pi}{2})$ to the tangent lines between the vehicle and the obstacle.

### 3.4.1 Extended vision cone

To keep a constant avoidance angle to the object, the vision cone is extended by the angle $\alpha_o$, which is shown in figure (3.1).

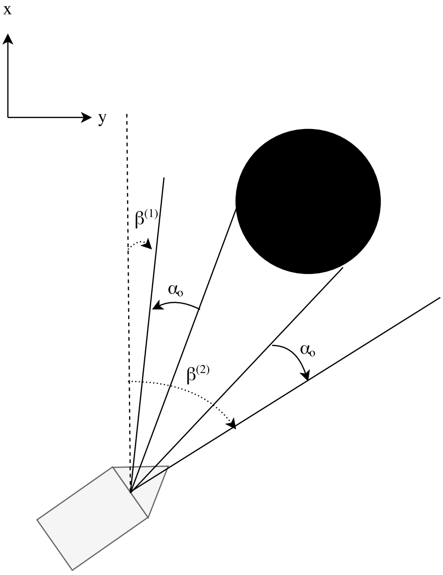

Figure 3.1: Extended vision cone

The two velocity vectors $\boldsymbol{v}_{\beta(j)}$ where $j = \{1, 2\}$ is defined along the sides of the extended vision cone

$$\boldsymbol{v}_{\beta(j)}(t) = u_{\beta(j)}(t)[\cos(\beta^{(j)}(t)), \sin(\beta^{(j)}(t))] \tag{3.16}$$

where $u_{\beta(j)}(t) \triangleq u$ to keep the forward speed of the vehicle constant.

### 3.4.2 Compensated vision cone

To compensate for moving obstacles, the extended vision cone is shifted along the direction of the obstacle velocity $\boldsymbol{v}_o$, which forms the definition of the compensated vision cone, $\mathcal{V}_o(t)$. This is illustrated in figure (3.2)

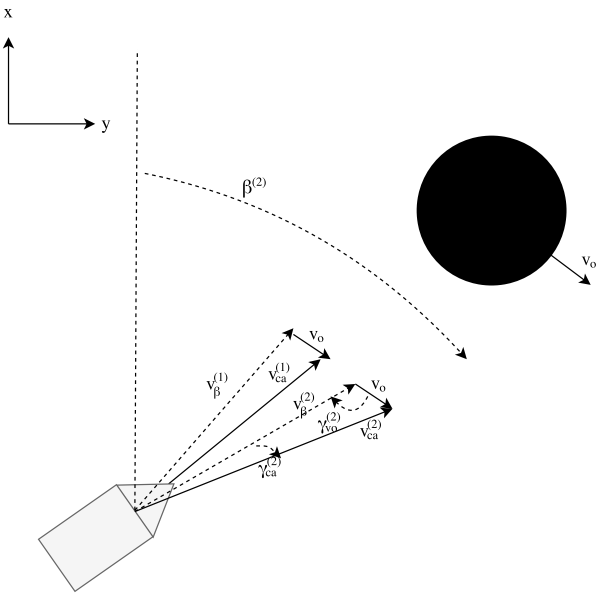

Figure 3.2: Compensated vision cone

The new velocity vector candidates are then given by

$$\boldsymbol{v}_{ca}^{(j)}(t) \triangleq \boldsymbol{v}_{\beta(j)}(t) + \boldsymbol{v}_o(t), \tag{3.17}$$

To keep a constant forward speed the length of these candidates is defined as $||\boldsymbol{v}_{ca}^{(j)}|| \triangleq u$. The angle between $\boldsymbol{v}_{\beta(j)}$ and $\boldsymbol{v}_o(t)$ is found by

$$\gamma_{\boldsymbol{v}_o}^{(j)}(t) = \pi - (\psi_o(t) - \beta^{(j)}(t)), \quad j = 1, 2, \tag{3.18}$$

the angle $\gamma_{ca}^{(j)}(t)$ in (3.2) is then given by

$$\gamma_{ca}^{(j)}(t) = \sin^{-1}\left(\frac{u_o(t)\sin(\gamma_{\boldsymbol{v}0}^{(j)}(t))}{u}\right), \quad j = 1, 2. \tag{3.19}$$

Finally, this leads to the Collision Avoidance (CA) law

$$\psi_{dca}^{(j)}(t) = \beta^{(j)}(t) + \gamma_{ca}^{(j)}(t), \quad j = 1, 2 \tag{3.20}$$

### 3.4.3 Switching rule

A switching rule is defined so that the vehicle enters CA mode at a time $t_1$ if the distance to an obstacle is smaller or equal to a chosen range $d_{switch}$ and the desired heading, $\psi_d(t)$, given by the guidance law (3.15) lies within the compensated vision cone, $\mathcal{V}_o(t_s)$

$$\psi_d(t) \in \mathcal{V}_o(t_1), \tag{3.21}$$

$$d_o(t_1) \leq d_{switch} \in (d_{\min}, d_{sense}] \tag{3.22}$$

The switching distance, $d_{switch}$, can be chosen as

$$d_{switch} = \frac{2u + \pi u_{o,max}}{r_{max}} + d_{\min} \tag{3.23}$$

This distance is mathematically proven [2] to be a sufficiently large distance for the vehicle to be able to avoid collision.

The vehicle will switch back to nominal guidance at a time, $t_2$, when the desired heading, $\psi_d(t)$ is no longer inside the vision cone. That is

$$\psi_d(t) \notin \mathcal{V}_o(t_2) \tag{3.24}$$

### 3.4.4 Turning direction

From the CA law (3.20) there are two alternatives for the desired heading. Both of these candidates avoids collision with the obstacle. However, it will often be beneficial to seek to move behind the obstacle. This is done by defining the directional parameter $j$ when the vehicle enters CA mode and choose the direction accordingly

$$j = \begin{cases} arg\max_{j=1,2} & |\psi_o(t) - \psi_{dca}^{(j)}(t)|, \quad d_o(t) = d_{switch} \\ arg\min_{j=1,2} & |\psi_o(t) - \psi_{dca}^{(j)}(t)|, \quad d_o(t) = d_{switch} \end{cases} \tag{3.25}$$

## Chapter 4

# Reciprocal CA of nonholonomic vehicles

The main contribution of this thesis is to extend the Collision Cone (CC) based Reactive Collision Avoidance (RCA) algorithm presented in Chapter 3 too the multi vehicle case. As previously mentioned in Section 2.2 the reactive nature of other vehicles raises new challenges such as oscillations and reciprocal dances. This chapter presents the alterations and extensions added to the algorithm to overcome this these challenges.

## 4.1 Inherent limitation in algorithm

To guarantee that a vehicle is able to avoid collision with an obstacle, an assumption made by [2] is that the obstacle's speed is slower or equal to the speed of the vehicle. This a reasonable assumption to make as faster and more maneuverable obstacles will always be able to reach the vehicle if it tries. However, the mathematical computation of the compensated vision cone relies on this assumption. If $u_o(t) > u$, equation (3.19) yields a result with imaginary parts and thus the algorithm breaks down. In its native intent, this is no large limitation to the algorithm [2]. However, when extending the algorithm to a reciprocal algorithm, this becomes a major limitation. In that case, all vehicles are limited to the same speed. In a real life scenario this is not a trivial assumption to make. Hence several strategies is presented so that the algorithm is able to consider hetrogeneous vehicles.

### 4.1.1 Let the faster moving agent do the collision avoidance

In a real-life scenario it would make sense that a faster, more agile vehicle will take the full responsibility of avoiding a possible collision with a more kinematic constrained vehicle. International Maritime Organization (IMO) have included some rules based on the classification of different vehicles e.g. a power-driven vehicle should always give way for a sailing vehicle and small vehicle should keep out of the way of larger vehicles in narrow passages such as channels. Automatic classification of vehicles can be hard, thus these rules are rarely considered in robotics. However, without loss of generality an easy classification can be achieved based on the forward speed of the vehicle, which is already assumed to be known from sensor data. Due to the algorithm's ability to handle dynamic obstacles, this can easily be implemented by letting vehicles ignore faster moving vehicles under the assumption that the faster moving vehicle will take full responsibility of avoiding a possible collision. Simlation from Chapter 5.7 identify some performance issues with this method, which is further discussed in Chapter 6.2.

### 4.1.2 Saturation

Another approach would be to alter equation (3.19) directly to ensure that it does not produce an imaginary result. This can be achieved by saturating the relation between $u_o(t)$ and $u$ in a manner such that the operand of $\sin^{-1}$ do not exceed 1. Thus, the equation (3.19) can be rewritten as:

$$\gamma_{ca}^{(j)}(t) = \sin^{-1}\left(sat\left(\frac{u_o(t)}{u}\right) \cdot \sin(\gamma_{\boldsymbol{v}0}^{(j)}(t))\right), \quad j = 1, 2. \tag{4.1}$$

where

$$sat(x) = \begin{cases} x & \text{if} \quad x \leq x_{max} \\ x_{max} & \text{if} \quad x > x_{max} \end{cases} \tag{4.2}$$

and the maximum bound, $x_{max}$ is equal to 1. Simulations in Chapter 5.7 illustrated the performance of this method and how it successfully extends the overall algorithm to support homogeneous vehicles. Results are discussed in greater detail in Chapter 6.2

## 4.2 CA law

When designing the Collision Avoidance (CA) law for a reciprocal CA algorithm an important aspect is to ensure that all vehicles involved in a possible collision reach the same decision regarding how to handle the CA maneuver. However, this should be achieved without any communication between the involved vehicles. By not relying on any form of communication between vehicles, the algorithm becomes completely distributed and more robust. To help define such a CA law an important symmetry property of the CC is stated:

$$\boldsymbol{v}_A \in CC_{A|B} \quad \text{if and only if} \quad \boldsymbol{v}_B \in CC_{B|A} \tag{4.3}$$

i.e. if vehicle A is on collision course with vehicle B, vehicle B is on collision course with vehicle A. As a side note; this property also exist for Velocity Obstacle (VO) [29, 28, 35, 36]. The following subsections use this symmetry property and several candidates for a new CA law are presented.

### 4.2.1 Minimizing function

According to the CA law from Equation (3.20) there exist two possible candidates for the desired heading when in CA mode. These candidates are equal to the two edges of the CC. Regardless of which candidate is chosen, the vehicle will be able to avoid the obstacle successfully. However, in most cases one of the candidates would be a far better choice than the other, e.g. it would be favorable to move behind a crossing vehicle rather than trying to move in front of it. Do to how the vision cone is compensated (3.17) by the velocity vector $\boldsymbol{v}_o(t)$, this behavior can be achieved by selecting the velocity candidate that is closest to the desired heading, $\psi_d(t)$. More formally, it can be expressed as a minimizing function.

$$\psi_{dca} = \underset{\psi(t)\notin CC}{\arg\min} |\psi_d(t) - \psi(t)| \tag{4.4}$$

This is similar to how the new velocity is chosen in both Optimal Reciprocal Collision Avoidance (ORCA) and Reciprocal Velocity Obstacle (RVO). Simulation from Section 5.7 show how vehicles often end up in deadlock when using this CA law. An analysis of why this happends is given in Section 6.1 while deadlocks is discussed in Section 6.5.

### 4.2.2 COLREGS

Due to the issues caused by the optimization function (4.4), which are discussed in Subsection 6.1.1, another CA law is suggested. Like others have done before [33], the CA law is constructed in such a manner that the vehicle upholds Rule, 13, 14 and 15 of Collision Regulations (COLREGS). By doing so, both safe and predictable behavior of the vehicles is achieved. The COLREGS states how a possible collision between two vessels should be handled by the involved parties in situations such as crossing, head-on and overtaking (2.3). It is defined that a vessel is overtaking another vessel if it comes up to the other vessel at more than 22.5 ° from the beam of the other vessel. However, the boundaries between crossing and head-on is not explicitly stated. By investigating what others have successfully implemented and tested, the boundary between crossing and head-on is defines as ±15° from the heading of the other vessel [37]. The boundaries is illustrated in Figure (4.1):

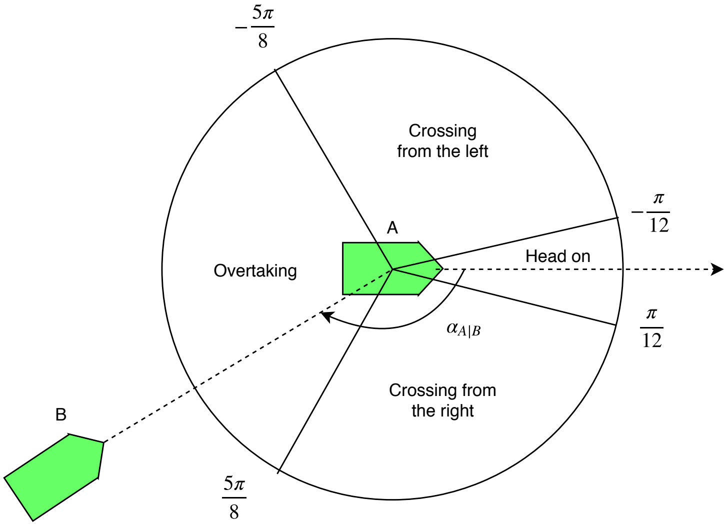

Figure 4.1: Boundaries between COLREGS situations

It is important that the involved vehicles agree on which COLREGS rule to apply. To help distinguish between the rules, two parameters are introduced:

$$\alpha_{A|B} = \arctan 2(y_A - y_B, x_A - x_B) - \psi_A \in (-\pi, \pi] \tag{4.5}$$

$$|\psi_{A,B}| = \text{abs}(\psi_A - \psi_B) \in [0, \pi] \tag{4.6}$$

where $\alpha_{A|B}$ is the relative position between vehicle A and B given in the body frame of vehicle A expressed as an angle from the heading of vehicle A and $|\psi_{A,B}|$ is the absolute relative bearing between the vehicles. It can be observed that the relative bearing have a symmetry property; $|\psi_{A,B}| = |\psi_{B,A}|$, which help to ensure that the vehicles can reach the same conclusion on which rule to apply. The same symmetry can not be found between $\alpha_{A|B}$ and $\alpha_{B|A}$ simply because they are translated into body coordinates. This translation will however make it easier and more intuitive to define which rule to apply. Table (4.1) shows how the different COLREGS situations is defined by the relative bearing and position.

| | | |
|---|---|---|
| **Head-on** | | |
| $\lvert\psi_{A,B}\rvert$ | $\in$ | $\left(-\frac{5}{8}\pi, \frac{5}{8}\pi\right)$ |
| **A is overtaking B** | | |
| $\lvert\psi_{A,B}\rvert$ | $\in$ | $\left[0, \frac{1}{12}\pi\right]$ |
| $\alpha_{A\vert B}$ | $\in$ | $\left(-\frac{5}{8}\pi, \frac{5}{8}\pi\right)$ |
| **B is overtaking A** | | |
| $\lvert\psi_{A,B}\rvert$ | $\in$ | $\left[0, \frac{1}{12}\pi\right]$ |
| $\alpha_{A\vert B}$ | $\in$ | $\left(-\pi, -\frac{5}{8}\pi\right] \cup \left[\frac{5}{8}\pi, \pi\right]$ |
| **B is crossing from left** | | |
| $\lvert\psi_{A,B}\rvert$ | $\in$ | $\left(\frac{1}{12}\pi, \frac{5}{8}\pi\right]$ |
| $\alpha_{A\vert B}$ | $\in$ | $(-\pi, 0]$ |
| **B is crossing from right** | | |
| $\lvert\psi_{A,B}\rvert$ | $\in$ | $\left(\frac{1}{12}\pi, \frac{5}{8}\pi\right]$ |
| $\alpha_{A\vert B}$ | $\in$ | $(0, \pi]$ |

Table 4.1: COLREGS parameters: $\alpha_{A|B}$ is the relative position and $|\psi_{A,B}|$ is the absolute relative bearing between vehicle A and B

Due to the definitions (4.5) and (4.6) and the bound given in Table (4.1) it is guaranteed that the involved vehicles reach the same conclusion on which rule to apply e.g. if vehicle A finds that a head-on collision with vehicle B is imminent, vehicle B will also conclude that it is in a head-on situation with A.

Rule 14 - Head-on (2.3.2) of COLREGS states that each vehicle should alter their course to starboard so that they pass on the port side of each other. Hence, the new velocity is given by (3.20) where $j = 2$.

Rule 15 - Crossing (2.3.3) states that the vehicle that have the other vehicle to its starboard side should aim to move behind that vehicle. According to COLREGS rule 17 (2.3.5) the other vehicle is not required to do any form of avoidance maneuver in a crossing situation (right-hand rule). However, as this is a reciprocal CA algorithm, it makes sense that this vehicle take part in the CA as well. Thus, it is defined that the vehicle which have the other vehicle on its port side should aim to pass in front of the other vehicle. Like for a head-on collision the new velocity for both vehicles is given by (3.20) where $j = 2$.

Rule 13 - Overtaking (2.3.1) does not define if a vehicle should overtake another vehicle on the starboard or port side. Hence, it would be natural to choose the side which requires the least amount of alteration to the current velocity. To avoid deadlock situations, both vehicles is required to choose the same velocity vector. Hence, a new optimization function that finds the velocity vector candidates that results in the least combined alterations to the desired heading of vehicle A and B while ensuring that the vehicles turn in the same direction is defined as:

$$j = \underset{j=1,2}{\arg\min} \, |(\psi_A - \psi_{dca,A}^{(j)}) + (\psi_B - \psi_{dca,B}^{(j)})| \tag{4.7}$$

Note that this assumes that the vehicles have access to each others CC. The CC of surrounding vehicles can be computed from information already known to the vehicle. Hence, there is no loss in generality. Simulations in Section 5.2 shows how the COLREGS rules from Section 2.3 is respected while ensuring safe, oscillation free navigation. The proposed CA law is further benchmarked in Monte Carlo simulation in Section 5.7, where some cases of reciprocal dances is observed. The CA law is further discussed in Section 6.1 and reciprocal dances is addressed in Section 6.6.

### 4.2.3 Roundabout policy

A simple yet effective way of guaranteeing that each vehicle involved in a possible collision chooses to pass the other vehicles on the same side as is to implement traffic rules corresponding to those found in roundabouts. Specifically, the control strategy ensures that each vehicle chooses the same velocity vector candidate and turns in a counter-clockwise fashion when avoiding other vehicles. A similar policy is implemented in [30] on a multiple homogeneous vehicles and it has been shown how all vehicles are able to reach their targets positions without deadlocks and reciprocal dances. To implemented this behavior the CA law (3.20) is adjusted:

$$\psi_{dca}^{(2)}(t) = \beta^{(2)}(t) + \gamma_{ca}^{(2)}(t) \tag{4.8}$$

i.e. the turning direction is always counter-clockwise regardless of the COLREGS situation. Hence, the parameters $\alpha_{A|B}$ and $|\psi_{A,B}|$ are not needed in this control strategy. Monte Carlo simulations is presented in Section 5.7 and the CA law is further discussed and compared to the aforementioned CA laws in Section 6.1.

### 4.2.4 Closest point of approach

It is possible to add a level of CA above the reciprocal CA algorithm, in order to decide which COLREGS rule to apply for each vehicle. This is for example done in [33] where VO is used as an RCA and the Closest Point of Approach (Closest Point of Approach (CPA)) is used as an upper layer CA. The CPA decides which COLREGS rule to apply to each vehicle under the assumption that each vehicles respects the COLREGS. Hence, the vehicle can respect the give-away and stand-on behavior defined in COLREGS rule 16 and 17. In the case that a vehicle do not respect the COLREGS the VO acts as a fall-back CA to ensure collision free navigation under all circumstances.

To the end of illustrating how the reciprocal CA in this thesis can be extended to respect the stand-on behavior defined by COLREGS Rule 17 the CPA are added as an upper-layer collision detection in a similar fashion as in [33]. The symmetry between $CC$'s defined in Equation (4.3) limits the possibility of incorporating this behavior directly into the reciprocal CA. It would be possible to ignore $CC$'s induced by vehicles on the port side under the assumption that all vehicles respect the COLREGS. However, this assumption greatly limits the use the algorithm and collisions will occur if the other vehicle or obstacle does not respect the COLREGS. Hence, a more general approach would be to add a second layer of CA and use the reciprocal CA as a fall-back algorithm.

The CPA is implemented according to its definition which is given in Section 2.4. If CPA detects a possible collision between vehicle A and B, each vehicles enters CPA mode and calculates which COLREGS situation they are in, according to the method described in Subsection 4.2.2. The CA law is equal for all situations with crossing from the left as the only exception. This is when the vehicle can ignore the collision under the assumption that the other vehicle will give away. The CA law described in Section 4.2.2 is still implemented for the regular CA mode which is activated by the switching rule given by Equation (3.23). The switching rule for CPA mode is given by:

$$d_{\text{cpa}} \leq (R_A + R_B) + d_{\text{min}} \tag{4.9}$$

$$t_{\text{cpa}} \leq t_{\text{cpa}\,\text{min}} \tag{4.10}$$

where $d_{\text{cpa}}$ is the minimum distance between the vehicle at any future time. Due to how the CPA considers points in a plane, a collision between vehicle A and B occurs when $d_{\text{cpa}}$ is less than their combined radius plus the $d_{\text{min}}$ bound. The time until collision bound, $t_{\text{cpa}\,\text{min}}$ should be defined such that CPA mode activates before the regular CA mode and gives the vehicles enough time to execute a CA maneuver without interference from the CA law.

This method do not fall in the category of reciprocal CA as the intended behavior is for only one vehicle to perform the avoidance maneuver. However, it does extend the usage of the algorithm. Simulations showing how the crossing situation is handled according to Rule 16 and 17 of the COLREGS is presented in Section 5.3. As this thesis focuses in reciprocal CA, this method is not further discussed.

## 4.3 Responsibility

In Chapter 3 it is assumed that all obstacles are non-reactive i.e. they do not take any part in resolving a possible collision. Thus, the algorithm [2] is design in a manner where the vehicle take full responsibility of avoiding any potential collisions. In a multi vehicle scenario where all vehicles are assumed to be implemented with the same algorithm it is both possible and reasonable to distribute the responsibility of avoiding a collision between the vehicles involved. This reasoning is used in many of the reciprocal CA algorithms reviewed in Chapter (2), such as ORCA, Bicycle Reciprocal Collision Avoidance (B-ORCA), RVO and Hybrid Reciprocal Velocity Obstacle (HRVO).

In [2] the CC is extended by an constant avoidance angle, $\alpha_o$. The authors present a bound on this angle to ensure that the vehicle can keep a minimum distance, $d_{\min}$ to any obstacle

$$\alpha_o \geq \sin^{-1}\left(\frac{R_o}{R_o + d_{\min}}\right) \tag{4.11}$$

Under the assumption that the obstacle is also a reactive vehicle which will take half the responsibility of avoiding a collision, this bound can be relaxed to

$$\alpha_o \geq 0.5\sin^{-1}\left(\frac{R_o}{R_o + d_{\min}}\right) \tag{4.12}$$

This is similar to how the VO is relaxed in RVO. The difference between the reduced cone, the extended vision cone (3.19) and the original CC is illustrated in Figure (4.3). Note that the reduced cone can not be used with the speed alteration proposed in Section 4.1.1.

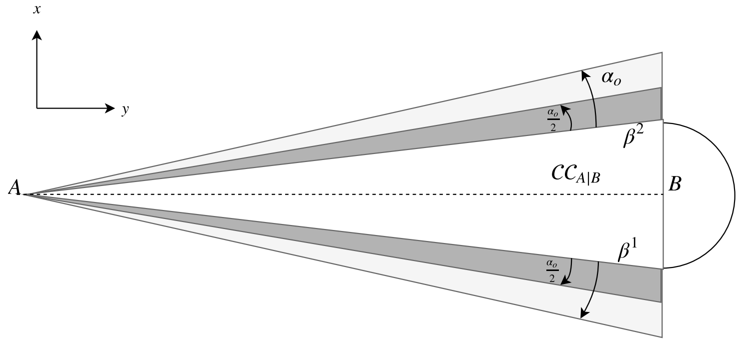

Figure 4.2: Reduced cone: The light gray cone is the extended vision cone (3.19), the white cone represents the CC as presented in (2.1.1) and the dark gray shown the newly proposed reduced vision cone

The vehicle response in each of the COLREGS situations defined in Section 2.3 is presented in form of a distance plot between the vehicles in Section 5.4. The reduced cone is further discussed in Section 6.3.

## 4.4 Hysteresis

Oscillation is a common issue caused by the reactive nature of other vehicles. To ensure that the vehicles not only reaches their target, but does so in an oscillation free manner, a hysteresis on the CA law is included. By only letting the vehicle select their turning direction (velocity vector candidate) when it enters CA mode, oscillations caused by the reactiveness of the other vehicle are avoided. Also, to not make the algorithm too restrictive, a vehicle is given permission to select a new turning direction if a new vehicle enters the collision situation. The new turning direction in the same way as when entering CA mode, as described in Section 4.2.

# Chapter 5

# Results

## 5.1 Simulation parameters and figure explanations

All vehicles are modeled by the nonholonomic vehicle model given by Equation (3.1). Unless other information is stated, the simulations in this chapter are done with the following parameters; the radius of all vehicles is, $R = 1$. The minimum safe distance is $d_{\text{min}} = 1$ m. The forward speed, $u = 1$ m/s and the maximum turning rate is $r_{max} = 1$ rad/s. In the simulation plots, the vehicles are illustrated with a black circle with radius, $R$, the current velocity is represented by a pink arrow with origin in the center of the vehicle. When a vehicle is in a Collision Avoidance (CA) mode, the edges of the Collision Cone (CC) is represented by two red lines i.e. the velocity vector candidates. The target positions is marked with a green ’x’.

## 5.2 COLREG scenarios

Simulations showing how the algorithm defined in Subsection 4.2.2 handles scenarios like overtaking, head-on, crossing from left and crossing from right as defined by Collision Regulations (COLREGS) in Section 2.3. Note that the vehicle response is identical when using the roundabout policy from Subsection 4.2.3 in these scenarios.

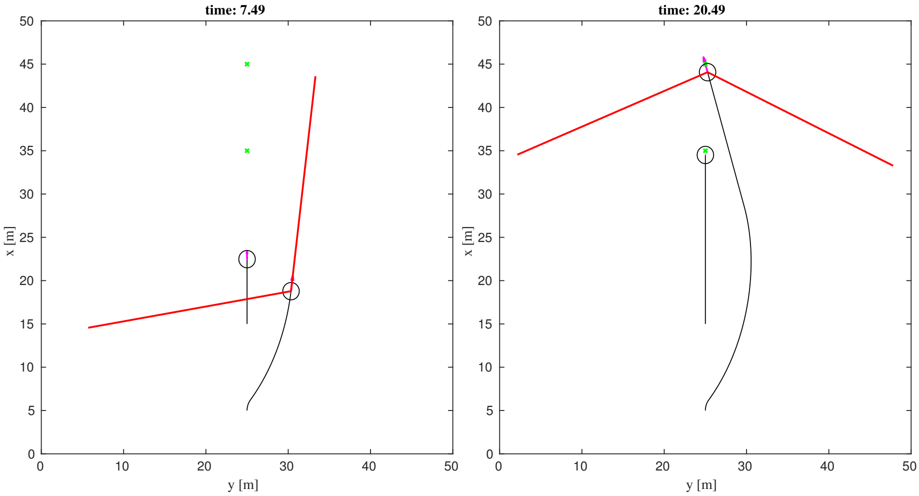

Figure 5.1: Overtaking as defined by COLREGS Rule 13 (2.3.1)

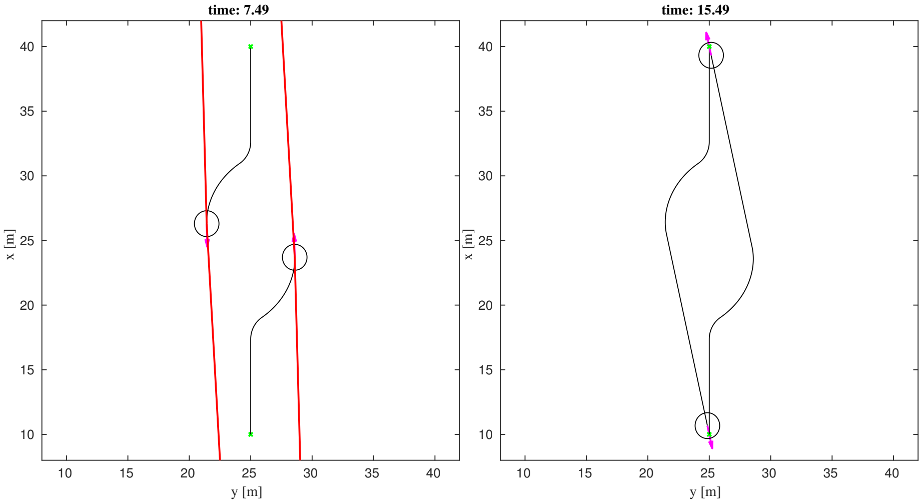

Figure 5.2: Head-on as defined by COLREGS Rule 14 (2.3.1)

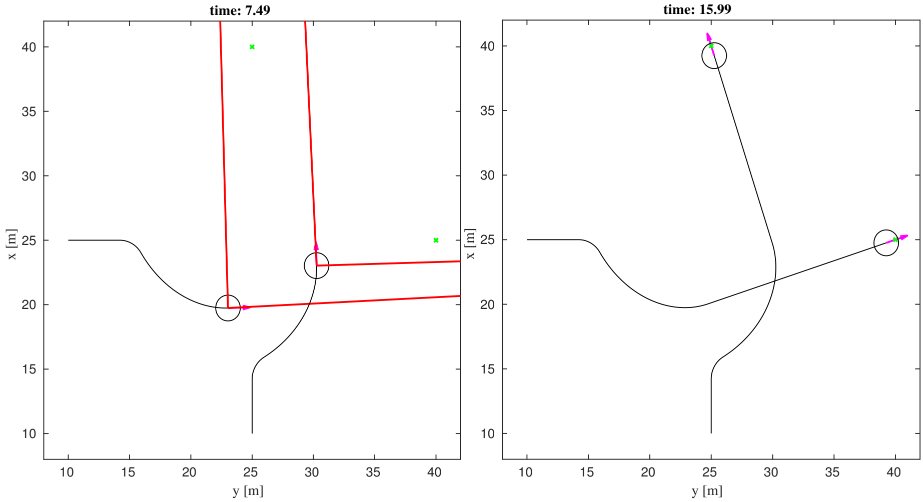

Figure 5.3: Crossing from the right as defined by COLREGS Rule 15 (2.3.3)

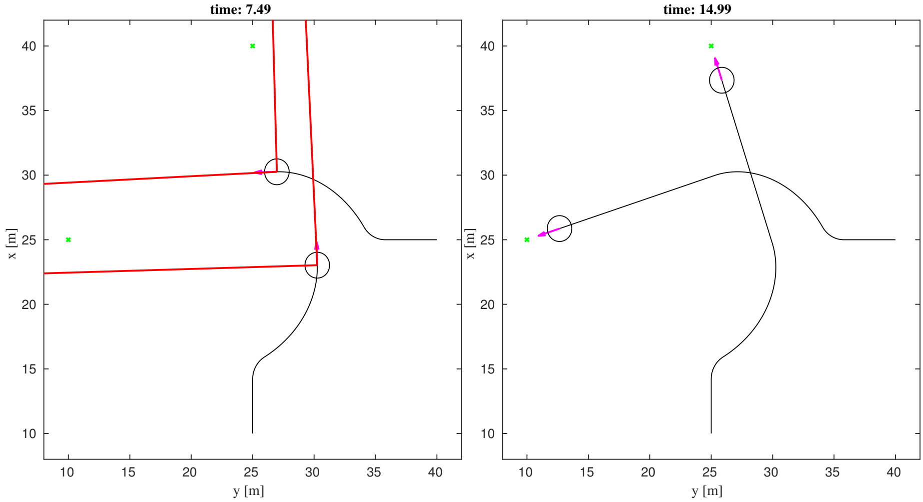

Figure 5.4: Crossing from the left as defined by COLREGS Rule 15 (2.3.3)

## 5.3 Crossing situation using CPA

Figure 5.5 illustrates how the upper layer collision detections proposed in Section 4.2.4 is able to break CC symmetry and let one vehicle stand-on while the other vehicle gives-away. This is the expected vehicle behavior in a crossing sitation according to COLREGS rule 16 and 17. See Section 2.3 for a full description of these rules.

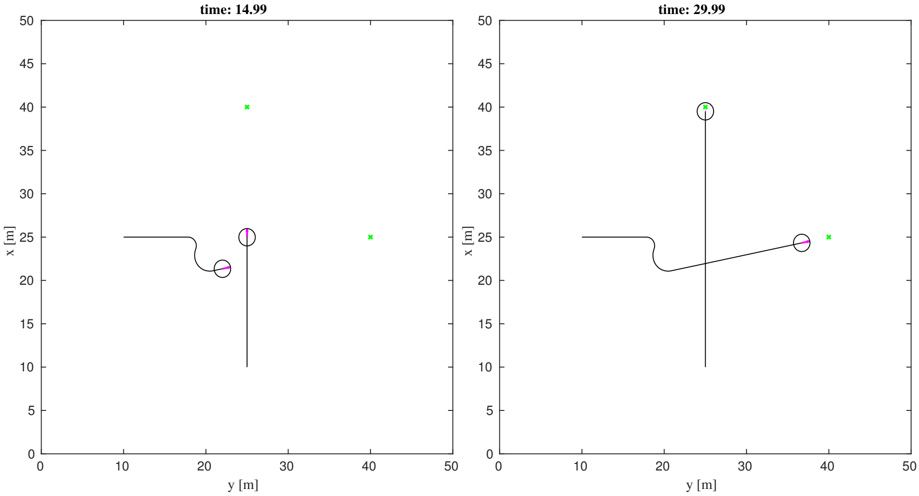

Figure 5.5: Crossing from the left as defined by COLREGS Rule 15, using the CPA to break CC symmetry. According to Rule 15, a vehicle with another vehicle on its starboard side should give away whilst the other vehicles stands by.

## 5.4 Reduced Cone

Simulations illustrating the difference between the reduced cone introduced in Section 4.3 and the original compensated vision cone defined in Subsection 3.4.2. The plots show the distance between two vehicles in the following COLREGS situations: crossing, head-on and overtaking. The time where the vehicles enters CA mode can be read as the point where the slopes changes and the time when the conflict is resolved can be read as the point where the distance between the vehicles is at its minima.

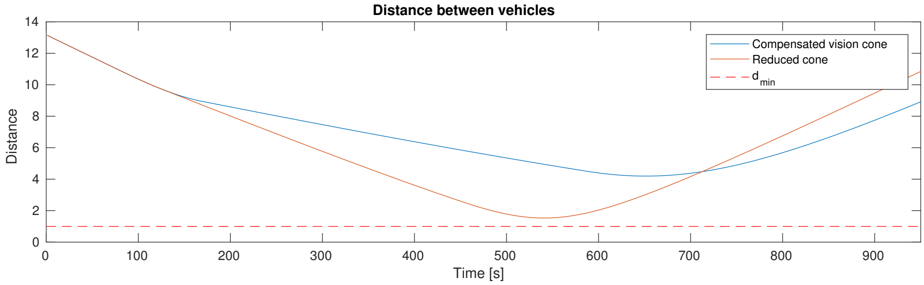

Figure 5.6: COLREGS crossing situation: distance comparison between the compensated vision cone (3.17) and the reduced vision cone (4.3)

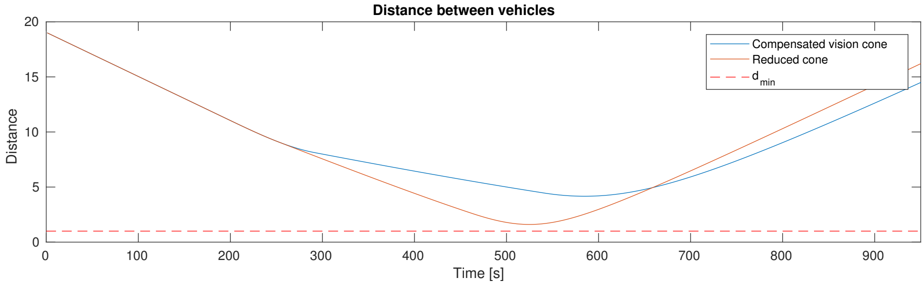

Figure 5.7: COLREGS head-on situation: distance comparison between the compensated vision cone (3.17) and the reduced vision cone (4.3)

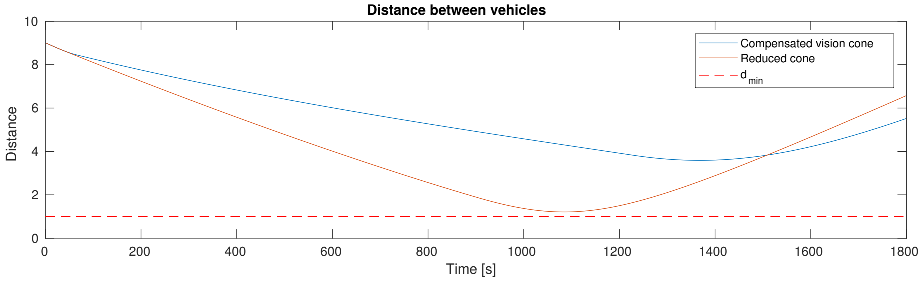

Figure 5.8: COLREGS overtaking situation: distance comparison between the compensated vision cone (3.17) and the reduced vision cone (4.3)

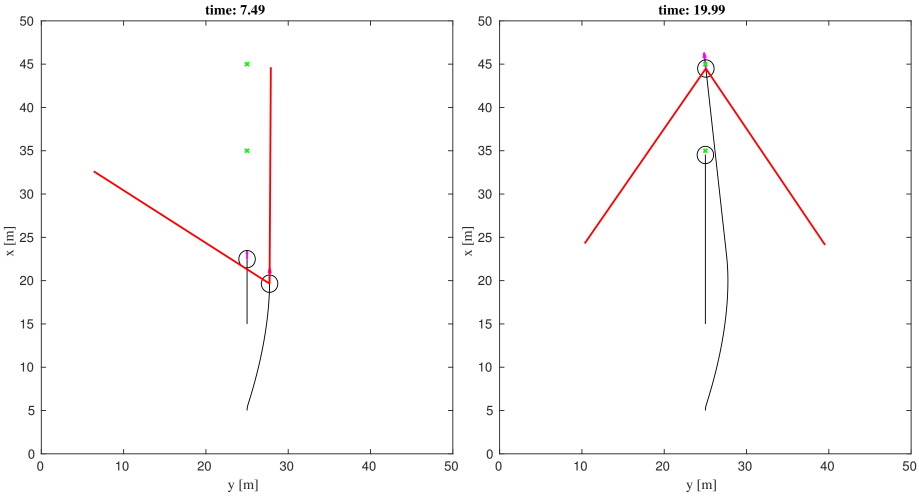

Figure 5.9: Reduced Cone: Overtaking as defined by COLREGS Rule 13 (2.3.1)

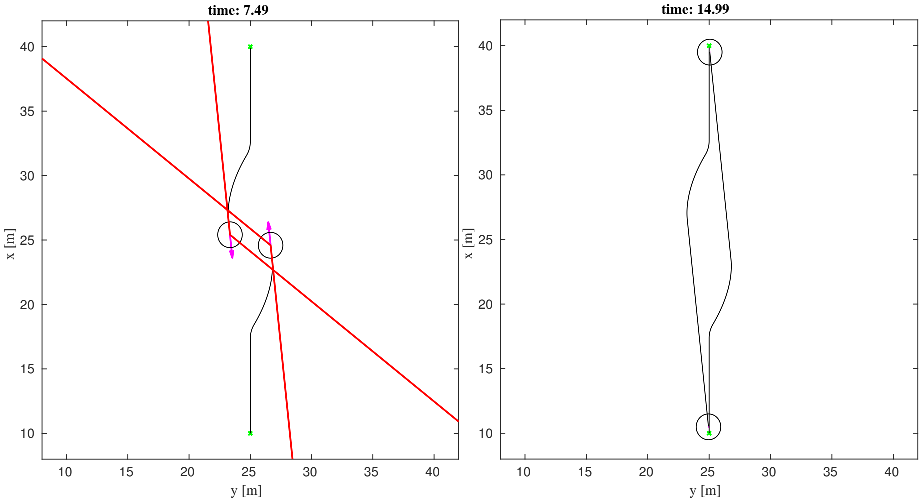

Figure 5.10: Reduced Cone: Head-on as defined by COLREGS Rule 14 (2.3.1)

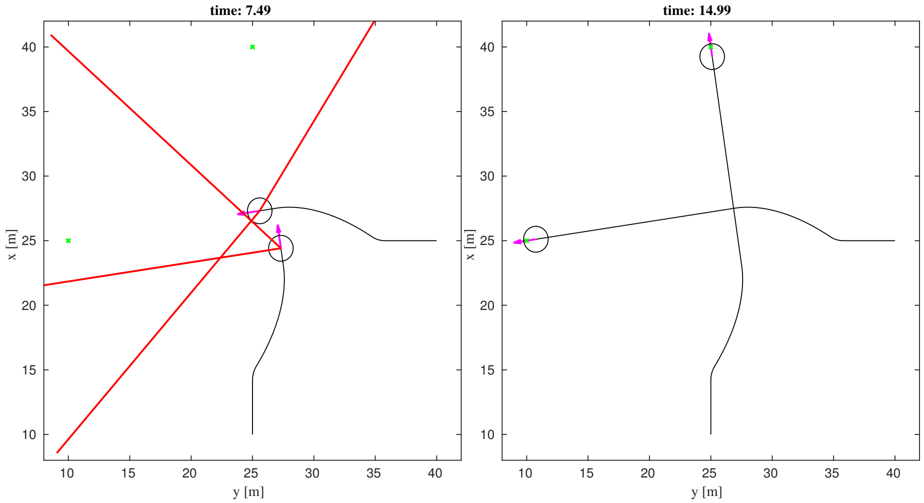

Figure 5.11: Reduced Cone: Crossing from the left as defined by COLREGS Rule 15 (2.3.3)

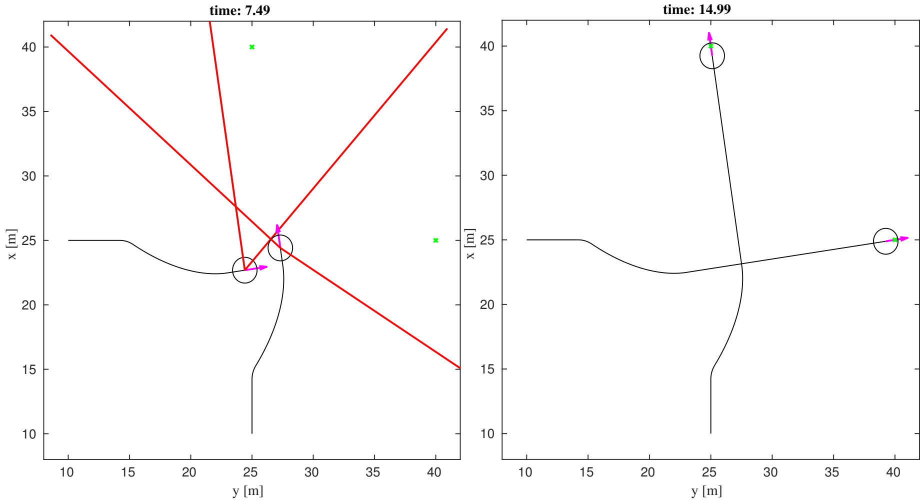

Figure 5.12: Reduced Cone: Crossing from the left as defined by COLREGS Rule 15 (2.3.3)

## 5.5 10 homogeneous vehicles

Figure 5.13 illustrates how 10 homogeneous vehicles, all implemented with the CA law introduced in Section 4.2.3, reach their respective targets without oscillations, reciprocal dances or deadlocks. The area of which the vehicles is generated within is 50x50.

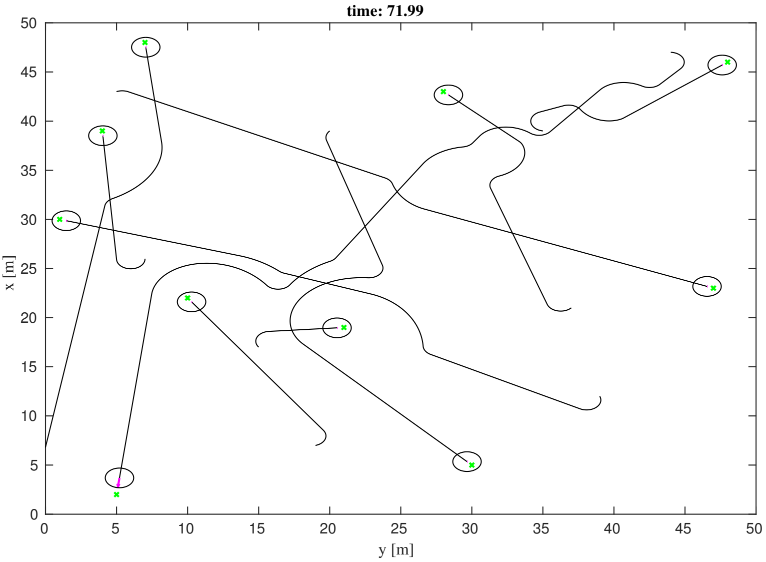

Figure 5.13: 10 homogeneous vehicles traversing an unknown environment. All vehicles is implemented with the CA law from Section 4.2.3

## 5.6 Exchange of antipodal positions on a circle

The exchange of antipodal positions on a circle is often regarded as a worst case scenario as all vehicles intend to pass through the same point at the same time instant. Each vehicles have a start and goal position on the radius of a circle, opposite of one another. Hence, the preferred trajectory heads through the center of the circle. The following simulations shows the vehicle response of 8 vehicles, all implemented with the CA law introduced in Subsection 4.2.2.

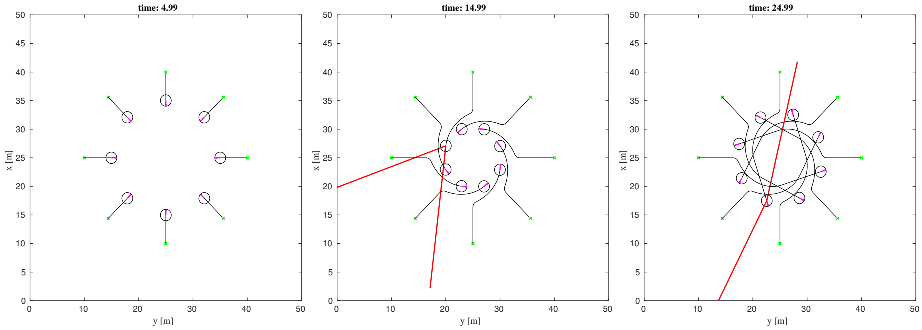

Figure 5.14: Exchange of antipodal positions on a circle w/ 8 vehciles using the COLREGS based CA law introduced in Subsection 4.2.2

## 5.7 Monte Carlo simulations

A Monte Carlo simulation is technique where repeated simulation with random values is executed to study the response of a model [38]. In this thesis Monte Carlo simulations are used to verify the robustness of the CA algorithm and to identify its weaknesses. The environment for the Monte Carlo simulations is constructed as an m-by-m area where both the starting position, $\boldsymbol{p}_0(0)$, and the target position, $\boldsymbol{p}_t$ is on the outside of the area. The area is 30x30 when simulating more than two vehicles. To increase the number of simulation where CA mode is activated, the area is reduced to 10x10 when the simulation includes only two vehicles. Both $\boldsymbol{p}_0(0)$ and $\boldsymbol{p}_t$ is generated with random values for each simulation. Unless stated otherwise, the forward speed $u$ is equal for all vehicles. To look for possible deadlocks and reciprocal dances, a simulation is marked as Did Not Finish (DNF) if not all targets have reached their target ($\boldsymbol{p}(t) = \boldsymbol{p}_t$) within a specified time, $t_{stop}$. This is defined as three times the average time it takes for all vehicles reach their targets. The average completion time is found from 10 successful simulation done a-priori to the Monte Carlo simulation.

| | |
|---|---|
| Number of simulations | 1000 |
| Number of vehicles | 2 |
| Success | 79.8 % |
| DNF | 20.2 % |
| $d_{\min}$ violations | 0.0 % |
| Crash | 0.0 % |
| CA Mode activated | 43.2 % |
| Average completion time | 20.3 sec |

| | |
|---|---|
| Number of simulations | 1000 |
| Number of vehicles | 4 |
| Success | 60.8 % |
| DNF | 38.8 % |
| $d_{\min}$ violations | 0.1 % |
| Crash | 0.3 % |
| CA Mode activated | 88.2 % |
| Average completion time | 38.7 sec |

Table 5.1: Simulations using the optimization function from Equation (4.4) as CA law

| | |
|---|---|
| Number of simulations | 1000 |
| Number of vehicles | 2 |
| Success | 100.0 % |
| DNF | 0.0 % |
| $d_{\min}$ violations | 0.0 % |
| Crash | 0.0 % |
| CA Mode activated | 41.0 % |
| Average completion time | 21.8 sec |

| | |
|---|---|
| Number of simulations | 1000 |
| Number of vehicles | 4 |
| Success | 97.4 % |
| DNF | 1.8 % |
| $d_{\min}$ violations | 0.8 % |
| Crash | 0.0 % |
| CA Mode activated | 87.7 % |
| Average completion time | 43.5 sec |

Table 5.2: Simulations using the COLREGS parameters form Equation (4.5) and (4.6) as described in Section 4.2.2

| | |
|---|---|
| Number of simulations | 1000 |
| Number of vehicles | 2 |
| Success | 100.0 % |
| DNF | 0.0 % |
| $d_{\min}$ violations | 0.0 % |
| Crash | 0.0 % |
| CA Mode activated | 42.8 % |
| Average completion time | 22.0 sec |

| | |
|---|---|
| Number of simulations | 1000 |
| Number of vehicles | 4 |
| Success | 98.5 % |
| DNF | 1.0 % |
| $d_{\min}$ violations | 0.25 % |
| Crash | 0.25 % |
| CA Mode activated | 88.9 % |
| Average completion time | 45.2 sec |

Table 5.3: Simulation using the roundabout policy presented in Section 4.2.3 as CA law

| Number of simulations | 1000 |
|---|---|
| Number of vehicles | 6 |
| Success | 87.8 % |
| DNF | 11.5 % |
| $d_{\min}$ violations | 0.5 % |
| Crash | 0.2 % |
| CA Mode activated | 99.8 % |
| Average completion time | 82.9 sec |

| Number of simulations | 1000 |
|---|---|
| Number of vehicles | 6 |
| Success | 93.5 % |
| DNF | 6.3 % |
| $d_{\min}$ violations | 0.1 % |
| Crash | 0.1 % |
| CA Mode activated | 99.1 % |
| Average completion time | 95.8 sec |

Table 5.4: Density comparison: The left table show simulation results with a environment size of 40x40. In the simulation in the right table the size is increased to 60x60. The CA law is defined by the roundabout policy from Section 4.2.3

| Number of simulations | 1000 |
|---|---|
| Number of vehicles | 4 |
| Success | 98.3 % |
| DNF | 0.6 % |
| $d_{\min}$ violations | 0.9 % |
| Crash | 0.2 % |
| CA Mode activated | 92.5 % |
| Average completion time | 44.0 sec |

| Number of simulations | 1000 |
|---|---|
| Number of vehicles | 4 |
| Success | 90.3 % |
| DNF | 4.1 % |
| $d_{\min}$ violations | 2.7 % |
| Crash | 2.9 % |
| CA Mode activated | 96.7 % |
| Average completion time | 48.4 sec |

Table 5.5: Simulation of 4 heterogeneous vehicles with different forward speed, $u \in [0.5, 1.5]$m/s. Left table shows results when the compensated vision cone is saturated. In the simulation in the table to the right, the vehicles only considers slower moving vehicles. The CA law is defined by the roundabout policy from Section 4.2.3

| | |
|---|---|
| Number of simulations | 1000 |
| Number of vehicles | 3 |
| Number of obstacle | 1 |
| Success | 93.8 % |
| DNF | 3.4 % |
| $d_{\text{min}}$ violations | 1.7 % |
| Crash | 1.1 % |
| CA Mode activated | 95.9 % |
| Average completion time | 62.9 sec |

Table 5.6: Three vehicles traverse an environment which also include a non-reactive dynamic obstacle. The CA law is defined by the roundabout policy from Section 4.2.3

## 5.8 Deadlocks

Figure 5.15 illustrates how the CA law defined by Equation 4.4 fails to preserve the liveness of two vehicles in a simple crossing situation. Both vehicles tries to pass each other on the same side resulting in a parallel configuration between the vehicles. As the vehicles have the same dynamics i.e. they are modeled by the same nonholonomic model (3.1) and have the same constant forward speed and turning rate, the vehicles are not able to escape the parallel configuration. Hence, none of the vehicles are able to reach their targets.

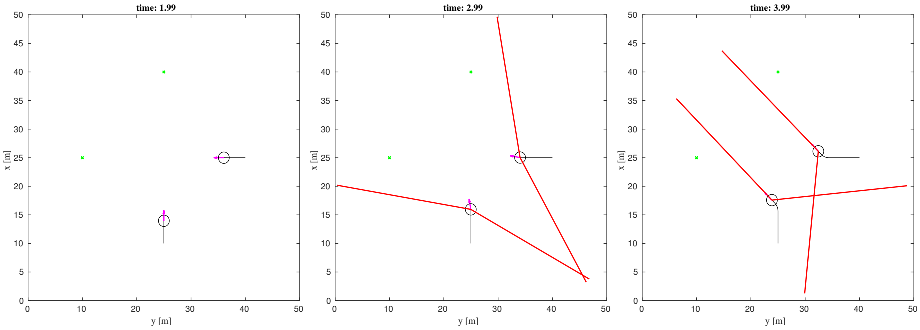

Figure 5.15: Deadlock: Both vehicles makes a greedy choice of turning direction by choosing to follow the CC edge closest to their desired heading. In this case that means they do not choose to cross each other on the same side and end in a parallel configuration

## 5.9 Reciprocal dances

Simulation showing how reciprocal dances affects the liveness of three vehicles. Section 6.6 gives an analysis of the origin of reciprocal dances and the different outcomes of such unwanted vehicle behavior. Figure 5.16 is collected from a Monte Carlo simulation that was registered as DNF, i.e. the three vehicles did not reach their target within a reasonable amount of time.

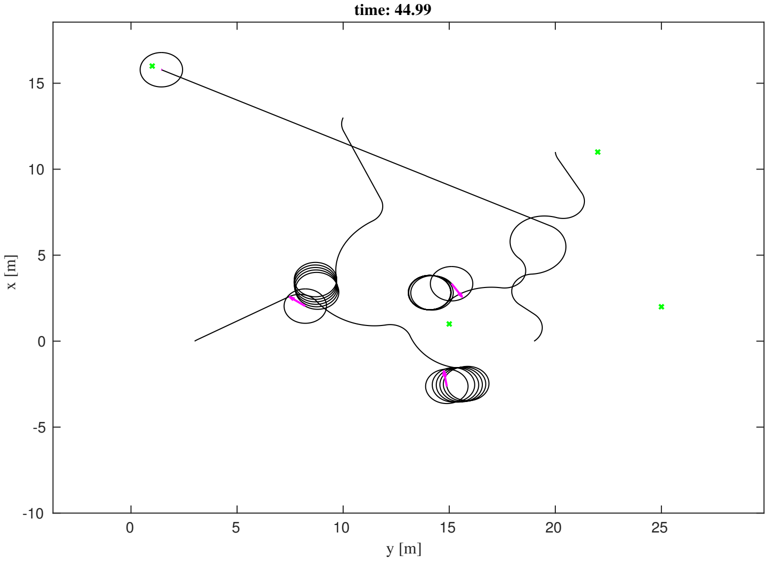

Figure 5.16: Reciprocal dance of three vehicles. This behavior is discussed in Section 6.6

# Chapter 6

# Discussion

This chapter gives a discussion of the reciprocal Collision Avoidance (CA) approach developed in this thesis. Section 6.1, 6.2 and 6.3 discusses the proposed extensions presented in Chapter 4 while evaluating the results presented in Chapter 5. These sections also offers reasoning for the choices made when developing this approach. Strengths and shortcomings of this approach is reviewed and suggestions for improvements are given. Section 6.4, 6.5 and 6.6 defines some common issues experienced by reciprocal CA, issues that is identified in the results from Chapter 5. An analysis of why these issues occur is also given.

## 6.1 CA law options

From the simulations in Chapter 5 a quite substantial difference in performance between the different control strategies presented in Section 4.2 is observed. A discussion on why the CA laws perform differently is given below.

### 6.1.1 Issues with the minimizing function

In other reciprocal CA algorithms such as Optimal Reciprocal Collision Avoidance (ORCA) and Reciprocal Velocity Obstacle (RVO), a minimizing function is successfully implemented to ensure optimal CA in the sense of minimizing the error between the desired velocity and the new velocity chosen by the CA law. However, when using a minimization function in a similar fashion with a Collision Cone (CC) approach, simulations from Table 5.1 show that in a large amount of situations the vehicles will not reach their target due to deadlocks. The reason this happens is tracked back to the difference between the $\mathcal{V}\rceil\updownarrow\wr\rfloor\rangle\sqcup\dagger\mathcal{O}\lfloor\int\sqcup\dashv\rfloor\updownarrow\rceil(\mathcal{VO})$ and the $CC$. Algorithms based on the VO make use of the following symmetry property between $\mathcal{VO}$'s: If the velocity of vehicle A, $\boldsymbol{v}_A$ is on the left side of the centerline of $\mathcal{VO}^A_B$, then the velocity of vehicle B, $\boldsymbol{v}_B$ is on the left side of the centerline of $\mathcal{VO}^B_A$. This implies that if both vehicles choose the optimal velocity, i.e. the velocity closest to the desired velocity, they will both decide to pass each other on the same side. The $CC$ does not possess this symmetry property and thus no guarantee that both vehicles will chose to pass each other on the same side can be made. This is illustrated in (5.15) in the second figure where both vehicles have just entered CA mode. The desired heading, $\psi_d(t)$ of the first vehicle is to the left of the centerline of the $CC_{A|B}$ while $\psi_d(t)$ of the other vehicle is to the right of the centerline of $CC_{B|A}$. Hence, when both vehicles chooses the optimal CA velocity for itself, the vehicles will pass each other on different sides, which eventually lead to a deadlock situation. Deadlocks are further discussed in Section (6.5).

### 6.1.2 COLREGS based steering

Using the Collision Regulations (COLREGS) parameters defined by Equation (4.5) and (4.6) and the boundaries defined in Table (4.1) simulations in Chapter 5 show how the algorithm ensures safe traversal while maneuvering in a predictable way relating to the COLREGS rules presented in Section 2.3. Furthermore, the reliability of this method is tested in a Monte Carlo simulation (Table 5.2) and the results show that this method guarantees both safety and liveness for any situation involving two vehicles. However, simulations shows that in some cases, the introduction of a third vehicle can cause deadlock. This can be explained by the break in symmetry between $CC_B^A$ and $CC_A^B$ when there also exist a cone, $CC_C^A$, which vehicle B does not know about. Hence, vehicle A and B have a different basis when determining how to pass each other which may cause the vehicles to choose different strategies when passing each other. This can in turn result in deadlocks. It has been shown that RVO suffers from a similar scalability problem [35]. Situations involving more than two vehicles can be complicated and many configurations of vehicles exist, thus the COLREGS do not spesify any rules for collisions involving more than two vehicles.

To counter the problem with scalability when using COLREGS parameters and optimization function given in Subsection 4.2.2 the roundabout policy presented in Subsection 4.2.3 is introduced as an alternative control strategy. By defining that all vehicle should move in a counter clock-wise fashion there is no need for complete symmetry between all CC's induced on the vehicles. A small improvement in performance in liveness can be observed when comparing Table 5.2 and 5.3 in the case with 4 vehicles. By examining the remaining Did Not Finish (DNF) cases in Table 5.3 it is confirmed that they are caused by reciprocal dances and not due to deadlocks. Thus, it can be concluded that the roundabout policy do solve the problem of asymmetry described above. For both approaches, the observed $d_{\min}$ violations and crashes are caused by reciprocal dances, which is discussed in more detail in Section 6.6.

Although Table 5.2 and 5.3 does not reflect it, it is not hard to imagine overtaking situations where the roundabout policy is suboptimal in a minimum-time and minimum-distance sense compared to the optimization function (4.7). Also it should be noted that the optimization function which is initially designed for the overtaking case from Section 2.3.1 could be used regardless of the COLREGS situation. In the case of resolving a collision between two vehicles, this would be a more optimal approach. However, there is no guarantee that the vehicles respect the rules defined by COLREGS in this case.

## 6.2 Candidates for handling hetrogeneous vehicles

Section 4.1 identifies a mathematical limitation in how the vision cone from Section 3.4.2 is compensated by the velocity of the obstacle. When the obstacle speed, $u_o(t)$, is larger than the vehicle speed, $u$, the operand of $\sin^{-1}$ in Equation (3.17) becomes larger than 1, which yields an imaginary result, thus the computation of the compensated vision cone breaks down. In the native intent of [2], it is trivial to assume that the vehicle moves faster than surrounding obstacle. However, in a reciprocal CA approach, this limits the algorithm to homogeneous vehicles, i.e. all vehicles have the same forward speed. Hence, two strategies for handling heterogeneous vehicles is presented in Section 4.1.

One strategy is based on classification of vehicles i.e. by letting faster moving vehicles take full responsibility of a potential collision. In this case, vehicles do not consider the faster moving vehicles under the assumption that the other vehicle will give-away to resolve the conflict. This method is supported by the COLREGS which include similar rules for different classes of vehicles. These rules are rarely used in robotics as it can be hard to automatically classify other vehicles. However, in this case, the classification only consist of the forward speed of other vehicles which is already assumed to be known through sensor data. Hence, there is no loss of generality. However, the assumption needed for the reduced cone presented in Section 4.3 does not hold in this case. Simulation results presented in Table 5.5 shows that the performance of the algorithm has dropped when vehicles ignore faster moving obstacles. These issues traces back the definition of the COLREGS based CA law. In crossing cases, it is defined that if vehicle A have vehicle B on its starboard side, then vehicle A should aim to move behind vehicle B and vehicle B should aim to move in front of vehicle A i.e. both vehicles move in a counter-clockwise manner. In the case that vehicle A is non-reactive, this situation will result in a deadlock. It is important to note these deadlocks will be resolved when the non-reactive vehicle reaches its target and the reactive vehicle can maneuver around it. Hence not all deadlocks is recorded as DNF in the results. This is considered to be an uncertainty in the simulations. However, in a real-life scenario heterogeneous vehicles will be able to resolve deadlocks when they have different forwards speed. The deadlocks caused by non-reactive vehicles is been identified as the reason behind the drop in performance compared the homogeneous case. The deadlocks often results in more densely packed areas e.g. two vehicles are in a deadlock and then encounters a third vehicle, which in turn increases the chances for reciprocal dances.

The other solution presented in Section 4.1 is to saturate the relation between $u_o(t)$ and $u$ such that Equation (3.17) do not produce an imaginary result. It is easy to imagine that this will cause some problems in the sense that the vision cone is not compensated enough, which could result in collisions. However, through simulations presented in Table 5.5 it have been verified that this is not the case. In fact, not a single instant where this is the case have been identified when investigating the failed simulations. Note that there have been done little research on this when $u << u_o(t)$. Simulations also show that the saturated cone yields similar results as the case where only homogeneous vehicles are considered.

## 6.3 The reduced collision cone

The introduction of the reduced CC is motivated by the assumption that all vehicles are reactive and will take half the responsibility of avoiding a potential collision. In contrast, the compensated vision cone (3.17) is design based on the assumption that a vehicle will take full responsibility of avoiding collision. In the case of reciprocal CA the compensated vision cone becomes overly restrictive and simulations (5.2) of vehicles in COLREGS situations shows how this leads to suboptimal behavior. The plots show that when each vehicle takes full responsibility i.e. when using the compensated vision cone, they pass each other at a much larger distance than what is required by the algorithm. It can also be observed that the vehicles enters CA mode earlier and that the CA maneuver takes longer time to execute. Although the reduced cone results in a more optimal CA, both in a minimum time and minimum distance sense the minimum distance bound, $d_{\min}$, set by the algorithm is being respected. It should be noted that faster conflict resolution times help improve performance of the overall approach in densely packed areas e.g. if vehicle A and B can resolve a conflict between them before vehicle C enters the conflict, the complexity of the conflict is greatly reduced.

The reduced cone improves the performance of the algorithm in dense environments. As discussed in Section 6.6 reciprocal dances occur due to large CC's, hence reducing the cone reduces the occurrences of reciprocal dances.

However, it is possible to construct cases where the reduced CC will violate the minimum distance bound, $d_{\min}$. Due to how the different CA laws presented in Section 4.2 are defined, there will be situations where a vehicle can choose a new heading that is not directly contributing to avoiding a collision. E.g. in the case of vehicles changing antipodal positions on a circle (5.6). The second time instant shows how the CA law chooses a velocity vector candidate that is pointing directly towards another vehicle, hence the assumption that it takes half the responsibility of avoiding the collision do not hold in this case and could lead to $d_{\min}$ violations or crashes.

Two suggestions on how to counter this issue are given here. One possibility is to redefine the CA law so that it can be guaranteed that the vehicles will always move away from each other. However, this can be hard to guarantee in densely packed environments. Another possibility is to design a more sophisticated weighing of the responsibility of each vehicle, rather than given 50% to each vehicle. The percentage of responsibility of a vehicle involved in a conflict can be calculated based on the configuration of the vehicles involved. E.g. in an overtaking case, the vehicle coming from behind have a larger responsibility than the one in front.

## 6.4 Oscillations

A well-know issue with using VO in reciprocal CA algorithms is that the VO cause unwanted oscillations due to the reactiveness of other vehicles [27, 39]. Even simple situations like a head-on situation between two vehicles is prune to oscillations caused by the VO. The oscillations can be mathematically described as follows: if $\boldsymbol{v}_A \in \mathcal{VO}_B^A$ and $\boldsymbol{v}_B \in \mathcal{VO}_A^B$ i.e. vehicle A and B is headed for a collision and thus both vehicles adapt new velocities ${\boldsymbol{v}_A}^{new}$ and ${\boldsymbol{v}_B}^{new}$ outside of their respective $\mathcal{VO}'s$. In this new situations, the old velocities $\boldsymbol{v}_A$ and $\boldsymbol{v}_B$ is outside of the $\mathcal{VO}'s$. Then, if these velocities is considered to be the optimal velocities i.e. they will lead the vehicles directly toward there target, the vehicles will adapt the old velocities again, thus the vehicles is again headed for a collision. This problem is in fact, the motivation for the development of RVO.

As the VO and CC are very similar, this type of oscillations is also an issue when using CC based algorithms such as in this thesis. To avoid the case where the vehicle enters and leaves CA mode at a frequent rate, a hysteresis on the CA law is presented in Section 4.4. Simulations of the different COLREGS situations presented in Section 5.2 show how the vehicles solves the conflict in a safe and oscillation free manner.

## 6.5 Deadlocks

A deadlock is defined as two or more vehicles who are stuck in a configuration where none of the involved vehicles is able to reach their targets. When considering homogeneous vehicles, simulation have shown this to be a major issue for the CC based algorithm investigated in this thesis. In the Monte Carlo simulations (5.7) most of the cases where a simulation is marked as DNF is due to deadlocks. More specifically, if two homogeneous vehicles choose to pass each other on different sides i.e. they choose different velocity candidates (3.20), they will end up in a parallel configuration. Because the vehicles have the same dynamics, forward speed and turning rate, the relative positions between them will not change over time nor will their $CC$'s. Thus, they are stuck in this parallel configuration where neither reach their target. This is illustrated in Figure 5.15. For heterogeneous vehicles, such a configuration will resolve itself over time do to difference in forward speed and turning rate.

Deadlocks between homogeneous vehicles can be resolved by defining some traffic rule e.g. the vehicle to the left could give away to the vehicle to the right. Thus the symmetry that keeps the vehicles on parallel trajectories is broken.

## 6.6 Reciprocal dances

While distinct from the oscillations caused by the vehicle entering and leaving CA mode as described in Section 6.4, reciprocal dances may be equally difficult for the vehicles to resolve and may even cause collisions. Reciprocal dances occur in cluttered environments where the a conflict includes more than two vehicles. An illustration of reciprocal dances is given in Figure 5.16. Let $CC_A = CC_C^A + CC_B^A$, i.e. vehicle B and C are within the switching distance, $d_{\text{switch}}$, of vehicle A. Consider the case where $\boldsymbol{v}_A \in CC_A$ and the new velocity given by the CA law, ${\boldsymbol{v}_A}^{new} \notin CC_A$, is far from $\boldsymbol{v}_A$. When the $CC_A$ consist of multiple cones and the vehicles are close to one another, the combined cone can become relatively large. Due to the nonholonomic nature of the vehicles and the bound on the turning rate, $r_{max}$, some time will pass before $\boldsymbol{v}_A = {\boldsymbol{v}_A}^{new}$. During this time, the $CC$'s of vehicle B and C will change due to the change in $\boldsymbol{v}_A$, i.e. $CC_A^B$ and $CC_A^C$ are compensated by $\boldsymbol{v}_A$ as given in Equation (4.1). The turning of vehicle B and C will cause new changes to $CC_A$. Hence, the possibility of $\boldsymbol{v}_A$ not reaching the new velocity from the CA law arises. This behavior of the $CC$'s can cause the vehicles to turn in circles without resolving the conflict, known as reciprocal dances. The reciprocal dance of one vehicle can be contagious to surrounding vehicle's as the rapid changes in the velocity vector of the vehicle cause rapid changes to the surrounding vehicle's $CC$'s.

Most reciprocal dances resolve them self after a few occurrences. However, collisions can occur as the definition of the switching distance given by Equation (3.23) assumes that the vehicle will try to leave the area of conflict. In situations where the vehicle is surrounded by multiple vehicles, this may not be the case. Also, reciprocal dances may result in deadlocks between vehicles. The occurrence of reciprocal dances is greatly correlated to the density of vehicles in an area, which is further discussed in Section 6.7.

## 6.7 Density

Most unsuccessful simulations from the Monte Carlo simulation presented in Section 5.7 is due to reciprocal dances. As reciprocal dances are directly correlated to the density of vehicles, so are the Monte Carlo results in Section 5.7. This is illustrated in Table 5.4 where both Monte Carlo simulations is executed with the exact same vehicle parameters. The table show how the success rate is directly correlated with the density of vehicles. Hence, it is important to note this relationship when considering the results.

# Chapter 7

# Conclusion and future work

In this thesis a reactive Collision Avoidance (CA) algorithm [2] that is especially designed for nonholonomic vehicles with constant speeds is extended to the multi vehicle case, i.e. to a reciprocal CA algorithm. The main focus have been on designing an algorithm which ensures predictable vehicle behavior for vessels at sea e.g. cargo ships and other large vessels. The proposed CA algorithm is designed such that the vehicles respects the International Regulations for Preventing Collisions at Sea (Collision Regulations (COLREGS)). The behavior of vehicles in the different COLREGS situations is verified in simulations.

As the COLREGS only define traffic rules for conflicts involving two vehicles, a more generalized CA law is introduced to handle the N-vehicle case. The performance of the algorithm is measured by a series of Monte Carlo simulations as well as simulations showing how the algorithm preserves the liveness and safety of all vehicles. Although the performance of the algorithm is concluded to be good, the simulations show how reciprocal dances can occur in densely packed areas. In such cases the liveness and safety of the vehicle can not be guaranteed. An analysis of such cases is given with suggestions for improvements.

The algorithm also features a reduced Collision Cone (CC). This is motivated by the reactive nature of other vehicles and let the vehicles share the responsibility of avoiding a collision among them self in a fair manner. By sharing the responsibility of avoiding a collision, the control action needed by each vehicle is reduced and the CA maneuver is more optimal in a minimal-time and minimal-distance sense.

In summary, the presented algorithm ensures predictable vehicle behavior with accordance to the well established COLREGS as well as safety and liveness of the vehicles. The algorithm have been tested and the performance is been verified through simulations. Limitations of the algorithm have been identified and a thorough review of the results is given along suggestions for improvements.

## Suggestions for future work

As a final remark, some suggestions for improvement of the algorithm is presented to the interested reader:

- Create a more sophisticated weighting function that shares the responsibility of avoiding collisions between the vehicles. The weighting may be based on the configuration of the vehicles, e.g. in an overtaking situation it would be reasonable that the vehicle coming from behind have a larger responsibility than the vehicle that is being overtaken.
- To solve the case of reciprocal dances in densely packed areas one option would be to define a reserved region for each vehicle. Then collision cones can be created based on reserved region rather than the vehicle itself. Thus if a vehicle is trapped and the reserved region is stopped, the turning of the vehicle inside the region will not affect the surrounding vehicles. However, the creation of reserved regions includes the turning rate of other vehicles which is not trivial to obtain from sensor data.
- To ensure liveness of all vehicles and solve the issue of deadlocks, an easy solution would be to implement a set of traffic rules in the case of deadlocks, e.g. the vehicle that has the other vehicle on its starboard side should give away and thus resolving the deadlock.

# References

[1] C. C. Baker and A. K. Seah, “Maritime accidents and human performance: the statistical trail,” in *Presented at MARTECH 2004, Singapore*, Sep 2004, p. 225–239.

[2] M. S. Wiig, K. Y. Pettersen, and A. V. Savkin, “A reactive collision avoidance algorithm for nonholonomic vehicles,” in *2017 IEEE Conference on Control Technology and Applications (CCTA)*, Aug 2017, pp. 1776–1783.

[3] A. Chakravarthy and D. Ghose, “Obstacle avoidance in a dynamic environment: a collision cone approach,” *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 28, no. 5, pp. 562–574, Sep 1998.

[4] A. K. Singh and K. M. Krishna, “Reactive collision avoidance for multiple robots by non linear time scaling,” in *52nd IEEE Conference on Decision and Control*, Dec 2013, pp. 952–958.

[5] B. Gopalakrishnan, A. K. Singh, and K. M. Krishna, “Time scaled collision cone based trajectory optimization approach for reactive planning in dynamic environments,” in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sept 2014, pp. 4169–4176.

[6] P. Fiorini and Z. Shiller, “Motion planning in dynamic environments using velocity obstacles,” *The International Journal of Robotics Research*, vol. 17, no. 7, pp. 760–772, 1998.

[7] ——, “Motion planning in dynamic environments using the relative velocity paradigm,” in *1993 Proceedings IEEE International Conference on Robotics and Automation*, May 1993, pp. 560–565 vol.1.

[8] J. O’Rourke and N. Badler, “Decomposition of three-dimensional objects into spheres,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-1, no. 3, pp. 295–305, July 1979.

[9] D. Wilkie, J. van den Berg, and D. Manocha, “Generalized velocity obstacles,” in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct 2009, pp. 5573–5578.

[10] J. van den Berg, J. Snape, S. J. Guy, and D. Manocha, “Reciprocal collision avoidance with acceleration-velocity obstacles,” in *2011 IEEE International Conference on Robotics and Automation*, May 2011, pp. 3475–3482.

[11] O. Khatib, “Real-time obstacle avoidance for manipulators and mobile robots,” in *Proceedings. 1985 IEEE International Conference on Robotics and Automation*, vol. 2, Mar 1985, pp. 500–505.

[12] Y. Koren and J. Borenstein, “Potential field methods and their inherent limitations for mobile robot navigation,” in *Proceedings. 1991 IEEE International Conference on Robotics and Automation*, Apr 1991, pp. 1398–1404 vol.2.

[13] A. A. Masoud, “Kinodynamic motion planning,” *IEEE Robotics Automation Magazine*, vol. 17, no. 1, pp. 85–99, March 2010.

[14] ——, “A harmonic potential approach for simultaneous planning and control of a generic uav platform,” *Journal of Intelligent & Robotic Systems*, vol. 65, no. 1, pp. 153–173, Jan 2012. [Online]. Available: https://doi.org/10.1007/s10846-011-9570-8

[15] J. Ren, K. A. McIsaac, and R. V. Patel, “Modified newton’s method applied to potential field-based navigation for mobile robots,” *IEEE Transactions on Robotics*, vol. 22, no. 2, pp. 384–391, April 2006.

[16] ——, “Modified newton’s method applied to potential field-based navigation for nonholonomic robots in dynamic environments,” *Robotica*, vol. 26, no. 1, p. 117–127, 2008.

[17] L. Valbuena and H. G. Tanner, “Hybrid potential field based control of differential drive mobile robots,” *Journal of Intelligent & Robotic Systems*, vol. 68, no. 3, pp. 307–322, Dec 2012. [Online]. Available: https://doi.org/10.1007/s10846-012-9685-6

[18] A. S. M. M. Hoy and A. V. Savkin, “Algorithms for collision-free navigation of mobile robots in complex cluttered environments: a survey,” *Robotica*, vol. 33, no. 03, pp. 463–497, 2014.

[19] W. B. D. Fox and S. Thrun, “The dynamic window approach to collision avoidance,” *IEEE Robotics & Automation Magazine*, vol. 4, no. 1, pp. 23–33, 1997.

[20] B. O. H. Eriksen, M. Breivik, K. Y. Pettersen, and M. S. Wiig, “A modified dynamic window algorithm for horizontal collision avoidance for auvs,” in *2016 IEEE Conference on Control Applications (CCA)*, Sept 2016, pp. 499–506.

[21] O. Brock and O. Khatib, “High-speed navigation using the global dynamic window approach,” in *Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No.99CH36288C)*, vol. 1, 1999, pp. 341–346 vol.1.

[22] M. Seder, K. Macek, and I. Petrovic, “An integrated approach to real-time mobile robot control in partially known indoor environments,” in *31st Annual Conference of IEEE Industrial Electronics Society, 2005. IECON 2005.*, Nov 2005, pp. 6 pp.–.

[23] P. Ogren and N. E. Leonard, “A convergent dynamic window approach to obstacle avoidance,” *IEEE Transactions on Robotics*, vol. 21, no. 2, pp. 188–195, April 2005.

[24] R. Simmons, “The curvature-velocity method for local obstacle avoidance,” in *Proceedings of IEEE International Conference on Robotics and Automation*, vol. 4, Apr 1996, pp. 3375–3382 vol.4.

[25] N. Y. Ko and R. G. Simmons, “The lane-curvature method for local obstacle avoidance,” in *Proceedings. 1998 IEEE/RSJ International Conference on Intelligent Robots*

*and Systems. Innovations in Theory, Practice and Applications (Cat. No.98CH36190)*, vol. 3, Oct 1998, pp. 1615–1621 vol.3.

[26] J. Fernández, R. Sanz, J. Benayas, and A. Diéguez, “Improving collision avoidance for mobile robots in partially known environments: the beam curvature method,” *Robotics and Autonomous Systems*, vol. 46, no. 4, pp. 205 – 219, 2004. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0921889004000120

[27] Y. Abe and M. Yoshiki, “Collision avoidance method for multiple autonomous mobile agents by implicit cooperation,” in *Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems. Expanding the Societal Role of Robotics in the the Next Millennium (Cat. No.01CH37180)*, vol. 3, 2001, pp. 1207–1212 vol.3.

[28] J. van den Berg, S. J. Guy, M. Lin, and D. Manocha, “Reciprocal n-body collision avoidance,” in *Robotics Research*, C. Pradalier, R. Siegwart, and G. Hirzinger, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 3–19.

[29] J. Alonso-Mora, A. Breitenmoser, P. Beardsley, and R. Siegwart, “Reciprocal collision avoidance for multiple car-like robots,” in *2012 IEEE International Conference on Robotics and Automation*, May 2012, pp. 360–366.

[30] L. Pallottino, V. G. Scordio, A. Bicchi, and E. Frazzoli, “Decentralized cooperative policy for conflict resolution in multivehicle systems,” *IEEE Transactions on Robotics*, vol. 23, no. 6, pp. 1170–1183, Dec 2007.

[31] E. J. Rodríguez-Seda, “Decentralized trajectory tracking with collision avoidance control for teams of unmanned vehicles with constant speed,” in *2014 American Control Conference*, June 2014, pp. 1216–1223.

[32] , “Convention on the international regulations for preventing collisions at sea.”

[33] Y. Kuwata, M. T. Wolf, D. Zarzhitsky, and T. L. Huntsberger, “Safe maritime autonomous navigation with colregs, using velocity obstacles,” *IEEE Journal of Oceanic Engineering*, vol. 39, no. 1, pp. 110–119, Jan 2014.

[34] T. I. Fossen, *Handbook of Marine Craft Hydrodynamics and Motion Control.* John Wiley & Sons, Ltd, 2011.

[35] J. Snape, J. v. d. Berg, S. J. Guy, and D. Manocha, "The hybrid reciprocal velocity obstacle," *IEEE Transactions on Robotics*, vol. 27, no. 4, pp. 696–706, Aug 2011.

[36] J. van den Berg, M. Lin, and D. Manocha, "Reciprocal velocity obstacles for real-time multi-agent navigation," in *2008 IEEE International Conference on Robotics and Automation*, May 2008, pp. 1928–1935.

[37] M. R. Benjamin, J. J. Leonard, J. A. Curcio, and P. M. Newman, "A method for protocol-based collision avoidance between autonomous marine surface craft," *Journal of Field Robotics*, vol. 23, no. 5, pp. 333–346. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/rob.20121

[38] MathWorks. Monte carlo simulation. [Online]. Available: https://se.mathworks.com/discovery/monte-carlo-simulation.html

[39] B. Kluge and E. Prassler, "Reflective navigation: individual behaviors and group behaviors," in *Robotics and Automation, 2004. Proceedings. ICRA '04. 2004 IEEE International Conference on*, vol. 4, April 2004, pp. 4172–4177 Vol.4.

[40] A. V. Savkin and C. Wang, "Seeking a path through the crowd: Robot navigation in unknown dynamic environments with moving obstacles based on an integrated environment representation," *Robotics and Autonomous Systems*, vol. 62, pp. 1568 – 1580, 2014.

# Appendix A: Algorithm comparison

To give some additional context to the performance of the proposed algorithm, a comparison with another similar algorithm is presented. The algorithm in question is developed by Andreas L. Aarvold as a part of his master thesis. The algorithm seeks to solve the same problem; reciprocal Collision Avoidance (CA) for nonholonomic vehicles with constant forward speed. This appendix offers a review of the alternative algorithm, followed by the results of Monte Carlo simulations of both algorithms and a case that illustrates similarities between the algorithms. Lastly an discussion of the results is given. The algorithm review is written by Andreas L. Aarvold.

## Algorithm review

The algorithm proposed by Andreas L. Aarvold is a multi-agent reactive and decentralized algorithm for nonholonomic vehicles without communication. It is based on the algorithm presented in [40] and rely on an integrated representation of the information about the environment. This results in a sensor disk in front of the vehicle as shown in Figure 1. A method is applied to compensate for obstacle velocity [2], illustrated by the shifted regions in the figure. When no obstacles are detected, the agents are guided towards the target by a pure pursuit guidance law. Otherwise, the middle value of a obstacle-free interval closest to the current heading is chosen.

Related to Figure 1, the desired heading is $C_i$ in the bottom of the figure. In addition, a braking rule mimicking a typical yield-pass maneuver is created to overcome typical multi-agent challenges, such as deadlock and oscillation. Note that the algorithm assume bounded forward speed $u(t) \in [u_{min}, u_{max}]$. However, constant agent speed is achieved by using $u_{min} = u_{max}$.

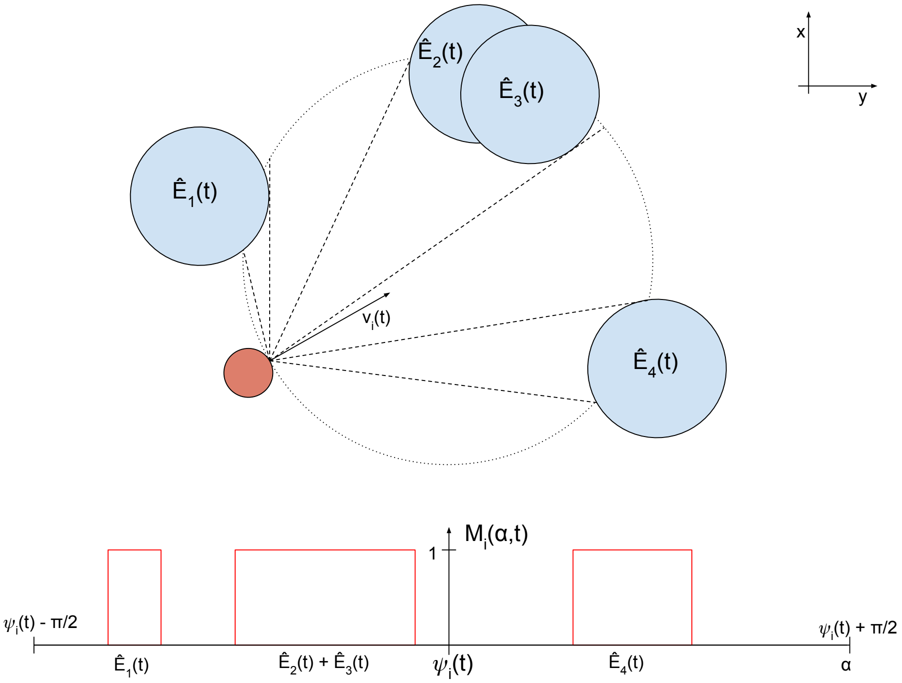

Figure 1: Example scenario illustrating multiple and overlapping obstacles (blue) with one agent (red), the dotted circle is the sensor disk, where $d_sen$ is the sensor range. The scenario results in a binary sensor function $\hat{M}(\hat{\alpha}, t)$ with four obstacle-free intervals $[A_i^-,\ A_i^+]$ for $i = 1, 2, 3, 4$.

# Simulations

The simulations presented in Table 1 is the results of a Monte Carlo simulation consisting of 1000 simulations of 10 nonholonomic vehicles modeled by the vehicle model from Equation (3.1). The vehicles are homogeneous i.e. all vehicles have equal forward speed, $u = 1$ m/s and maximum turning rate, $r_{max} = 1$ m/s$^2$. Both master thesis have developed a script for initialization of Monte Carlo simulations. To ensure that the simulations is carried out in a fair manner, a new script that simulates both algorithms with the same parameters is created by Andreas L. Aarvold.

| | |
|---|---|
| Number of simulations | 1000 |
| Number of vehicles | 10 |
| Success | 89.8 % |
| Did Not Finish (DNF) | 4.4 % |
| Crash | 5.8 % |
| Average completion time | 56.9 sec |

| | |
|---|---|
| Number of simulations | 1000 |
| Number of vehicles | 10 |
| Success | 99.9 % |
| DNF | 0.0 % |
| Crash | 0.1 % |
| Average completion time | 51.7 sec |

Table 1: Monte Carlo: Performance comparison. The table to the left presents the result from the algorithm presented in this thesis. The table to the right presents result from the alternative algorithm

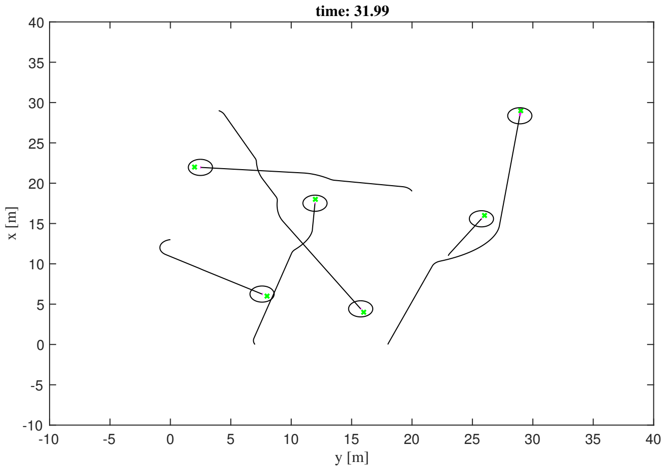

Figure 2: Algorithm comparison: 10 homogeneous vehicles with constant speed implemented with the reciprocal CA algorithm proposed in this thesis

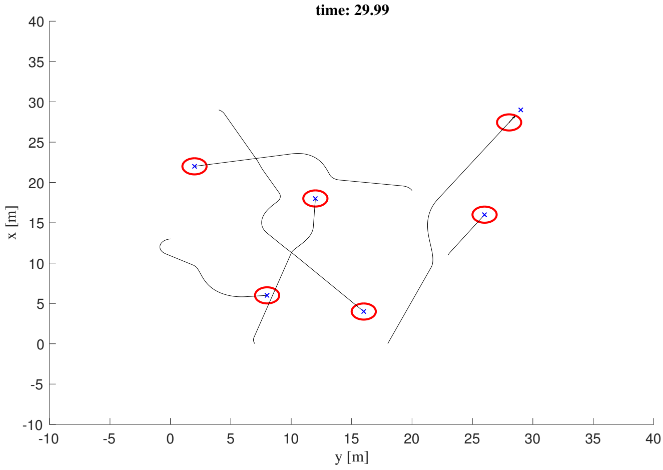

Figure 3: Algorithm comparison: 10 homogeneous vehicles with constant speed implemented with the reciprocal CA algorithm proposed by Andreas L. Aarvold

# Discussion

Table 1 shows that the alternative algorithm outperforms the algorithm proposed in this thesis in a cluttered environment. As the second algorithm is based on an reactive CA algorithm designed to handle cluttered environments [40], these results are as expected. There is a slight difference in completion time between the algorithms in the favor of the second algorithm. This difference is due to the algorithm in this thesis is more prune to reciprocal dances. Some reciprocal dances resolves them self and the vehicles is able to reach their target, hence the simulation is considered to be a success. However, reciprocal dances result in suboptimal trajectories which affects the average completion time.

Figure 2 and 3 shows a simulation where the liveness and safety of all vehicles is preserved by both algorithms. In general the algorithm presented in this thesis applies less control input to avoid a collisions and is able to pass other vehicles at a smaller distance. This results in more optimal trajectories both in a minimal distance and time sense. However, in some cases the second algorithm makes more optimal choices on how to pass other vehicles. E.g. the two right-most vehicles in Figure 2 and 3.

# Appendix B: MATLAB simulator

The proposed algorithm in this Thesis have been implemented in a MATLAB simulator, all results presented in Chapter 5 is obtained from this simulator. The simulator was originally developed by the author of [2] and supervisor of this Thesis, Martin Syre-Wiig (FFI). Initially it supported simulation of the Collision Avoidance (CA) algorithm presented in Chapter 3 [2]. The simulator was given as a starting point for this Thesis. Below follows a review of the initial simulator and the changes done to the simulator throughout this thesis. Lastly the main loop of the simulator is presented as pseudo-code in (1) to illustrate the flow of the CA algorithm.

## Features of the original simulator

The handout of the simulator included scripts for initialization of a single simulation, and a simulation script that simulated a single vehicle and multiple obstacles in an environment created by the initialization script. The sim script makes use of a several classes such as an vehicle and obstacle class as well as a collision cone class. The vehicle class included the vehicle model, vehicle states, a function for calculating new states and a heading controller. The collision cone class included functions for cone creations and detection if a heading is inside a cone. The simulator also included a plot script which plots the vehicle and obstacles in the xy-plane.

## Adaptations to the simulator

The simulation script is adapted to support simulation of multiple vehicles by iterating over a list of all vehicles. Each vehicle is simulated by considering the other vehicles as obstacles, either static, moving or reactive. Similar to the original script. Algorithm (1) illustrated the full simulation loop. An alternative sim script that includes the upper layer CA law presented in Section 4.2.4 is also added.

The vehicle class have been extended such that each vehicle keeps track of the distance to surrounding vehicles. Several plot options have been added for individual plotting of vehicles. The initial heading controller included in the class was designed with a high gain to simulate a bang-bang controller. Due to some issues regarding the combination of a high controller gain and the time step of the simulations, the controller have been re-tuned.

The collision cone class is implemented with support for cone ID's i.e. each cone is directly linked with the vehicle that induced that cone. An numerical error in the function that computes if a heading is inside a cone is identified and fixed. Also, a function that determines the Collision Regulations (COLREGS) situation, calculate the COLREGS parameters defined in Section 4.2.2 and determines the turning direction is added.

Several initialization scripts have been added. A script that creates random positions and targets for a user specified number of vehicles and simulation instances is created to initiate Monte Carlo simulations. The script includes logging of results and saves all necessary variables needed for re-simulations. Another scrips lets the user re-simulate failed Monte Carlo simulations. A script that lets the user create preset configurations of vehicles and simulate them, e.g. the COLREGS situations is also added. Lastly a script that lets the user simulate N-vehicles exchanging antipodal positions on a circle is added.

The plot scripts have been given extended features that gives the the user more options regarding which information that is included in the plots. Each vehicle and obstacle have its unique plot settings. An new plot script for plotting of distance between vehicles is added. Also, the initialization scripts includes setting for automatic plot saving at a user specified interval or time instant.

## Psudo code of simulation

**Algorithm 1** Reciprocal CA algorithm

```
for all $A_A \in \mathcal{A}$ do
    Sense $\boldsymbol{p}_A$ and $\boldsymbol{v}_A$
    for all $A_B \in \mathcal{A}$ such that $B \neq A$ do
        Sense $\boldsymbol{p}_B$ and $\boldsymbol{v}_B$
        distance ← $|\boldsymbol{p}_A - \boldsymbol{p}_B|$
        if distance ≤ $d_{\text{switch}}$ then
            Construct C≀↕↕⟩∫⟩≀\C≀\⌉$(CC)_B^A$
        end if
        if $CC_B^A \neq$ NULL and $\boldsymbol{v}_A \in CC_B^A$ then
            if inCollision$_{A,B}$ = False then
                inCollision$_{A,B}$ ← True
                Calulate turningDirection
            end if
        else
            if inCollision$_{A,B}$ = True then
                inCollision$_{A,B}$ ← False
            end if
        end if
        if inCollision$_{A,B}$ then
            Compute new velocity ${\boldsymbol{v}_A}^{new}$ from $CC_B^A$(turningDirection)
        else
            Compute new velocity ${\boldsymbol{v}_A}^{new}$ from guidance law
        end if
        Apply control input ${\boldsymbol{v}_A}^{new}$
    end for
end for
```