

Golnaz Raja

VISION-BASED COST MAPS FOR SAFE AUTONOMOUS NAVIGATION

Design and Evaluation of Vision-based Control Barrier
Functions

Master of Science Thesis
Faculty of Information Technology and Communication Sciences
Examiner: Prof. Reza Ghabcheloo
Dr. Nataliya Strokina
October 2023

ABSTRACT

Golnaz Raja: Vision-based Cost Maps for Safe Autonomous Navigation
Master of Science Thesis
Tampere University
Signal Processing and Machine Learning, Computing Sciences
October 2023

Designing safety-critical systems for unfamiliar environments is a substantial challenge in the field of robotics. Control Barrier Functions (CBF) serve as a common tool for addressing this challenge. However, the definition of CBF based on perceptual input remains a relatively unexplored and complex area of research.

This thesis extends prior work, where the authors, including the author of this thesis, introduced the innovative concept of Vision-based Control Barrier Functions (V-CBF). V-CBF defines Control Barrier Functions using perceptual input obtained from an RGBD camera, enabling the avoidance of obstacles with arbitrary shapes in unknown environments. A pivotal element of V-CBF is the 2D customized cost map, which transforms the segmented unsafe sets into an appropriate format that satisfies the requirements of CBF. The design and evaluation of these cost maps play a crucial role in the proper generation of V-CBFs. However, this aspect has not been thoroughly extensively in previous research.

The thesis embarks on a comprehensive investigation of diverse methodologies for generating these essential cost maps. The proposed methods are rigorously implemented and assessed within the CARLA simulator. To offer a thorough evaluation, both qualitative and quantitative comparisons are conducted, drawing from industry-standard ISO 22737 guidelines and custom-designed metrics within the CARLA simulator environment.

Furthermore, to substantiate the practical applicability of V-CBF, it is implemented on an industrial mobile robot. This real-world deployment serves as a tangible demonstration of the effectiveness of V-CBF in unknown environments, emphasizing its potential beyond simulated contexts. The transition from simulation to tangible real-world implementation underscores the portability and robustness of V-CBF, signifying its relevance and promise in real-world scenarios.

Keywords: Vision-based Cost Map, Perception-based Safe Autonomous Navigation, Obstacle Avoidance, Control Barrier Functions, Local Perception for Navigation

The originality of this thesis has been checked using the Turnitin OriginalityCheck service.

PREFACE

I would like to express my sincere gratitude to my esteemed supervisors, Professor Reza Ghabcheloo and Dr. Nataliya Strokina, for their guidance, and invaluable insights throughout the course of my research.

Additionally, I am deeply thankful to Teemu Mökkönen for his crucial assistance during the real-world implementation phase and Parsa Miraghaei for his support during writing the thesis.

Finally, I would like to thank my parents for their encouragement and support on my academic journey. Their belief in my potential has been a constant source of motivation and strength for me.

Tampere, 30th October 2023

Golnaz Raja

CONTENTS

1.	Introduction	1
1.1	Problem Statement	1
1.2	Research Questions	2
1.3	Thesis Outline	3
2.	Related work	4
2.1	End-to-End Safe Navigation Methods	4
2.1.1	RL-based End-to-End Safe Navigation Methods	4
2.1.2	Vision-Based Cost Map Approaches	5
2.2	Cost Maps and Control Barrier Functions	6
2.2.1	Vision-based Control Barrier Function	7
3.	Theoretical Background	8
3.1	CBF Control Methodology	8
3.2	V-CBF Control Methodology	10
3.3	Multiple V-CBFs	11
3.4	Depth-based Cost Map Definition for V-CBF	12
3.5	Image-based Cost Map Definition for V-CBF	13
4.	Image-based Cost Map Generation Methods for V-CBF	15
4.1	Gaussian Blur	15
4.1.1	Theoretical Background	15
4.1.2	Gaussian Blur as an Image-based Cost Map	16
4.2	Distance Transform	17
4.2.1	Theoretical Background	17
4.2.2	Signed Distance Transform as an Image-based Cost Map	19
4.3	Learning-based methods.	20
4.3.1	Conditional Generative Adversarial Networks	20
5.	Evaluation in Simulated Environment.	24
5.1	ISO 22737 Standard	24
5.2	Scenario-based Test Platform based on ISO 22737 Standard	25
5.2.1	Scenario Test Setup and Tools.	25
5.2.2	Scenario Generation Process	28
5.2.3	V-CBF Hyperparameters and Configuration in Scenarios	29
5.3	Evaluation Metrics	30
5.4	Evaluation and Result Analysis	30
5.4.1	Analysis of Scenario 1	31

5.4.2	Analysis of Scenario 2	31
5.4.3	Analysis of Scenario 3	33
6.	V-CBF Validation on Industrial Mobile Robot	37
6.1	Experiment Setup	37
6.1.1	Hardware Setup	37
6.1.2	Software Setup	38
6.2	Goal of the Experiment	39
6.3	V-CBF Hyperparamters and Configuration in the Experiment	39
6.4	Results and Performance Analysis	39
7.	Discussion and Conclusion.	43
	References.	46

1. INTRODUCTION

The safe operation of autonomous systems in complex and unfamiliar environments presents a significant challenge within the realm of robotics. As the demand for autonomous systems continues to rise across various applications, ensuring their safe and reliable functioning has become an utmost priority. Control Barrier Functions (CBF) [2] emerge as a potent mathematical tool for guaranteeing the safety of autonomous navigation. However, a notable gap exists within the research community, particularly with regard to the utilization of sensor data in the definition of CBF. In a preceding paper [1], a team of authors, including the author of this thesis, introduced a pioneering concept known as Vision-based Control Barrier Functions (V-CBF). This innovative approach redefines CBF within the domain of image space, introducing the novel concept of image-based safety cost maps to the robotics community. V-CBF offers an exceptionally robust framework for ensuring safety in unknown environments, in the presence of obstacles with arbitrary shapes, making use of state of the art segmentation models. This thesis builds upon V-CBF, with the objective of exploring various methods for generating V-CBF and validating the performance of it through real-world robotic experiments.

1.1 Problem Statement

The generation of V-CBF, necessitates the appropriate mapping of input images to a space that encapsulates essential information for V-CBF. During this process, proper cost values are assigned to pixels of the input image. Therefore, this process is referred as "cost map generation" in this thesis. While one method is proposed in the V-CBF paper for generating such cost maps, a more thorough exploration of potential mapping methods remains uncharted territory within the prior research. This work is set to investigate viable cost map generation methods that are well-suited to meet the prerequisites of V-CBF.

Furthermore, the examination of impact of different cost map designs on the overall performance of V-CBF is a key aspect of this research. To address this inquiry, a series of obstacle avoidance scenarios are designed and implemented in compliance with the ISO 22737 standard [19] within the CARLA simulator [12]. The designed framework is utilized to test and compare the candidated methods for generating V-CBFs rigorously.

However, a qualitative comparison based solely on scenario-based assessments may not provide a comprehensive understanding of the behavior of the controller. Therefore, there arises a compelling need for a well-designed metric system to analyze and evaluate V-CBF and analogous safety assurance controllers. Remarkably, the research community has hardly addressed the development of a unified metrics framework for evaluation of safety controllers. This absence makes it inherently challenging to compare the performance of diverse obstacle avoidance controllers, as none of them have been assessed under a standardized metrics system. Therefore, this thesis introduces a comprehensive metrics system that considers essential performance metrics for analysis of obstacle avoidance algorithms. Subsequently, this evaluation platform is employed to compare the performance of V-CBF under distinct cost map methodologies.

This qualitative and quantitative evaluation system introduced in this thesis serves as a foundational framework for comparing the performance of forthcoming obstacle avoidance controllers. Scenarios in this evaluation system are derived from ISO 22737, a widely accepted ISO standard within the community, coupled with vital metrics for quantitative analysis of controller performance.

Ultimately, due to the gap that often exists between simulated environments and real-world applications, assessing V-CBF exclusively in a simulated context cannot fully reveal its strengths and limitations. Hence, this thesis extends the evaluation of V-CBF to encompass real-world scenarios, validating its practicality and efficacy. To accomplish this, V-CBF is implemented on an industrial mobile robot, showcasing its capabilities through real-world experiments.

1.2 Research Questions

To summarize, the goal of this thesis is to contribute to the advancement of safe and efficient autonomous navigation in complex and unfamiliar environments, with potential implications for various industries and applications. This goal is being addressed by improving and enhancing design, evaluation, and validation of the state-of-the-art safety controller, VCBF, as well as introducing a well-structured scenario-based and metrics-based evaluation system for later safety assurance and obstacle avoidance controllers.

To be specific, following research questions are addressed in this thesis:

- what are suitable methods for defining control barrier functions in the image space based on RGB-D images?
- what is the comparative evaluation of candidate V-CBF generation methods, drawing from ISO 22737 standard and custom-designed metrics in CARLA simulator?
- what are practical details and results of implementing V-CBF on an industrial mobile robot?

The following chapters will delve into each of these research questions, providing a

detailed exploration of the methodology, findings, and contributions to the field.

1.3 Thesis Outline

The structure of this thesis is as follows: Chapter 2 offers an extensive review of related work within the field of CBF, V-CBF, and other vision-based navigation and obstacle avoidance methods introduced in the community, highlighting key contributions and gaps that motivate this study. In chapter 3 presents the theoretical background on CBF, and V-CBF, laying the foundation for understanding their role in ensuring safe autonomous navigation. Chapter 4 explores various methods for generating cost maps in the image space, a pivotal component of V-CBF. Chapter 5 takes a deep dive into the assessment of cost maps, employing comprehensive evaluation criteria drawn from custom-designed metrics inspired by relevant literature, and the design and execution of obstacle avoidance scenarios based on ISO 22737 standard within the CARLA simulator, providing a rigorous testing ground for comparing different methods for generating V-CBF. Chapter 6 documents practical details in implementing V-CBF on an industrial mobile robot (MiR), serving as a real-world experiment. Finally, chapter 7 revisits research questions, and discusses implications and contributions, concluding with insights into potential future directions in this dynamic field.

2. RELATED WORK

Safe autonomous navigation through unknown and dynamic environments poses a significant challenge in the field of robotics. Such scenarios often necessitate innovative solutions to ensure the safety and stability of autonomous robots. One of the key components to build such navigation systems is local cost maps, derived from sensor data. This chapter reviews the literature concerning the development of such cost maps in the context of safe autonomous navigation, specifically with regard to V-CBF and CBF, which are the focal points of this thesis.

2.1 End-to-End Safe Navigation Methods

End-to-end sensor data-based navigation solutions have gained prominence in the field, demonstrating their effectiveness in mapping sensor data to control commands. For example, in [35], the authors present a straightforward yet effective end-to-end approach that leverages sensor inputs, including RGB images, and point cloud data, to generate steering commands for safe navigation in challenging environments. In such end-to-end platforms, the whole neural network can be considered as a cost map, since it is mapping sensor inputs to a proper level of feature maps to produce safe control command. However, the effectiveness of these methods depends on their learning strategies and data biases. Furthermore, end-to-end methods face challenges in training convergence, as they attempt to simultaneously learn positioning, traversability maps, and desired control behavior within a single network.

2.1.1 RL-based End-to-End Safe Navigation Methods

Reinforcement learning (RL) methods have also been employed to achieve end-to-end sensor data-based navigation without relying on pre-built maps. In [50], low-dimensional range findings are used to introduce a mapless motion planner that employs Actor-Critic deep RL methods and distance-based reward functions for safe navigation. The reward functions used in this method can be seen as a form of cost function, assigning a cost based on distances to potential collisions.

Similarly, in BADGR [24], an end-to-end learning-based method is developed using various sensor inputs, such as RGB, IMU, and 2D LIDAR, to map sensor data to environmental events, such as collision, bumpiness, and robot position. These events are used to generate a reward function that encapsulates the cost of each event, resembling the cost function generation from high-dimensional sensor data.

Likewise, [18] introduced a multiscale belief generation system that extracts spatial information from first-person camera images. This spatial information, such as predictions of free space in a 2D top view of the environment, serves as a cost map used for planning and navigation in complex environments.

In [47], Conditional Generative Adversarial Networks (cGANs) are harnessed to produce depth maps from RGB images. Later on, learned maps are used in their deep RL platform to employ safe autonomous navigation. The depth maps generated from RGB images in this method can be considered as cost maps, as they reward the RL model for maintaining distance from unsafe zones.

In all the RL-based end-to-end solutions mentioned above, various formats of cost functions are extracted from fundamental features of sensor data, contributing to the reward functions used in RL. These methods share the limitations of end-to-end solutions mentioned earlier as well as the challenge of employing them in real-world scenarios.

2.1.2 Vision-Based Cost Map Approaches

In contrast to end-to-end methods, in the approaches mentioned in this section, vision-based cost maps are explicitly integrated into the problem formulation, rather than being hidden within an end-to-end solution. These approaches derive local cost maps from sensor data through learning-based techniques, primarily for navigation controllers such as Model Predictive Control (MPC).

For example, in [14], RGBD input is used to estimate traversability coefficients through methods like the nonlinear moving horizon estimator (NMHE). This traversability map is then integrated into MPC for safe navigation in real-world scenarios.

One of the most prevalent approaches to safe navigation involves the use of occupancy grid maps. For instance, in [52], stereo vision is used to obtain terrain measurements, which are employed to construct a global occupancy grid map for autonomous robots. The local information derived from this approach can be considered as a cost map that contributes to the global occupancy grid map.

In [6], vision-based costs are computed based on contrast distance feature maps and the non-uniformity of textures within images captured by a camera mounted on the robot. In their work, a Gaussian Process-based predictor is employed to model and learn the final traversability cost.

Additionally, vision-based cost maps have also been used in aiding path planning. For example, in [29], identification of unknown obstacles is achieved through camera images, particularly by generating an optical flow-based map. This optical flow-based map can be seen as another representation of cost maps.

2.2 Cost Maps and Control Barrier Functions

Cost maps, in their various formats, play a crucial role in safe autonomous navigation, aiding in different ways depending on the specific method employed. On the other hand, Control Barrier Functions (CBF) is one of the most promising control theories for safety assurance, which guarantees the remaining in the safe set. Despite CBFs advantages, the mapping of sensor inputs to cost maps suitable for these barrier functions remains a challenge, addressed in limited research papers.

In [48], Support Vector Machines (SVM) were employed to classify LiDAR data points into safe and unsafe categories, showcasing the potential of data-driven CBF generation. In this context, SVM itself can be seen as a binary cost map (obstacle map), as it classifies 3D point clouds as obstacles or non-obstacles. However, SVMs are sensitive to the choice of kernel and hyperparameters, and may not perform well with imbalanced datasets.

In [42], authors introduces an innovative approach using subspace clustering algorithms to cluster data points into different linear subregions that specify the shape of unsafe sets. These clusters can then be used as CBFs. This approach presents cost maps in the form of the learned clustered subspaces, defining the shape and position of unsafe sets. However, the limitation of this method is that it is highly environment-specific, making it challenging to generalize to unknown environments.

Some works have attempted to formulate CBF as a neural network layer. In [53], Differentiable Control Barrier Functions were introduced, encapsulating CBF formulations within a neural network layer called BarrierNet. This formulation enables CBFs to be environment-dependent and trainable. In this context, the features learned from the environment in previous network layers can be seen as a form of cost map.

Later, in [54], the same BarrierNet layer was used within an end-to-end vision-based neural network. This architecture maps sensor inputs within the network layers, essentially generating a suitable cost map for the final BarrierNet layer. However, it is important to note that this research, while bridging the gap between vision-based navigation and CBFs, suffers from disadvantages of end-to-end solutions.

2.2.1 Vision-based Control Barrier Function

Finally, V-CBF [1] is specifically designed to generate barrier functions within the image space. It receives the shape of unsafe sets from segmentation models and translates it into a cost map, adjusting it to CBFs prerequisites accordingly.

The modular approach adopted in V-CBF enables it to leverage recent advancements in foundational models and state-of-the-art segmentation techniques. Noteworthy examples include the Segment Anything Model (SAM) [28], Segment Everything Everywhere All at Once (SEEM) [61], and DINOv2 [37]. These ongoing developments in segmentation models promise high-accuracy results, even in uncharted environments. This advancement fosters the creation of generalizable and adaptable solutions for V-CBF since segmentation is the fundamental component of the approach.

The cost map generation represents another crucial aspect of V-CBF. Although the previous work introduced methods like Gaussian blur and conditional Generative Adversarial Networks (cGAN) for generating suitable cost maps, a thorough exploration and evaluation of candidate cost map methods remain relatively uncharted. This thesis addresses this gap by delving into the cost map generation module of V-CBF and evaluating its performance in various scenarios.

3. THEORETICAL BACKGROUND

This chapter lays the groundwork for the subsequent discussions within this thesis. The primary focus of this thesis revolves around cost map generation methods for Vision-based Control Barrier Functions (V-CBF). Therefore, as V-CBF is an extension of Control Barrier Functions (CBF), first an in-depth understanding of the fundamental principles of CBFs is provided. Subsequently, control theory of V-CBF is explained, with a specific focus on the role and definitions of cost maps within the V-CBF framework.

3.1 CBF Control Methodology

As CBF is a central concept referenced throughout the subsequent chapters, this section aims to provide a basis for comprehending its role in safe autonomous navigation. This theoretical groundwork will be crucial in explaining the principles behind Vision-based Control Barrier Functions (V-CBF) and the necessary specifications of V-CBF cost maps in following sections. In this section, an introductory overview of CBFs and their formal safety guarantees is provided. First, key concepts such as safe control, safe sets, and forward invariant sets are introduced. Furthermore, the principles underlying control barrier functions are presented.

To begin, consider a non-linear affine control system in the form bellow:

$$\begin{aligned}\dot{\mathbf{x}} &= f(\mathbf{x}) + g(\mathbf{x})\mathbf{u} \\ \zeta &= q(\mathbf{x}, \mathbf{m})\end{aligned}\tag{3.1}$$

which $f(\mathbf{x})$ and $g(\mathbf{x})$ are locally Lipschitz continuous functions, $\mathbf{x} \in \mathbb{X} \subseteq \mathbb{R}^n$ is state vector, and $\mathbf{u} \in \mathbb{U} \subseteq \mathbb{R}^m$ is control input. In addition, $\zeta \in \mathbb{R}^{W \times H \times C}$ is the observation from RGB-D camera where images width, height, and number of channels are denoted as W , H , C . And q is a function of states and environment variables \mathbf{m} that generates ζ .

In the system mentioned above, Lipschitz is defined as follows:

Definition 1. A function $f : D \subseteq \mathbb{R}^n \rightarrow \mathbb{R}^m$ is said to exhibit local Lipschitz continuity at any given arbitrary point $x \in D$ if there exist positive constants M and δ such that for any pair of points x and x' within a distance of δ , $\|f(x) - f(x')\| \leq M\|x - x'\|$ holds.

In this study, "safe control" means creating control inputs that ensure two key conditions: i) keeping the states of the system out of unsafe sets, and ii) staying within the physical constraints of the system, stated more formally as follows:

Definition 2. *The system (3.1) is considered safe if $\mathbf{x}(t) \in \mathbb{X}_s \subseteq \mathbb{X}$, $\mathbf{u}(t) \in \mathbb{U}_s \subseteq \mathbb{U}$, $\forall t \geq 0$, where \mathbb{X}_s and \mathbb{U}_s represent the sets of safe states and admissible control inputs, respectively.*

Let $S \subseteq \mathbb{X}_s$ be the zero-superlevel set of a smooth and continuously differentiable function $h(\mathbf{x}) : \mathbb{X} \rightarrow \mathbb{R}$ satisfying the following conditions:

$$S = \{x \in \mathbb{X} \mid h(x) \geq 0\} \quad (3.2a)$$

$$\partial S = \{x \in \mathbb{X} \mid h(x) = 0\} \quad (3.2b)$$

$$\text{Int}(S) = \{x \in \mathbb{X} \mid h(x) > 0\}, \quad (3.2c)$$

where ∂S represents the boundary of the set, and $\text{Int}(S)$ denotes the interior of set S . In order to ensure the long-term safety of system (3.1), it is necessary for the set S to be a forward invariant set.

Definition 3. *An arbitrary set $S \subseteq \mathbb{R}^n$ is considered a forward invariant set for the dynamic system $\dot{\mathbf{x}} = f(\mathbf{x})$ if for any initial condition $\mathbf{x}(0) \in S$, it satisfies $\mathbf{x}(t) \in S$, $\forall t \geq 0$ [8].*

Theorem 1. *As proven in [2], the function $h(\mathbf{x})$ qualifies as a control barrier function when certain conditions are met. Specifically, if there exists an extended class κ function $\alpha(\cdot)$ such that the following condition holds for the control system (3.1):*

$$\sup_{\mathbf{u} \in \mathbb{U}} [L_f h(\mathbf{x}) + L_g h(\mathbf{x}) \mathbf{u}] \geq -\alpha(h(\mathbf{x})), \quad (3.3)$$

where $\dot{h}(\mathbf{x}, \mathbf{u}) = L_f h(\mathbf{x}) + L_g h(\mathbf{x}) \mathbf{u}$, and L_f and L_g are the Lie derivatives. This condition ensures that the safe set S remains forward invariant, meaning that if the initial state belongs to the safe set, $\mathbf{x}(0) \in S$, the trajectories of the system will always stay within this safe set.

Definition 4. *In the context of this study, a function $\alpha(\cdot) : [0, a) \rightarrow [0, \infty)$, where $a > 0$, is considered a member of the κ class if it is a continuous function with a strictly increasing*

nature and satisfies $\alpha(0) = 0$. Additionally, a continuous function $\beta(\cdot) : \mathbb{R} \rightarrow \mathbb{R}$ belongs to the extended κ class if it is strictly increasing and $\beta(0) = 0$.

Definition 5. The notation $L_\xi \eta(x)$ is commonly used to denote the Lie derivative of $\eta(x)$ along the vector field $\xi(x)$, i.e., $L_\xi \eta(x) = \frac{\partial \eta(x)}{\partial x} \xi(x)$.

3.2 V-CBF Control Methodology

To provide a comprehensive understanding of the problem tackled in this thesis, it is essential to delve into Vision-based Control Barrier Functions (V-CBF) control methodology[1]. In this dedicated section, the concept of CBFs is defined when applied in the image coordinates. This fundamental step serves as the backbone upon which this thesis is built, as it lays the theoretical foundation essential for understanding characteristics of V-CBF.

Assuming the RGBD image ζ mentioned in (3.1), vision-based control barrier function $h(\zeta)$ is defined as following:

$$h(\zeta) = h_{img}(u, v) + h_d(d) \quad (3.4)$$

where (u, v) is pixel coordinates of ζ , and $h_{img}(u, v) : \mathbb{R}^2 \rightarrow \mathbb{R}$ is defined as a scalar function in the image coordinate. And $h_d(d) : \mathbb{R} \rightarrow \mathbb{R}$ is a strictly increasing function defined on depth domain, which d denotes the depth value at the pixel (u, v) . In this context, $h_{img}(u, v)$ is defined based on segmentation of unsafe regions in the image coordinate. And finally, $h_d(d)$ is added to it in such a way that $h(\zeta)$ satisfies the control barrier function properties mentioned in (3.2). Detailed specification of $h_{img}(u, v)$ and $h_d(d)$ is presented in subsequent sections.

With the correct design of h_d and h_{img} , the value of $h(\zeta)$ will be zero at a safe distance (d_s) to the border of unsafe region ($h_{img}(u, v) + h_d(d_s) = 0$). As a result, the necessary requirements of CBF (3.2) will be automatically fulfilled for $h(\zeta)$ with a more conservative safe set $S \subseteq \mathbb{X}_s$ defined in the 3D Euclidean workspace.

To summarize, the correlation between different states of the system and $h(\zeta)$ can be described as follows:

- When $h(\zeta) > 0$, the system is in a safe set. This indicates that either the system is within the safe set in the image space, where $h_{img}(u, v) > 0$, or the system is in an unsafe area in the image space but at a distance d from the unsafe region in Cartesian space (u, v, d) , satisfying $h_{img}(u, v) < 0$ and $d > d_s$.
- When $h(\zeta) = 0$, the system is on the border of the unsafe set. This means that the

system is either located on the boundary of the unsafe region in pixel coordinates and Cartesian coordinates, where $h_{img}(u, v) = 0$ and $d = 0$, or it is within the unsafe region in pixel coordinates but on the border of the unsafe region in Cartesian coordinates, satisfying $h_{img}(u, v) < 0$ and $d = d_s$.

- When $h(\zeta) < 0$, the system is in an unsafe set. This occurs when $h_{img}(u, v) < 0$ and $d < d_s$.

Mentioning the above properties for $h(\zeta)$, CBF requirement condition mentioned in(3.3) can be re-written as bellow:

$$\dot{h}(\zeta) = \nabla_p h_{img}^T [\dot{u}, \dot{v}]^T + \frac{\partial h_d}{\partial d} \dot{d} \geq -\alpha(h(\zeta)) \quad (3.5)$$

where in the pixel coordinate, $\nabla_p h_{img}^T$ represents the transpose of the gradient of the function $h_{img}(u, v)$. This gradient can be computed based on $h_{img}(u, v)$.

Theorem 2. *As it is proven in the previous work[1], \dot{u} , \dot{v} , and \dot{d} can be written as a function of actual control signals of the robot, linear and angular velocity, v and ω in the robot's body coordinate. This relation is computed based on simple coordinate transformation from the camera coordinate \dot{u} , \dot{v} , and \dot{d} to world coordinate and getting time derivative from the coordinate transformation. Consequently, (3.5) can be written based on control inputs.*

Therefore, having the control system (3.1), and control inputs $\mathbf{u} = [v, \omega]^T$ the control policy (3.6) will guarantee safety for any trajectory starting in the safe set S .

$$\pi_{vcbf} = \{\mathbf{u} \in \mathbb{U} \mid \nabla_p h_{img}^T [\dot{u}, \dot{v}]^T + \frac{\partial h_d}{\partial d} \dot{d} \geq -\alpha(h(\zeta))\} \quad (3.6)$$

3.3 Multiple V-CBFs

To enhance this control strategy, it is possible to consider multiple points of obstacles throughout the image and utilize multiple cameras to cover a wider range of the robot's surroundings.

In the case of employing multiple V-CBFs, $S_k = \mathbb{X} \setminus \mathbb{X}_{u_k}$ is defined as a safe set that is relevant to the system (3.1). This safe set is the complement of the k^{th} unsafe set (\mathbb{X}_{u_k}). By constructing the control barrier functions $h_k(\zeta)$ to satisfy the necessary conditions stated in (3.2) for each S_k , an overall safe set can be established by taking the intersection of individual safe sets S_k , as shown in Equation (3.7):

$$S = S_1 \cap S_2 \cap \dots \cap S_{N_o} \quad (3.7)$$

Clearly, the set S fulfills the conditions specified in (3.2) for all $k \in \{1, 2, \dots, N_o\}$, where N_o represents the number of constructed V-CBFs.

Therefore, to ensure safety, the control policy can be established as $\pi_{vcbf} = \pi_{vcbf}^1 \cap \pi_{vcbf}^2 \cap \dots \cap \pi_{vcbf}^{N_o}$.

Having provided a brief background on V-CBF control theory, the focus of subsequent chapters of the thesis is on the methods employed for generation of V-CBFs. To enhance the readability of the discussion in later sections, $h_{img}(u, v)$ is referred as the "Image-based Cost Map" and $h_d(d)$ as the "Depth-based Cost Map". The combination of these safety functions results in $h(\zeta)$, the ultimate Control Barrier Function (3.4).

3.4 Depth-based Cost Map Definition for V-CBF

In this section, a concise overview of the fundamental attributes of the depth-based cost map, referred to as $h_d(d)$, is provided.

The fundamental property of the function $h_d(d)$ is delineated as follows:

1. **Spatial Characteristics:** It must exhibit negative values within the unsafe set, positive values outside the unsafe set, and remain zero on the boundary, as articulated in (3.2).
2. **Differentiability Requirement:** The function $h_d(d)$ must be continuously differentiable, an essential condition for the construction of $h(\zeta)$ as a Control Barrier Function (CBF).

Subsequently, the depth-based cost map is defined as follows for this specific design:

$$h_d(d) = c_\beta b_\beta d^{p_\beta} + a_\beta b_\beta d, \quad (3.8)$$

where c_β , p_β , a_β and b_β are scalar values that can be fine-tuned for improved performance based on the minimum and maximum range of the depth sensor, as well as the magnitude of $h_{img}(u, v)$.

As visible in (3.8), all necessary prerequisites of the depth-based cost map is satisfied. The constructed function is non-negative for positive depth values, exhibits $h_d(0) = 0$ on the border of unsafe set, and demonstrates continuous differentiability, affirming its suitability for the intended application.

3.5 Image-based Cost Map Definition for V-CBF

Within this section, a brief summary of the core characteristics of the image-based cost map is presented. In the context of the control system defined in (3.1) and V-CBF control theory described in (3.4), the image-based cost map can be defined as a function that maps the RGBD image ζ to $h_{img}(u, v)$:

$$h_{img} = c_{\zeta} \Gamma_{\mathbf{w}}(\zeta') + a_{\zeta}, \quad (3.9)$$

where c_{ζ} and a_{ζ} are scalar values, \mathbf{w} denotes the mapping parameters, and $\zeta' \in \mathbb{R}^{W \times H \times 4}$ is the preprocessed RGB-D image ζ , and function $h_{img} \in \mathbb{R}^{W \times H}$ represent the values of function h_{img} in each pixel (u, v) .

The mapping function $\Gamma_{\mathbf{w}}$ should be designed in a way that it generates $h_{img}(\cdot)$ so that $h(\zeta)$ satisfies the CBF requirements, mentioned in (3.2). In pixel coordinates, this implies that $h_{img}(\cdot)$ should exhibit the following characteristics:

1. **Spatial Characteristics:** $h_{img}(\cdot)$ should be zero on the boundary of the unsafe set. Additionally, it should take on positive values outside the unsafe set, signifying safety, and negative values inside the unsafe set, reflecting the heightened risk, in accordance with (3.2).
2. **Differentiability Requirement:** As mentioned in (3.2), $h(\zeta)$, the CBF function, should be smooth and continuously differentiable. Consequently, $h_{img}(\cdot)$ should adhere to the same rule. This ensures that $h(\zeta)$ meets the essential requirements to be a CBF, as it is constructed from $h_{img}(\cdot)$ and $h_d(\cdot)$.

In the design phase, additional properties of $h_{img}(\cdot)$ have been considered for practical implementation, although they are not mandatory but have been defined to facilitate the design process:

1. **Gradient Preservation:** Notably, $h_{img}(\cdot)$ can be designed to exhibit increasing positivity within the safe set and increasing negativity within the unsafe set. This approach prevents vanishing gradients in $h_{img}(\cdot)$, ensuring that the gradient of $h_{img}(\cdot)$ contributes effectively to the avoidance of unsafe sets.
2. **Bounds on Values:** Furthermore, in this thesis, $h_{img}(\cdot)$ is constrained within upper and lower bounds to ensure the proper design of V-CBF. This constraint is particularly relevant when the robot is at a distance d_s from the unsafe set. Under such circumstances, $h_{img}(\cdot)$ will produce a bounded negative value, while $h_d(\cdot)$ at an arbitrary point d_s will add an enough positive value. Consequently, h_{ζ} will be zero at a conservative point d_s , representing the boundary of the unsafe sets.

The design and optimization of $h_{img}(\cdot)$ in accordance with the essential criteria mentioned above are crucial for the successful development of V-CBFs in an autonomous

navigation system. In later sections, possible methods for generating such a cost map are investigated.

4. IMAGE-BASED COST MAP GENERATION METHODS FOR V-CBF

As discussed in the previous chapters, an image-based cost map designed for V-CBF should possess specific characteristics. In this section, the theoretical background and implementation details of three candidate methods for generating image-based cost maps is explored. These three methods include Gaussian Blur, Distance Transform, and learning-based approaches, which employ a variety of techniques from image processing and machine learning to transform raw sensor data into suitable representations for safety functions.

4.1 Gaussian Blur

In this thesis, Gaussian blur is used as one method to generate image-based cost maps for V-CBF. In this section, the theory and detailed implementation of this cost map generation method is presented.

4.1.1 Theoretical Background

Gaussian blur is a widely used image filtering technique that helps to reduce noise and smooth out details in images. The Gaussian blur operation in a 2D image involves convolving the image with a Gaussian filter. The Gaussian filter is defined by the Gaussian function, which represents a bell-shaped curve. The shape of this curve is determined by two parameters: the standard deviation (σ) and the kernel size. This filter is a square matrix of size $N \times N$, where N represents the kernel size. Each element of the kernel corresponds to a weight that determines the contribution of the neighboring pixels. And the weights are calculated based on their distance from the center of the kernel, following the Gaussian distribution. The amount of blurring achieved by the Gaussian filter depends on the standard deviation parameter σ . A larger σ value leads to a wider Gaussian distribution, resulting in more blurring. [36]

To perform Gaussian blur on a 2D image, each pixel is convolved with the Gaussian kernel by computing the weighted average of its neighboring pixels. The result is a new image with reduced high-frequency components and smoother transitions between pixels.

The Gaussian blur function in two dimension can be expressed as follows:

$$f(x, y) = Ae^{-\frac{(x-x_0)^2}{2\sigma_x^2} - \frac{(y-y_0)^2}{2\sigma_y^2}}, \quad (4.1)$$

where (x, y) represents the distance from the origin of the horizontal and the vertical axis, (x_0, y_0) represents center of the kernel, σ_x and σ_y are the standard deviations of the gaussian distribution in x and y directions.

This filter is convolved to pixels of the image like other image processing filters using convolution formula bellow:

$$y[i, j] = \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} h[m, n]x[i - m, j - n], \quad (4.2)$$

where, x represents the input image matrix to be convolved with the kernel matrix h to result in a new matrix y , representing the output image. Here, the indices i and j are concerned with the image matrices while those of m and n deal with that of the kernel.

Gaussian blur finds applications in various computer vision tasks such as image preprocessing, feature extraction, and image enhancement. It helps to remove noise, eliminate high-frequency details, and create a smoother representation of the image.

4.1.2 Gaussian Blur as an Image-based Cost Map

Creating image-based cost maps with Gaussian blur involves a series of essential steps to meet the requirements outlined in Section 3.5. The following steps illustrate how Gaussian blur can be effectively used to generate image-based cost maps.

1. Instance Segmentation Mask: The process begins by obtaining an instance segmentation mask of unsafe sets from the RGB image. Utilizing state-of-the-art computer vision techniques, this mask accurately identifies regions containing obstacles or potential hazards.
2. Variance Selection: Choosing an appropriate variance value for the Gaussian filter is crucial. One effective strategy is to base this choice on the ratio of the area covered by the unsafe instance mask. This approach aligns the behavior of the Gaussian filter with the size and shape of the unsafe sets within the environment.
3. Gaussian Filtering: The next step involves the application of the Gaussian filter to the instance segmentation mask using the selected parameters. This action introduces controlled blurring to the mask, creating a spatial distribution that reflects potential hazards within the environment.

4. Rescaling and Bounding for CBF Compliance: To ensure that the blurred mask complies with the requirements of the image-based cost maps defined in Section 3.5, a rescaling and bounding operation is performed. Specifically, the values within the mask are adjusted to be zero at the boundary of the unsafe set.
5. Minimum Value Selection from blurred masks: As a final step, the minimum values among all the blurred masks for each pixel are considered. This approach ensures that V-CBF receives the most conservative values.

It is worth to emphasize that the choice of variance for the Gaussian filter should be tailored to the size and shape of the instance segmentation mask. Different masks may require different variance values to ensure a smooth gradient.

To sum up, the constructed cost map via this method is a continuously differentiable map that gradually decreases to bounded negative values inside the unsafe set, and progressively increases to bounded positive values outside the unsafe set. Fig. 4.1 illustrates an example of the Gaussian blur as an image-based cost map, given a segmented mask as an input.

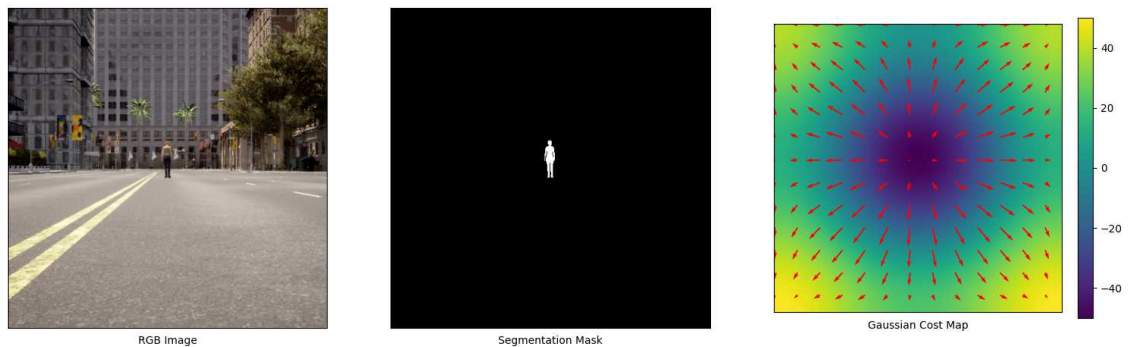


Figure 4.1. *Gaussian Blur as an Image-based Cost Map*

4.2 Distance Transform

This section investigates the theory and implementation of the distance transform method for generating the image-based cost map for V-CBF.

4.2.1 Theoretical Background

Distance transform, often referred to as a distance map or distance field, is a derived representation of a digital image. Distance fields can be categorized as either signed or unsigned, depending on whether it is necessary to distinguish whether a point is inside or outside the shape.

This transformation assigns a distance value to each pixel in the image, representing the

distance to the nearest obstacle pixel. In binary images, boundary pixels are typically regarded as obstacle pixels.

The choice of metric plays a crucial role in specifying the distance transform. Common metrics include Euclidean distance, Taxicab geometry, and Chebyshev distance [13].

Distance transforms find applications in various fields such as digital image processing (e.g., blurring effects, skeletonization), motion planning in robotics, medical image analysis for prenatal genetic testing, and pathfinding.

Several algorithms exist to compute the distance transform. Computing the exact Euclidean distance transform (EEDT) on the image grid requires special treatment [49]. Recently, an alternative approach has been proposed that utilizes a static Schrodinger equation to compute distance transforms analytically [44]. This method offers the advantage of obtaining a closed-form solution to distance transforms and computing the average distance transform over a set of distance transforms, enhancing its versatility.

Signed Distance Transform

Signed distance function (SDF), also known as the oriented distance function, represents the orthogonal distance from a given point x to the boundary of a set Ω in a metric space. The sign of the distance indicates whether x is inside or outside Ω . The function takes positive values for points inside Ω , decreases in value as x approaches the boundary (where the distance is zero), and assumes negative values outside Ω [10]. An alternative convention sometimes takes the opposite sign convention (negative inside Ω and positive outside) [31]. Efficient algorithms like the fast marching method and the fast sweeping method [60], along with the level-set method, can be employed to calculate signed distance functions [38].

If Ω is a subset of a metric space X with metric d , the signed distance function f is defined as follows:

$$f(x) = \begin{cases} d(x, \partial\Omega) & \text{if } x \in \Omega \\ -d(x, \partial\Omega) & \text{if } x \in \Omega^c \end{cases} \quad (4.3)$$

where $\partial\Omega$ denotes the boundary of set Ω , and $d(x, \partial\Omega)$ represents the minimum distance from x to points on the boundary.

Signed distance functions (SDF) find applications in various areas, including real-time rendering, computer vision, and UI frameworks. They are used for rendering smooth fonts, global illumination, and interpenetration error minimization in pixel rendering. In recent years, signed distance functions have been utilized in various domains. For example, the FOSS game engine Godot 4.0 introduced SDF-based real-time global illumi-

nation, providing a compromise between voxel-based and baked global illumination [16]. Additionally, a GPU-based UI framework called "GPUI," released in 2023, utilizes SDF extensively for drawing UI elements, including a rounded rectangle SDF, approximated Gaussian blur, and geometric primitives [43].

Properties in Euclidean Space

In the case where Ω is a subset of Euclidean space \mathbb{R}^n with a piecewise smooth boundary, the signed distance function is differentiable almost everywhere, and gradient of the function satisfies the eikonal equation ($|\nabla f| = 1$). [15]

4.2.2 Signed Distance Transform as an Image-based Cost Map

Utilizing the signed distance transform (SDF) as an image-based cost map provides a straightforward and effective approach to generate V-CBFs. This method ensures that the resulting safety function aligns with the requirements of V-CBF, making it an appealing choice.

Two key steps are sufficient to apply this method:

1. **Semantic Segmentation:** The process should be initiated by obtaining a semantic segmentation of unsafe sets from sensor data using state-of-the-art methods. This segmentation accurately identifies areas in the environment that may contain potential obstacles or hazards, laying the foundation for subsequent cost map generation.
2. **Signed Distance Transform Application:** Once the semantic segmentation is available, applying the signed distance transform on the binary segmentation mask is all that is required.

Cost maps generated via SDF automatically satisfy V-CBF prerequisites mentioned in Section 3.5, attaining a value of zero at the boundary of unsafe sets, gradually transitioning to positive values outside the unsafe set and increasing in negativity within them. Furthermore, as the maximum size of the segmented mask is limited to the size of the input image, maximum distance to the boundary of unsafe sets is also limited, which means the designed cost map is bounded. Also, SDF is a continuously differentiable function in the Euclidean space, as mentioned in Section 4.2.1. Therefore, it satisfies all the requirements for image-based cost maps.

In conclusion, combination of semantic segmentation and signed distance transform results in a robust and adaptable image-based cost map, making a significant contribution to the overall success of V-CBF. The outcome of SDF as an image-based cost map is

depicted in Fig. 4.2 when provided a segmented mask of unsafe set as an input.

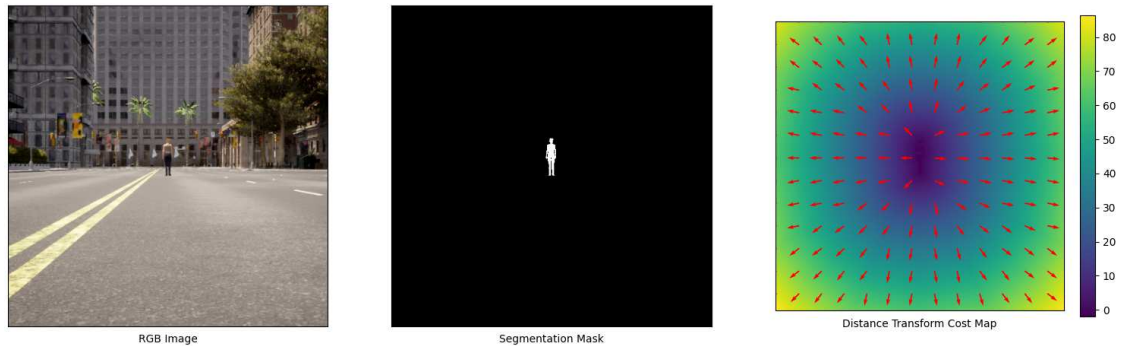


Figure 4.2. Signed Distance Transform as an Image-based Cost Map

4.3 Learning-based methods

One effective approach for generating image-based cost maps is through learning-based methods. The utilization of these methods is motivated by the flexibility and expandability offered by neural networks. Learning-based methods can also be employed to predict various parameters from one trained network. For example, the velocity and position of non-stationary unsafe sets can be derived from a single integrated framework as well as the desired cost map. Additionally, these techniques facilitate the joint learning of image-based cost maps and segmentation, eliminating the need for post-processing layers like distance transform or Gaussian blur after the neural network.

As a concrete example of learning-based methods, the application of conditional Generative Adversarial Networks (cGANs) [33] is explored in this research. In this section, the theoretical foundation of cGANs and a detailed explanation of their implementation as an image-based cost map generation method is stated.

4.3.1 Conditional Generative Adversarial Networks

Theoretical Background

Conditional Generative Adversarial Networks (cGANs) [33] are an extension of the traditional Generative Adversarial Networks (GANs) [17]. The key idea behind cGANs is to condition the image generation process on additional information, enabling control and guidance of the generated outputs.

In the standard GAN framework, there are two neural networks: the generator and the discriminator. The generator aims to create data samples that are indistinguishable from real data, while the discriminator tries to distinguish between real and generated data. These two networks engage in a minimax game, where the generator tries to improve its

ability to generate realistic data, while the discriminator strives to improve its capacity to distinguish real from generated data.

cGANs introduce conditional information, typically in the form of additional input, allowing the generator to produce data that meets specific conditions or criteria. This condition can be, for instance, a class label for image generation, a text description for image-to-text synthesis, or any other relevant information that guides the generation process. The discriminator, in turn, receives both real and generated data along with the conditional information, enabling it to evaluate the adherence of the generated data to the provided conditions more easily.

The application of cGANs extends to various fields, including image synthesis [11], image-to-image translation [22], and even style transfer [39]. Researchers have leveraged cGANs for tasks such as generating high-resolution images [34], turning sketches into colorful images [46], and creating art in a specific style [5].

The cGAN framework has opened new avenues for data generation and manipulation by allowing for the integration of conditional information into the generative process. This theoretical background lays the foundation for understanding the cGAN methodology and its application in the context of image-based cost maps for V-CBF.

cGAN as an Image-based Cost Map

To generate appropriate image-based cost map, in general, a conversion should be applied to the segmented mask to move it to a new image space in which V-CBF requirements is satisfied. This transformation can be seen as an image-to-image translation problem. Due to the capability of cGANs in image-to-image translation solutions [22], they appear as a strong choice for generating image-based cost maps for V-CBF.

cGAN can be applied in various ways: it has the capacity to simultaneously learn both segmentation and cost map generation in an end-to-end fashion, or it can be employed as a post-processing layer, focusing exclusively on cost map generation. In the context of this thesis, cGAN is utilized as a post-processing layer following semantic segmentation.

The input image provided to the network is the semantically segmented image of unsafe regions, while the output represents the image-based cost map. The ground truth dataset employs the signed distance transform cost map method, elaborated in Section 4.2. Since the ground truth for training cGAN aligns with the specified criteria for image-based cost map generation, the inferred cGAN output also satisfies the mentioned requirements, exhibiting minimal error in the form of prediction loss within the network.

The architectural design of the cGAN network is based on the pix2pix paper [22]. The detailed architecture of the cGAN remains consistent with the prior design used in the context of V-CBF [1], as illustrated in Fig. 4.3.

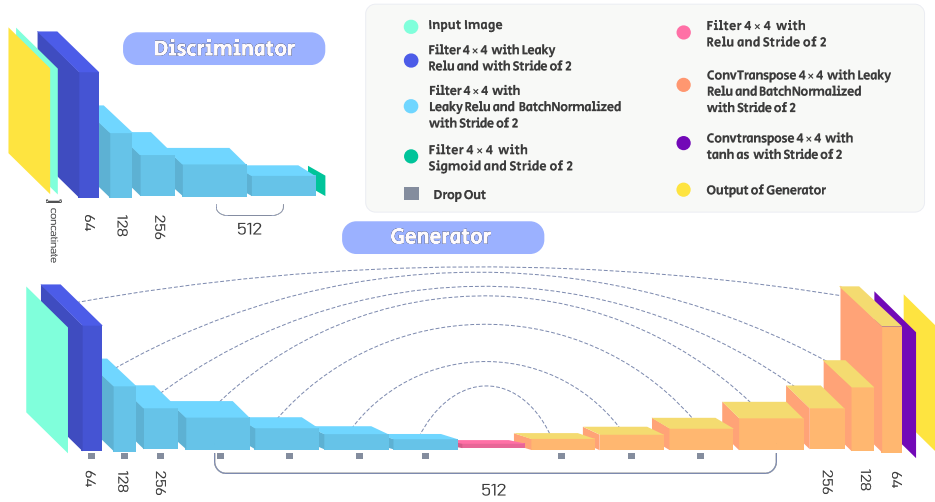


Figure 4.3. *cGAN Architecture*

To train the cGAN, the following loss function is employed:

$$\mathcal{L}_{cGAN}(G, D) = \mathbb{E}_{\zeta_i, \zeta_o} [\log D(\zeta_i, \zeta_o)] + \mathbb{E}_{\zeta_i, \xi} [\log (1 - D(\zeta_i, G(\zeta_i, \xi)))] \quad (4.4)$$

Here, ζ_i represents the input image (preprocessed RGB-D image), ζ_o denotes the output image (h_{img}), and ξ indicates the random noise vector. The generator $G : \{\zeta_i, \xi\} \rightarrow \zeta_o$ maps the observed image ζ_i and random noise vector ξ to the output image ζ_o , and D serves as the discriminator network. The objective is to train the map $\Gamma_w(\cdot)$ so that G minimizes the loss function and D maximizes it. Therefore, the final trained model can be expressed as:

$$\mathbf{G}^* = \arg \min_G \max_D \left(\mathcal{L}_{cGAN}(G, D) + \lambda_1 \mathcal{L}_h(G) + \lambda_2 \mathcal{L}_{\nabla h}(G) \right) \quad (4.5)$$

In this equation, $\mathcal{L}_h(G) = \mathbb{E}_{\zeta_i, \zeta_o, \xi} [\|\zeta_o - G(\zeta_i, \xi)\|_1]$, and $\mathcal{L}_{\nabla h}(G) = \mathbb{E}_{\zeta_i, \zeta_o, \xi} [\|\partial \zeta_o / \partial u - \partial G(\zeta_i, \xi) / \partial u\|_1 + \|\partial \zeta_o / \partial v - \partial G(\zeta_i, \xi) / \partial v\|_1]$. With this loss function, the cGAN will be trained to generate maps near the ground truth of ζ_o , $\partial \zeta_o / \partial u$, and $\partial \zeta_o / \partial v$. Furthermore, λ_1 and λ_2 represent the regularization weights.

The results obtained from the application of cGAN for image-based cost map generation is visualized in Fig. 4.4.

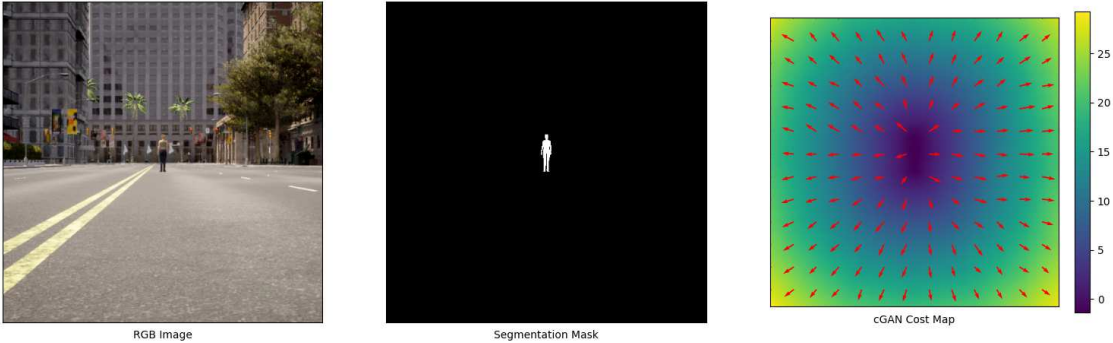


Figure 4.4. *cGAN as an Image-based Cost Map*

5. EVALUATION IN SIMULATED ENVIRONMENT

In this chapter, proposed methods for the image-based cost map generation is compared both in a quantitative and qualitative manner. The objective of this assessment is to comprehend the strengths and weaknesses of the proposed method, and to assess their influence on the controller's performance.

To perform this comparison, the candidated image-based cost maps are integrated with V-CBF in various scenarios based on ISO standard 22737 [19] and the performance of the controller is analyzed based on custom-designed metrics. To delve deeper into the topic, first ISO standard 22737 and the process of generating scenarios in CARLA simulator is introduced, followed by presenting evaluation metrics and analysis of image-based cost maps in the designed scenarios.

5.1 ISO 22737 Standard

In the context of developing automated driving systems, the establishment of standards and guidelines is imperative to ensure the safe deployment of these systems. Recently, a diverse range of standards has been introduced, such as ISO 17757 for earth-moving machinery and mining [20], and ISO 18497 for agricultural machinery and tractors [21]. One of widely used standards is ISO standard 22737 [19], which provides a guideline for assessing the performance of Low-Speed Automated Driving (LSAD) systems. LSAD systems are autonomous vehicles that are specifically designed for predefined routes, following regulations advised in ISO 22737 such as predetermined speed limits.

While ISO 22737 has been utilized for generating test scenarios and frameworks in previous works [59, 25, 58], its application for assessing the performance of state-of-the-art safe navigation controllers, such as Control Barrier Functions (CBFs), has been limited. Leveraging and testing safety controllers based on this standard is desirable since it is a well-established and widely accepted standard. Successful deployment of LSAD systems following this standard reflects their genuine safety performance and provides a common ground for comparing developments in safe autonomous navigation based on the same evaluation framework.

Therefore, the objective is to assess the performance of V-CBF and, more specifically, the impact of the proposed image-based cost maps in chapter 4 on the controller's behavior within the framework of ISO 22737.

It is important to note that the detailed guidelines for the design and evaluation of LSAD systems, as specified in the ISO 22737 standard, are not provided in this thesis due to copyright protection.

5.2 Scenario-based Test Platform based on ISO 22737 Standard

As mentioned in the previous section, ISO 22737 standard for LSAD systems has multiple aspects, including assessing safety of the system in case of hazardous situation and formulating proper scenarios for this aim. In this study, three distinct scenarios are inherited from the standard, characterized by pedestrian involvement in hazardous situations. General description of these scenarios is as followed:

1. Scenario 1: Stationary pedestrian abruptly enters the path of the ego vehicle when the vehicle approaches a distance threshold to the pedestrian. Fig. 5.1 illustrates the configuration of this scenario in CARLA schematically.
2. Scenario 2: Stationary pedestrian suddenly crosses the path of the ego vehicle when the same condition in scenario 1 is satisfied while having an occluded line of sight. Visual design of this scenario can be seen in Fig. 5.2.
3. Scenario 3: Non-stationary pedestrian walks directly in front of the ego vehicle when a certain distance threshold to the pedestrian is passed. The scenario setup in CARLA is demonstrated in Fig. 5.3.

Due to the copyright protection of this standard, specific details of scenario setups is not disclosed, including vehicle and pedestrian speeds, and exact formulations. More detailed information about the formulation of scenarios can be found in the standard document.[19]

5.2.1 Scenario Test Setup and Tools

For production of scenarios, RoadRunner software [40], CARLA [12], CARLA ScenarioRunner[9], and ASAM OpenSCENARIO [4] is used. To clarify the scenario generation process, first the mentioned tools are introduced, following the outline of scenario generation.

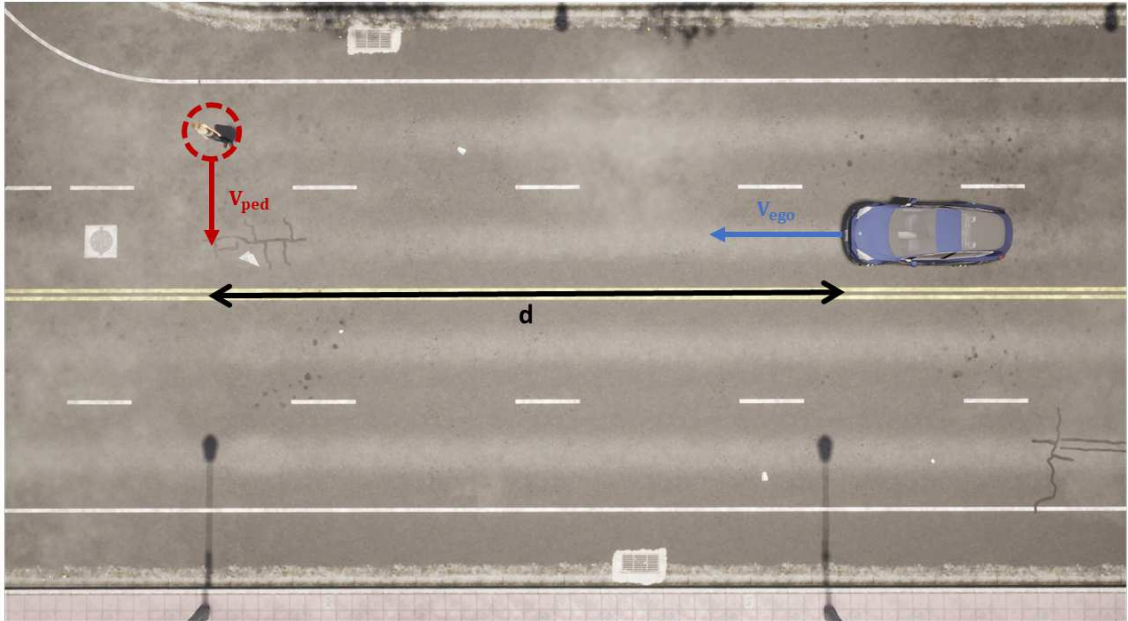


Figure 5.1. Scenario 1, derived from ISO Standard 227373 where d is the distance threshold to trigger the scenario, and V_{ped} and V_{ego} are velocities of the pedestrian and the ego vehicle.

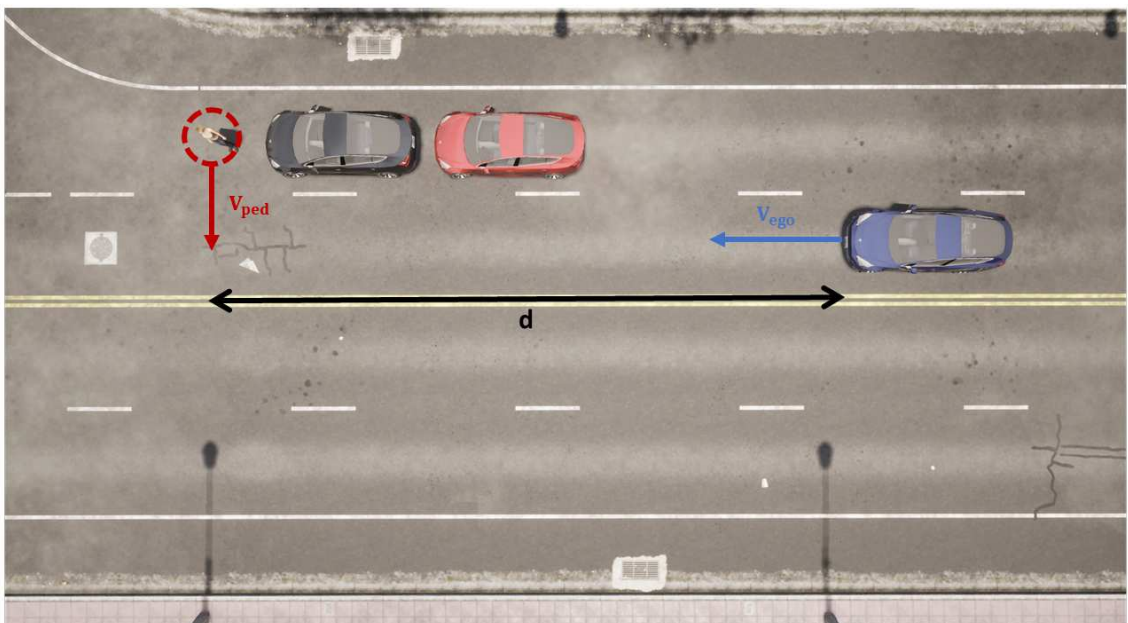


Figure 5.2. Scenario 2, derived from ISO Standard 227373 where d is the distance threshold to trigger the scenario, and V_{ped} and V_{ego} are velocities of the pedestrian and the ego vehicle.

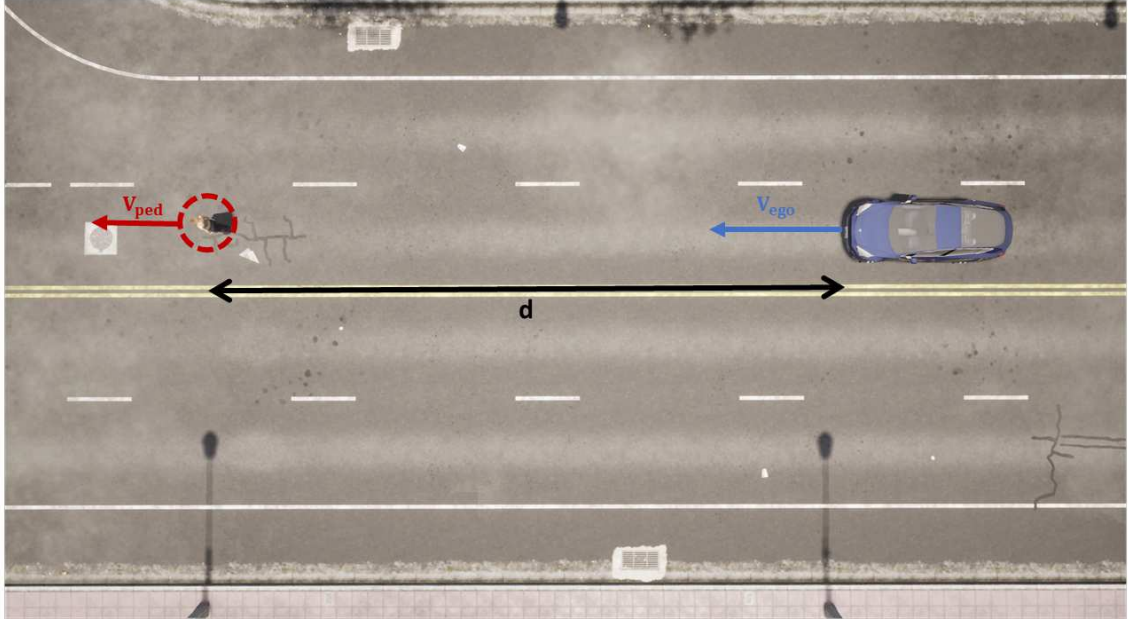


Figure 5.3. Scenario 3, derived from ISO Standard 227373 where d is the distance threshold to trigger the scenario, and V_{ped} and V_{ego} are velocities of the pedestrian and the ego vehicle.

CARLA

CARLA [12] is an open-source simulation platform for autonomous driving research and development, particularly in urban settings. It is built on top of Unreal Engine [51], offering realistic graphics and a wide range of sensor models. CARLA employs a Remote Procedure Call (RPC) protocol, enabling multiple clients on less powerful devices to connect to a central server in a distributed manner.

The platform is extendable and integrable, allowing users to create custom environments, sensors, and vehicles. Moreover, CARLA seamlessly integrates with popular robotics tools including Robot Operating Systems (ROS, and ROS2) [41], ASAM OpenSCENARIO [4] and OpenDRIVE [3], and CarSIM [7], making it a versatile and invaluable resource for academic research in autonomous driving.

ASAM OpenSCenario

ASAM OpenSCENARIO [4] introduces a file format for depicting the dynamic components of driving and traffic simulators. It primarily serves to define intricate, synchronized maneuvers involving multiple entities like vehicles, pedestrians, and other participants in traffic scenarios. These maneuvers can be based on either driver actions, such as lane changes, or on trajectories derived from recorded driving patterns. This standard also encompasses supplementary details, including descriptions of the ego vehicle, driver attributes, pedestrians, traffic, and environmental conditions.

RoadRunner

RoadRunner [40] is an interactive editor that allows you to create and customize 3D scenes for testing automated driving systems. In this platform, specific road layouts with signs, signals, guardrails can be designed and integrated to CARLA. RoadRunner also provides tools to adjust traffic signal timing and vehicle paths at intersections, as well as scenario generation capabilities, making it a useful tool for academic research on autonomous vehicles.

ScenarioRunner

ScenarioRunner [9] is a module within the CARLA simulator that facilitates the creation and execution of traffic scenarios. These scenarios can be specified either through a python interface or by adhering to the OpenSCENARIO standard, which can be exported from RoadRunner.

5.2.2 Scenario Generation Process

The process of scenario creation involves multiple steps across various software platforms. In this section, a comprehensive description of the essential steps required to prepare scenarios within the CARLA simulator is provided. As illustrated in Fig. 5.4, the systematic approach to generate scenarios in compliance with ISO 22737 within the CARLA simulator can be summarized as follows:

1. **Scenario Creation:** Scenarios were precisely crafted employing MathWorks RoadRunner software, taking into account the characteristics of the ego vehicle, pedestrian speed, and their relative spatial positions according to the formulation of the ISO standard.
2. **Scenario Export:** The scenarios were exported in the ASAM OpenSCENARIO format directly from the RoadRunner software, preserving the details of each test case.
3. **Integration with CARLA:** The exported OpenSCENARIO files were executed in CARLA using CARLA ScenarioRunner module.
4. **V-CBF Controller Assignment:** The V-CBF controller was designated as the external controller for the ego vehicle in the scenario, accompanied by the incorporation of the desired cost map.

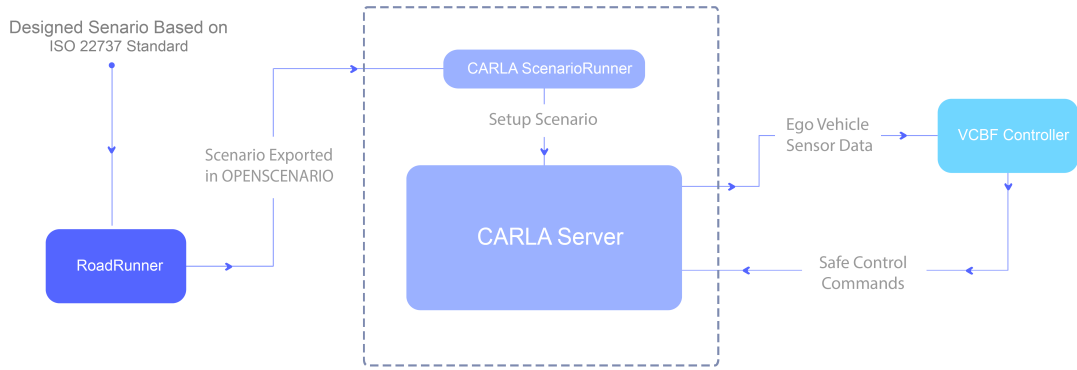


Figure 5.4. Scenario Generation Process

5.2.3 V-CBF Hyperparameters and Configuration in Scenarios

For generating test scenarios described in Section 5.2, CARLA simulator [12] is chosen as the simulation platform. The primary focus is to evaluate the performance of image-based cost maps within the V-CBF framework. Consequently, ground truth semantic and instance segmentation data from CARLA is used to remove the impact of the potential prediction model for segmentation on the final V-CBF performance in all scenarios, for all the candidate cost map methods. Additionally, V-CBF hyperparameters are fixed during scenario-based tests for the same reason.

To be more specific, for all the Image-based cost map methods, cost maps with a size of 128x128 pixels are selected, with value of $c_\zeta = 0.015$ and $a_\zeta = 0$ for the specified hyperparameters in the formulation mentioned in (3.9). For the depth-based cost map, the formulation mentioned in (3.8) is utilized, where $c_\beta = 0.015$, $p_\beta = 2$, $a_\beta = 0.025$, and $b_\beta = 1$. The designed hyperparameters ensure that the range of values in the image-based cost map is in a balanced correlation with the range of values in the depth-based cost map. Furthermore, $\alpha = 0.9$ in (3.6).

Additionally, similar to the previous work [1], Control Lyapunov Function (CLF) is integrated with V-CBF to ensure both the stability and safety of the system. For the test scenarios, the heading target is set to $-\frac{\pi}{2}$ as the stability goal for CLF, while the objective of V-CBF is to avoid defined unsafe sets. Consequently, the desired behavior of the robot is to navigate in the environment while maintaining minimal error from the nominal heading as the stability target and avoiding pedestrians as obstacles. The detailed formulation for the integration of CLF and V-CBF can be found in the previous work [1].

5.3 Evaluation Metrics

The scenarios designed in the previous section are valuable for analyzing the safety controller. However, they alone may not provide a comprehensive evaluation, as they primarily focus on meeting specific criteria without delving into the details of how these criteria are fulfilled. Furthermore, despite the significant progress in research and development related to safety assurance and obstacle avoidance controllers, there is a lack of widely agreed, uniform metrics for comparing the performance of various methods.

In this section, a set of custom-designed metrics is introduced to the community. The designed metrics are aimed to assess the performance of safety assurance controllers quantitatively.

List of the selected evaluation metrics is as followed:

1. **Distance to Collide (DTC):** DTC is a useful metric for analyzing safety controllers, providing insight into the behavior of the safety controller. It has also been used in [30] to assess the performance of their controller in dynamic obstacle avoidance scenarios.
2. **Deviation from Target Tracking:** Stability and safety are crucial aspects of control in robotics, and autonomous systems must balance safety considerations (such as obstacle avoidance) with stability concerns (e.g., reaching the goal). There often exists a trade-off where safety controllers may relax stability constraints to satisfy safety requirements. Analyzing the degree to which safety constraints impact stability controllers provides insights into the performance of the safety controller.
3. **Rate of Deviation from Target Tracking:** To expand the scope of analysis, the derivative of the deviation from the stability target is also considered to assess the intensity of deviations.
4. **Acceleration of Ego Vehicle:** This metric helps in evaluating the smoothness of velocity commands generated by the controller.
5. **Accumulated Duration in High-Risk Zone:** As emphasized in [55], the time spent in high-risk zones is a critical factor to consider, indicating the overall time the controller spent close to the unsafe sets.

5.4 Evaluation and Result Analysis

In this section, impact of the introduced image-based cost maps on the final performance of V-CBF is analyzed based on the designed metrics. During the experiments, V-CBF utilized consistent hyperparameters mentioned in Section 5.2.3, while the generated cost maps varied. Finally, the candidated cost maps are applied to all three scenarios described in Section 5.2 while capturing the metric values.

Metrics	Gaussian	SDT	cGAN
Average DTC	21.44	21.17	21.29
Total Deviation from target tracking	15.93	16.08	17.75
Max Rate of Deviation from target tracking	0.824	0.851	0.859
Max Acceleration	0.36	5.03	0.40
Accumulated Duration in High-Risk Zone	0.65	0.35	0.3

Table 5.1. Summary of Metrics Evaluation in Scenario 1

5.4.1 Analysis of Scenario 1

Fig. 5.5 depicts the results of custom-designed metric measurements on three candidate image-based cost maps while V-CBF controlled the ego vehicle in scenario 1. Furthermore, to improve the analysis, summarized information of metric values is visualized in Table 5.1.

As visible in Table 5.1 and Fig. 5.5, all three methods successfully avoided the obstacle and moved to an obstacle-free path. The V-CBF controller, which used a Gaussian cost map, had the lowest "average Distance to Collide (DTC)" and "total Deviation from target tracking". The controller also exhibited smooth behavior with respect to acceleration. However, it had non-smooth behavior concerning the "rate of deviation from target tracking". Additionally, the controller spent the longest time in the High-Risk zone compared to the other two methods (0.65 seconds).

On the other hand, the V-CBF integrated with the Signed Distance Transform (SDT) exhibited smoother behavior in "rate of deviation from target tracking" and finished the scenario successfully in a faster pace compared to other methods. However, the distance transform exhibited more oscillations in acceleration as well as higher acceleration values, indicating that the controller was attempting to stay within the safe set by adjusting linear velocity rather than angular velocity.

Finally, the V-CBF that utilized cGAN had behavior similar to the controller with SDT since it was using SDT as ground truth data while training. Additionally, this scenario showcased that learning-based methods can also have stable performance, similar to image processing methods. For example, cGAN had the lowest "accumulated duration in the High-Risk zone" compared to other methods.

5.4.2 Analysis of Scenario 2

In Fig. 5.6, the outcomes of custom-designed metric measurements in scenario 2 are visualized, while the V-CBF with different image-based cost map methods controlled the ego vehicle. Furthermore, for enhanced analysis, a concise summary of metric values

Metrics	Gaussian	SDT	cGAN
Average DTC	17.31	17.82	20.53
Total Deviation from target tracking	23.98	20.59	21.67
Max Rate of Deviation from target tracking	0.91	0.77	0.86
Max Acceleration	0.15	0.71	0.19
Accumulated Duration in High-Risk Zone	0.5	0	0.75

Table 5.2. Summary of Metrics Evaluation in Scenario 2

Metrics	Gaussian	SDT	cGAN
Average DTC	11.34	9.97	7.79
Total Deviation from target tracking	22.53	22.64	9.91
Max Rate of Deviation from target tracking	0.70	0.65	0.36
Max Acceleration	3.06	6.69	1.46
Accumulated Duration in High-Risk Zone	0	0	0

Table 5.3. Summary of Metrics Evaluation in Scenario 3

can be found in Table 5.2.

It is evident from Table 5.2 and Fig. 5.6 that all three methods effectively avoided the obstacle, moving to an obstacle-free path to follow their target. In particular, the controller integrated with SDT had the lowest "total deviation from the target tracking" as well as the lowest "maximum rate of deviation from the target tracking". This indicates that SDT successfully avoided the obstacle with minimal deviation from the target. Additionally, same as previous scenario, the controller utilizing SDT successfully avoided the obstacle faster and did not enter the high-risk zone during the whole scenario. However, similar to the previous scenario, this controller suffers from a high rate of change in acceleration values.

Moreover, in this scenario, V-CBF with Gaussian and cGAN exhibits very similar behavior regarding the "maximum rate of deviation from target tracking." However, the controller with a Gaussian cost map avoided the obstacle with stronger commands, resulting in less time spent in the high-risk zone compared to cGAN. Due to this behavior, it had more "total deviation from target tracking" compared to the integrated controller with cGAN. Additionally, similar to the previous scenario, the non-smoothness of behavior in the "rate of deviation from target tracking" is evident in the metrics measurements of V-CBF designed using Gaussian cost maps.

5.4.3 Analysis of Scenario 3

In this section, the performance of the V-CBF controller utilizing the designed cost maps in scenario 3 is investigated.

As illustrated in Fig. 5.7 and summarized in Table 5.3, the controller incorporated with cGAN exhibits different behavior compared to the one employing SDT and Gaussian. The controller employed with cGAN maintains a safe distance from obstacles while following the target, without moving to an obstacle-free path. Therefore, the scenario takes longer as the controller reduces the linear velocity throughout the scenario to maintain a safe distance. This results in significantly lower values for "total deviation from target tracking," "maximum rate of deviation from target tracking," and "average distance to collision" compared to the other two methods. However, "deviation from target tracking" is non-smooth and exhibits more oscillations in the behavior of the controller.

On the other hand, SDT and Gaussian both avoid the obstacle and return to tracking their target after obstacle avoidance. However, similar to previous scenarios, SDT avoids the obstacle and completes the scenario more quickly than the other methods, albeit with higher maximum acceleration and acceleration changes.

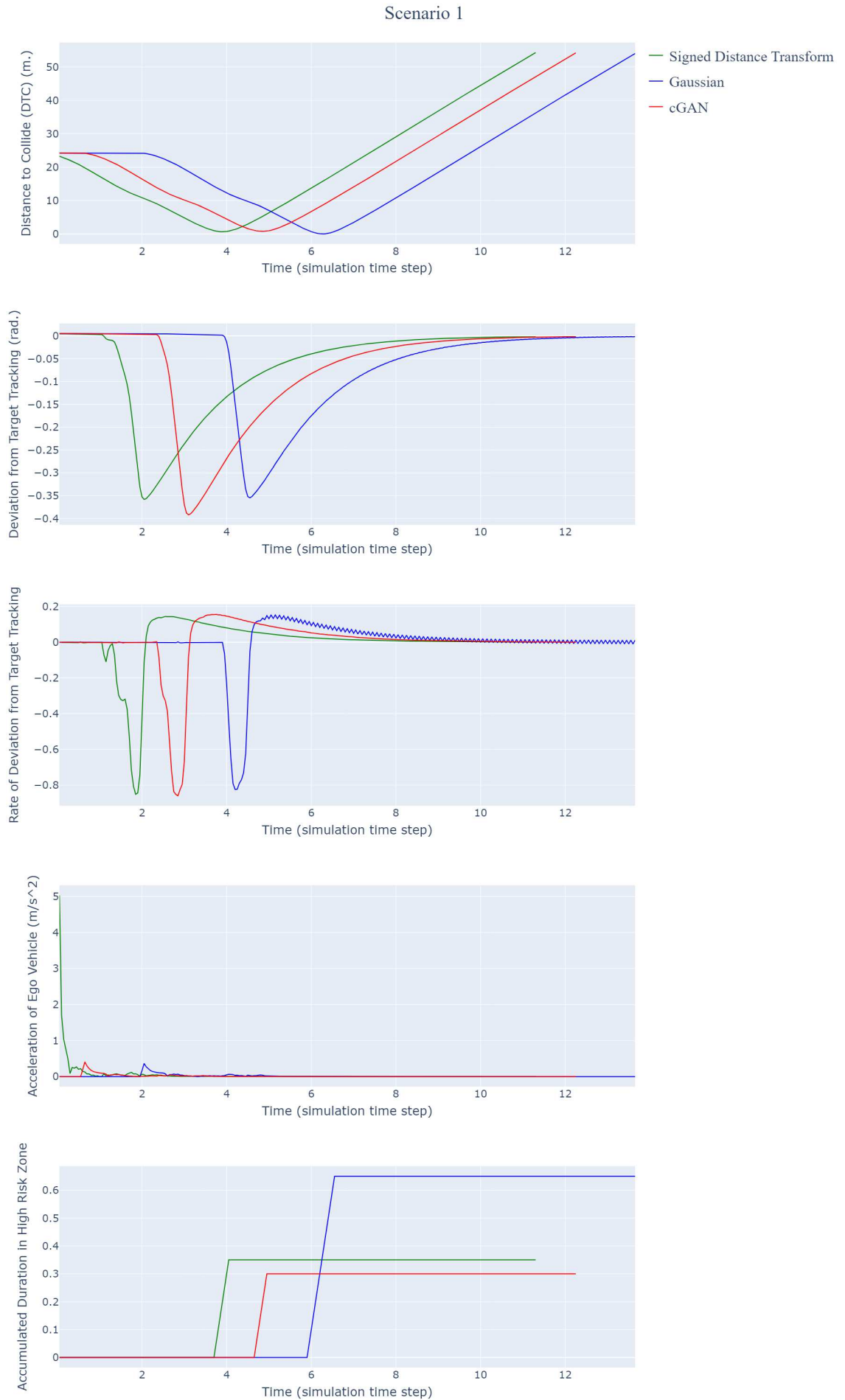


Figure 5.5. Metrics Analysis on Scenario 1

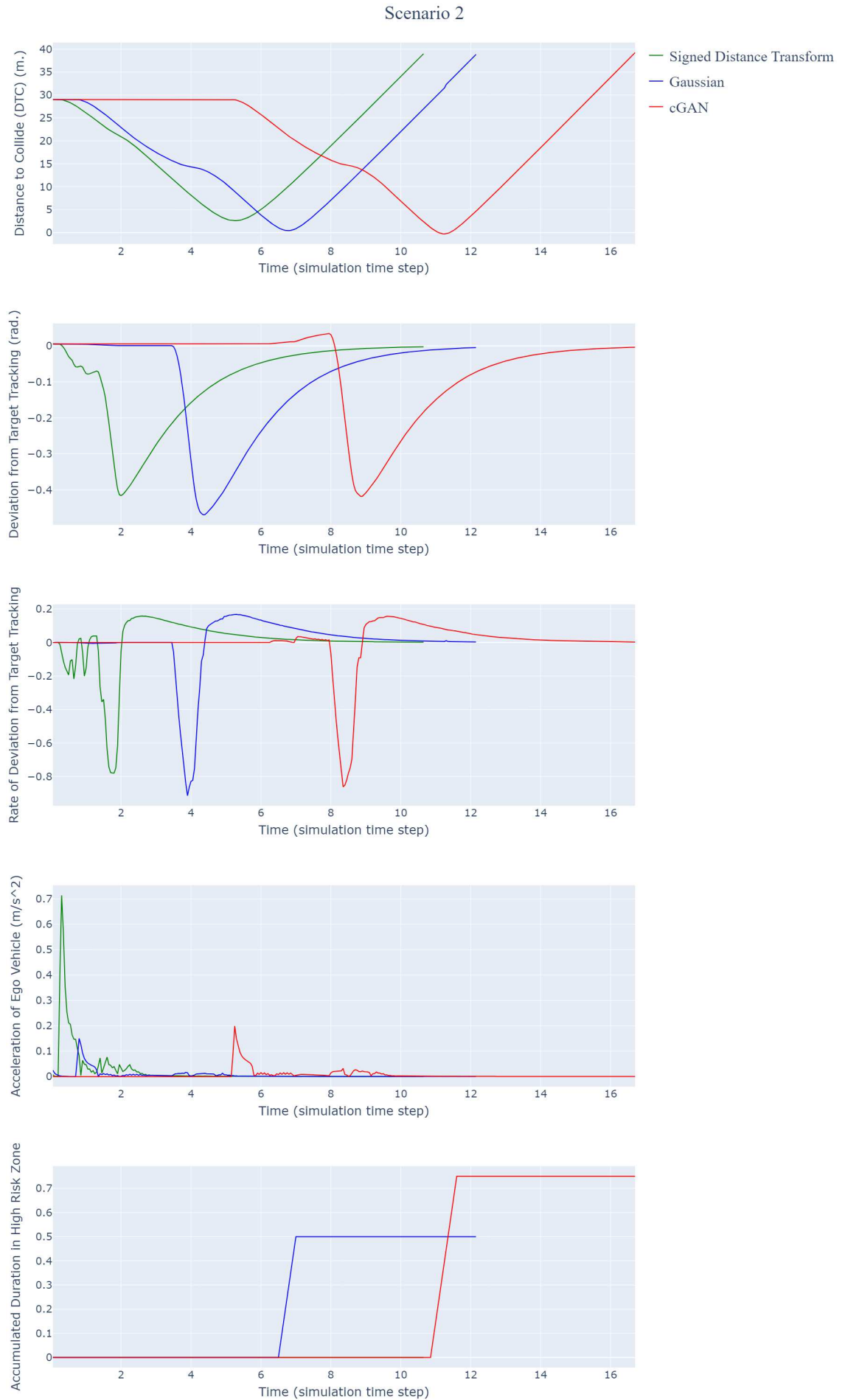


Figure 5.6. Metrics Analysis on Scenario 2

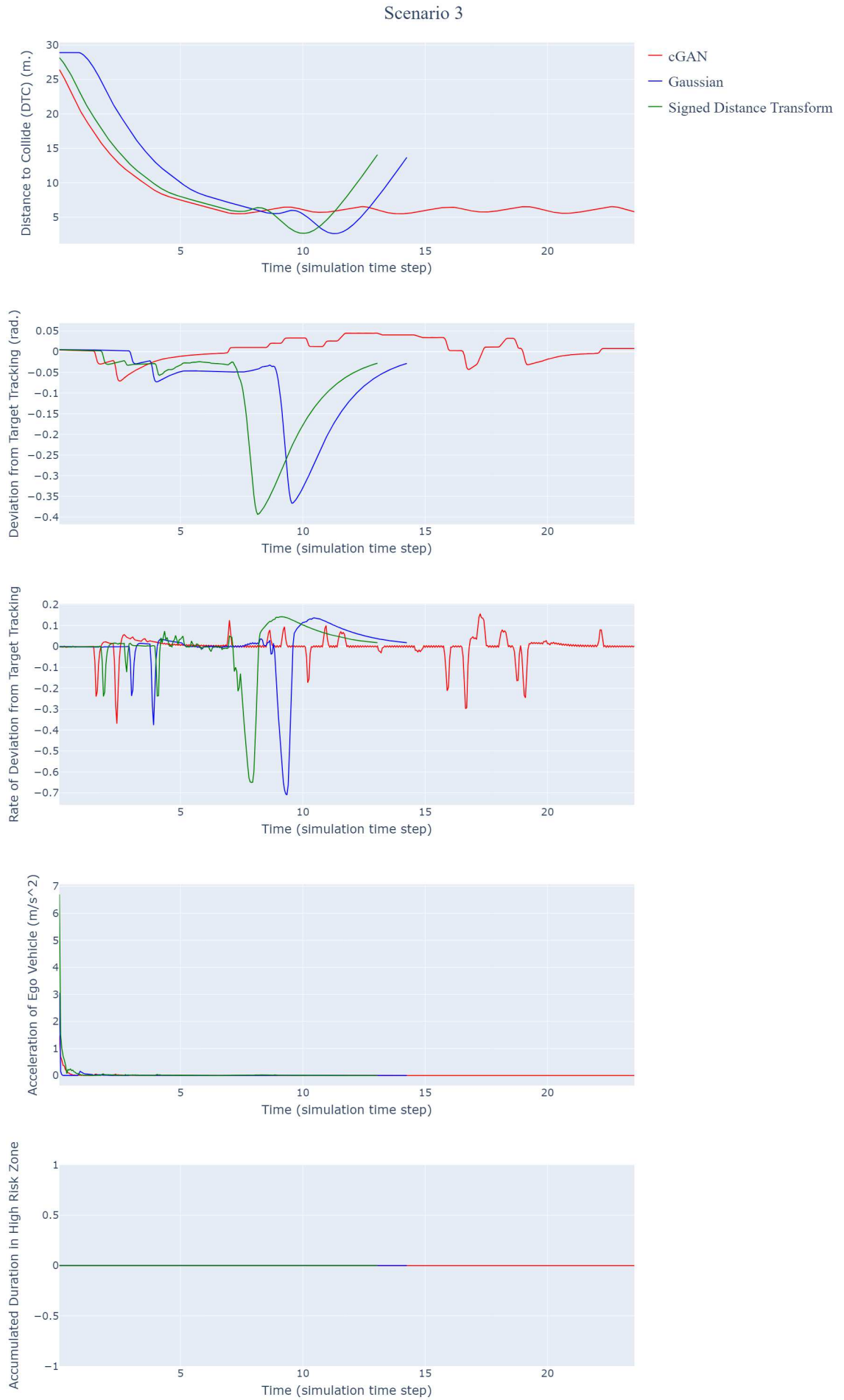


Figure 5.7. Metrics Analysis on Scenario 3

6. V-CBF VALIDATION ON INDUSTRIAL MOBILE ROBOT

Implementing safety assurance controllers in simulators provides valuable insights, but it is vital to recognize the limitations of simulations. For instance, sensor models and physics simulations, while improving, may not always accurately replicate complex real-world phenomena like motion blur in RGB cameras or accurate friction simulation. Despite continuous improvements in simulators, a noticeable gap still exists between the simulated environment and the real world. Because of this disparity, relying solely on controller verification within simulations is insufficient.

To address the mentioned gap, this chapter explores the implementation of V-CBF on a physical robot to assess the real-world effectiveness of it. The implemented experiment aims to bridge the gap between simulation and reality, ensuring that the performance of V-CBF is validated in practical, real-world scenarios. The chapter begins by explaining the hardware and software setup of the experiment, providing the foundation for the real-world testing. Subsequently, results are presented alongside a comprehensive analysis of the performance of V-CBF on one obstacle avoidance scenario.

6.1 Experiment Setup

In this section, a suitable configuration for deploying V-CBF on a real robot is presented, addressing both the hardware and software aspects. Providing such information establishes a comprehensive understanding of the practical setup, in which V-CBF demonstrates promising results.

6.1.1 Hardware Setup

As a hardware platform, MiR [32], an industrial mobile robot suitable for indoor environments is utilized for the experiments, shown in Fig. 6.1. To deploy test scenarios, the built-in mapping capability and automatic safe maneuver functions of the robot are deactivated. And an NVIDIA Jetson AGX Orin [23] is configured and connected to MiR via an onboard Ethernet connection, on which V-CBF is deployed. The V-CBF controller is developed in ROS2 [41] platform and commands generated by the controller are trans-

mitted from Jetson to MiR as manual control commands through a ROS2 interface and WebSocket protocol.

The sensor setup utilized for the experiments involves a Microsoft Azure Kinect DK camera [26] attached on MiR. This camera features a 12-MP RGB sensor aligned with a depth stream. The Time-of-Flight (ToF) depth camera of Kinect operates in a range from 0.5 to 5.46 meters, with a Nominal Field of View (NFOV) in binned mode. Additionally, the final RGB-D images are configured with the resolution of 1920x1080 pixels.



Figure 6.1. Photo of the MiR robot [32] utilized in the experiment

6.1.2 Software Setup

All software components involved in the experiment including controllers and sensor data streaming modules are based on ROS2 [41], enabling modular and reliable software stack. The main software elements includes the Azure Kinect ROS driver [27], which enables the flow of sensor data within the ROS2 platform. For object segmentation, YoloV8 [57] is integrated with ROS2, with a specific focus on segmenting humans as obstacles. To avoid complexities of non-convex mask shapes, instead of segmentation masks, bounding box masks are used. An illustrative example of images captured by the camera, along with their corresponding segmentation masks is depicted in Fig. 6.3.

In summary, the software configuration for the experiments entails the development of V-CBF and all related modules within the ROS2 platform. The controller operates by receiving sensor data and segmentation masks from the Azure Kinect sensor and the YOLOv8 segmentation model. Subsequently, V-CBF publishes the relevant linear and angular velocity commands to control the MiR robot.

6.2 Goal of the Experiment

During the experiments, same as the previous work [1], Control Lyapunov Function (CLF) is integrated with V-CBF to ensure both the stability and safety of the system. The current heading of the robot serves as the stability goal for CLF, while the objective of V-CBF is to avoid humans, selected as obstacles to illustrate the system's capabilities. Consequently, the desired behaviour of the robot is to navigate in the environment while maintaining the current heading as the stability target, and avoiding humans as obstacles. The detailed formulation for the integration of CLF and V-CBF can be found in the previous work [1].

6.3 V-CBF Hyperparameters and Configuration in the Experiment

For the designed experiments on MiR, signed distance transform is used as the image-based cost map generation method, described in Section 4.2. This choice is motivated by the modularity and portability of this cost map. Signed distance transform image-based cost map can be implemented as a post-processing function after the segmentation. Additionally, no re-tuning is needed in case of transitioning the robot to different environments, enabling a straightforward implementation of V-CBF in unknown environments.

To balance the values of image-based cost map and depth-based cost map, signed distance transform is applied on a cropped section of images with the size of 64x64 pixels, multiplied by a scaling hyperparameter $c_\zeta = 0.74$ and added by a shifting hyperparameter $a_\zeta = 8.2$, according to the formulation in (3.9). Also the formulation described in (3.8) is applied for depth-based cost maps, with parameter values $c_\beta = 6.47$, $p_\beta = 2$, $a_\beta = -1$, and $b_\beta = 0.6$. These chosen hyperparameters are designed to ensure a balanced correlation between the range of values in the image-based cost map and the depth-based cost map. Additionally, the value of α in (3.6) is set to 0.3.

6.4 Results and Performance Analysis

This section presents a comprehensive analysis of one iteration of the obstacle avoidance scenario, capturing all pertinent internal variables associated with the controller to offer in-depth insights into the system's behavior during obstacle avoidance.

In Fig. 6.2, a velocity-time plot is presented, depicting the MiR robot's response when encountering an obstacle. As indicated in (3.4), the image-based cost map denoted as $h_{img}(u, v)$, and the depth-based cost map denoted as $h_d(d)$ jointly contribute to the construction of the final control barrier function, denoted as $h(\zeta)$. In Fig.6.2, the values of these three functions are plotted for the closest obstacle point to the robot during the experiment.

As observed in the figure, for the initial 8 seconds of the experiment, $h_{img}(u, v)$, and consequently, $h(\zeta)$, exhibit a decreasing trend, transitioning from positive values to zero and eventually to negative values. This trend signifies that the robot was approaching an unsafe set. Around the 7-second mark, as $h(\zeta)$ crossed from positive to negative values, the V-CBF controller increased the robot's angular velocity to steer away from the obstacle while simultaneously reducing its linear velocity, causing the robot to nearly come to a halt due to its proximity to the unsafe set. After successfully avoiding the obstacle, the angular velocity gradually returned to zero, allowing the robot to continue moving in its current direction as directed by the stability target within the Control Lyapunov Function (CLF). Once the robot entered a safe set after 8.5 seconds, $h_{img}(u, v)$, $h_d(d)$, and $h(\zeta)$ gradually increased to reach significantly positive values, indicating that the robot was safely situated, and there were no nearby unsafe sets.

Figure 6.3 provides visual documentation of RGB images, depth images, segmentation masks, and the image-based cost maps generated during the experiment at the specified timestamps. These visual representations clearly illustrate the robot's avoidance behavior, demonstrating its successful evasion of the unsafe set around the 7-second mark.

In terms of system performance, the segmentation process operates at an impressive rate of 0.02 seconds per image, while the entire V-CBF controller takes approximately 0.08 seconds per image to compute linear and angular velocities. These timing characteristics play a crucial role in emphasizing the real-world applicability of V-CBF on the MiR robot.

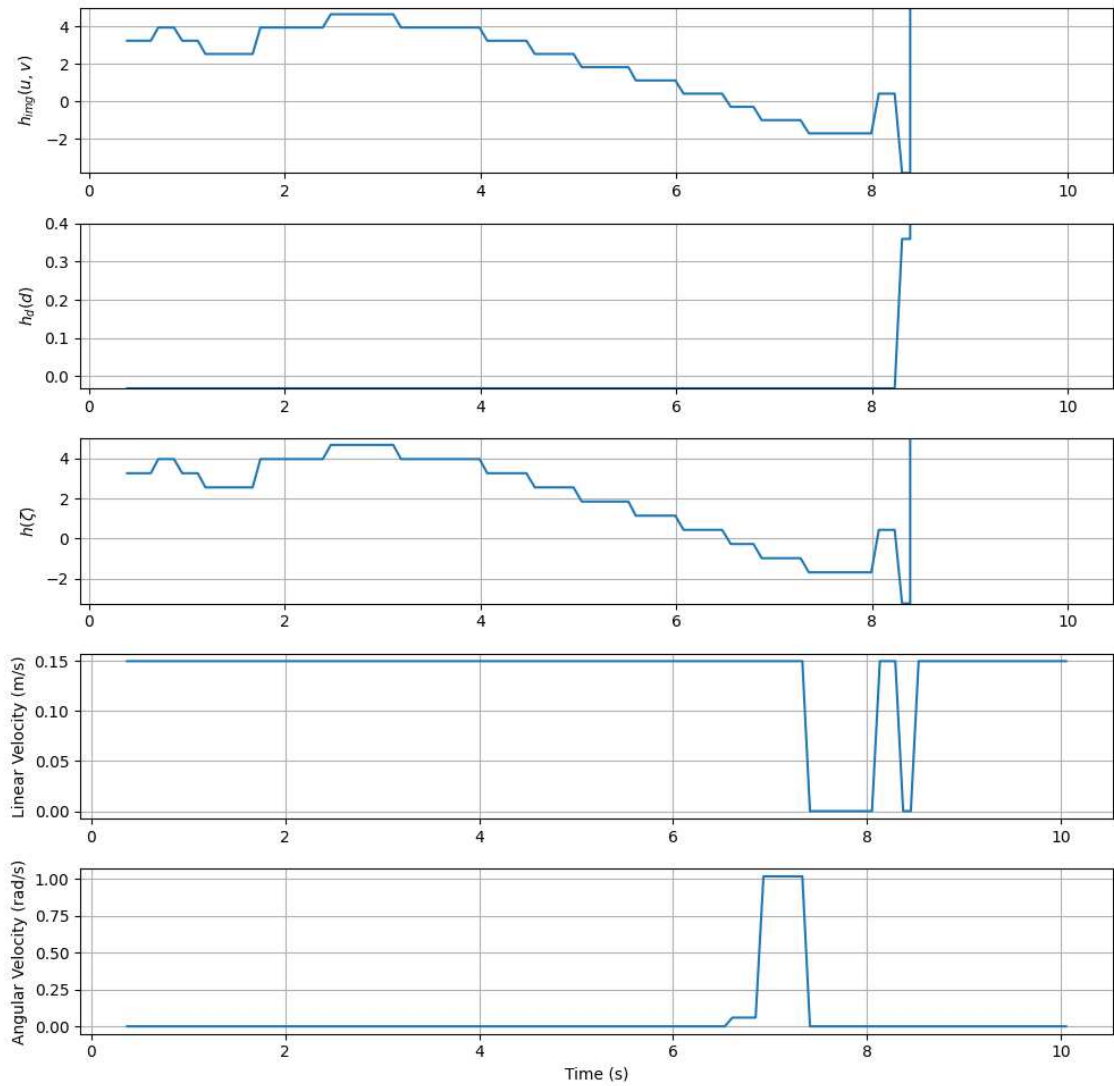


Figure 6.2. Image-based cost value (first from top), depth-based cost value (second from top), and V-CBF value (third from top) for the closest obstacle point vs time, and robot's linear (forth from top) and angular velocity commands (fifth from top) vs time

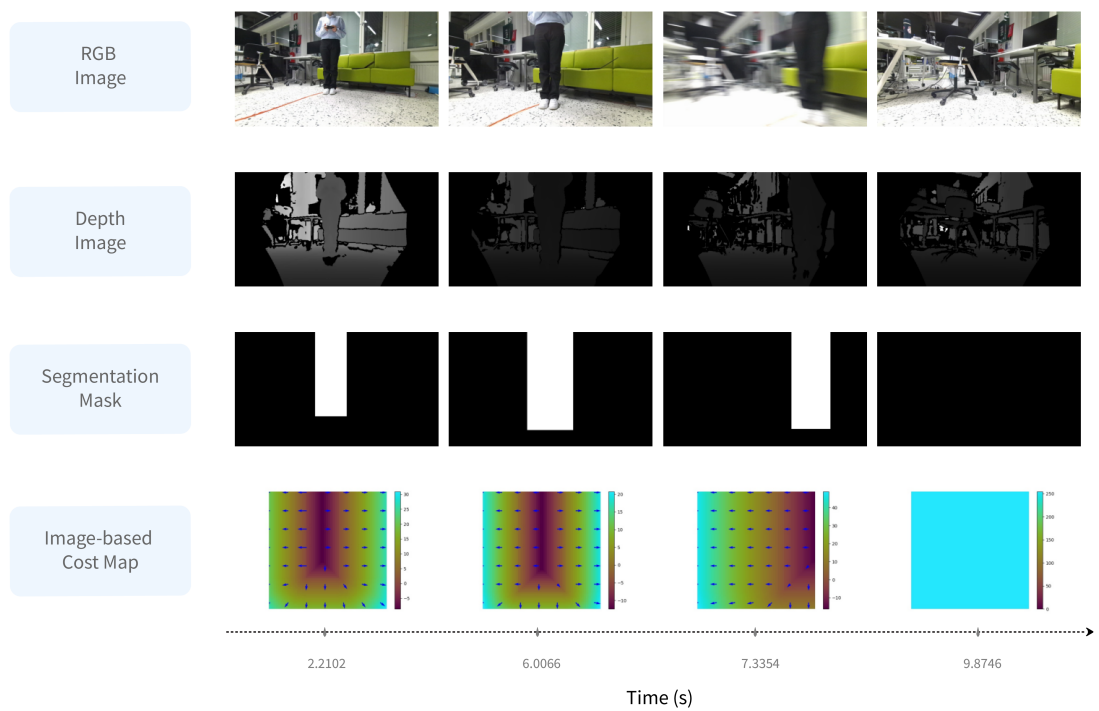


Figure 6.3. Robot's view while avoiding a human as an obstacle using V-CBF

7. DISCUSSION AND CONCLUSION

This master's thesis has comprehensively investigated a novel obstacle avoidance method, V-CBF (Vision-based Control Barrier Function), which was introduced in previous work. The research aimed to address three main questions, as outlined in Section 1.2.

The first question sought to identify suitable methods for defining control barrier functions in the image space based on RGB-D images. To tackle this question, a literature review was conducted in Chapter 2, focusing on state-of-the-art methodologies for defining local cost maps for safe navigation controllers using sensor input, such as RGB-D cameras. Chapter 3 provided a detailed exploration of control barrier functions (CBFs) and vision-based control barrier functions (V-CBF), elucidating the key characteristics required for V-CBF. It was established that V-CBF could be divided into image-based and depth-based cost maps, each with specific characteristics. A design for depth-based cost maps was proposed. Later on, chapter 4 introduced three methods based on image processing and machine learning, including Gaussian blur, signed distance transform (SDT), and cGAN, as examples of learning-based methods.

The second question involved a comparative evaluation of candidate V-CBF generation methods, drawing from the ISO 22737 standard and custom-designed metrics in the CARLA simulator. Chapter 5 introduced ISO standard 22737 and highlighted its advantages in forming the basis for evaluating state-of-the-art safe navigation controllers. Moreover, custom metrics were designed to extend the test framework quantitatively. In this chapter, a scenario-based and metric-based test framework was developed in the CARLA simulator, aligning with the ISO 22737 standard, and was used to evaluate the candidate cost map methods. The results in Chapter 5 demonstrated that, with the specified cost map configuration and V-CBF setup, all three proposed methods enabled safe autonomous navigation in the test framework. Detailed quantitative metrics results for the cost map methods in various scenarios were presented.

In summary, the comparison revealed that the SDT method exhibited faster success in completing scenarios and maintained consistency across all scenarios. The V-CBF controller integrated with SDT demonstrated smooth behavior with respect to the target goal, higher acceleration values, and changes compared to the other methods. However, it is important to note that the current implementation of SDT had limitations, as the bound

of the cost map was dependent on its size. Additionally, SDT has same value of gradients all over the cost map. Although this does not have any conflict with the criteria of image-based cost maps for V-CBF, the presence of same constant gradients in safe and unsafe sets is not desirable. To address these limitations, a new distance metric for SDT calculation, involving non-linearity, can be designed in future work.

On the other hand, the V-CBF integrated with Gaussian image-based cost map exhibited undesired oscillations in the rate of deviation from the target tracking in all scenarios in the current setup. As Gaussian blur performance is tunable, this issue can be investigated in future work with tuning the Gaussian blur cost map with respect to distance to obstacle and bound of depth-based cost map to improve its behaviour.

Furthermore, the results of using cGAN as an image-based cost map for V-CBF indicated that learning-based methods have the potential for stable integration with safety assurance controllers. Comparative results demonstrated that cGAN was on par with the SDT and Gaussian, indicating its effectiveness as an image-based cost map method. Learning-based methods can also offer the advantage of integrating segmentation and cost map generation within a single network, eliminating the need for post-processing methods.

It is worth to mention that order of complexity and time of complexity of cost maps may effect in the selection of suitable cost map, as well as the selected hardware platforms. For example, both SDT and Gaussian methods can be implemented in various ways. To elaborate more, SDT can be estimated with acceptable error via methods such as fast marching [45], quantized fast marching [56], and fast sweeping [60] with varied order of complexity from $O(n)$ to $O(n \log n)$, where $n \times n$ is size of image. Also, Gaussian blur can be implemented with various techniques such as converting it to 1D convolution. As a result, with different implementation of Gaussian blur, the order of complexity of it may vary from $O(nr^2)$ to $O(n)$, where $n \times n$ is size of image and $r \times r$ is the kernel size. To sum up, with respect to the hardware platform running the algorithm (CPU or GPU), and the chosen implementation method, the selection of desired cost map may vary from case to case.

The final question of the thesis explored the practical details and results of implementing V-CBF on an industrial mobile robot. Chapter 6 addressed the implementation of V-CBF on an industrial mobile robot, providing insights into the detailed segmentation model, sensor setup, and software and hardware platform used for V-CBF implementation, along with the experimental results.

In conclusion, this master's thesis investigated various methods for designing V-CBF, a novel control method for safe navigation. It introduced a solid scenario-based and metric-based test framework based on the ISO 22737 standard for the evaluation of safe navigation controllers. The designed V-CBF methods were rigorously tested within this

framework. Moreover, the thesis demonstrated the practical implementation of V-CBF on an industrial mobile robot, showcasing the solutions to integrate V-CBF in unknown real-world environments.

REFERENCES

- [1] Hossein Abdi, Golnaz Raja, and Reza Ghabcheloo. “Safe Control using Vision-based Control Barrier Function (V-CBF)”. *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2023, pp. 782–788.
- [2] Aaron D Ames et al. “Control barrier functions: Theory and applications”. *2019 18th European control conference (ECC)*. IEEE. 2019, pp. 3420–3431.
- [3] *ASAM OpenDrive*. URL: <https://www.asam.net/standards/detail/openscenario/>.
- [4] *ASAM OpenSCENARIO*. URL: <https://www.asam.net/standards/detail/openscenario/>.
- [5] Seho Bae et al. “Efficient generation of multiple sketch styles using a single network”. *IEEE Access* 7 (2019), pp. 100666–100674.
- [6] Mohammed Abdessamad Bekhti and Yuichi Kobayashi. “Regressed terrain traversability cost for autonomous navigation based on image textures”. *Applied Sciences* 10.4 (2020), p. 1195.
- [7] Rahim F Benekohal and Joseph Treiterer. “CARSIM: Car-following model for simulation of traffic in normal and stop-and-go conditions”. *Transportation research record* 1194 (1988), pp. 99–111.
- [8] Franco Blanchini. “Set invariance in control”. *Automatica* 35.11 (1999), pp. 1747–1767.
- [9] *CARLA ScenarioRunner*. URL: <https://carla-scenariorunner.readthedocs.io/en/latest/>.
- [10] Tony Chan and Wei Zhu. “Level set based shape prior segmentation”. *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05)*. Vol. 2. IEEE. 2005, pp. 1164–1170.
- [11] Salman UH Dar et al. “Image synthesis in multi-contrast MRI with conditional generative adversarial networks”. *IEEE transactions on medical imaging* 38.10 (2019), pp. 2375–2388.
- [12] Alexey Dosovitskiy et al. “CARLA: An open urban driving simulator”. *Conference on robot learning*. PMLR. 2017, pp. 1–16.
- [13] Pedro F Felzenszwalb and Daniel P Huttenlocher. “Distance transforms of sampled functions”. *Theory of computing* 8.1 (2012), pp. 415–428.
- [14] Mateus V Gasparino et al. “Wayfast: Navigation with predictive traversability in the field”. *IEEE Robotics and Automation Letters* 7.4 (2022), pp. 10651–10658.
- [15] David Gilbarg et al. *Elliptic partial differential equations of second order*. Vol. 224. 2. Springer, 1977.

- [16] *Godot 4.0 gets SDF based real-time global illumination*. <https://godotengine.org/article/godot-40-gets-sdf-based-real-time-global-illumination/>.
- [17] Ian Goodfellow et al. “Generative adversarial networks”. *Communications of the ACM* 63.11 (2020), pp. 139–144.
- [18] Saurabh Gupta et al. “Cognitive mapping and planning for visual navigation”. *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 2616–2625.
- [19] ISO. *Intelligent transport systems — Low-speed automated driving (LSAD) systems for predefined routes Performance requirements, system requirements and performance test procedures*. 2021. URL: <https://www.iso.org/standard/73767.html>.
- [20] *ISO 17757 standard*. URL: <https://www.iso.org/standard/76126.html>.
- [21] *ISO 18497 standard*. URL: <https://www.iso.org/standard/62659.html>.
- [22] Phillip Isola et al. “Image-to-image translation with conditional adversarial networks”. *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 1125–1134.
- [23] *jetson*. URL: <https://developer.nvidia.com/embedded/learn/jetson-agx-orin-devkit-user-guide/index.html>.
- [24] Gregory Kahn, Pieter Abbeel, and Sergey Levine. “Badgr: An autonomous self-supervised learning-based navigation system”. *IEEE Robotics and Automation Letters* 6.2 (2021), pp. 1312–1319.
- [25] Siddhartha Khastgir et al. “Systems approach to creating test scenarios for automated driving systems”. *Reliability engineering & system safety* 215 (2021), p. 107610.
- [26] *kinect*. URL: <https://azure.microsoft.com/en-us/products/kinect-dk>.
- [27] *Kinect ROS Driver*. URL: https://github.com/microsoft/Azure_Kinect_ROS_Driver.
- [28] Alexander Kirillov et al. “Segment anything”. *arXiv preprint arXiv:2304.02643* (2023).
- [29] Huei-Yung Lin and Xin-Zhong Peng. “Autonomous quadrotor navigation with vision based obstacle avoidance and path planning”. *IEEE Access* 9 (2021), pp. 102450–102459.
- [30] Björn Lindqvist et al. “Nonlinear MPC for collision avoidance and control of UAVs with dynamic obstacles”. *IEEE robotics and automation letters* 5.4 (2020), pp. 6001–6008.
- [31] Ravi Malladi, James A Sethian, and Baba C Vemuri. “Shape modeling with front propagation: A level set approach”. *IEEE transactions on pattern analysis and machine intelligence* 17.2 (1995), pp. 158–175.
- [32] *MiR*. URL: <https://www.mobile-industrial-robots.com/solutions/robots/mir100/>.
- [33] Mehdi Mirza and Simon Osindero. “Conditional generative adversarial nets”. *arXiv preprint arXiv:1411.1784* (2014).
- [34] Sahar Almahfouz Nasser et al. “Perceptual cGAN for MRI Super-resolution”. *2022 44th Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*. IEEE. 2022, pp. 3035–3038.

- [35] Anh Nguyen et al. "Autonomous navigation in complex environments with deep multimodal fusion network". *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2020, pp. 5824–5830.
- [36] Mark Nixon and Alberto Aguado. *Feature extraction and image processing for computer vision*. Academic press, 2019.
- [37] Maxime Oquab et al. "Dinov2: Learning robust visual features without supervision". *arXiv preprint arXiv:2304.07193* (2023).
- [38] Stanley Osher, Ronald Fedkiw, and K Piechor. "Level set methods and dynamic implicit surfaces". *Appl. Mech. Rev.* 57.3 (2004), B15–B15.
- [39] Zhiwei Qin et al. "Style transfer in conditional GANs for cross-modality synthesis of brain magnetic resonance images". *Computers in Biology and Medicine* 148 (2022), p. 105928.
- [40] *RoadRunner*. URL: <https://se.mathworks.com/products/roadrunner.html>.
- [41] *ROS*. URL: <https://www.ros.org/>.
- [42] Matteo Saveriano and Dongheui Lee. "Learning barrier functions for constrained motion planning with dynamical systems". *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2019, pp. 112–119.
- [43] Antonio Scandurra. *Leveraging Rust and the GPU to render user interfaces at 120 FPS - Zed Blog*. <https://zed.dev/blog/videogame>. 2023.
- [44] Manu Sethi, Anand Rangarajan, and Karthik Gurumoorthy. "The Schrödinger distance transform (SDT) for point-sets and curves". *2012 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE. 2012, pp. 198–205.
- [45] James A Sethian. "A fast marching level set method for monotonically advancing fronts." *proceedings of the National Academy of Sciences* 93.4 (1996), pp. 1591–1595.
- [46] Felipe Coelho Silva et al. "Mangan: Assisting colorization of manga characters concept art using conditional gan". *2019 IEEE International Conference on Image Processing (ICIP)*. IEEE. 2019, pp. 3257–3261.
- [47] Abhik Singla, Sindhu Padakandla, and Shalabh Bhatnagar. "Memory-based deep reinforcement learning for obstacle avoidance in UAV with limited environment knowledge". *IEEE transactions on intelligent transportation systems* 22.1 (2019), pp. 107–118.
- [48] Mohit Srinivasan et al. "Synthesis of control barrier functions using a supervised machine learning approach". *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2020, pp. 7139–7145.
- [49] Tilo Strutz. "The distance transform and its computation". *arXiv preprint arXiv:2106.03503* (2021).
- [50] Lei Tai, Giuseppe Paolo, and Ming Liu. "Virtual-to-real deep reinforcement learning: Continuous control of mobile robots for mapless navigation. In 2017 IEEE". *RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 31–36.

- [51] *Unreal Engine*. URL: <https://www.unrealengine.com/en-US>.
- [52] David Wooden. "A guide to vision-based map building". *IEEE Robotics & Automation Magazine* 13.2 (2006), pp. 94–98.
- [53] Wei Xiao et al. "BarrierNet: A safety-guaranteed layer for neural networks". *arXiv preprint arXiv:2111.11277* (2021).
- [54] Wei Xiao et al. "Differentiable control barrier functions for vision-based end-to-end autonomous driving". *arXiv preprint arXiv:2203.02401* (2022).
- [55] Lin Yao et al. "Path planning obstacle avoidance algorithm based on wheeled robot". *2020 International Workshop on Electronic Communication and Artificial Intelligence (IWECAI)*. IEEE. 2020, pp. 61–65.
- [56] Liron Yatziv, Alberto Bartesaghi, and Guillermo Sapiro. "O (N) implementation of the fast marching algorithm". *Journal of computational physics* 212.2 (2006), pp. 393–399.
- [57] *YOLOv8*. URL: <https://github.com/ultralytics/ultralytics>.
- [58] Xizhe Zhang et al. "A novel scenario-based testing approach for cooperative-automated driving systems". *IEEE SMC (systems, man, and cybernetics) 2023* (2023).
- [59] Xizhe Zhang et al. "Test framework for automatic test case generation and execution aimed at developing trustworthy avs from both verifiability and certifiability aspects". *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*. IEEE. 2021, pp. 312–319.
- [60] Hongkai Zhao. "A fast sweeping method for eikonal equations". *Mathematics of computation* 74.250 (2005), pp. 603–627.
- [61] Xueyan Zou et al. "Segment everything everywhere all at once". *arXiv preprint arXiv:2304.06718* (2023).