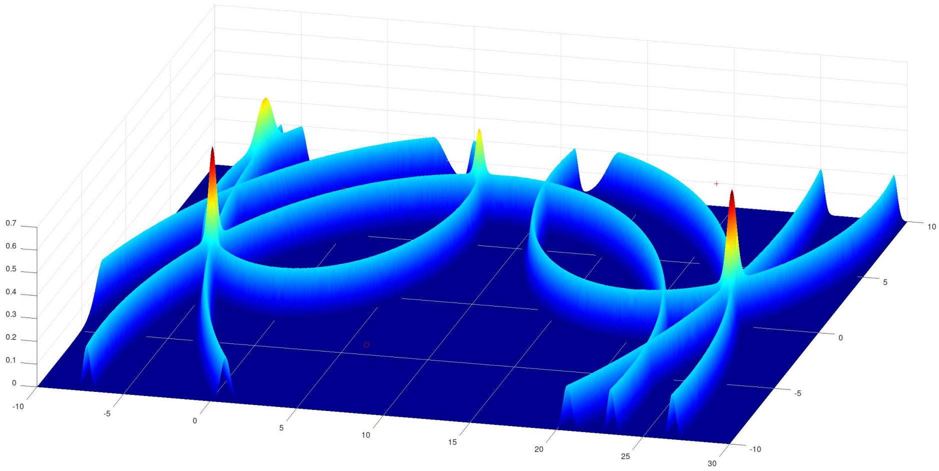
CHALMERS
UNIVERSITY OF TECHNOLOGY



# Statistical Safeguards: Redefining Collision Avoidance with Probability Theory

## Employing Statistical Decision-Making to Enhance Safety in Mixed Traffic Environments

Master's thesis in Systems, Control and Mechatronics


SEBASTIAN HJERTÈN BRINK

DEPARTMENT OF MECHANICS AND MARITIME SCIENCES

CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2024
www.chalmers.se

MASTER'S THESIS 2024

# Statistical Safeguards: Redefining Collision Avoidance with Probability Theory

Employing Statistical Decision-Making to Enhance Safety in Mixed Traffic Environments

SEBASTIAN HJERTÈN BRINK

Department of mechanics and maritime sciences
*Vehicle Engineering and Autonomous Systems*
Applied Artificial Intelligence Group
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2024

Statistical Safeguards: Redefining Collision Avoidance with Probability Theory
Employing Statistical Decision-Making to Enhance Safety in Mixed Traffic Environments
Sebastian Hjertén Brink



Supervisor: Martin Hedvall Fogelquist, Combitech AB
Examiner: Peter Forsberg, Department of Mechanics and Maritime Sciences

Master's Thesis 2024
Department of Mechanics and Maritime Sciences
Vehicle Engineering and Autonomous Systems
Applied Artificial Intelligence Group
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: Distinct spherical probability distributions on a 2D map, where the z-dimension represents the probability. The joint distributions are represented by the area where the individual distributions overlap.

Typeset in LaTeX
Printed by Chalmers Reproservice
Gothenburg, Sweden 2024

Statistical Safeguards: Redefining Collision Avoidance with Probability Theory
Employing Statistical Decision-Making to Enhance Safety in Mixed Traffic Environments
SEBASTIAN HJERTÈN BRINK
Department of Mechanics and Maritime Sciences
Chalmers University of Technology

# Abstract

This thesis introduces a probabilistic collision avoidance system that employs statistical decision-making in order to enhance the safety of mixed traffic environments. Central to this approach is the representation of vehicle positions as normal probability distributions, which are convolved with real-time sensor data to assess risks more accurately and reduce the unnecessary emergency stops.
The research develops and implements a dynamic collision probability threshold, that is derived from safety integrity levels (SIL), which is imperative for complying with the rigorous safety standards and regulations. Simulations and analytical methods were used to validate the effectiveness of the proposed algorithm and demonstrating its potential in decision-making in emergency situations.
Thus a a scalable solution for collision avoidance is presented in the form of an algorithm that can be integrated into existing safety systems, in order to enhance the operational efficiency for mixed traffic environments.

Keywords: Collision avoidance systems, Probabilistic risk assessment, Safety integrity levels, Statistical decision making.

# Acknowledgements

I would like to preface this thesis by expressing my gratitude to everyone who made it possible.
First and foremost I would like to thank Peter Forsberg for helping me shape the research and giving valuable insight, guidance and advise through this thesis.
Martin Hedvall Fogelquist and Christian Grante, for both tutelage and continuous feedback but also an unwavering vision on how the end product should behave and fit into existing systems.
I would like to thank my family, my loving wife Frida and our two wonderful daughters Alice & Melanie. Without your support and encouragement i would still be stuck doing manual labour in some factory.
Last but not least, i would like to thank Combitech AB and Boliden AB for giving me the opportunity to work on this fascinating and enlightening project.

Sebastian Hjertén Brink, Gothenburg, September 23, 2024



**Thesis advisor:** Martin Hedvall Fogelquist, Combitech AB
**Thesis examiner:** Peter Forsberg, Department of Mechanics and Maritime Sciences

# List of Acronyms

Below is the list of acronyms that have been used throughout this thesis listed in alphabetical order:

| | |
|---|---|
| FFT | Fast Fourier Transform |
| IMU | Inertial Measurement Unit |
| MOZ | Mixed Operating Zone |
| PDF | Probability density function |
| PFH | Probability of Failure per Hour |
| SIL | Safety Integrity Level |
| STD | Standard Deviation |
| UWB | Ultra-Wide band |

# Nomenclature

Below is the nomenclature of indices, sets, parameters, and variables that have been used throughout this thesis.

| | |
|---|---|
| $f_e$ | frequency of exposure |
| $i$,$j$ | Array indices |
| $t$ | Index for time step |
| $\varnothing$ | Empty set |
| $\mu$ | Mean value |
| $p_c$ | Probability constant for unforeseen risk exposure |
| $\sigma$ | Standard deviation |
| $V_{th}$ | Probability threshold |

# Contents

# List of Figures

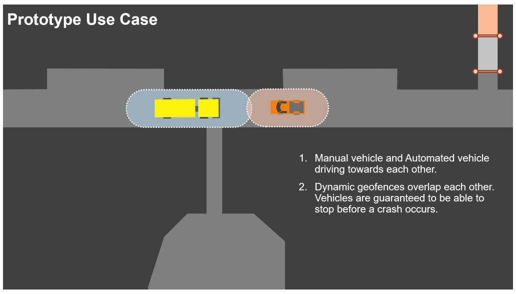
# List of Tables

# 1
# Introduction

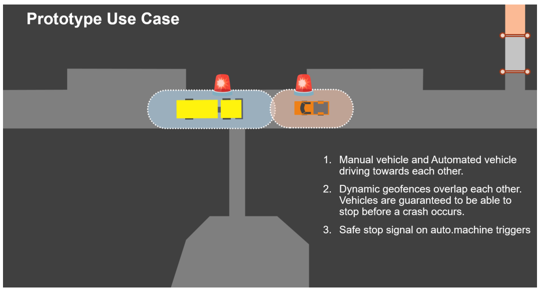
To formulate an analytical collision avoidance algorithm using statistical decision making that is computationally light-weight is a challenging task. By formulating the positions as normal probability distributions, applying them to a euclidean space and then enhancing them with additional distributions representing errors, unknown variables and predefined collision avoidance strategies, the goal is to end up with a formally provable collision avoidance algorithm that can be deployed in a microcontroller for real-world applications.

## 1.1 Background

Boliden AB is a company working with the first steps of the metal processing chain, which includes prospecting, mining, enrichment, smelting refining and recycling [19]. Combitech AB is a consultancy company and subsidiary to the Swedish defence and security group Saab. They are focused on providing smart, resilient and sustainable solutions for a better society.
Boliden AB and Combitech AB have a joint venture to design and implement a safety system that enables mixed traffic (manually driven vehicles and autonomous vehicles operating together) which live up to rigorous regulations and safety standards. One of the technologies leveraged in order to achieve this is to implement an external emergency stop system, which would monitor all autonomous vehicles within the site and if an accident is likely to occur trigger a site wide emergency brake for all vehicles. The emergency braking system is a very costly action since it affects all vehicles in the site, halting all transports for a time, thus it is of great import to minimize the number of unnecessary stops from the system. The emergency braking supervisor system is comprised of Ultra-wideband (UWB) anchors placed throughout the site, that will pick up signals from tags which every vehicle in the site is equipped with. The anchors are placed such that there will always be at least three within range at any time to enable triangulation. After a vehicles position is estimated an area of risk is created around the vehicle which represents the area in which a collision is inevitable. This area is based on the vehicle dynamics and if two vehicles risk area overlap the emergency brakes will trigger. See figures 1.1.
The possibility of using probabilities to express the risk areas of each vehicle separately instead has the potential benefit of reducing the area of risk for the vehicles, thus reducing the amount of false positives of the emergency braking system. It also has the nifty attribute that if the sum of the risk is expressed in probabilities it can be directly correlated to the levels set by the regulations and standards, thus prov-




**(a)** Two vehicles moving towards each other




**(b)** Emergency system triggered

**Figure 1.1:** Emergency system concept

ing mathematically that they are fulfilled. The standard IEC 61508-1:2010 classify different Safety Integrity Levels (SIL) with dictated probability of failure per hour [20]. By using these levels and the defined probabilities per hour and combining it with an expected frequency of vehicles meeting each other, a probability value can be defined such that if two vehicles probabilistic collision distribution surpass it the emergency brake system will trigger.

## 1.2 Related work

The field of research relating to a positioning and collision avoidance for autonomous vehicles is vast and ever expanding. There are a large number of components that needs to work in order to gain confidence in an autonomous vehicles safety credibility. This section describes previous research topics, questions and findings done by other authors which is of direct or indirect value in the research being done in this thesis.
Markkula, G., Benderius, O., Wolff, K., and Wahde, M [13] reviewed different near-collision driver behaviour models with the purpose of improving computer simulations of accident situations in an effort to improve traffic safety. They accomplished this by grouping articles by main emphasis behaviour, which where: avoidance by braking, steering, a combination of the two, effects of driver states and characteristics on avoidance and simulation platforms. They go on to conclude that there is a lot of proposed driver behaviour models for near-collisions proposed but validation using human driving data is limited. The research area is fragmented but simulation-based comparisons of the model hints that the different models may have more similarity than the equations indicate.
A. Goudar, T. D. Barfoot and A. P. Schoellig, "Continuous-Time Range-Only Pose Estimation" [17] presents a way to estimate the pose of a vehicle using range-only positional technology, e.g no IMU or odometry data is fused with the external positional sensors. They accomplish this by combining the use of several anchors for triangulation (3+ external and 2/3+ on the robot for 2D and 3D respectively), along with using the *motion priors* as constraints between measurements to constrain the full state. Then a *maximum a posteriori* (MAP) estimation is made to infer the robot state. Both a motion model and range measurement model is needed to calculate the motion prior used in the estimation.

Francesco Riz et al [18] claim that when solving the problem of simultaneous trilateration using ranging sensors, three measurements are not sufficient to localise a moving target in a given environment even in ideal conditions. They argue that when using several sensors the measurements are not made on the same time-step and thus the object has moved between each measurement. Thus there are multiple trajectories that are compliant with the motion of the target and the measurements.
James T. Klosowski et al. [5] presents an efficient collision detection algorithm for moving objects in complex environments based on bounding volume hierarchies. The choice of bounding volume is a "discrete orientation polytope" or "k-dop", it is "a convex polytope whose facets are determined by halfspaces whose outward normals come from a fixed set of k orientations". Increasing k gives a more fitted polygon, but they also propose algorithms for maintaining an effective bounding volume tree of k-dops for moving objects, their rotation and for performing fast collision detection.
Jonas Janson has written or co-written several publications on statistical and probabilistic approaches in the collision avoidance space. A few of them include:

- The chapter "Dealing with Uncertainty in Automotive Collision Avoidance" discuss a driver assist system for autonomous braking when close to colliding. Considerations are made for limited computational capacity of microprocessors. Due to the uncertainty of driver actions and sensor measurements the decisions in such as system is inherently uncertain and how to deal with estimated parameters in the decision making process is also discussed [9].
- The thesis "Model-based statistical tracking and decision making for collision avoidance application" look into a collision mitigation system where the decision is based on a Bayesian approach to estimation and use an extended Kalman filter and particle filter to solve the tracking problem [8].
- "A probabilistic approach to collision risk estimation for passenger vehicles" presents a method for risk estimation on which to base decision making. The criteria for the decision making is formed in terms of probability of collision [6].
- A general method for calculating the risk of collision and activating a brake maneuver if the collision probability reaches 1 is presented in [7].
- Theory for tracking and decision making in collision avoidance systems is presented in the thesis "Collision Avoidance Theory with Application to Automotive Collision Mitigation". This include an algorithm to calculate the probability of collision by simulating the system for future time instants and calculate the probability for collision at each time step [11].

The topic of UWB is especially active the last couple of years due to its low cost and energy consumption paired with its relatively accurate measurements, making it ideal for in door positioning systems.
Alarifi et al [14] analyzed the strengths, weaknesses, opportunities and threats of different UWB positioning technologies. They compare the different positioning algorithms used in UWB systems: Time of Arrival (TOA), Angle of Arrival (AOA), Received Signal Strength (RSS), Time Difference of Arrival (TDOA) and hybrid algorithms. The different algorithms have different strengths and weaknesses, and by utilizing hybrid solutions they can complement each other at the expense of complexity and cost.

However, to the authors knowledge no research has been done on the topic of using probability distribution functions to calculate a vehicles probabilistic risk zone. A probability function like this for all vehicles in an enclosed space could then be combined to calculate joint probability on the probability that two (or more) vehicles would occupy the same space, resulting in a collision.

## 1.3 Research questions

1. Can a probability distribution be defined such that it represents a vehicles potential position encompassing potential evasive maneuvers, measurement errors and other uncertainties?
2. Can the probability distribution be directly correlated to relevant standards and regulations?
3. How does the precision and performance of the model change if the vehicles current heading is known?
4. How does a probabilistic approach compare to a geometric for collision avoidance in terms of unnecessary stops?

## 1.4 Risks

Since we are dealing with probabilities there is always a judgement call on what threshold is good enough to continue without triggering the safety system. There is always the possibility that an accident happens. In order to mitigate such risks there should be several safety- and sanity checks in place to minimize the possibility of such errors. Some variables that can yield uncertain results are:

- Measurement errors
- Poorly chosen hyper-parameters of the distributions, such as amplitude (A) or standard deviation ($\sigma$)

Measurement errors would yield data that is wrong and thus the probabilities and estimates based on them would not accurately reflect reality. Poorly chosen hyper-parameter would instead result in poor probability calculations. The amplitude scales the height of the Gaussian distribution and if it is poorly picked that could result in sub-optimal maximal measurement values or likelihood estimate. The standard deviation (std) parameter instead controls the range around the peak of the Gaussian distribution. A small value would result in a sharp peak and narrow range, while a large one would yield a flatter peak and wider range. It could be represented as the uncertainty of our sensor measurements. Given very precise sensors, a small std is suitable to reflect that. Thus a poorly picked std would yield result in either false confidence (in the event that the sensors are poor but sigma small) or low probabilities due to low peak value (in the event of large sigma and good sensors).

## 1.5 Limitations

The main research is on probability functions and how they can be used on measurement data from different sources in order to estimate the position of vehicles and to model different variables as potential positions in the euclidean space. Other solutions for position or pose estimate is not studied. The scenario under study is a single-lane road with meeting-pockets at regular intervals to enable vehicles to pass each other. This presents a natural limitation on the scope of study, mainly that traffic should only be meeting head-on, and rear- or side collisions is assumed to be non existent. It is also assumed that the vehicles types and information about them is known, such as weight, brake force and dimensions. This is a natural extension of the purpose which states that the scope is for a closed and controlled site, so the objects and agents moving within it can be controlled.

# 2

# Theory

## 2.1 Getting positional data

By measuring the time of flight between a uniquely tagged transmitter and receiver a maximum distance between them can be calculated. For example the Double Sided Two-way-Ranging technique works as follows:

1. Tag transmits a message
2. Anchor(s) receives message
3. Anchor(s) transmits a message
4. Tag receives message
5. Tag transmits message containing round-trip times and reply times for each anchor respectively
6. Anchor(s) receives the message

With the round-trip and reply times the time-of-flight can be calculated. Because the algorithm uses both round-trip and reply time to calculate the time-of-flight both clock-time of the transmitters/receivers and synchronization requirements can be eliminated.

$$T = \frac{T_{round1} * T_{round2} - T_{reply1} * T_{reply2}}{T_{round1} + T_{round2} + T_{reply1} + T_{reply2}} \tag{2.1}$$

$$d = T * c \tag{2.2}$$

Where $c$ is the speed of light. Since a communication has been done it can be confidently assumed that the vehicle is within this distance. If the transmission has bounced before being picked up by the receiver the calculated distance would be longer than the actual distance, but the measurement can not be shorter than the actual distance and hence, the vehicle cant be further away. Given a stationary sensor with this ranging theory, the potential position of a vehicle would be described as a circle, with the origin at the sensor position and the radius equal to the distance calculated by the time of flight measurement. Given two (or more) sensors, the area in which the tag can be in would be the intersection of the resulting circles, see figure 2.1.

## 2.2 Statistical Concepts

This section describes the statistical concepts which is the foundation of the work done in this paper.

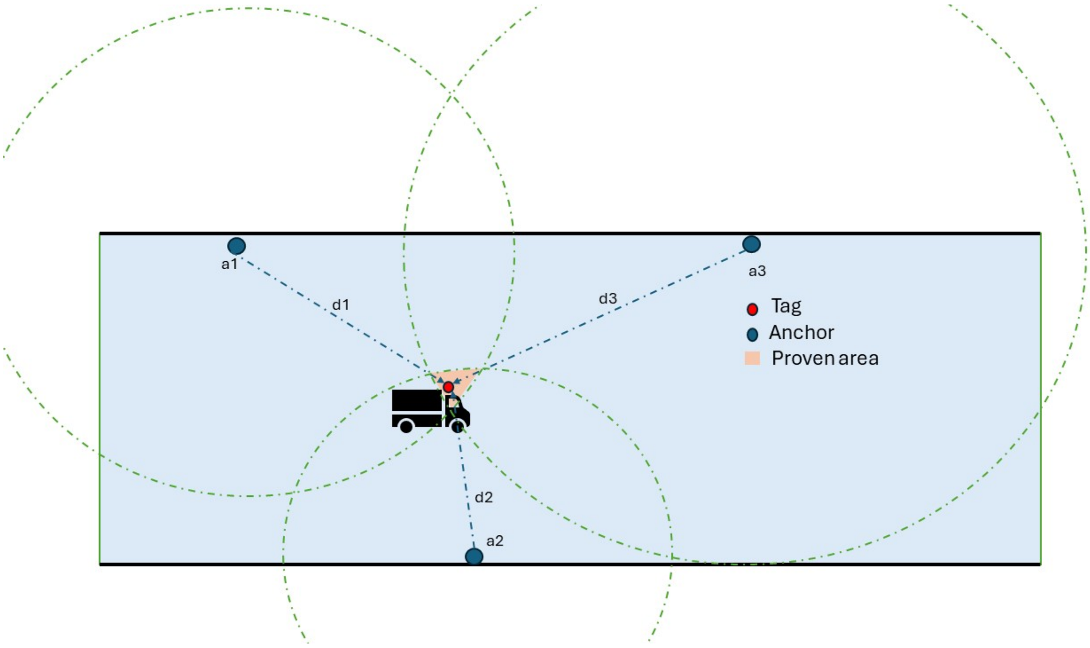



**Figure 2.1:** Measurement representation for anchor/tag communication.

### 2.2.1 Probability distributions

A probability distribution is a mathematical function that gives the probabilities of the different outcomes for an experiment. A random variable takes values from a sample space and probabilities describe which value or values are more likely taken. An event is a set of possible values (or outcomes) of a random variable within a certain probability [12] [15]. All probabilities should be non-negative and the sum of the distribution should be equal to one.

$$\begin{aligned}\sum_x p(x) &= 1 \\ p(x) &\geq 0\end{aligned}$$

A probability distribution can be both continuous or discrete. For example, a discrete distribution could be one where the outcomes describe the value of a dice throw, each outcome is a discrete value. Some examples of a continuous distribution is heights of people in a group, IQ scores or measurement errors.
A distribution can have one or several random variables associated with it, a distribution with only one random variable is called *univariate* while a distribution with several is called *multivariate*.

#### 2.2.1.1 Absolutely continuous probability distribution

The absolutely continuous probability distribution is defined on real numbers with an infinite number of possible values, such as an interval in the real line, where the probability of an event is described as an integral.

$$P(a \leq X \leq b) = \int_a^b f(x)dx \tag{2.3}$$

This is also the definition of the probability density function.

#### 2.2.1.2 Normal distributions

A normal or Gaussian distribution is a type of continuous probability distribution. The formula for the univariate distribution is shown in equation 2.4 and the multivariate case is defined in equation 2.5.

$$p(x) = \frac{1}{\sigma\sqrt{2\pi}} * e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{2.4}$$

$$p(\mathbf{x}) = \frac{1}{(2\pi)^{\frac{\pi}{2}}|\Sigma|^{\frac{1}{2}}} exp\left(-\frac{1}{2}(\mathbf{x}-\mu)^T\Sigma^{-1}(\mathbf{x}-\mu)\right) \tag{2.5}$$

where $\mu$ is the mean (and median and mode) and $\sigma$ is the standard deviation. The normal distribution is bell-shaped and symmetric about the mean, which is located in the center of the distribution. The normal distribution is an important distribution in statistics because many real-world scenarios have variables which are approximately normally distributed, such as height or test scores in a population.

### 2.2.2 Probability functions

A *probability mass function* gives the probability that a discrete random variable is equal to some value. $p(x) = P(X = x)$
A *probability density function* describes the infinitesimal probability of any given value, and by integrating the density function over a given interval the probability that the outcome lies in that interval can be calculated.
The *cumulative density function* on the other hand describes the probability that the random variable is no larger than the given value x. This is given by the area under the density function from $-$ inf to x.
The *joint distribution* of two random variables describe the probability of both variables taking certain values simultaneously. For discrete random variables it is usually represented as a probability mass function p(x,y). It is defined for all pairs of values that the random variables can take and for each pair of outcomes it assigns a probability. For continuous random variables the joint distribution is described by the probability density function.
*Convolution* is used when we want to find the distribution of the sum of two independent random variables. It can be viewed as addition, forming a linear combination or combining the information of both distributions., see equation 2.6 and figure 2.2. The convolution operation is commutative and thus it does not matter which function is shifted and reflected, the end result is the same. The mean of the sum is simply the sum of the individual distributions means, e.g $\mu_z = \mu_x + \mu_y$. The same idea applies to the variance of the sum if the distributions are independent, $\sigma_z^2 = \sigma_x^2 + \sigma_y^2$.

$$f_Z(z) = \int_{-\infty}^{\infty} f_X(x) f_Y(z-x) dx \tag{2.6}$$

When performing a convolution of two densities applied in a eucledian space it is important to center the functions correctly. Since the mean of the convolution is a

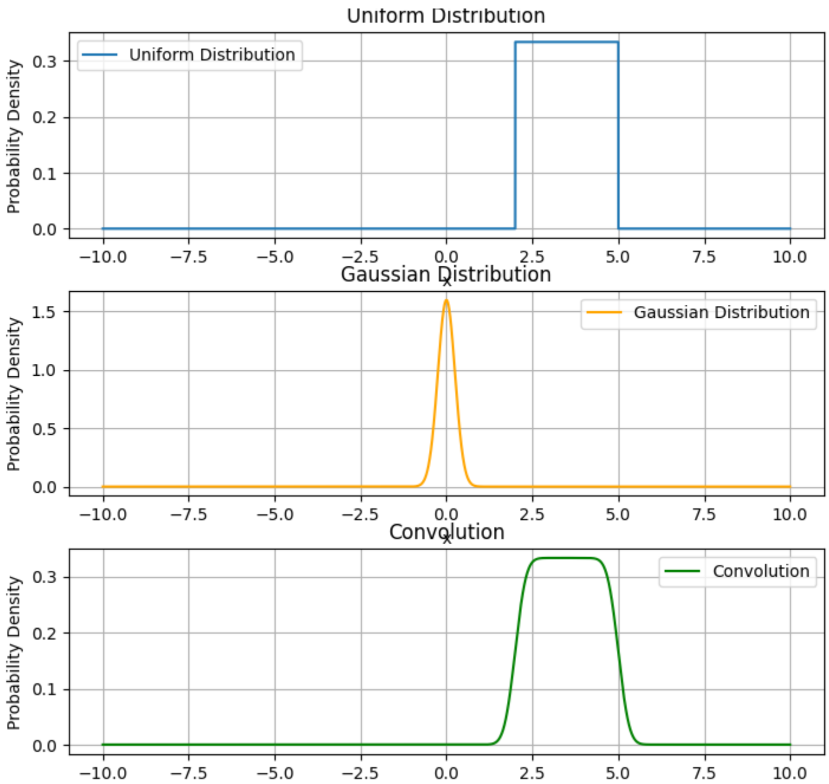



**Figure 2.2:** Convolution of two distributions. Figure 1 is a uniform distribution and figure 2 is a normal distribution. Figure 3 is the convolution of figure 1 and 2.

sum of its components if the densities is not centered the convolution will introduce an offset, in other words the resulting density will have shifted. This behaviour can also be utilized to account for expected movement of the distribution in a direction, e.g applying a brake action with known heading, the expected outcome would be shifted in the vehicles angle of movement. The same can be said for the variance or standard deviation.

## 2.3 Probabilistic positioning

Using a probability density function (PDF) to describe the position of a moving agent is a novel approach to the positioning problem. By defining a PDF for the grid (X,Y) where $P_{v_1}(X_{i,j})$ describes the probability that the vehicle is at that point $(x_i, y_j)$. Then by extension if one combines the probability function for two moving vehicles the joint PDF would describe the probability that both vehicles occupy the same grid-point, i.e a collision has occurred.

$$P_{v_1,v_2}(X,Y) = P_{v_1}(X,Y) * P_{v_2}(X,Y) \tag{2.7}$$

Given the distance calculations described in section 2.1, the probability would be greatest at the edge of the circle, and diminish as you move towards the origin. This is based on the assumption that the measurement is close to the actual distance, it is possible that the measurement has been reflected before returning, but not to the

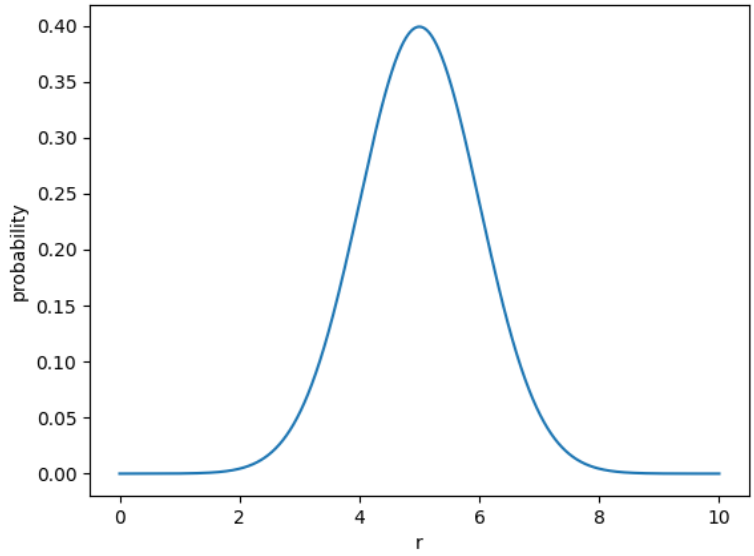



**Figure 2.3:** Spherically symmetric pdf in 1D

degree that the vehicle is likely to be close to the sensor if the circle is sufficiently large. This can be described with a normal distribution, where the mean is set to the measurement value. This would result in a symmetric distribution around the measurement. When calculating the density over the grid, first the distribution needs to be centered on the measurement source. To do this the euclidean distance between the grid point and sensor is calculated and used as the input to the distribution. See figures 2.4, 2.3 to see this distribution in both 1D and 2D with the hyper-parameters Amplitude (A) = 1, Sigma = 1 and $r_{max} = 5$.
For sensor measurements $\bar{\mathbf{x}}$, a vehicles position $(\hat{x}_i, \hat{y}_i)$ in the grid (X,Y) can be estimated by taking the product of the distribution for each measurement and then integrating the distribution for the interval we want to observe:

$$P(\bar{\mathbf{x}}) = \prod_{i=0}^{n} f(\bar{x_i}) \tag{2.8}$$

$$(\hat{x}_i, \hat{y}_i) = \int_a^b P(\bar{\mathbf{x}})dx \tag{2.9}$$

If there is more than one vehicle in the system, then calculating the probability distribution for each and then combining them yields the joint probability distribution describing the probability that both vehicles are in the same space in the grid.

$$P_{v_1,v_2}(X,Y) = P_{v_1}(X,Y) * P_{v_2}(X,Y) \tag{2.10}$$

Algorithm 1 details how to estimate vehicle positions in a system with stationary anchor measurement devices using statistical theory. For each anchor that has a successfully done a communication with a transmitter and thus calculated a distance, a probability distribution is calculated. These distributions are combined into a joint

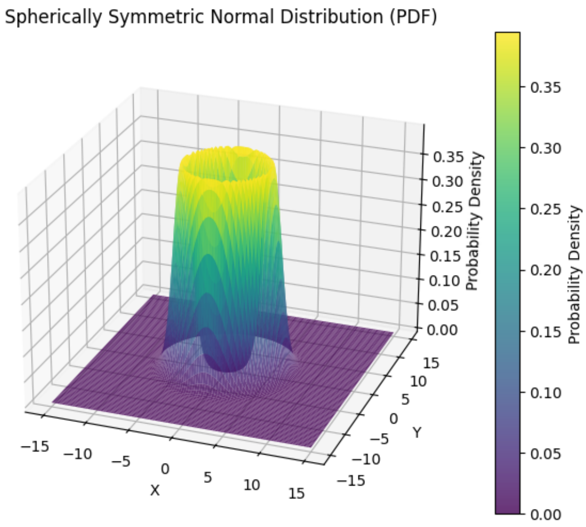



**Figure 2.4:** Spherically symmetric pdf in 2D

distribution, with the outcomes representing the probability that the transmitter is at a certain geographic place. By then integrating the joint probability distribution between two points the probability that the transmitter is within this interval is calculated.

**Algorithm 1** Calculate joint probability density

**Require:** $n_{measurements} > 0, \mathbf{x}_{anchors} > 0$
**Ensure:** $\sigma > 0$
  $P_{joint} \leftarrow \mathbf{1}$
  $i \leftarrow 0$
  **while** $i < n_{measurements}$ **do**
    $r \leftarrow euclidean\_distance(anchor_i, \mathbf{X})$
    $\mu \leftarrow n_{measurements_i}$
    $P_i = normal(r, \mu, \sigma)$
    $E_i \leftarrow uniform(0, (v_i * t_{interval})$
    $P_i \leftarrow convolution(P_i, E_i)$
    $P_{joint} \leftarrow P_{joint} * P_i$
    $i \leftarrow i + 1$
  **end while**
  $(\hat{x}_i, \hat{y}_i) = \int_a^b P(\bar{\mathbf{x}}) dx dy$

## 2.4 Local map calculations

Doing calculations on the whole map grid is unfeasible in real-world scenarios due to several different factors, such as map size or performance constraints. In order to avoid doing these costly calculations on a large matrix several different techniques can be used. A common approach is to do calculations on a smaller, local coordinate system and create a linear transformation that maps a point in the global coordinate system to a point in the local coordinate system. For a three dimensional system see equation 2.11 where $p = [x, y, z]^T$ is a point in the global coordinate system and $\hat{p} = [\hat{x}, \hat{y}, \hat{z}]^T$ is a point in the local coordinate system. The matrix T describes the homogeneous transformation matrix, which contains a rotational and translational element that maps p to $\hat{p}$.

$$T = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix} \tag{2.11}$$

$$p_b = T p_a \tag{2.12}$$

$$p_a = T^{-1} p_b \tag{2.13}$$

Another way is to quickly find a relevant subsection of the global grid and do the calculations on this small subset of the coordinate system. This has the benefit of not requiring transformations between the local and global, but has the drawback of using the same scale as the global grid.
In order to quickly and efficiently find the correct section of the map different relational data structures can be pre-computed and later queried.

## 2.5 Spatial databases and indices

A spatial database is a database that has been enhanced with spatial data that represents geometric objects. It also includes tools to analyze the data and query the contents. Most spatial databases include representations of simple geometric objects, such as points, lines and polygons.
In a spatial database, ordinary sorting schemes (e.g alphabetical) is not ideal for spatial queries, instead a spatial index is created for multi-dimensional ordering. [16] Many different methods of spatial indexing exists, such as

- **Bounding volume hierarchy** (BVH) is a tree structure on a set of geometric objects. The leaf nodes of the tree are represented by the geometric objects which are wrapped in *bounding volumes*. The leaf nodes are grouped as small sets and enclosed in another bounding volume which encompass the set, this is then grouped and enclosed in another larger bounding volume recursively until a single bounding volume contain the whole set of objects in the database. BVH is commonly used in collision detection and ray tracing. [10]
- **k-dimensional tree** (kd-tree) is a binary tree where every node is a k-dimensional point and every non-leaf node is thought of as a hyperplane used to divide the space into two parts called *half-spaces*. The left and right part of this hyperplane are represented as the left and right side of the subtree. Every node is associated with one of the k dimensions and the hyperplane is perpendicular to that dimensions axis. For example if the tree is built with a split on the x axis, then the all points in the subtree with a smaller x value than the

node will appear on the left subtree and the larger values on the right. The kd-tree is commonly used to create point clouds or multidimensional searches such as range or nearest neighbor. [2]

- **Quadtree** is a tree data structure for two dimensional data where each internal node has exactly four child nodes recursively. The leaf cell data type varies by application but can be seen as a unit with spatial information. Each cell has a maximum capacity, when this capacity is reached the cell splits. [1] Some common applications for quadtrees include image processing, mesh generation, collision detection and spatial indexing.
- **Octtree** is the same as quadtree but for three dimensions and instead of four cells per node it has eight to facilitate the extra dimension.
- **R* tree** is a variant of R-Trees and used for spatial indexing. They are used for indexing multi-dimensional data such as rectangles, polygons or coordinates. The data is structured by grouping nearby objects and representing them by their minimum bounding rectangle the the next level up in the tree hierarchy. The leaf node contains a single object, and higher levels contain an increasing number of objects, until the top node bounding box contain the whole set. This structure provides efficient querying since if a query does not intersect a bounding box then it cannot intersect its contents.[3] The r* tree improves on the regular r-tree by updating the node split algorithm and forced reinsertion at node overflow. [4]

# 3
# Method

In this section the methodology is explained in detail, from data measurements, what assumptions are made and how the probabilities are calculated. The methodology is described as a system of operations triggered in sequence based on different criteria, starting with a sensor measurement being picked up and finishing with a risk estimate describing the likelihood of a collision.

## 3.1 Assumptions

- It is assumed that the measurement devices are placed throughout the map without any measurement gap.
- Each measurement device has a max measurement distance, either defined based on the hardware or by its position in the geography (e.g blocked by walls or corners).
- It is assumed that each agent moving in the grid has a unique tag id such that each measurement registered by the supervising system can be mapped to a specific agent.
- The supervising system is doing the risk estimation queries at a known frequency (e.g 10 hertz).
- The vehicle meeting frequency is known and can be used to calculate the probability threshold.

## 3.2 Sensor measurements

As described in (2.1) the sensor data is received and used as a distance measurement based on double sided two way ranging. Measurements from several sensors are batched within each sampling period of the supervising system, e.g all measurements recorded between $t_n$ and $t_{n+1}$ are used for the risk estimation query $q_{n+1}$. Since each agent is equipped with a unique tag the measurements are grouped together by the tag id and used to estimate the position of the agent. This setup is shown in figure 2.1, where one agent is traveling down a path with three close anchors that picks up the signal. These measurements are represented as circles, since the individual measurements does not have a heading other way of knowing in which direction the signal came from. The intersection of the circles will have higher probabilities of agent occupancy. By batching the measurements, a small offset of the vehicle position is introduced in the calculations since the different measurements is done at different times, e.g measurements $m_1, m_2, m_3$ is done at time $t_1, t_2, t_3$ but passed into the system as if they occurred simultaneously. Given a query frequency of 10 hertz a vehicle traveling at 20m/s may have traveled up to 2 meters from the first measurement to the last in a given batch. This offset is later compensated for by introducing a probability distribution which reflects the uncertainty.

## 3.3 Local map extraction

There are many different techniques for doing calculations on a subset of a larger grid map, as described in section (2.4). A combination of a kd-tree and bounding boxes are utilized to create a local subsection of the global map grid containing the relevant agents and sensors. A kd tree is built from all the sensor positions in the system and then used for nearest neighbor and range queries. When an anchor is triggered by an agent tag, the system does a nearest neighbor search for n additional sensors within a distance d and creates a list of the triggered sensor(s) and the neighbors. The system then compares the lists created for the other agents that triggered sensors in the same batch, if any of the lists contain a common sensor the system begins a risk estimate calculation, otherwise the batch is discarded. Given that there is a common sensor in two lists, a local map extraction procedure is started. The relevant subsection of the global map is found by calculating the minimum rectangle which encompass all the bounding boxes of the sensors in the list, see figure 3.1. The intersection of the calculated rectangles are used as the local map grid. The algorithm is detailed in algorithm 2.

## 3.4 Agent position probabilities

With the local grid extracted, a probability density can be calculated from the sensor measurements. For each measurement, a normal probability density is calculated by taking the euclidean distance between the anchor position and each grid point in the local grid to get a vector $r$, and applying a normal distribution with mean $\mu = measurement$ and standard deviation $\sigma$.

$$\mathbf{r} = \sqrt{(X-x)^2 + (Y-y)^2} \tag{3.1}$$

$$P(x) = \mathcal{N}(\mathbf{r}, \mu, \sigma) \tag{3.2}$$

These densities are then combined with a joint operation for a resulting density that represents the combined probabilities for each grid point, as described in algorithm 1.

## 3.5 Vehicle geometry

The measurements used to calculate the probability distribution for the position of a vehicle is measured between a transmitter and receiver, with no regard for the actual space the vehicle occupies. In order to accurately estimate potential collisions, the geometry of the agents needs to be considered. This is done by creating a binary grid

**Algorithm 2** Find overlapping grid for local grid calculations

```
procedure CalculateOverlapGrid(SensorsA, SensorsB, GridSize)
    if SensorsA ∩ SensorsB = ∅ then
        return
    end if
    BoundingBoxA ← GetTotalBoundingBox(SensorsA)
    BoundingBoxB ← GetTotalBoundingBox(SensorsB)
    Overlap ← CalculateOverlap(BoundingBoxA, BoundingBoxB)
    if Overlap = ∅ then
        return
    end if
    Grid ← GenerateGrid(Overlap, GridSize)
    return Grid
end procedure
function GetTotalBoundingBox(Sensors)
    minX ← min(sensor.get_bbox()[0] for sensor in Sensors)
    minY ← min(sensor.get_bbox()[1] for sensor in Sensors)
    maxX ← max(sensor.get_bbox()[2] for sensor in Sensors)
    maxY ← max(sensor.get_bbox()[3] for sensor in Sensors)
    return (minX, minY, maxX, maxY)
end function
function CalculateOverlap(BoxA, BoxB)
    overlapMinX ← max(BoxA.minX, BoxB.minX)
    overlapMinY ← max(BoxA.minY, BoxB.minY)
    overlapMaxX ← min(BoxA.maxX, BoxB.maxX)
    overlapMaxY ← min(BoxA.maxY, BoxB.maxY)
    if overlapMaxX ≤ overlapMinX ∨ overlapMaxY ≤ overlapMinY then
        return ∅
    end if
    return (overlapMinX, overlapMinY, overlapMaxX, overlapMaxY)
end function
function GenerateGrid(OverlapBox, GridSize)
    x ← Linspace(OverlapBox.minX, OverlapBox.maxX, GridSize)
    y ← Linspace(OverlapBox.minY, OverlapBox.maxY, GridSize)
    Grid ← CreateMeshGrid(x, y)
    return Grid
end function
```

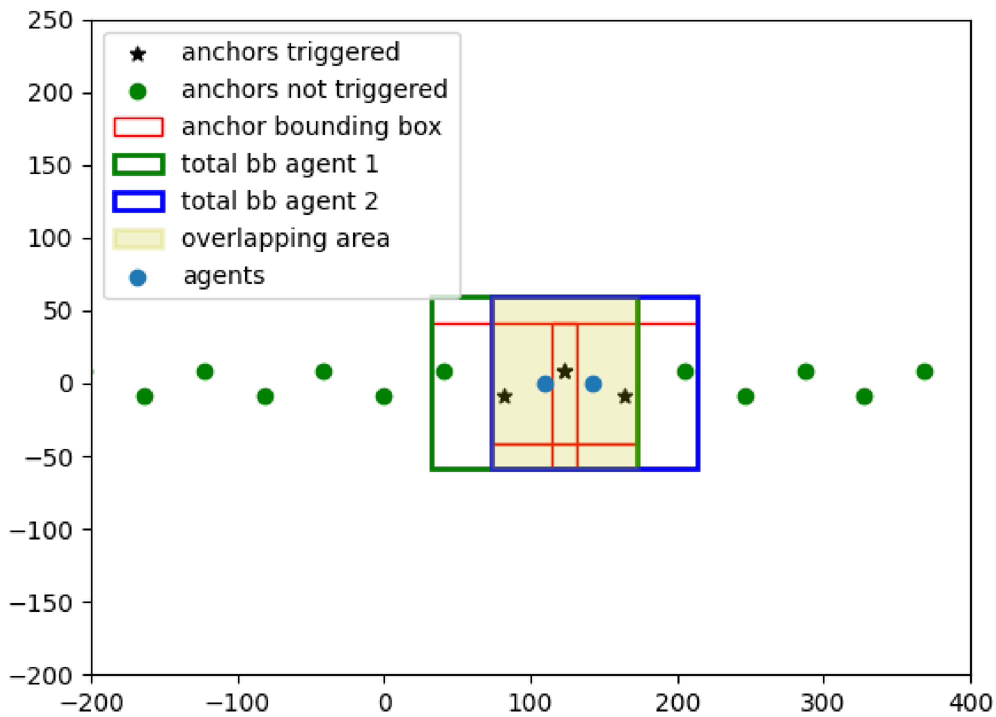



**Figure 3.1:** Local grid extraction from sensor bounding boxes

of the space the agent occupies with the same resolution as the other calculations. If the heading is known then this space is rotated to reflect this orientation and if the orientation is not known, then the space is "inflated" to a square with the sides set to the largest of its dimensions, as seen in algorithm 3. This footprint is then convolved with the position density to create a new density that incorporates the vehicle dimensions in the probabilities.

**Algorithm 3** Vehicle geometry matrix

**Ensure:** $l, w > 0, 0 \leq \theta \leq 2\pi, c = gridcenter$
  $M \leftarrow \mathbf{0}$
  **if** $\theta = \varnothing$ **then**
    $l = w = \max\{l, w\}$
  **end if**
  $dx \leftarrow \lfloor \frac{l}{2} \rfloor$
  $dy \leftarrow \lfloor \frac{w}{2} \rfloor$
  $M[c - dx : c + dx, c - dy : c + dy] \leftarrow 1$
  **if** $\theta \; ! = \varnothing$ **then**
    $M \leftarrow rotate(\theta)$
  **end if**
  **return** M

## 3.6 Collision avoidance system

In order to avoid collisions between the vehicle and another object different avoidance strategies may be employed. The strategy that is utilized is dependant on several factors:

- *Velocity*, both of host and other vehicles on the road.
- *Vehicle properties*, e.g brake force, weight, dimensions etc.
- *Surroundings*, such as other vehicles, barriers, pedestrians etc.

A set of actions can be defined for the agents to perform if the risk of collision is high enough. These actions could be braking, swerving or another action with the intent to avoid a collision. Each action can be described as a probability distribution and given the circumstances be performed in order to mitigate or avoid a collision.

$$\mathbf{a} = \begin{bmatrix} action_1 & action_2 & ... & action_n \end{bmatrix}^T \tag{3.3}$$

The simplest action for a vehicle to take is to apply the brakes and generate a deceleration with the intent to avoid the collision entirely or minimize the damage. This project will focus on the strategy of breaking, which is analogous with activating an emergency stop for a site, resulting in the vehicles applying its brakes.

### 3.6.1 Brake action

A distribution that describes a vehicles brake action will be dependant on several variables, such as the heading, speed and brake force. The brake force and speed will

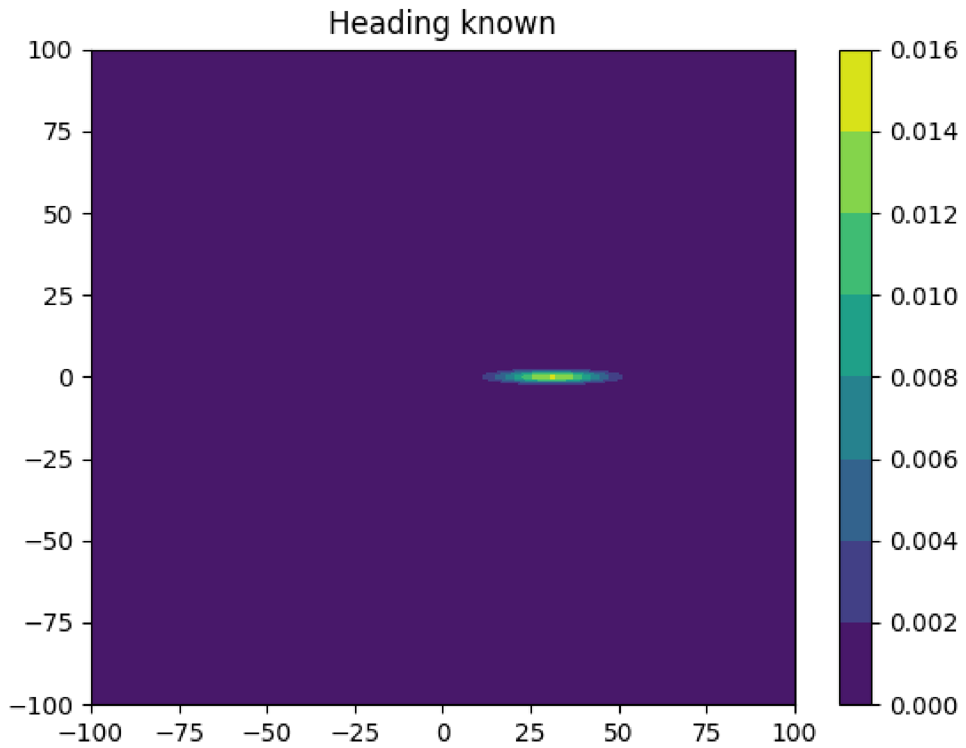



**Figure 3.2:** Probability density for brake action with known heading.

affect the brake distance for the vehicle. If the heading is known, the distribution can be defined as a multivariate distribution with the covariance set such that the distribution is elongated in the heading direction. Thus the probability that the vehicle ends up perpendicular to the heading is significantly less likely than in the direction of movement, see figure 3.2. If the heading is unknown then the distribution is assumed a univariate normal distribution with an appropriate standard deviation that reflects the use case, see figure 3.3. In both cases the estimated brake distance set as the mean, to reflect that the most likely distance it will travel after performing the brake action is the estimated brake distance. The equation used to calculate the brake distance is described in equation 3.4, where $\mu$ is the friction coefficient, g is gravitational acceleration, $F_{brake}$ is the vehicles brake force and v is the vehicle speed.

$$a = \mu * g * F_{brake} \tag{3.4}$$

$$d = \frac{v^2}{2a} \tag{3.5}$$

Algorithm 4 describes how the brake distribution is defined. It is dependant on if the heading is known or not. If the speed of the vehicle is unknown, then a pre-defined max velocity for the vehicle is chosen for the distance calculation. This

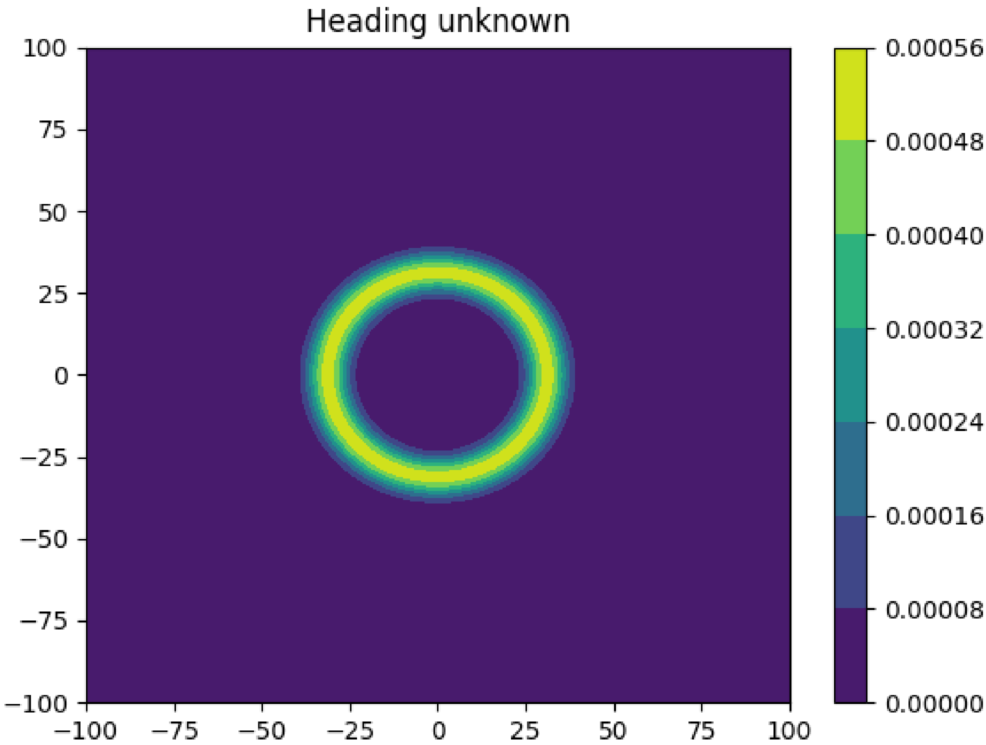



**Figure 3.3:** Probability density for brake action with unknown heading.

---

**Algorithm 4** Define brake action probability distribution

---

$\mu \leftarrow brake\_distance$
**if** $\theta = \varnothing$ **then**
  $p(x) \leftarrow normal(x, \mu, \sigma^2)$
**else**
  $\mu \leftarrow \mu * \begin{bmatrix} cos(\theta) & sin(\theta) \end{bmatrix}^\top$
  $p(x) \leftarrow multivariate\_normal(x, \mu, cov)$
**end if**
$P_{joint}(x) \leftarrow convolve(P_{joint}(x), p(x))$

---

yields a more conservative distribution so that it can confidently be assumed that the density contains all possible positions it can assume in the event of a brake. The last step of the algorithm adding the brake distribution to the total joint distribution representing the potential position of the vehicle.

## 3.7 Combined probability density

When all relevant probabilities are calculated and defined, combining them will result in the final density reflecting the combined probability of the vehicles potential position. The way to combine the different densities varies depending on what they represent. In the case of calculating the position probability density the joint operation is done between each sensor measurement. Adding the vehicle footprint, e.g "inflating" the density with respect to the vehicle dimensions is done by calculating the convolution between the position density and the footprint. The same is done when adding the brake action distribution. Adding another distribution representing the uncertainties inherent in the system which is not accounted for in the previous distributions may also be done in this way, as defined in equation 3.6.

$$P_{tot} = P_{position} * P_{footprint} * P_{brake} * P_{uncertainties} \tag{3.6}$$

### 3.7.1 Convolution using Fast Fourier Transform

When the size of the input arrays increases so too does the computational demand for doing the calculations. In order to mitigate this the Fast Fourier Transform (FFT) can be used to reduce the computational complexity from $O(N^2)$ to $O(Nlog(N))$. The FFT works by first transforming the functions from the time domain to the frequency domain using the Fourier transform, then multiplying the transforms element-wise and finally doing the inverse FFT back to the time domain. The FFT performs circular convolution, that means that the output wraps around the edges of the grid, and thus if a linear convolution is desired it is important to adequately pad the input data before doing the transform.
The FFT is a more efficient algorithm for calculating the convolution, but since it requires more steps than a regular convolution it may introduce some latency to the system. For applications where latency requirements are strict it may still be advantageous to perform regular convolutions.

## 3.8 Dynamic collision probability threshold

The standards dictate the probability of failure per hour for a site, and this probability is dependant on the level of safety integrity. This thesis is working with SIL 2, with a probability of failure per hour of $10^{-6}$ to $10^{-7}$. The expected exposure frequency for this scenario is 1 per hour. Adding a constant representing the possibility of unforeseen additional exposure equation 3.7 is defined.

$$V_{th} = \frac{PFH}{f_e * p_c} \tag{3.7}$$

The frequency of exposure is based on data supplied by the site, where it reflects the expected number of additional vehicles driving through the autonomy zone per hour. By using this value as is a good baseline probability threshold can be calculated.

# 4 Results

In this chapter the results of the thesis is detailed. It describes the theoretical results of applying the method and theory.

## 4.1 Analytical Results

This section details the analytical results gained from running the algorithm(s) locally on generated measurements and values. The following scenario is defined:

- Global grid defined as 2000x2000.
- 40 Anchors positioned alternating between two straight lines through the global system, with 41 units between each alternating pair (82 between two on the same line) and 18 units between the "lines".
- Two vehicles, positioned with opposite heading at position (5, 0) and (300, 0) respectively. They have the same load and properties except speed.
- Vehicle 1 has speed 5 and Vehicle 2 has speed 7.
- A maximum of 5 anchor measurements is used when performing the position distribution calculation.
- The maximum measurement distance for the anchors is 70.
- When calculating the probability distribution for position standard deviation $\sigma = 5$ was used.
- When calculating the brake distance the following parameters was used:
  - Friction coefficient $\mu = 0.7$
  - System assume no knowledge of the speed so the vehicle max speed is used: $v_{max} = 20$
  - This results in a brake distance of  42 units.
- When calculating the probability density for the brake distribution the following parameters was used:
  - Heading known: covariance matrix: $\begin{bmatrix} 100 & 0 \\ 0 & 1 \end{bmatrix}$
  - Heading unknown: standard deviation $\sigma = 4$
- The vehicle footprint is x=2, y=5
- The probability of failure per hour is set to $10^{-8}$ for the heading unknown case and $10^{-7}$ for the heading known case. The difference is because of the more spread out distribution of the unknown heading case.
- The expected number of meetings per hour is set to 1

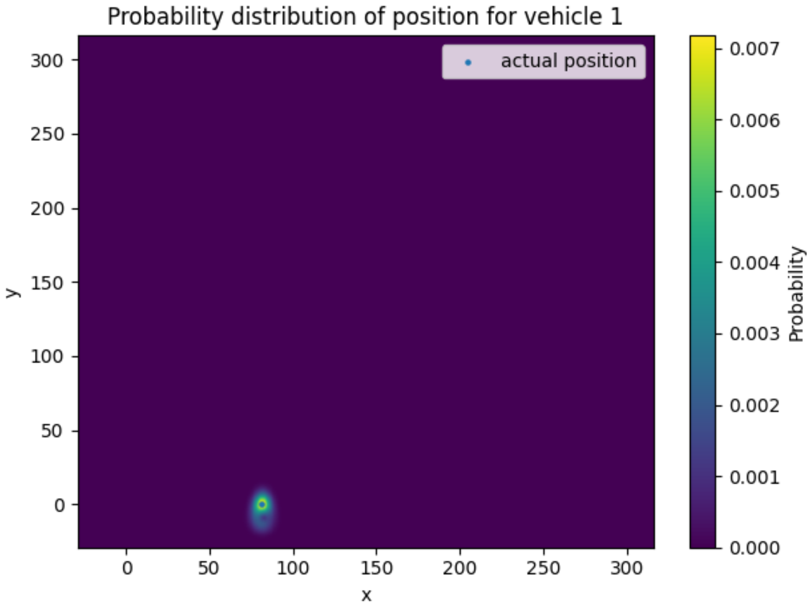



**Figure 4.1:** Probability density representing the measured position of an agent.

The values chosen for the various hyper-parameters (such as standard deviation, friction coefficient and covariance) is best-guesses based on a cursory research on similar environments. They are not based on specific measurement data or simulations.
The joint probability distribution representing the potential position of vehicle 1 is shown in figure 4.1. The ground truth position of the vehicle is indicated by the blue dot, surrounded by the probabilities calculated from the sensor measurements. The densities representing the brake actions are shown in figures 4.2, 4.3 along with the position probabilities and the resulting density from convolving the two.
Figure 4.4 shows the result of convolving the position distribution with a vehicle dimension footprint in different orientations. When the orientation of the vehicle footprint is rotated, the resulting probability distribution is also rotated.
Combining these two distributions and the footprint for both vehicles and calculating the joint distribution between them the system monitors for potential collisions. Figures 4.5, 4.7 shows the final densities for both vehicles and the joint distribution in 1d. The vertical lines in the figure represents the ground truth of the corresponding vehicle, in this case the x-position. The curves represents the final probabilities for each vehicle and the joint. while 4.6, 4.8 shows the same in 2d.
The table 4.1 contains the results from running the algorithm 200 times, measuring the average time to perform a system query and the first warning trigger for a potential collision. The system used to run it is a intel core i7-10805H with 2.7Ghz and 32Gb of RAM. No parallelisation was done.

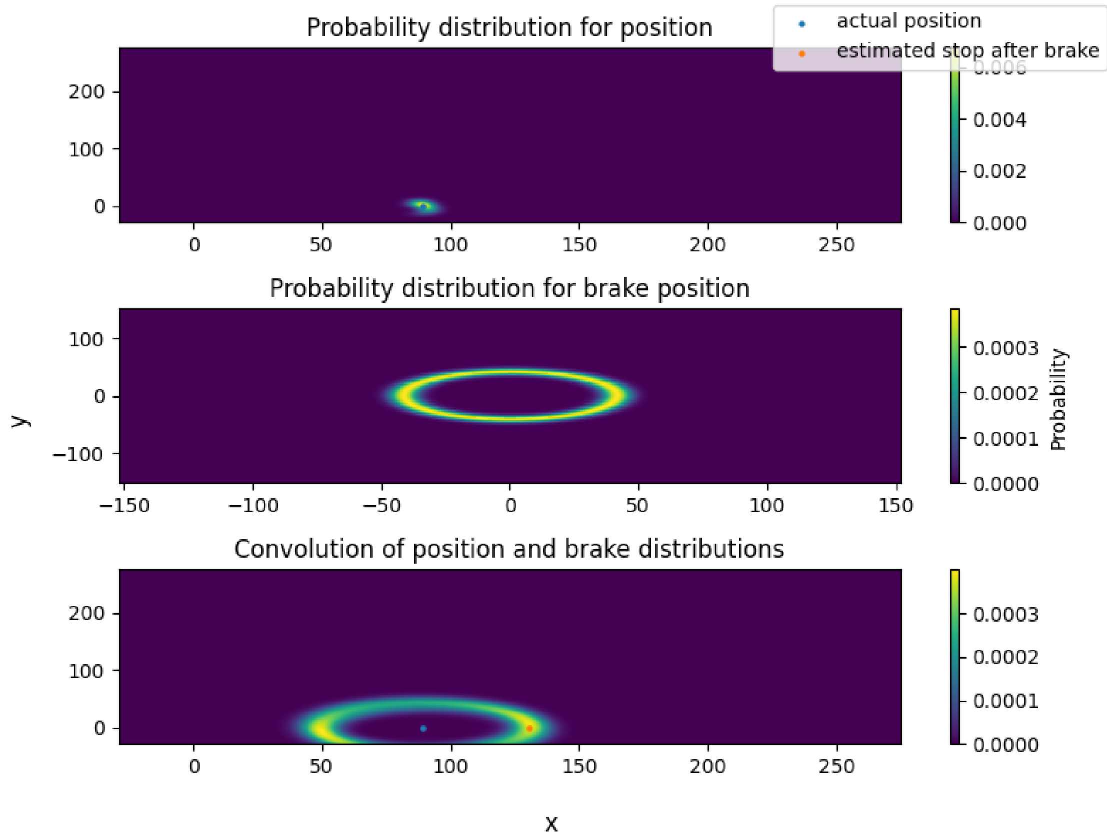



**Figure 4.2:** Probability density for brake action with unknown heading.

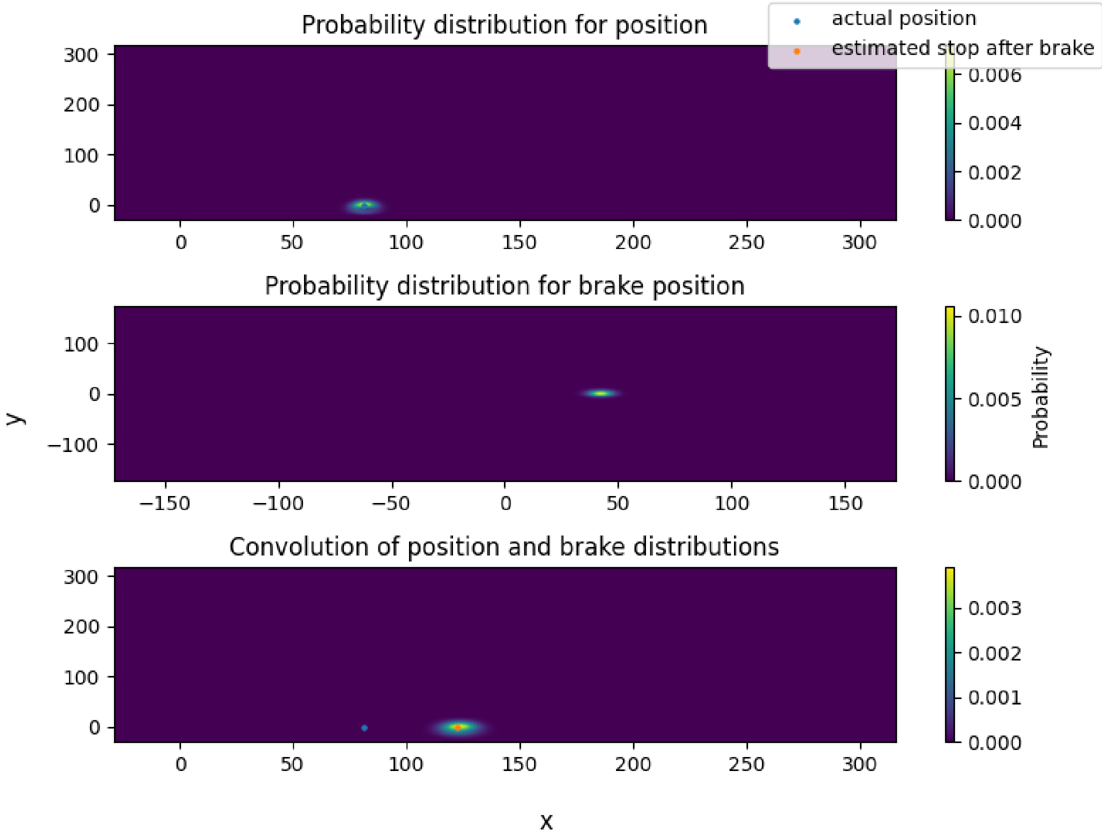



**Figure 4.3:** Probability density for brake action with known heading.

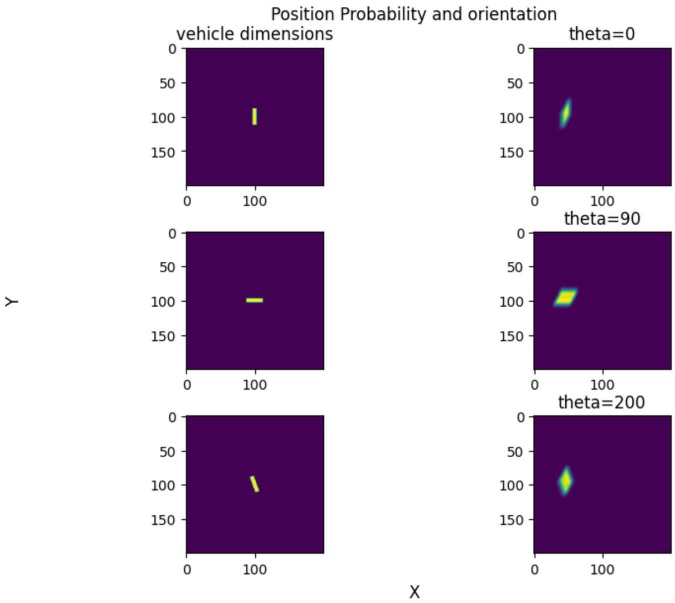



**Figure 4.4:** Probability densities for measured position with vehicle shape and orientation included.

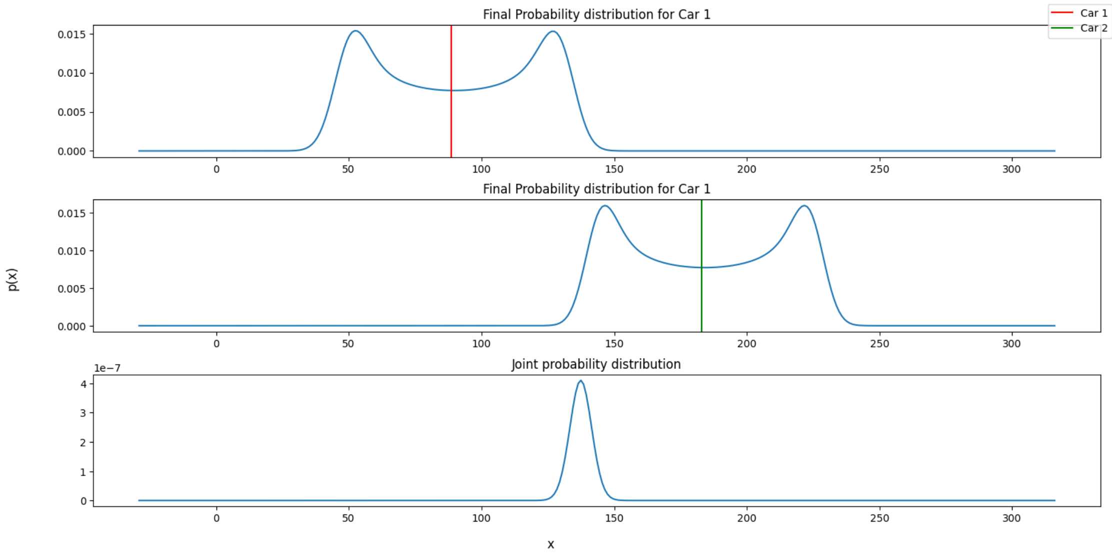



**Figure 4.5:** Combined Probability densities in 1d when the heading of the vehicles is unknown to the system. The vertical lines indicate the x position of the vehicle.

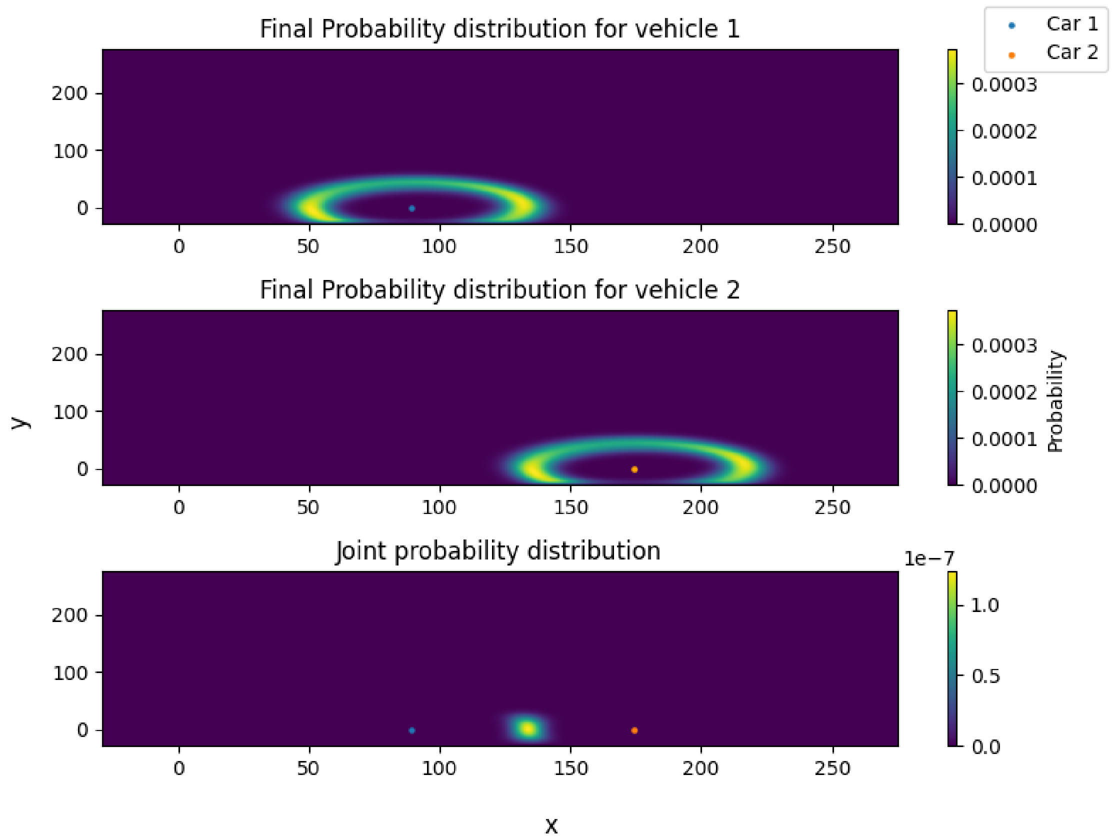



**Figure 4.6:** Combined Probability densities in 2d when the heading of the vehicles is unknown to the system. The dots represents the position of the vehicles.

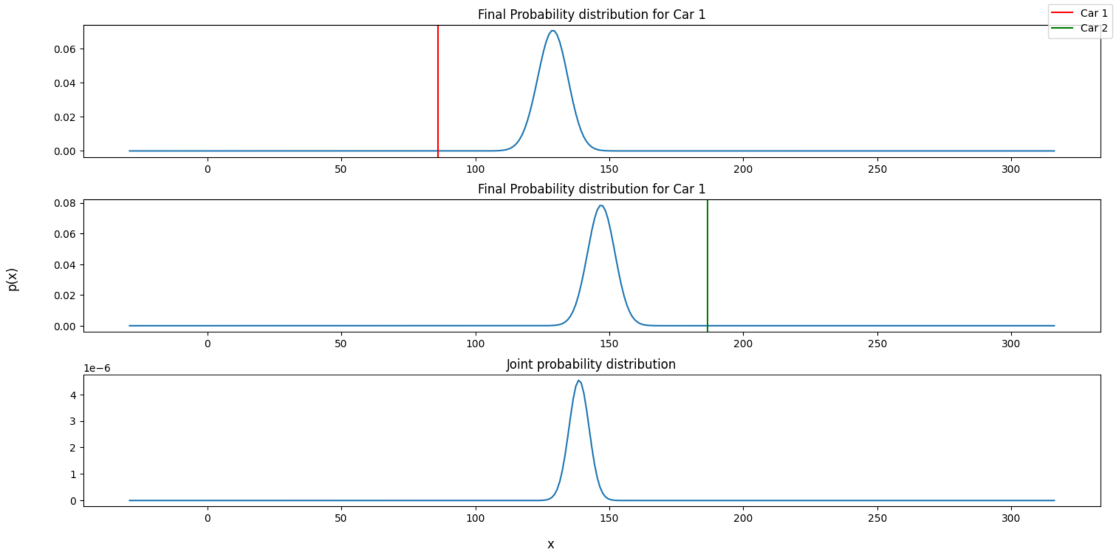



**Figure 4.7:** Combined Probability densities in 1d when the heading of the vehicles is known to the system. The vertical lines indicate the x position of the vehicle.

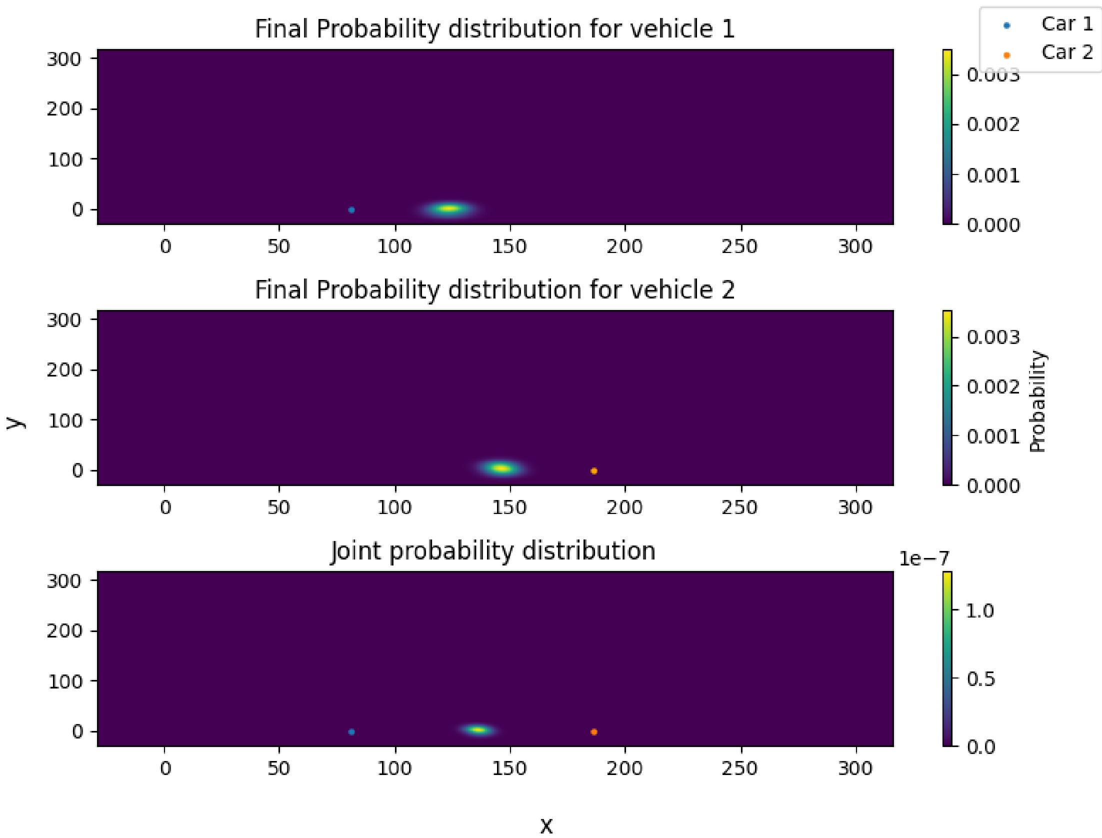



**Figure 4.8:** Combined Probability densities in 2d when the heading of the vehicles is known to the system. The dots represents the position of the vehicles.

| Heading known | Avg. Query time | avg. First Collision Warning time |
|---|---|---|
| True | 0.0021408s | 20.0740 |
| False | 0.0011394s | 19.2045 |

**Table 4.1:** Average query time and first collision warning when heading is known or unknown to the system.

# 5
# Discussion

This project set out to investigate the feasibility of using a statistical approach using probability distributions to model the risk of collisions using only external sensor data. By modelling the different variables as random variables and applying it to a euclidean space in the form of distributions, the sum of an agents potential position at a given time can be viewed as a geometry of probabilities. Thus if two or more of these geometries overlap and the sum of that overlap is above a certain safety threshold, anti-collision measurements can be taken in order to prevent the collision. This has the potential to create a smaller area of risk compared to the alternatives, reducing the amount of false positives thus making the whole system more efficient. The alternative (and current) solution is to model the different parameters as rectangles that represents the vehicles area of risk. The algorithm is mostly the same, as the one described in this paper, first the localization is done by taking distance measurements from the sensors and representing them as circles and calculating the intersection of them. This represents the area in which the vehicle is assumed to occupy. This localization geometry is then expanded with regards to first the vehicle dimensions, and then the dynamics. The idea is the same with regards to the heading, if it is unknown then the geometry is expanded with the largest vehicle dimension in all axis. The same idea applies to the dynamics, if the heading is unknown then the geometry is expanded in all directions equally. This results in a rather large volume representing the vehicles potential position and space occupation. However because every point in the volume has equal probability, it is a rather crude representation which lends itself to possible false positives. On the other hand, the calculations are simple and the hardware requirements are thus relatively small. The approach described in this thesis is more demanding of the hardware than doing a binary geometry representing the area it may be in, since its doing calculations on fairly large matrices representing the grid. However the potential gain by reducing the number of false positives, i.e. the number of unnecessary emergency stops more than outweighs the added cost of higher hardware requirements.

By using a statistical model to represent the states of real objects, we need to collect a lot of data for the different scenarios to accurately set the model parameters, such as variance and mean. This can be done initially by collecting data from the environment it is supposed to be used at and calculating the mean and standard deviation for the given scenario. This could include braking effect and distances and measurement errors. Another set of parameters to consider when tuning a system such as this is the grid size and resolution.

The results gained from this study mostly falls within the expectations, we managed to define a distribution that represents a vehicles total positional probabilities and combine them such that a distribution can be calculated that reflects the joint probability of a collision between several vehicles. The algorithm to combine the individual distributions representing various parameters is defined and adding more to the mix is as straightforward as defining a distribution representing the parameter, apply it to the grid as a euclidean distance and calculate the convolution with the rest. As mentioned previously the use of statistical functions to accurately describe real world objects require a lot of data in order to properly set the hyper-parameters. The parameters set when running the analytical simulations are best-guesses and not based on measurements or interpolation done on our own data. The focus was always on finding a way to express the distribution and finding an algorithm to do it.

Looking at the different results, we can see that the probability density representing the positional measurements is very accurate, where the ground truth is on top of the center of the distribution. This is heavily dependant on the number of anchors that pick up the tag, if less than three is used the distribution gets poorly defined and the resulting densities is very spread out and uncertain. The brake action distribution is also very accurate to the expected values, where we can see that the distributions are centered on the expected brake point in the case of knowing the vehicles heading, and in the middle of the deviation in the case of the heading unknown. The algorithm assumes that we don't know the vehicles current speed, and instead uses the predefined max speed of the vehicle type to calculate the brake distance. By enabling the system to know the speed nothing in the algorithm changes, and the brake distance would decrease or increase depending on the vehicles speed.

The addition of vehicle dimensions to the distribution is also behaving as expected, where the distributions gets elongated or stretched out in the shape of the vehicle. In the case of not knowing the heading the distributions get stretched equally in all dimensions and in the other case it gets stretched as the vehicles proper dimensions and also rotated to reflect its heading. This worked out very well and is an important addition to the algorithm to make it more applicable for real-world scenarios.

When calculating the joint between the vehicles different distributions the resulting distribution looks as expected where it represents the density where the individual densities overlap. The magnitude of this distribution and its maximum value is what is compared to the collision probability threshold. This is where the choice of hyper-parameters becomes apparent, if the parameters are poorly picked, the magnitude may be off and collisions that is likely to happen may or may not be reflected by the calculated probabilities. So it will be very important to rigorously investigate and research the proper values for the given scenarios this will be running for. We can see by comparing the figures 4.6 and 4.8 that the case of knowing the heading yields higher probabilities and the joint is one magnitude larger than the case of not knowing the heading. This is because inherently knowing the heading results in a smaller volume (since the distribution is centered on the brake distance) compared to the unknown heading case, where the distribution is centered on the position distribution with the brake distance set as the mean, i.e. a spherically symmetric distribution.

Another point of discussion is the fact that we don't normalize the final joint den-

sity, so it is technically not a valid distribution. We do it this way because we want the density to reflect the individual probabilities per grid cell to be preserved when comparing it to the threshold. If we normalize the joint then the cell probabilities would be changed, no matter how small the individual probabilities were the normalization would change the values, resulting in results that don't reflect the individual distributions probabilities. For example say that the standard deviation of the individual vehicles distributions is large, such that small probabilities are far out from the mean, they would show up "first" on the joint distribution, and by normalizing these very small probabilities would be inflated and trigger the collision avoidance system when in reality the vehicles may not have been in a dangerous trajectory at all.
The dynamic Probability of failure per hour calculations done to break it down to a per-meeting basis is a quite basic equation which has room for improvement. We start with the requires SIL level and get the required PFH interval, take the lowest value, divide with how many expected risk exposures (or meetings) per hour for the site and a constant to reflect some uncertainty. This results in an estimated per meeting probability that the max value of the joint density for each risk calculation have to reach in order for the system to trigger an emergency brake for the site. This is quite clumsy and is certainly an avenue for further research. By making the calculations more dynamic in the sense that the probabilities can be updated on the fly when the number of vehicles in the site changes for example, or tying it more tightly to the standards by integrating the SIL's or risk exposure frequencies better.

# 6
# Conclusion

By leveraging statistical decision-making this thesis presents a novel approach to collision avoidance. This research has been able to successfully develop a method that reduces the number of false positives while maintaining the accuracy of predicting potential collisions. This was achieved by defining a vehicles potential position at a given time as a combination of probability distributions, and integrating these with real-time sensor data.
By the development of a dynamic probability of failure threshold based on safety integrity levels we successfully integrated the system with established safety standards and regulations. The results from an analytical evaluation indicate potential of this statistical approach for adoption in real-world application of mixed traffic environments.
The application and integration of probabilistic risk assessment for collision avoidance systems holds promise for the future of vehicular safety protocols.

## 6.1 Future work

This research focused on the core idea of representing a vehicle risk volume using statistical properties in order to get a more accurate external collision avoidance system. The method can be further improved in various ways to get a more robust and efficient system. It was shown that knowing the heading yields a more accurate representation and thus better results. In order to consistently get the heading for each agent that is subject to risk estimation is to either have the vehicles Inertial Measurement Unit (IMU) data available or to save the previous estimated position of the agent and interpolate data from this. By saving the previous estimate the agents velocity can also be interpolated and used to better model the brake distance thus increasing the accuracy of the estimation.
On the subject of saving previous states and use in calculations, there are several techniques that can be used to optimize the algorithm. One such technique could be to utilize temporal coherence in order to reduce the number of vehicle pairings to check. In the current implementation all vehicles are paired up and checked if they are in the vicinity of each other. Since vehicles move a relatively small distance between queries, thus if a vehicle pair is in different parts of the map last query, it is unlikely that they are in close proximity this query, and thus the pairing can be discarded before doing any further calculations. This would result in a smaller subset of vehicle pairing being calculated each iteration.
Another avenue for further research is the addition of other avoidance maneuvers,

such as swerving or regulating the vehicle speeds dynamically. The swerving maneuver in particular may require certain requirements of the environment the vehicle operates in, but it would most likely be a valid option for a last resort avoidance maneuver. Dynamic regulation of vehicle speed is the idea that looking at a larger picture of the map, if there is a potential for collision in the future given two vehicles current trajectories and speed, then reducing one or both vehicles speed would result in the "close collision avoidance calculations" to not be needed since the scenario would be avoided.

A limitation set on this thesis was that only two vehicles will be calculated simultaneously and that they are assumed to be meeting head on. It would be beneficial to add the ability for the algorithm to check more then two vehicles probability of collisions at the same time, even if it is an unlikely scenario. The limitation of only checking head-on collisions would be a good way to increase the robustness of the algorithm. Collisions from the back, e.g that a vehicle comes up from behind with a higher speed than the one in front. For example by checking if the vehicles has the same or similar heading. In such a case the distribution of the "vehicle in front" would forego the moving of the distribution by the brake action so it would stay centered on the vehicle.

Another point to research further is the dynamic collision threshold calculations as mentioned in the discussion. By improving the equation or creating an algorithm to dynamically update this threshold based on the number of vehicles in the site would yield a significant improvement to the overall system. By making a distinction between autonomous and manual vehicles checked in to the site and having different weights for them in the calculations would also be a better reflection of the reality. On the topic of the threshold, there is also the possibility of adding a system to update the meeting frequencies and related data points dynamically by utilizing historical data to fit the values better to the specific case.

# Bibliography


[1] Raphael Finkel and Jon Bentley. “Quad Trees: A Data Structure for Retrieval on Composite Keys.” In: *Acta Inf.* 4 (Mar. 1974), pp. 1–9. DOI: 10.1007/BF00288933.

[2] Jon Louis Bentley. “Multidimensional binary search trees used for associative searching”. In: *Commun. ACM* 18.9 (Sept. 1975), pp. 509–517. ISSN: 0001-0782. DOI: 10.1145/361002.361007. URL: https://doi.org/10.1145/361002.361007.

[3] Antonin Guttman. “R-trees: a dynamic index structure for spatial searching”. In: *Proceedings of the 1984 ACM SIGMOD International Conference on Management of Data*. SIGMOD ’84. Boston, Massachusetts: Association for Computing Machinery, 1984, pp. 47–57. ISBN: 0897911288. DOI: 10.1145/602259.602266. URL: https://doi.org/10.1145/602259.602266.

[4] Norbert Beckmann et al. “The R*-tree: an efficient and robust access method for points and rectangles”. In: *SIGMOD Rec.* 19.2 (May 1990), pp. 322–331. ISSN: 0163-5808. DOI: 10.1145/93605.98741. URL: https://doi.org/10.1145/93605.98741.

[5] J.T. Klosowski et al. “Efficient collision detection using bounding volume hierarchies of k-DOPs”. In: *IEEE Transactions on Visualization and Computer Graphics* 4.1 (1998), pp. 21–36. DOI: 10.1109/2945.675649.

[6] Jonas Jansson. “A probabilistic approach to collision risk estimation for passenger vehicles”. In: *IFAC Proceedings Volumes* 35 (July 2002). DOI: 10.3182/20020721-6-ES-1901.01505.

[7] Jonas Jansson, Jonas Johansson, and Prof Gustafsson. “Decision Making for Collision Avoidance Systems”. In: (Apr. 2002). DOI: 10.4271/2002-01-0403.

[8] Rickard Karlsson and Jonas Jansson. “Model-based statistical tracking and decision making for collision avoidance application”. In: Apr. 2003, 3435–3440 vol.4. ISBN: 0-7803-8335-4. DOI: 10.23919/ACC.2004.1384441.

[9] Jonas Jansson. “Dealing with Uncertainty in Automotive Collision Avoidance”. In: Jan. 2004, pp. 165–180. ISBN: 978-3-540-20586-9. DOI: 10.1007/978-3-540-76989-7_12.

[10] Christer Ericson. *Real-Time collision detection*. Morgan Kaufmann, 2005. ISBN: 1-55860-732-3.

[11] Jonas Jansson. “Collision Avoidance Theory with Application to Automotive Collision Mitigation”. PhD thesis. June 2005.

[12] R.B. Ash. *Basic Probability Theory*. Dover Books on Mathematics Series. Dover Publications, Incorporated, 2012. ISBN: 9780486135199. URL: https://books.google.se/books?id=fmNbFnfrb14C.

[13] Gustav Markkula et al. "A Review of Near-Collision Driver Behavior Models". In: *Human Factors* 54.6 (2012). PMID: 23397819, pp. 1117–1143. DOI: 10.1177/0018720812448474. eprint: https://doi.org/10.1177/0018720812448474. URL: https://doi.org/10.1177/0018720812448474.

[14] Abdulrahman Alarifi et al. "Ultra Wideband Indoor Positioning Technologies: Analysis and Recent Advances". In: *Sensors* 16.5 (2016). ISSN: 1424-8220. DOI: 10.3390/s16050707. URL: https://www.mdpi.com/1424-8220/16/5/707.

[15] Nick Thomopoulos. *Statistical Distributions: Applications and Parameter Estimates.* Jan. 2017. ISBN: 978-3-319-65111-8. DOI: 10.1007/978-3-319-65112-5.

[16] X Zhang and Z Du. "Spatial Indexing". In: *The Geographic Information Science & Technology Body of Knowledge (4th Quarter 2017 Edition), John P. Wilson (ed).* (2017). DOI: 10.22224/gistbok/2017.4.12.

[17] Abhishek Goudar, Timothy D. Barfoot, and Angela P. Schoellig. "Continuous-Time Range-Only Pose Estimation". In: *2023 20th Conference on Robots and Vision (CRV).* 2023, pp. 29–36. DOI: 10.1109/CRV60082.2023.00012.

[18] Francesco Riz, Luigi Palopoli, and Daniele Fontanelli. "Why Three Measurements are not Enough for Trilateration-based Localisation". In: *2023 IEEE International Instrumentation and Measurement Technology Conference (I2MTC).* 2023, pp. 01–06. DOI: 10.1109/I2MTC53148.2023.10176002.

[19] *Boliden AB operations.* https://www.boliden.com/operations/. Accessed: 2024-04-22.

[20] *Functional safety of electrical/electronic/programmable electronic safety-related systems - Part 1: General requirements (see Functional Safety and IEC 61508).* Standard. Geneva, CH: International Electrotechnical Commission, Apr. 2010.

**DEPARTMENT OF MECHANICS AND MARITIME SCIENCES**
**CHALMERS UNIVERSITY OF TECHNOLOGY**
Gothenburg, Sweden
www.chalmers.se

CHALMERS
UNIVERSITY OF TECHNOLOGY