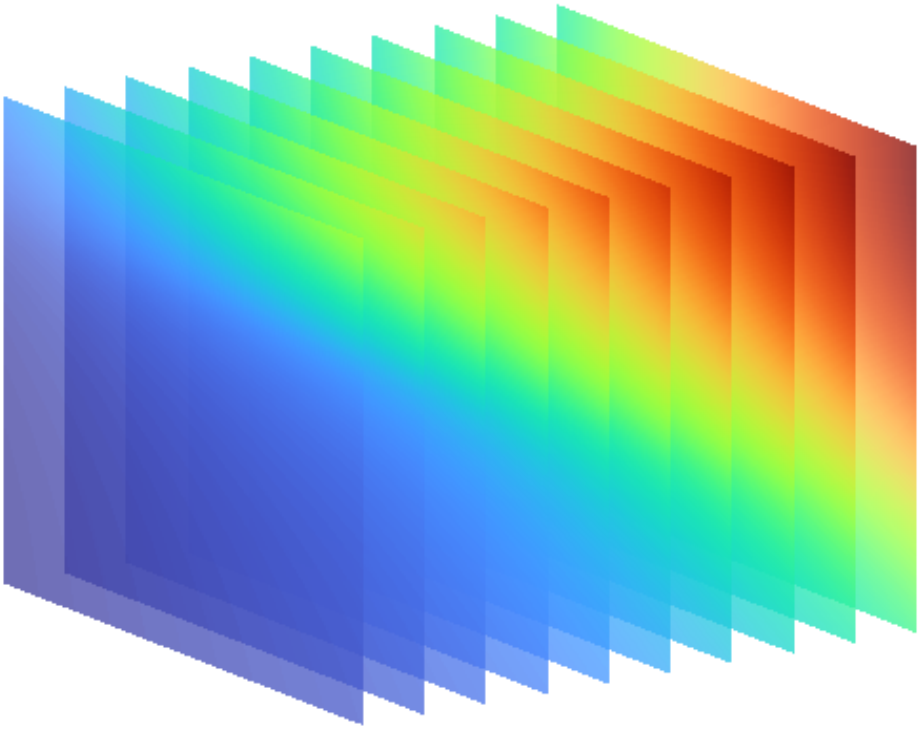
Fault Tolerant Control Barrier Functions

Master of Science Thesis

Quinten Eric Johannes van Hilten

17 January 2024

Technical University of Delft
Faculty of Aerospace Engineering



TU Delft

Master of Science Thesis
Fault Tolerant Control Barrier Functions

Q.E.J. van Hilten, MSc student TU Delft
supervisor: Dr. ir. C.C. de Visser, TU Delft

January 2024

Members of the thesis committee:
Dr. ir. C. Borst, TU Delft
Dr. ir. X. Wang, TU Delft
Dr. ir. C.C. de Visser, TU Delft

Acronyms

Acronym	Definition
aCBF	adaptive Control Barrier Function
CLF	Control Lyapunov Function
DCBF	Discrete Control Barrier Function
DLCF	Discrete Lyapunov Control Function
ECBF	Exponential Control Barrier Function
ftCBF	fault tolerant Control Barrier Function
HJB	Hamilton Jacobi Bellman
HOCBF	Higher Order Control Barrier Function
MPC	Model Predictive Control
QP	Quadratic Programming
PC	Personal Computer

Glossary

Glossary of thesis paper

Symbol	Description
$\mathcal{L}_f$	Directional derivative of f
$\alpha(\cdot)$	Extended class κ_∞ function
$\vec{x}$	State vector
$\vec{u}$	Input vector
$p_i(t)$	Penalty function
$\vec{\pi}_i$	Auxiliary system state vector for p_i
$\nu(t)$	Virtual controller
$z(t)$	Distance between ego vehicle and preceding vehicle
$v(t)$	Velocity of ego vehicle
v_p	Constant velocity of preceding vehicle
l_p	Constant describing the minimum allowed value of $z(t)$
a	A failure constant related to the change of the control bounds
γ	A failure constant related to the change of the system dynamics parameters
ζ	Used as an integration variable
$V(t)$	The velocity of the Dubin's Car vehicle
$\theta(t)$	Angle of the Dubin's Car vehicle
$[\cdot]_0$	Value of the variable at t_0
$[\cdot]_s$	Starting value of variable during ftCBF computation
$u^*(t)$	Recovery function
$\mathcal{C}_1$	Predefined safe set
$\mathcal{S}$	Set for which the recovery function is defined
$I(x_0)$	Interval of existence
χ_0^i	HOCBF for penalty function p_i
$u^*(\vec{x}, \vec{\lambda})$	Recovery function
t^*	Time after which $\psi_m \geq 0, \forall t \geq t^*$, when $u^* = u(t)$ is used
u_{min}	lower control bound
u_{max}	upper control bound
$\vec{\lambda}$	System parameters vector

Glossary of preliminary report

Symbol	Description
$\mathcal{L}$	Backwards reachable set
$\vec{x}$	State vector
χ	System state space
γ	Class κ function
$V(\cdot)$	Lyapunov function
$h(x)$	Control barrier function
L_f	Directional derivative of f

Contents

General Introduction

This document is a summary of the thesis work done from December 2022 to December 2023.

Control Barrier Functions (CBFs) are a powerful tool in the control theory community that provide a systematic way to ensure safety in dynamical systems. CBFs enable the design of control laws that guarantee the system trajectories remain within predefined safe sets, which is also called set invariance [1]. The range of applications for CBFs are vast, ranging from autonomous vehicles, industrial processes, safer manually controlled vehicles and in general for safety critical systems.

The main objective of the research was to contribute to this field of safety critical control, where the main focus was on using control barrier functions to provide set invariance to a predefined safe set, for changing control bounds and changing system dynamics. This would allow the CBFs to be used for fault tolerant purposes, as an anticipated failure could be described by its corresponding changes in the system dynamics and changes in the control bounds. The formal main research question was thus formulated as follows:

Main Research Question:
How can a predefined safe set be made invariant for changing control bounds and changing system dynamics?

This research question has been answered by the introduction of a novel CBF, called the fault tolerant Control Barrier Function (ftCBF), which has the necessary properties to guarantee set invariance of a predefined safe set with changing control bounds, and changing system dynamics, defined by a lipschitz continuous failure function describing these changes in system dynamics and control bounds for an anticipated failure. In addition to answering the main research question, the following research questions were also answered.

Research Question 1. *Is the adaptive Control Barrier Function (aCBF) [2] able to guarantee set invariance for changing control bounds, and changing system dynamics?*

In the thesis paper (Part I), it has been shown that the aCBF is not able to guarantee set invariance for constant control bounds, and thus would not be able to do so for changing control bounds. Furthermore, it has been shown that the relaxation of the constraint for the input, will eventually result in a singularity for the virtual control input, thus it would also not be suitable for changing system dynamics, as the same effect would occur, and the system would leave the predefined safe set.

Research Question 2. *What effect does preventing anticipated failures with the ftCBF have on the performance of the vehicle in nominal conditions?*

This has been partially answered in the thesis paper (Part I), where an analysis has been made for different anticipated failure scenarios, and in general it could be said that the more aggressive the anticipated failure is, for which the ftCBF has to account for, the more the performance of the vehicle will suffer as a result. Thus a trade-off between safety risks and performance should be made, as to not limit the nominal performance unnecessarily.

Research Question 3. *How will the invariant part of the predefined safe set change when a sudden or very aggressive failure happens to the system?*

This question has been answered in the context of the definition of the recovery function (Part I Definition 10), for which a recovery function is only defined if there exists a finite time t^* for which the derivative of the left hand side of a HOCBF constraint $\psi_m \geq 0$ (Part I (9)) is always larger or equal than zero, after this time t^*. In this context, as the invariant part of the

predefined safe set is the intersection of the set $\mathcal{S}$, for which the recovery function is defined, and the original defined safe set $\mathcal{C}_1$ (Part I (4)), if the recovery function is not defined for all $\vec{x} \in \mathcal{C}_1$ due to a sudden or very aggressive failure, then the invariant part of the set $\mathcal{C}_1$ will be smaller than $\mathcal{C}_1$ itself.

This document contains as the main deliverable the thesis paper, which is presented in Part I. The thesis paper is a stand-alone document, and contains the main result of the research.

The preliminary report is presented in Part II. The preliminary report contains an Introduction, a Literature Review, Research Questions, Methodology, Set-up of the research, Results and Outcomes planning, Project planning and finally a conclusion of the preliminary report. It should be noted that the preliminary report was made before the actual research began, and thus contains outdated information e.g. the project planning and the research questions.

The final part of this document are the Appendices, presented in Part III. Herein extra work and details are contained, which were not presented within the thesis paper.

Acknowledgement

I would like to thanks Dr. ir. Coen de Visser for the guidance during this thesis project. Furthermore, I would like to thank all the members of the VIDI group for providing feedback and inspiration for this research project.

Conclusions & Recommendations

In this chapter the main results of the research and the recommendations of the research will be summarized.

Main Results of Research

This main research question has been answered by the introduction of the novel CBF constraint, with the name fault tolerant Control Barrier Function (ftCBF), which has the necessary properties to keep a predefined safe set invariant in the event of changing control bounds and changing system dynamics parameters. The main property that allows the ftCBF to guarantee set invariance, is that it is dependent on the changing control bounds and changing system parameters, as opposed to the current state of the art CBF constraints.

The ftCBF can be used on general non-linear systems, and has been tested on the (non-linear) Dubin's Car model, for which it was clearly shown that it is capable of providing set invariance, even in the event of a failure.

Recommendations

For future work it is recommended that system dynamics parameter uncertainties as well as measurement noise from the sensors are incorporated into the ftCBF. This will allow the ftCBF to be even more generally applicable to systems with measurement noise and system dynamic parameter uncertainties.

The mitigation of the curse of dimensionality is another topic that may be researched in the future. Due to the curse of dimensionality it takes exponentially more computational time to compute the ftCBF constraint for higher dimensional systems.

Furthermore, additional research may be done on finding an optimal recovery function for general systems. This optimum could be in terms of least required energy or could be defined in terms of least time required for recovery.

Additionally, research may be done on optimizing the trade off between safety and performance.

Part I

Thesis paper

Fault Tolerant Control Barrier Functions

Quinten van Hilten, MSc Student TU Delft
supervisor: Dr. ir. C.C. de Visser, TU Delft

***Abstract*—In this article a novel Control Barrier Function (CBF) named the fault tolerant Control Barrier Function (ftCBF) is introduced. The ftCBF is able to keep a vehicle within a predefined safe set with changing control bounds and changing system dynamics. The ftCBF is shown to be feasible in fault tolerant control applications, as opposed to existing CBF methods. This novel constraint is tested on a double integrator system, and on a non-linear Dubin's Car system with changing system dynamics and changing control bounds. In the simulations it is shown that the ftCBF is able to keep the vehicle in the safe set with failure events occurring at any place in the timeline. The ftCBF contains design parameters that allow a trade-off between safety and performance.**

***Index Terms*—Control Barrier Function, Fault Tolerant Control, Safety Critical Control.**

I. Introduction

CONTROL Barrier Functions (CBFs) are a powerful tool in the control theory community that provide a systematic way to ensure safety in dynamical systems. CBFs enable the design of control laws that guarantee the system trajectories remain within predefined sets, which is also called set invariance[4]. CBFs have many applications in the field of robotics (e.g. Agrawal et al.[1]) and are able to be used in conjunction with other control techniques, such as Model Predictive Control (MPC) [12], to provide robust and safe control strategies for dynamical systems. There exist multiple definitions of a CBF, but for this research the definition given in A.D. Ames et al. [2] will be used.

The state of the art CBF methods[9][10][11] do not provide a way to ensure set invariance with changing system dynamics and changing control bounds. In this article the adaptive CBF (aCBF) is shown to be ineffective for fault tolerant control, as in this paper it is shown that set invariance is not guaranteed for constant control bounds, let alone changing control bounds. The application of the aCBF as of now is only done on a simple linear system, and has not been implemented on more general non-linear systems.

The main result of this research is the development of a new CBF method that is able to maintain set invariance in case of changing control bounds and changing system dynamics, and is thus able to guarantee safety during a failure event. This novel CBF constraint has been applied to both a linear and a non-linear system, and is generally applicable to any control affine system.

This article will first go over the necessary background that is needed to understand this article in Section II. In Section III, the adaptive Control Barrier Function will be briefly explained, and it will be shown that it is ineffective for fault tolerant control purposes, because set invariance of a predefined safe set is not guaranteed for changing control bounds and changing system dynamics. Section IV will introduce the novel CBF method, and an application of this method on a non-linear system will be shown in Section V. Finally this article will be concluded in Section VI.

II. Background

A. General Definitions

Definition 1 (Class κ_∞ function[5])**.** *A continuous function $\alpha : [0, \infty) \to [0, \infty)$, that is strictly increasing, and is such that $\alpha(0) = 0$ and $\lim_{r\to\infty} \alpha(r) = \infty$.*

Definition 2 (Extended class κ_∞ function [2])**.** *A class κ_∞ function for which the domain is extended to the entire real line $\mathbb{R} = (-\infty, \infty)$.*

Definition 3 (Lipschitz continuity)**.** *A function $f : \mathbb{R} \to \mathbb{R}$ is Lipschitz continuous if there exists a positive real number L such that the equation below holds:*

$$|f(x) - f(y)| \leq L|x - y| \tag{1}$$

Definition 4 (Interval of existence[2])**.** *The interval for which a differential equation has a unique solution $x(t)$ on $I(x_0) = [t_0, \tau_{max})$.*

Definition 5 (Forward completeness[3])**.** *A system is said to be forward complete if the interval of existence is equal to $I(x_0) = [t_0, \infty)$.*

Definition 6 (Invariance and safety[2])**.** *Considering a system with feedback controller $u = k(x)$*

$$\dot{\vec{x}} = f(\vec{x}) + g(\vec{x})k(\vec{x}) \tag{2}$$

a set $\mathcal{C}$ is forward invariant if for every $x_0 \in \mathcal{C}$, $x(t) \in \mathcal{C}$ for $x(t_0) = x_0$ and $\forall t \in I(x_0)$, where $I(x_0)$ is the interval of existence $I(x_0) = [t_0, \tau_{max})$ such that $x(t)$ is a unique solution for (2) on $I(x_0)$. The set $\mathcal{C}$ provides safety for its related system, if set $\mathcal{C}$ is forward invariant.

Definition 7 (Relative Degree[11])**.** *the relative degree of a function is the amount of times a function $d : \mathbb{R}^n \to \mathbb{R}$ has to be differentiated with respect to (3) along its dynamics in order for the input $\vec{u}$ of the system (3) to explicitly show up in the derivative. Provided that the function d can be differentiated sufficiently many times.*

B. Control Barrier Functions

Definition 8 (Control Barrier Function[2])**.** *Consider a general control affine system of the form*

$$\dot{\vec{x}} = f(\vec{x}) + g(\vec{x})\vec{u} \tag{3}$$

with $\vec{x} \in X \subset \mathbb{R}^n$ and $\vec{u} \in U \subset \mathbb{R}^m$. Then consider a set $\mathcal{C}_1$ defined by a continuously differentiable function $\psi_0 : \mathbb{R}^n \rightarrow \mathbb{R}$:

$$\mathcal{C}_1 := \{\vec{x} \in \mathbb{R}^n | \psi_0(\vec{x}) \geq 0\} \tag{4}$$

The function ψ_0 is a Control Barrier Function (CBF), if there exists an extended class κ_∞ function α_1 for which the following holds for all $x \in X \subset \mathbb{R}^n$:

$$\sup_{u \in U} [\mathcal{L}_f \psi_0(\vec{x}) + \mathcal{L}_g \psi_0(\vec{x}) \vec{u} + \alpha_1(\psi_0(\vec{x}))] \geq 0 \tag{5}$$

In the above equation (5) the symbols $\mathcal{L}_f$ and $\mathcal{L}_g$ denote the directional derivatives of the system in the direction of the functions f and g respectively.

Theorem 1 (from [2])**.** *Let $\mathcal{C}_1$ be defined as in (4), if ψ_0 is a Control Barrier Function and $\frac{\partial \psi_0}{\partial \vec{x}} \neq 0$ for all $\vec{x}$ on the boundary of $\mathcal{C}_1$, then any lipschitz continuous controller that satisfies (5), renders the set $\mathcal{C}_1$ invariant and thus safe (by Definition 6).*

C. Higher Order Control Barrier Functions

When the relative degree of the system is higher than one, (5) will not contain the input $\vec{u}$ explicitly as the term $\mathcal{L}_g \psi_0(\vec{x})$ will be zero by Definition 7. To still come up with a constraint explicit in the input, the Higher Order Control Barrier Function (HOCBF)[10] can be used.

To come up with a constraint that is explicit in $\vec{u}$, first (5) is made to define a new set

$$\mathcal{C}_2 := \{\vec{x} \in \mathbb{R}^n | \psi_1 \geq 0\} \tag{6}$$

where ψ_1 is defined as the left hand side of (5)

$$\psi_1 := [\mathcal{L}_f \psi_0(\vec{x}) + \mathcal{L}_g \psi_0(\vec{x}) \vec{u} + \alpha_1(\psi_0(\vec{x}))] \tag{7}$$

In order for ψ_1 to be a control barrier function for the set $\mathcal{C}_2$ it should by Definition 8 adhere to

$$\sup_{u \in U} [\mathcal{L}_f \psi_1(\vec{x}) + \mathcal{L}_g \psi_1(\vec{x}) \vec{u} + \alpha_2(\psi_1(\vec{x}))] \geq 0 \tag{8}$$

for all $x \in X \subset \mathbb{R}^n$, where α_2 in the above equation is again an extended class κ_∞ function (Definition 2) that may or may not be distinct from α_1. The set $\mathcal{C}_2$ is invariant, if there exists a lipschitz continuous controller that satisfies (8) and $\frac{\partial \psi_1}{\partial x} \neq 0$ for all x on the boundary (Theorem 1) . The invariance of the set $\mathcal{C}_2$ leads to the invariance of set $\mathcal{C}_1$, because the invariance of $\mathcal{C}_2$ by the definition of $\mathcal{C}_2$ (6) means that $\psi_1 \geq 0$, which is by the definition of ψ_1 (7) the same as satisfying (5), and thus in turn provides set invariance for $\mathcal{C}_1$ provided that $\frac{\partial \psi_0}{\partial x} \neq 0$ for all $\vec{x}$ on the boundary of $\mathcal{C}_1$.

If the constraint (8) still does not contain the input explicitly (i.e. $\mathcal{L}_g \psi_1(\vec{x}) = 0$), the above approach can be repeated until it does and the following will then be achieved with a relative degree of m:

$$\begin{aligned} \psi_0 &:= b(\vec{x}) \\ \psi_1 &:= \dot{\psi}_0 + \alpha_1(\psi_0) \\ &\vdots \\ \psi_m &:= \dot{\psi}_{m-1} + \alpha_m(\psi_{m-1}) \geq 0 \end{aligned} \tag{9}$$

When $\vec{x}_0 \in \mathcal{C}_1 \cap \cdots \cap \mathcal{C}_m$ and $\psi_m \geq 0$ holds for $\forall t \in I(x_0)$, the trajectories of the system (3) will remain in the set $\mathcal{C}_1$ (Theorem 2). In (9), ψ_m is actually not a CBF, but is used for ease of communication and denotes the constraint which is explicit in the input and is used to provide set invariance of the original predefined safe set $\mathcal{C}_1$ (4). In (9) the directional derivatives are replaced by the full derivatives (e.g. $\mathcal{L}_f \psi_0(\vec{x}) + \mathcal{L}_g \psi_0(\vec{x}) \vec{u} = \dot{\psi}_0$) for ease of communication. The function $b(\vec{x})$ in (9) denotes an arbitrary continuously differentiable function of $\vec{x}$ that defines the safe set $\mathcal{C}_1$ (4).

The more formal definition of the HOCBF and the corresponding theorem can be seen below, but first the formal definition of the sets $\mathcal{C}_i, i\{1, \ldots, m\}$ is given in (10).

$$\begin{aligned} \mathcal{C}_1 &:= \{\vec{x} \in \mathbb{R}^n | \psi_0 \geq 0\} \\ \mathcal{C}_2 &:= \{\vec{x} \in \mathbb{R}^n | \psi_1 \geq 0\} \\ &\vdots \\ \mathcal{C}_m &:= \{\vec{x} \in \mathbb{R}^n | \psi_{m-1} \geq 0\} \end{aligned} \tag{10}$$

Definition 9 (Higher order control barrier function[10])**.** *Let $\mathcal{C}_i, i \in \{1, \ldots, m\}$ be defined in (10) and $\psi_i, i \in \{1, \ldots, m\}$ be defined in (9). A function $b : \mathbb{R}^n \times I(x_0) \rightarrow \mathbb{R}$ is a higher order control barrier function of relative degree m for system (3) if there exist extended κ_∞ functions $\alpha_1, \ldots, \alpha_m$ such that $\psi_m \geq 0$ (9) holds for all $\forall t \in I(x_0)$ and for $\forall \vec{x} \in \mathcal{C}_1 \cap \cdots \cap \mathcal{C}_m$.*

Theorem 2 (from [10])**.** *Given a HOCBF (from Definition 9) with the sets $\mathcal{C}_i, i \in \{1, \ldots, m\}$ (defined in (10)), if $x_0 \in \mathcal{C}_1 \cap \cdots \cap \mathcal{C}_m$, then any Lipschitz continuous controller $u(t) \in U$ that satisfies $\psi_m \geq 0$ (from (9)) for all $t \in I(x_0)$, renders the set $\mathcal{C}_1 \cap \cdots \cap \mathcal{C}_m$ forwards invariant for system (3).*

III. Adaptive Control Barrier Functions

A. The workings of the aCBF

The Adaptive Control Barrier Function (aCBF) was originally designed to guarantee feasibility for Quadratic Programming (QP) problems[11]. In such problems, apart from satisfying the HOCBF constraint $\psi_2 \geq 0$ (9), also for instance $\int_0^T u^2(t) dt$ needs to be minimized. However, with the presence of constant control boundaries $u_{min} \leq u \leq u_{max}$, this could lead to in-feasibility of the QP problems. In Xiao et al. [11] it is also stated that the aCBF is validated for adaptivity for time varying control bounds and noisy system dynamics, however later in this section it will be shown that the aCBF does not even provide set invariance for constant control bounds and only provides very limited adaptivity. For this research the QP problems are not particularly of interest and thus will not be addressed further in this article. The adaptivity

properties however are interesting for this research, as they might have provided necessary fault tolerant control properties. In Xiao et al. [11] two different types of aCBF controllers are introduced, for which both are very similar in concept. For this research only the Parameter Adaptive Control Barrier Function (PACBF) will be used.

The main idea of the aCBF is to multiply the extended class κ_∞ functions with penalty functions $p_i(t) \geq 0, \forall t > 0$ in order to make the constraints adaptive, and thus be able to relax the constraints when they would otherwise fail for an input within the control bounds $u_{min} \leq u(t) \leq u_{max}$. These penalty functions are themselves designed to be HOCBF's for their own auxiliary systems and are controlled by virtual controllers ν_i to ensure that they will not reach below zero. These penalty functions have to be above zero as to not conflict with the definition of a CBF (Definition 8), because a function with a negative value multiplied by an extended class κ_∞ function does not belong to the extended class κ_∞ functions (e.g. $-\alpha(\cdot) \notin \kappa_\infty, \alpha(\cdot) \in \kappa_\infty$), as the function would not be strictly increasing (Definition 1). The auxiliary system is augmented with the original system (3) which results in the following HOCBF:

$$\begin{aligned} \psi_0 &:= b(\vec{x}) \\ \psi_i &:= \dot{\psi}_{i-1} + p_i(t)\alpha_i(\psi_{i-1}) \geq 0, i \in \{1, \dots, m\} \end{aligned} \tag{11}$$

The auxiliary systems can be designed in multiple ways, but an easy way is to make each state of the auxiliary state vector π_i, a derivative of the penalty function p_i. Below is an example of how an auxiliary system could be designed[11]:

$$\begin{aligned} \vec{\pi}_i &:= [p_i, \dots, p_i^{(m-i-1)}]^T, i \in \{1, \dots, m-2\} \\ \pi_{m-1} &:= [p_{m-1}] \\ \nu_i &:= \dot{\pi}_{i,m-i}, i \in \{1, \dots, m-1\} \end{aligned} \tag{12}$$

In (12) $p_i^{(m-i-1)}$ is the (m-i-1)th derivative of $p_i(t)$ and $\pi_{i,j} \in \mathbb{R}, j \in \{1, \dots, m-i\}$ are the auxiliary state variables. The amount of states in the auxiliary state vector $\vec{\pi}_i$, is equal to the amount of times the penalty function $p_i(t)$ needs to be differentiated in obtaining ψ_m, and thus $\vec{\pi}_i$ has m-i states. The derivative of the (m-i)th state of the auxiliary state vector $\vec{\pi}_i$ is made equal to the virtual controller ν_i (as can be seen in the last line of (12)), such that the final constraint ψ_m is explicit for the virtual controllers $\vec{\nu}$. The penalty function $p_m(t)$ can directly be controlled with its own virtual controller ($\nu_m = p_m(t)$), which always must be made greater than zero. Below the general form of the auxiliary system is given

$$\dot{\vec{\pi}}_i = F(\vec{\pi}_i) + G(\vec{\pi}_i)\nu_i, i \in \{1, \dots, m-1\} \tag{13}$$

which for the auxiliary system design of (12) for e.g. $i = 1$ and $m = 3$ would be:

$$\dot{\vec{\pi}}_1 = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} p_1(t) \\ \dot{p}_1(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \nu_1(t) \tag{14}$$

The HOCBF constraints for the penalty functions $p_i, i \in \{1, \dots, m-1\}$ are given below:

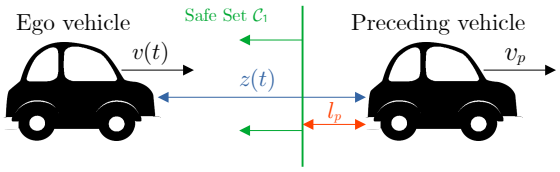


Fig. 1. This figure gives a visualisation of the simple double integrator system given in (16), with safe set $\mathcal{C}_1$ (4), defined by $\psi_0 := z(t) - l_p \geq 0$.

$$\begin{aligned} \chi_0^i &:= p_i(t) \\ &\vdots \\ \chi_{m-i}^i &:= \dot{\chi}_{m-i-1}^i + \alpha_{i,m-i}(\chi_{m-i-1}^i) \geq 0 \end{aligned} \tag{15}$$

In (15), χ_0^i is the HOCBF for the general auxiliary system (13) belonging to $p_i(t)$, which describes the set $\mathcal{B}_1^i := \{\vec{\pi}_i \in \mathbb{R}^{m-i} | \chi_0^i \geq 0\}$. By satisfying the constraint $\chi_{m-i}^i \geq 0$, which is explicit in ν_i, the set $\mathcal{B}_1^i$ will be invariant (Theorem 2) and thus $p_i(t)$ will remain above zero if $p_{i0}(t) \geq 0$.

Hereafter, the constraint $\psi_m \geq 0$ (11) of the augmented system can be satisfied, and because ψ_m will be a function of the input $\vec{u}$ and the virtual controllers $\vec{\nu}$, the constraint can be relaxed for $\vec{u}$ by controlling the virtual controls of $\vec{\nu}$.

B. Application of aCBF

1) Dynamic System: The aCBF will be applied on a simple double integrator system to illustrate its operation. The augmented system that will be used can be seen below:

$$\begin{aligned} \dot{z}(t) &= v_p - v(t) \\ \dot{v}(t) &= u(t) \\ \dot{p_1}(t) &= \nu_1(t) \end{aligned} \tag{16}$$

In the above equation, $p_1(t)$ is the penalty function that will be used in the HOCBF constraint, v_p is the constant velocity of a preceding vehicle and $v(t)$ is the velocity of the ego vehicle and $u(t)$ is the input of the ego vehicle, which has the control boundary constraints $-2 \leq u(t) \leq 5$. The safe set (4) will be defined by the function $\psi_0 := z(t) - l_p$, where l_p is some positive constant and $z(t)$ describes the distance between the ego vehicle and the vehicle in front of the ego vehicle. A visualisation of the system is given in Figure 1. The derivation of the HOCBF constraint $\psi_2 \geq 0$ resulting from the definition of HOCBF ψ_0 and (16) can be seen below:

$$\begin{aligned} \psi_0 &:= z(t) - l_p \\ \psi_1 &:= (v_p - v(t)) + p_1(z(t) - l_p) \\ \psi_2 &:= p_1(t)(v_p - v(t)) + \nu_1(t)(z(t) - l_p) \\ &-u(t) + (v_p - v(t)) + p_1(z(t) - l_p) \geq 0 \end{aligned} \tag{17}$$

In (17) the penalty function $p_2(t) = \nu_2(t)$ for the constraint ψ_2 is made to be constant and equal to one. As can be seen in

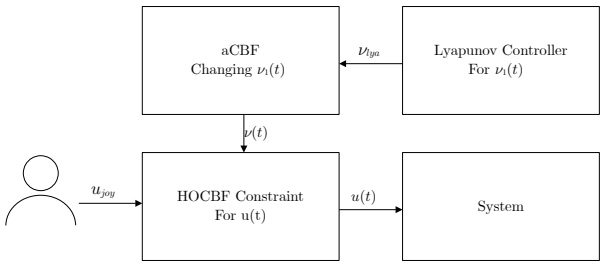


Fig. 2. This figure shows the general schematics of the implementation. The user provides a joystick input u_{joy} to the HOCBF constraint $\psi_2 \geq 0$ (17), which is only passed on to the system if it does not conflict with $\psi_2 \geq 0$ (17), otherwise it will provide an input to the system that does satisfy the constraint. If the HOCBF constraint is not able to be satisfied with an input within the control bounds $u_{min} \leq u(t) \leq u_{max}$, the aCBF increases $\nu(t)$ to relax the HOCBF constraint for u(t). If the aCBF is not active, the lyapunov controller controls $\nu(t)$ to achieve a preferred value for the penalty function.

the constraint $\psi_2 \geq 0$ (17), the input $u(t)$ can be relaxed by increasing $\nu_1(t)$. In order to provide invariance for the defined safe set $\mathcal{C}_1$ (4), in this case defined by $\psi_0 := z(t) - l_p \geq 0$, the constraint $\psi_2 \geq 0$ (17) should be satisfied (Theorem 2). The class κ_∞ functions $\alpha_1(\cdot)$ and $\alpha_2(\cdot)$ in (17) are defined as $\alpha_1(\psi_0) := \psi_0$ and $\alpha_2(\psi_1) := \psi_1$ respectively.

The HOCBF constraint $\chi_1^1 \geq 0$ that should be satisfied for the penalty function $p_1(t)$ can be seen below:

$$\begin{aligned} \chi_0^1 &= p_1(t) \\ \chi_1^1 &= \nu_1(t) + p_1(t) \geq 0 \end{aligned} \tag{18}$$

2) Implementation schematics: An overview of the implementation of the aCBF on the double integrator system can be seen in Figure 2. The constraint $\psi_2 \geq 0$ (17) with input $u(t)$ and virtual input $\nu_1(t)$ is implemented by first trying to satisfy the constraint by changing $u(t)$ within its control bounds. If the control bounds for $u(t)$ prevents the satisfaction of the constraint, the constraint will then be satisfied by changing the virtual control input $\nu_1(t)$. This will start to increase the penalty function $p_1(t)$. To keep $p_1(t)$ from ever increasing and close to a desired value after the constraint has turned off, a lyapunov controller for ν_1 is active when the aCBF constraint is not active and returns the penalty function to a predefined value.

3) Simulation Results: The simulation results can be seen in Figure 3 and Figure 4 and the parameter values used in the simulation are given in Table I. As can bee seen in Figure 4, the aCBF is not able to prevent a failure from occurring with constant control bounds, as the virtual control $\nu_1(t)$ reaches a singularity. This is because the aCBF does not contain the control bounds explicitly in the constraints and the constraint activates too late in order to prevent the failure from happening, even with infinite relaxation of the HOCBF constraint $\psi_2 \geq 0$ (17). In section IV a novel approach will be discussed that does contain the control bounds explicitly in the constraints.

TABLE I
SIMULATION PARAMETERS ACBF

p_0	z_0	v_0	l_p	v_p	a	u_{min}	u_{max}
0.4	10	11	2	8	1	-2	5

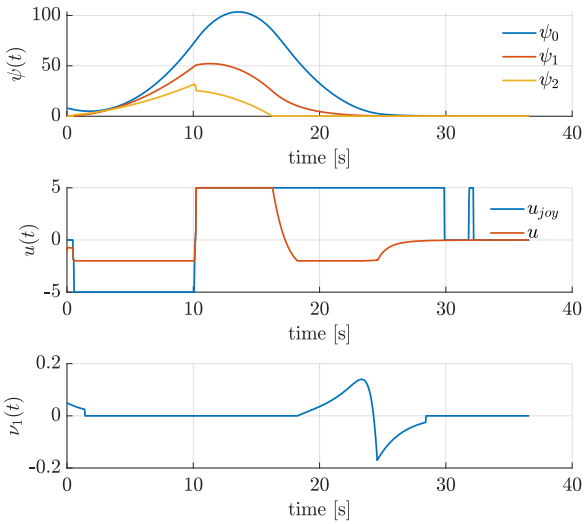


Fig. 3. In this figure a simulation of the aCBF is shown. For this simulation a human controller was used to demonstrate the workings of the aCBF. The function $\psi_0 := z(t) - l_p \geq 0$ describes the predefined safe set $\mathcal{C}_1$ (4), and physically describes the distance between the ego vehicle and the preceding vehicle minus a safety constant l_p (as can also be seen in Figure 1). The other CBF ψ_1 and the final constraint $\psi_2 \geq 0$ (17) are also shown and if $\psi_2 \geq 0$ (17) both ψ_0 and ψ_1 are as a result above zero, which guarantees that the safe set $\mathcal{C}_1$ stays invariant. In this simulation the distance between both vehicles ($z(t) = \psi_0 + l_p$), was first increased by providing a negative controller input u_{joy} for about 10 seconds. After around 10 seconds, the controller input u_{joy} provided a positive input, which accelerated the ego vehicle towards the preceding vehicle, until around 18 seconds, where the HOCBF constraint first activates. Quickly thereafter, the aCBF increases $\nu_1(t)$ to relax the HOCBF constraint for $u(t)$. The effects of the lyapunov controller can be seen after around 23 seconds where $\nu_1(t)$ briefly turns negative, to lower the penalty function $p_1(t)$ to its base value again.

IV. Fault Tolerant Control Barrier Functions

The Fault Tolerant Control Barrier Function (ftCBF) is a novel type of control barrier function and is the main contribution of this research. It guarantees set invariance for a predefined safe set $\mathcal{C}_1$ (4) for a system (3) that has changing control bounds and changing system dynamics. This makes it well suited for fault tolerant control, as a type of failure can be anticipated by describing how the failure would change the control bounds and system dynamics, in what will be hereafter called failure functions, describing these changes. The ftCBF can then with those failure functions provide invariance of the predefined safe set $\mathcal{C}_1$ (4).

First to explain the general concept, consider a system (3), with $\psi_m := \dot{\psi}_{m-1} + \alpha_m(\psi_{m-1}) \geq 0$ from (9) being the constraint that needs to be satisfied in order to guarantee safety for the system. Then a control function $u^*(t)$, within the changing control bounds $u_{min}(t) \leq u^*(t) \leq u_{max}(t)$, can be defined as a recovery function of the system that eventually

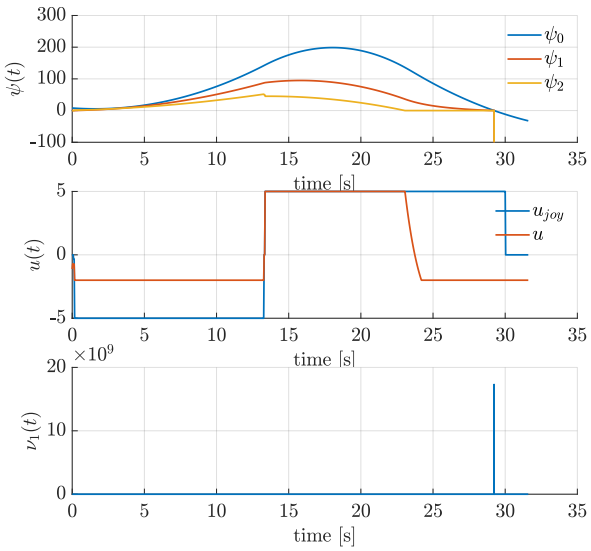


Fig. 4. This figure shows a simulation of the aCBF, where the vehicle is made to go outside of the predefined safe set $\mathcal{C}_1$ (4), defined by $\psi_0 := z(t) - l_p \geq 0$, and thus shows that the aCBF fails to guarantee set invariance with constant control bounds $u_{min} \leq u(t) \leq u_{max}$. The maneuver performed by the human controller was quite similar as the maneuver performed in Figure 3, however the distance ($z(t) = \psi_0 + l_p$) was increased even further, such that the ego vehicle was able to accelerate towards the preceding vehicle for a longer amount of time before the constraint $\psi_2 \geq 0$ (17) was activated. As can be seen in the figure at around 29 seconds the virtual controller tries to relax the constraint $\psi_2 \geq 0$ for the input, but reaches a singularity and the vehicle leaves the safe set, because $\psi_0 < 0$.

at a time t^* is able to stop a system from moving closer to the boundary of the predefined safe set $\mathcal{C}_1$ (4) for $\forall x_0 \in S \subset X$. Where S is the set for which the recovery function is defined. The recovery function is made to be a function of the system states $\vec{x}(t)$ and the changing system parameters $\vec{\lambda} \in \Lambda \subset \mathbb{R}^p$, which includes the changing control boundaries $u_{min}(t)$ and $u_{max}(t)$. As the failure functions describe the changes of $\vec{\lambda}$, the recovery function will also be defined for anticipated failure events. The formal definition of a recovery function, as well as the theorem that states when the set $\mathcal{C}_1 \cap \cdots \cap \mathcal{C}_m \cap \mathcal{S}$ will be invariant can be found below.

Definition 10 (Recovery function)**.** *A lipschitz continuous control function $u^*(\vec{x}, \vec{\lambda}) : \mathbb{R}^n \times \mathbb{R}^p \to \mathbb{R}^m$ within the changing control bounds $u_{min}(t) \leq u^*(t) \leq u_{max}(t)$ such that $\forall \vec{x}_0 \in \mathcal{S} \subset X$, $\exists t^* \in I(x_0)$ such that for $\forall t \in I(x_0) | t \geq t^*$ the derivative $\dot{\psi}_m \geq 0$.*

Theorem 3. *If there exists a recovery function (Definition 10) for a system (3) with a HOCBF ψ_0 and a corresponding HOCBF constraint $\psi_m \geq 0$, such that*

$$\min_{t \in I(x_0) | t_0 \leq t \leq t^*} \psi_m(u^*(\vec{x}(t), \vec{\lambda}(t)), \vec{x}(t), \vec{\lambda}(t)) \geq 0 \tag{19}$$

and $\vec{x}_0 \in \mathcal{C}_1 \cap \cdots \cap \mathcal{C}_m \cap \mathcal{S}$, then the set $\mathcal{C}_1 \cap \cdots \cap \mathcal{C}_m \cap \mathcal{S}$ is forward invariant.

Proof. By Theorem 2, the set $\mathcal{C}_1 \cap \cdots \cap \mathcal{C}_m$ is invariant if for $\vec{x}_0 \in \mathcal{C}_1 \cap \cdots \cap \mathcal{C}_m$, any lipschitz continuous controller $u(t)$ ensures that $\psi_m(u(t), \vec{x}(t), \vec{\lambda}(t)) \geq 0$ (9) for $\forall t \in I(x_0)$. If for $u(t)$ the recovery function $u^*(\vec{x}(t), \vec{\lambda}(t))$ (Definition 10) is used, and the minimum value of $\psi_m(u^*(\vec{x}(t), \vec{\lambda}(t)), \vec{x}(t), \vec{\lambda}(t))$ is greater then zero for $t \in I(x_0) | t_0 \leq t \leq t^*$, the HOCBF constraint $\psi_m \geq 0$ (9) will be satisfied for $\forall t \in I(x_0)$, because for $t \in I(x_0) | t \geq t^*$ the derivative of the HOCBF constraint $\dot{\psi}_m \geq 0$ (by Definition 10), while the recovery function is used. However, because the recovery function u^* is only defined for $x_0 \in S$, only the intersection of the set $\mathcal{C}_1 \cap \cdots \cap \mathcal{C}_m$ and $\mathcal{S}$ is made invariant. □

Definition 11 (fault tolerant Control Barrier Function (ftCBF))**.** *A HOCBF $\psi_0 : \mathbb{R}^n \to \mathbb{R}$ is a ftCBF if there exists a recovery function u^* (Definition 10) such that (19) holds $\forall t \in I(x_0)$ and $\forall \vec{x} \in \mathcal{C}_1 \cap \cdots \cap \mathcal{C}_m \cap \mathcal{S}$.*

For the practical application of the ftCBF, the system is simulated offline for various different starting conditions (denoted with subscript $[\cdot]_s$), of the system states $\vec{x}_s$ and different starting system parameters $\vec{\lambda}_s$ through time to compute the minimum of the HOCBF constraint through time. This is done because the constraint can typically not be found analytically. The obtained constraint (20) which is a function of the starting parameters, is then interpolated in real time with current values for the system states and current system parameters substituted as starting values.

$$\min_{t \in I(x_s) | t_s \leq t \leq t^*} \psi_m(u^*(\vec{x}_s, \vec{\lambda}_s, t), \vec{x}(t), \vec{\lambda}(t)) \geq 0 \tag{20}$$

A. Example on Double Integrator System

For this example the system (16) will be used, without the penalty functions:

$$\begin{aligned} \dot{z}(t) &= v_p - v(t) \\ \dot{v}(t) &= u(t) \end{aligned} \tag{21}$$

The HOCBF of the system (21) that needs to be satisfied to guarantee set invariance then results in the below equation:

$$\psi_2(t) = -u(t) + 2(v_p - v(t)) + (z(t) - l_p) \geq 0 \tag{22}$$

In this relatively simple system, a good recovery function would be the smallest input $u(t)$, which is by definition the lower control bound $u_{min}(t)$, and thus $u(t) = u_{min}(t)$. To simulate a failure in the control effectiveness, u_{min} is modeled to change over time during a failure by the following failure function

$$u(t) = u_{min}(t) = u_{min_0} \cdot a^t \tag{23}$$

where a is a parameter between zero and one. To prove that $u(t) = u_{min_0} \cdot a^t$ is a recovery function of the system (16), the derivative of (22) should be greater than zero after some time t^* (Definition 10). The derivative of (22) is given below.

$$\dot{\psi}_2 = -\dot{u}(t) - 2u(t) + (v_p - v(t)) \geq 0 \tag{24}$$

In above equation the derivative of the input is

$$\dot{u}(t) = u_{min_0} \cdot a^t ln(a) \tag{25}$$

and the velocity of the ego vehicle $v(t)$ can be written as

$$\begin{aligned} v(t) &= \int_0^t \dot{v}(\zeta)d\zeta + v_0 \\ &= \int_0^t u_{min_0} \cdot a^{\zeta} d\zeta + v_0 \\ &= u_{min_0}(\frac{a^t}{ln(a)} - \frac{1}{ln(a)}) + v_0 \end{aligned} \tag{26}$$

Then by substituting (23),(25) and (26) into (24), the following inequality will be obtained:

$$-u_{min_0}(a^t ln(a) + 2a^t + (\frac{a^t}{ln(a)} - \frac{1}{ln(a)})) \geq -(v_p - v_0) \tag{27}$$

The system (21) is assumed to be forward complete (Definition 5), which makes the interval of existence $I(x_0)$ equal to the interval $[t_0, \infty)$, and thus to find the set $\mathcal{S} \subset X$ for which (23) is a recovery function (Definition 10), the limit of $t \to \infty$ can be taken of (27), which results in the following:

$$\begin{aligned} \lim_{t\to\infty} -u_{min_0}(a^t ln(a) + a^t + (\frac{a^t}{ln(a)} - \frac{1}{ln(a)})) = \frac{u_{min_0}}{ln(a)} \\ \frac{u_{min_0}}{ln(a)} \geq -(v_p - v_0) \end{aligned} \tag{28}$$

From the above equation it can be deduced that (23) is a recovery function (by Definition 10) for $v_0 \leq \frac{u_{min_0}}{ln(a)} + v_p$.

Then to provide set invariance for $\forall \vec{x}_0 \in \mathcal{C}_1 \cap \cdots \cap \mathcal{C}_m \cap \mathcal{S}$ by Theorem 3, (19) should be satisfied. Which for the system (21) and recovery function (23) would be

$$\begin{aligned} \min_{t\in I(x_0)|t_0\leq t\leq t^*} & -(u_{min_0} \cdot a^t) \\ & +2(v_p - v_0 - u_{min_0}(\frac{a^t}{ln(a)} - \frac{1}{ln(a)})) \\ & +(z_0 + \int_0^t \dot{z}(\zeta)d\zeta - l_p) \geq 0 \end{aligned} \tag{29}$$

where in the above equation $z_0 + \int_0^t \dot{z}(\zeta)d\zeta = z(t)$.

A simulation of system (21) with the constraint (29) can be seen in Figure 5. The parameters used for the simulation are given in Table I, where it can be seen that $a = 1$. This is to keep the control bounds constant (as can be seen in the failure function (23)), such that Figure 5 (with the ftCBF implementation) can be more easily compared to Figure 4 (aCBF implementation), with the exact same parameters and maneuver.

A more in depth application of the ftCBF on a non-linear system with changing system dynamics and changing control bounds will be shown in the next section.

V. ftCBF Application: Dubin's Car

A. System Dynamics

The Dubin's Car system can be visualized as a car that has a constant velocity and has the ability to change direction. In this variation of a Dubin's car system, the velocity will not be constant, but will be allowed to change as to demonstrate that the ftCBF is able to handle changing system dynamics as well as changing control bounds. Below the equations of motion of the system that will be used in this application are shown:

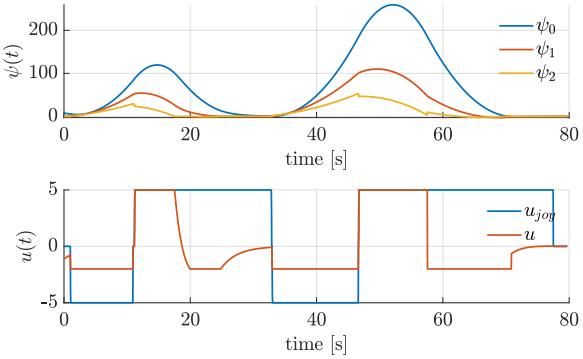


Fig. 5. In this figure the implementation of the ftCBF is shown on the same system as Figure 4, with the same parameters and maneuver being used for better comparison between this simulation and the simulation of Figure 4 (thus no failure is induced in this simulation). As can be seen the ftCBF ensures that the vehicle remains in the safe set $\mathcal{C}_1$ ($\psi_0 \geq 0$ (4)), with control boundaries $-2 \leq u(t) \leq 5$, as opposed to the aCBF for which the system failed (Figure 4). For this simulation the constant a in the failure function (23) was set to 1, thus u_{min} remained constant.

$$\begin{aligned} \dot{x} &= V(t)\cos(\theta(t)) \\ \dot{y} &= V(t)\sin(\theta(t)) \\ \dot{\theta} &= u(t) \\ u_{min}(t) \leq u(t) &\leq u_{max}(t) \end{aligned} \tag{30}$$

In the above equations $u(t)$ is the input that is able to change the angle θ at which the vehicle is driving. The speed will be a simple function of time:

$$V(t) = V_0 + \gamma t \tag{31}$$

where gamma is some positive design constant. A visualization of the system (30), can be seen in Figure 6, where also a visualisation of the safe set $\mathcal{C}_1$ (4) is given. The safe set for this application will be defined by $\psi_0 := x(t) \geq 0$

B. HOCBF & ftCBF constraints

For $\psi_0 := x(t)$ the system (30) will result in the the following HOCBF:

$$\begin{aligned} \psi_0 &:= x(t) \\ \psi_1 &= V(t)\cos(\theta(t)) + x(t) \\ \psi_2 &= \gamma\cos(\theta(t)) - V(t)\sin(\theta(t))u(t) \\ &\quad +2V(t)\cos(\theta(t)) + x(t) \geq 0 \end{aligned} \tag{32}$$

where $\alpha_1(\psi_0) := \psi_0$ and $\alpha_2(\psi_1) := \psi_1$.

For the system (30) a recovery function is defined below:

$$\begin{aligned} u^*(t) := u_{min}(t) = u_{min_0} \cdot a^t, \frac{\pi}{2} \leq \theta(t) \leq \pi \\ u^*(t) := u_{max}(t) = u_{max_0} \cdot a^t, \pi < \theta(t) \leq \frac{3\pi}{2} \\ u^*(t) := 0, -\pi/2 < \theta(t) < \pi/2 \end{aligned} \tag{33}$$

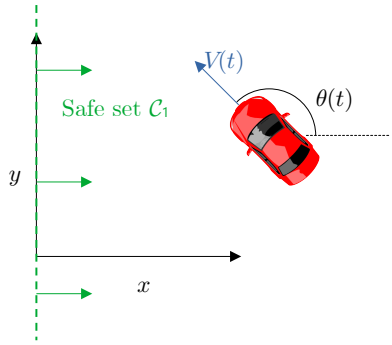


Fig. 6. This figure gives a visualisation of the Dubin's Car system (30), with safe set $\mathcal{C}_1$ (4), where $\psi_0 := x(t) \geq 0$.

As the system behaves very similar in both domain $\frac{\pi}{2} \leq \theta(t) \leq \pi$ and the domain $\pi < \theta(t) \leq \frac{3\pi}{2}$ (only the direction of rotation will be different), only the domain $0 \leq \theta(t) \leq \pi$ will be considered for simplicity.

To prove that

$$\begin{aligned} u^* = u_{min_0} \cdot a^t, \frac{\pi}{2} \leq \theta(t) \leq \pi \\ u^*(t) := 0, 0 \leq \theta(t) < \pi/2 \end{aligned} \tag{34}$$

is a recovery function for $0 \leq \theta(t) \leq \pi$, the derivative of ψ_2 in (32) must be greater or equal to zero after a time t^* (Definition 10). The derivative of ψ_2 (9) is given below

$$\begin{aligned} \dot{\psi}_2 = -2\gamma \sin\theta(t)u(t) \\ -V(t)\cos\theta(t)u^2(t) - V(t)\sin\theta(t)\dot{u}(t) \\ +2\gamma\cos\theta(t) - 2V(t)\sin\theta(t)u(t) \\ +V(t)\cos\theta(t) \end{aligned} \tag{35}$$

and for $0 \leq \theta(t) \leq \pi/2$ it can be shown that $\dot{\psi}_2 \geq 0$, by substituting $u^* = 0$ (34) into (35).

$$\dot{\psi}_2 = 2\gamma\cos\theta(t) + V(t)\cos\theta(t) \tag{36}$$

In the above equation γ and $V(t)$ are always positive or zero, and $\cos\theta(t)$ is also always positive or zero in the domain $0 \leq \theta(t) \leq \pi/2$, thus if the control function (34) is able to get the vehicle into the domain $0 \leq \theta(t) \leq \pi/2$, (34) is a recovery function as $\dot{\psi}_m \geq 0$ after a time t^*. The angle of the vehicle $\theta(t)$ can be made into a function of the control function u^* (34) as can be seen below

$$\begin{aligned} \theta(t) &= \theta_0 + \int_0^t \dot{\theta}(\zeta)d\zeta \\ &= \theta_0 + \int_0^t u(\zeta)d\zeta \\ &= \theta_0 + u_{min_0}(\frac{a^t}{ln(a)} - \frac{1}{ln(a)}) \end{aligned} \tag{37}$$

When assuming a negative lower control bound, and if θ_0 is not in the domain $0 \leq \theta(t) \leq \pi/2$, the angle $\theta(t)$ will enter the domain $0 \leq \theta(t) \leq \pi/2$ at $\theta(t^*) = \frac{\pi}{2}$. Then by finding the limit of (37) for t approaching infinity (assuming that the system (30) is forward complete (Definition 5)), and setting this limit smaller or equal to $\frac{\pi}{2}$, the set $\mathcal{S}$ can be found, for which (34) is defined as a recovery function (Definition 10).

$$\begin{aligned} \lim_{t\to\infty} \theta_0 + u_{min_0}(\frac{a^t}{ln(a)} - \frac{1}{ln(a)}) \leq \frac{\pi}{2} \\ \theta_0 - \frac{u_{min_0}}{ln(a)} \leq \frac{\pi}{2} \end{aligned} \tag{38}$$

From the equation above it can thus be deduced that (34) is a recovery function (by Definition 10) for $\theta_0 \leq \frac{\pi}{2} + \frac{u_{min_0}}{ln(a)}$ (38).

Then to provide set invariance for $\forall x_0 \in \mathcal{C}_1 \cap \cdots \cap \mathcal{C}_m \cap \mathcal{S}$ by Theorem 3, (19) should be satisfied. Which for the system (30) and recovery function (34) would be

$$\begin{aligned} \min_{t\in I(x_0)|t_0\leq t\leq t^*} \gamma\cos\theta(t) - V(t)\sin\theta(t)(u_{min_0}\cdot a^t) \\ +2V(t)\cos\theta(t) \\ +x(t) \geq 0 \end{aligned} \tag{39}$$

for which $x(t)$ can be described as

$$\begin{aligned} x(t) &= x_0 + \int_0^t \dot{x}(\zeta)d\zeta \\ &= x_0 + \int_0^t \dot{x}(\zeta)d\zeta \end{aligned} \tag{40}$$

Substituting (40), (31) and (37) in (39) results in the equation below.

$$\begin{aligned} \min_{t\in I(x_0)|t_0\leq t\leq t^*} \gamma\cos(\theta_0 + u_{min_0}(\frac{a^t}{ln(a)} - \frac{1}{ln(a)})) \\ -(V_0 + \gamma t)(u_{min_0}\cdot a^t)\sin(\theta_0 + u_{min_0}(\frac{a^t}{ln(a)} - \frac{1}{ln(a)})) \\ +2(V_0 + \gamma t)\cos(\theta_0 + u_{min_0}(\frac{a^t}{ln(a)} - \frac{1}{ln(a)})) \\ + \int_0^t (V_0 + \gamma\zeta)\cos(\theta_0 + u_{min_0}(\frac{a^\zeta}{ln(a)} - \frac{1}{ln(a)}))d\zeta \\ +x_0 \geq 0 \end{aligned} \tag{41}$$

For practical reasons the above ftCBF constraint is computed offline and then interpolated online during a simulation.

To make the computation a bit less computationally expensive and the constraint a bit more intuitive, x_0 in (41) is subtracted from both sides of the inequality sign and then both sides are multiplied with -1, which results in the following constraint

$$-\min_{t\in I(x_0)|t_0\leq t\leq t^*} (\psi_2(u^*) - x_0) \leq x_0 \tag{42}$$

which is easier to compute, as the constraint now does not have to be calculated offline for different starting conditions

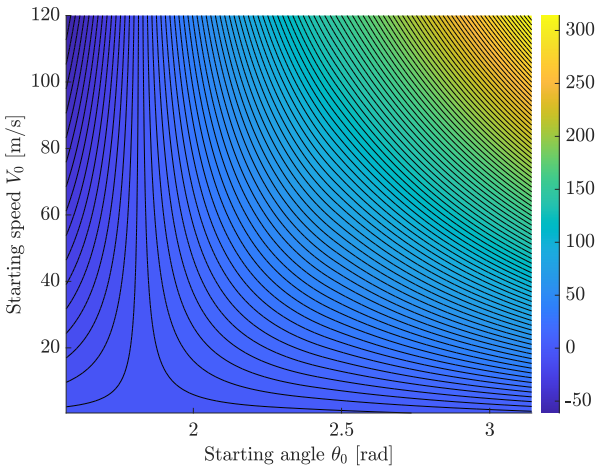


Fig. 7. This figure gives a slice of left hand side of the constraint of (42) for $u_{min_0} = -0.5$ [rad/s]. The color-bar on the right indicates the value of the left hand side of (42) and physically represents the required x-distance from the boundary the satisfy the constraint (42), which in turn is required to ensure safety for the system (30). For the computation of the constraint, $a = 0.9$ and $\gamma = 2$ were used. The lines that can be seen throughout the figure indicate different height levels of the function for visualization purposes.

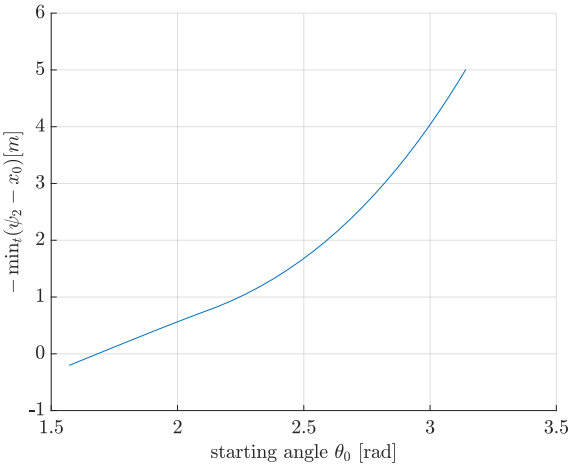


Fig. 8. This figure gives the function $-\min_{t\in I(x_0)|t_0\leq t\leq t^*}(\psi_2(u^*) - x_0)$ (left hand side of (42)) for $u_{min_s} = -0.5$ [rad/s] and $V_s = 0.5$ [m/s], and physically represents the required x-distance from the boundary to satisfy the constraint (42).

x_0, as the current value for $x(t)$ can be substituted for x_0 during the simulation to check if it satisfies the constraint. The new form of the constraint is also a bit more intuitive, as the value of the left hand side of (42) represents the required distance from the x-axis in order for (41) to be satisfied, and thus has a more physical meaning.

In Figure 7 and Figure 8 visualizations of the left hand side of the constraint (42) can be seen.

C. Simulation Setup

The simulation was set-up in Matlab and Simulink[7]. In Figure 9 the general layout of the setup that was used can be seen. The setup works with a joystick controller and gives

TABLE II
SIMULATION PARAMETERS FTCBF

x_0	y_0	θ_0	u_{min_0}	u_{max_0}	V_c	γ	a
20	1	π	-0.5	0.5	0.5	2	0.9

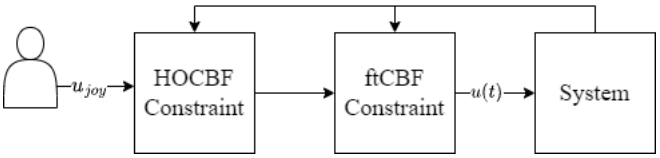


Fig. 9. General setup of the ftCBF implementation on the Dubin's Car system

input u_{joy} to the system if both the HOCBF and the ftCBF constraint are not active. If the ftCBF constraint is active the output that is given by the HOCBF constraint is overruled and the ftCBF provides the final input $u(t)$ to the system. The ftCBF thus has the highest hierarchy in the system, and always provides the input once it is activated, ensuring that the system will always remain in the safe set. The ftCBF constraint and the HOCBF constraint get state updates from the system. The HOCBF constraint is not necessary for safety, but in practice does provide a bit smoother behavior in approaching the boundary.

The parameters that are used in the simulation will be displayed in Table II.

D. Results

The main results of the simulation can be seen in Figure 11 and Figure 12, which uses xy-plots to show that the system (30) does not leave the safe set $\mathcal{C}_1$ (4), defined by $\psi_0 := x(t) \geq 0$. The defined safe set and the top down view of the system used in Figure 11 and Figure 12, can be visualized in Figure 6 for more clarity. The effect of the induced failure on $V(t)$, $u_{min}(t)$ and $u_{max}(t)$, can be seen in Figure 10. The Figures 11 and 12 show that the ftCBF controller is capable of providing safety for a vehicle in the case of a failure as the vehicle does not leave the safe set $\mathcal{C}_1$, defined by $\psi_0 := x(t) \geq 0$, during the simulation, regardless of when the failure occurs. This can be seen in Figure 11 where it shows a failure that happens well before the constraint would normally activate and in Figure 12 which shows an induced failure after the ftCBF constraint (42) has been activated.

The effect the ftCBF parameters γ and a on the maneuverability of the vehicle in nominal conditions (thus with no induced failure), is shown in Figure 13 and Figure 14. The effect of γ on the maneuverability is quite significant (Figure 13) as the ftCBF does not allow the system to go near to boundary $x = 0$ as much for higher values of γ. Therefore, a trade-off between safety and maneuverability has to be made as the higher the value for γ, the more prepared the system is for possible rapid changes in the system dynamics, but the less maneuverability it has. The effect of a on the maneuverability (Figure 14) in nominal conditions (thus without an induced failure) is not very significant in the range of $0.8 \leq a \leq 0.9$, as the boundary $x = 0$ is almost equally well approachable for

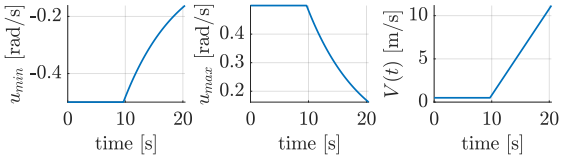


Fig. 10. This figure indicates what happens during an induced failure described by (31) and $u_{min}(t) = u_{min_0} \cdot a^t$. In the figure the failure is induced at around t=10. As can be seen the velocity $V(t)$ increases quickly and the control effectiveness decreases exponentially. The failure parameters used for this figure are $a = 0.9$ and $\gamma = 2$.

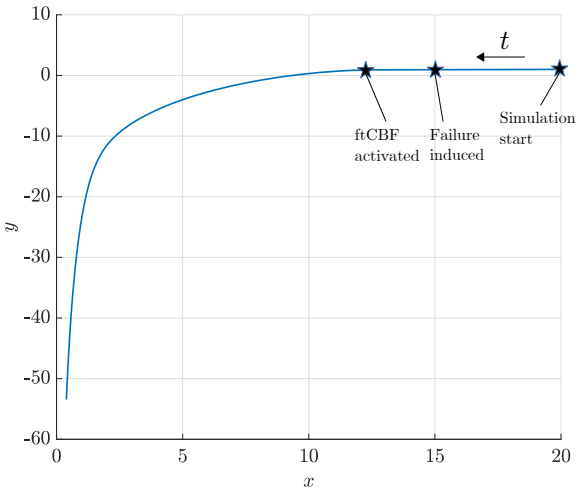


Fig. 11. This figure shows a xy-plot of the system (30). In the figure it can be seen that the ftCBF is able to keep the vehicle in the safe set $\mathcal{C}_1$, defined by $\psi_0 := x(t) \geq 0$ (Figure 6), with a failure resulting in changed system dynamics and changed control bounds (as can be seen in Figure 10). In this simulation there was no joystick input given and the failure was induced before the ftCBF constraint (42) was activated. The parameters used for the simulation can be found in Table II.

$0.8 \leq a \leq 0.9$. It should be noted however that if the value of a becomes too small the recovery function (34) is not defined for all $0 \leq \theta \leq \pi$, as can be seen from (38). From (38) it can be deduced that if the recovery function is to be defined for all $0 \leq \theta \leq \pi$, then $e^{\frac{-0.5}{0.5\pi}} \leq a \leq 1$ for $u_{min_0} = -0.5$ and more generally $e^{\frac{u_{min_0}}{0.5\pi}} \leq a \leq 1$. However, if $0 \leq a \leq e^{\frac{u_{min_0}}{0.5\pi}}$ the vehicle can still be made safe (provided that $\theta_0 \leq \frac{\pi}{2} + \frac{u_{min_0}}{ln(a)}$), but the maneuverability will be significantly impaired as $\theta(t)$ cannot reach the whole domain $0 \leq \theta(t) \leq \pi$.

E. Discussion

The results have shown that the ftCBF constraint is able to keep systems within their safe set, even when a failure is induced.

As can be seen in Figure 13, choosing higher values for γ has an effect on maneuverability, thus a trade-off can be made between safety and performance, where higher values of γ are more in favour of safety, and lower values of γ more favourable for performance.

Another point that should be mentioned, is that the computational time required to calculate the constraint will go

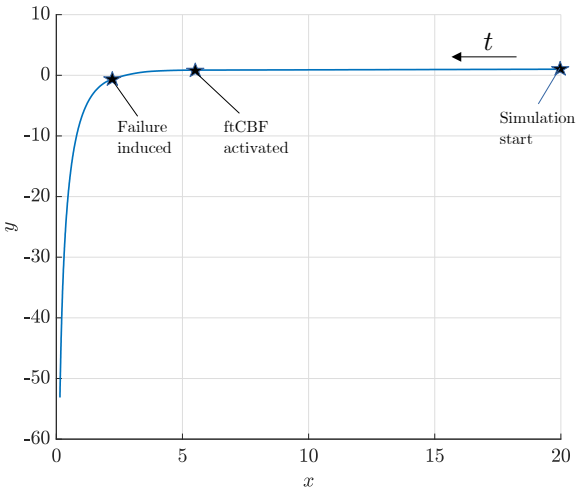


Fig. 12. This figure shows a xy-plot of the system (30). In this figure it can be seen that if the failure is induced after the ftCBF has been activated, the constraint is also able to make sure that the vehicle remains in the safe set $\mathcal{C}_1$ (4), defined by $\psi_0 := x(t) \geq 0$ (Figure 6). In this simulation there was no joystick input given. The parameters used in the simulation can be found in Table II.

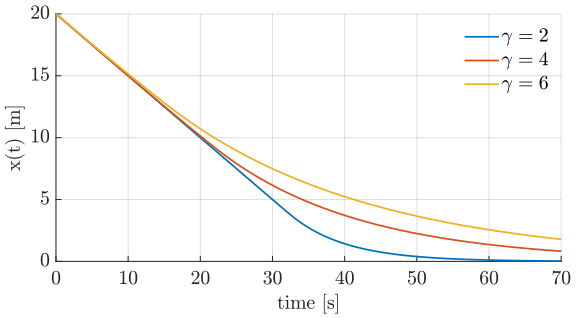


Fig. 13. This figure shows the effect of higher values of γ for (31) in the constraint (42) on vehicle maneuverability when no failure is induced. As can be seen, the faster the system dynamics are thought to be able to change (by setting a higher value for γ), the less maneuverability the vehicle has, as can be seen in the figure by the fact that the vehicle is not allowed to go near to boundary ($x = 0$) as much, as lower values of γ will allow. Setting the value for γ is thus a trade off between safety and maneuverability of the vehicle. For this figure no joystick input was given and $a = 0.9$ was used.

up exponentially for higher dimensional systems. For the Dubin's Car model only three dimensions had to be considered for the constraint, however for more complex systems like for instance full state drone models, the amount of states required are much higher than three, which would very likely require very long computational times. This phenomenon, called curse of dimensionality, also occurs in the related field of Reachability[6], for which reachable sets are solved from the Hamilton Jacobi PDE's with the level-set methods[8].

VI. Conclusion

In this work it has been shown that the adaptive Control Barrier Function[11] is not suitable for fault tolerant control, as it does not guarantee safety for changing system dynamics and changing control bounds. This paper has presented a new

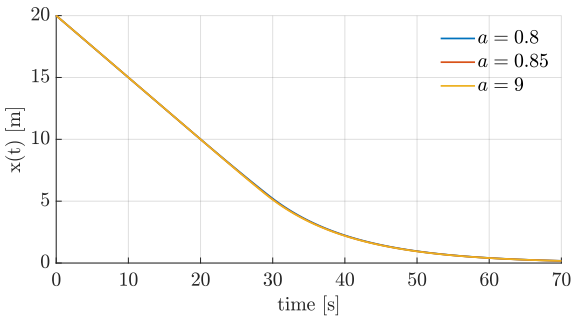


Fig. 14. This figure shows the effect of the value of a used in the failure function $u_{min}(t) = u_{min_0} \cdot a^t$, on the maneuverability of the vehicle. As can be seen in this figure, the changes in a do not effect the maneuverability of the vehicle in nominal conditions (where no failures are induced), thus it seems that a lower value of a would be most suitable, as this would mitigate more aggressive failures. However, a value of a that is lower then $a \leq e^{\frac{u_{min_0}}{0.5\pi}}$ (deduced from (38)), will result in the recovery function (34) to be not defined for all $\theta(t)$ on the domain $0 \leq \theta(t)\pi$, which will have a large impact on the maneuverability of the vehicle.

type of control barrier function called the fault tolerant Control Barrier Function (ftCBF), that is able to keep a system within its safe set $\mathcal{C}_1$ (4), with a failure event happening at any time during the simulation.

The failure function, used in the ftCBF, have design parameters that should be tuned according to a trade-off between safety and performance of the system.

For future work it is recommended to incorporate measurement noise and system dynamic parameters uncertainties within the ftCBF, to further increase the safety critical behavior of the ftCBF. Further work should also be done on mitigating the curse of dimensionality, such that the ftCBF could also be used practically for higher dimensional systems.

REFERENCES

[1] Ayush Agrawal and Koushil Sreenath. Discrete control barrier functions for safety-critical control of discrete systems with application to bipedal robot navigation. In *Robotics: Science and Systems*, volume 13. Cambridge, MA, USA, 2017.

[2] Aaron D. Ames, Samuel Coogan, Magnus Egerstedt, Gennaro Notomista, Koushil Sreenath, and Paulo Tabuada. Control barrier functions: Theory and applications. *2019 18th European Control Conference, ECC 2019*, pages 3420–3431, 6 2019.

[3] David Angeli and Eduardo D. Sontag. Forward completeness, unboundedness observability, and their lyapunov characterizations. *Systems Control Letters*, 38:209–217, 12 1999.

[4] F. Blanchini. Set invariance in control. *Automatica*, 35:1747–1767, 11 1999.

[5] Hassan K Khalil. *Nonlinear systems; 3rd ed.* Prentice-Hall, Upper Saddle River, NJ, 2002.

[6] John Lygeros. On reachability and minimum cost optimal control. *Automatica*, 40:917–927, 6 2004.

[7] MATLAB. *version R2021a*. The MathWorks Inc., Natick, Massachusetts, 2021.

[8] Ian M. Mitchell. The flexible, extensible and efficient toolbox of level set methods. *Journal of Scientific Computing*, 35:300–329, 6 2008.

[9] Quan Nguyen and Koushil Sreenath. Exponential control barrier functions for enforcing high relative-degree safety-critical constraints; exponential control barrier functions for enforcing high relative-degree safety-critical constraints. 2016.

[10] Wei Xiao and Calin Belta. Control barrier functions for systems with high relative degree. volume 2019-December, pages 474–479. Institute of Electrical and Electronics Engineers Inc., 12 2019.

[11] Wei Xiao, Calin Belta, and Christos G. Cassandras. Adaptive control barrier functions. *IEEE Transactions on Automatic Control*, 67:2267–2281, 5 2022.

[12] Jun Zeng, Bike Zhang, and Koushil Sreenath. Safety-critical model predictive control with discrete-time control barrier function. *Proceedings of the American Control Conference*, 2021-May:3882–3889, 5 2021.

Part II

Preliminary Report

Chapter 1

Introduction

This document is a deliverable for the course AE4010 Research Methodologies, given at the faculty of Aerospace Engineering of the Technical University of Delft. The deliverable of this course is a thesis proposal that serves as a preparation for the AE Control & Simulation MSc Thesis.

Control Barrier Functions (CBFs) are a powerful tool in the control theory community that provide a systematic way to ensure safety in dynamical systems. CBFs enable the design of control laws that guarantee the system trajectories remain within predefined safe sets, which is also called set invariance [1]. CBFs have also been used in conjunction with other control techniques, such as Model Predictive Control (MPC) [3], to provide robust and safe control strategies for dynamical systems.There exist multiple definitions of a CBF, but throughout this document and in the research the definition given in A.D. Ames et al. [4] will be used. The main goal of the proposed thesis is to investigate the behavior of CBFs on a subclass of time varying systems that have undergone event based changes (e.g. systems that are suddenly damaged).

The largest challenge in the study and application of the field of reachability is the so called curse of dimensionality. This hurdle makes it intractable to calculate the safe set for more than four dimensions[5]. This unfortunately also impacts the CBFs as they are based upon the safe set, which is calculated from reachability calculation methods. Contributing in reducing the computational complexity within the field of reachability is unfortunately not within the scope of this proposed thesis. Thus, to still be able to contribute to the field, smaller dimensional dynamical systems are chosen that are more tractable for calculations and simulation.

This document will go over a literature review in chapter 2, after which the research question of the proposed thesis is discussed in chapter 3. The proposed methodology will be discussed in chapter 4, and in chapter 5 the set-up of the simulations will be presented. In chapter 6 the data management and the verification & validation process of the project will be given. Hereafter, in chapter 7 the project planning will be discussed, and finally in chapter 8 the conclusions of this proposal will be presented.

Chapter 2

Literature Review

In this section the current literature that is relevant for the proposed thesis topic will be reviewed. This section will first go over the field of reachability and the calculation of the so called "safe set", as this will provide the necessary background for Control Barrier Functions (CBFs). The literature study was done mostly in the context of airplanes and therefore reachability was also researched mostly in this context. However, the methods described below are also applicable for dynamical systems in general. In the context of air vehicles the set of safe states is called the safe flight envelope, and for dynamic systems in general it is often called the safe set. There are many calculation methods available and also more being researched, but because of limited space, only the level set method will be discussed, as this is currently the most used method to calculate the safe set. After reachability has been discussed, the control barrier functions will be discussed.

2.1 Safe Flight Envelope/Safe Set Estimation

There are several techniques for the estimation of Safe Flight Envelopes. Most of the literature on this topic is primarily focused on the flight envelope estimation of aircraft. However, the techniques could also be applied to other vehicles as well. This section will briefly go over the most used method for calculating the Safe Flight Envelope, and briefly go over the field of reachability.

2.1.1 Reachability - Level Set Methods

Reachability can be solved as a minimal optimal control problem[6], which transforms the problem into the Hamilton Jacobi Bellman equations which can be solved numerically by the level set method[7]. A visualization of the forward and backward reachable set, and the intersection thereof, The Safe Flight Envelope, is given in Figure 2.1. The forward reachable set is defined as the set of states that can be reached from an initial set in t seconds. The backwards reachable set in this work is defined as the set of states for which the vehicle is still able to return to a target set of states. The backwards and forwards reachable set are often calculated from the trim envelope as target set or initial set, as this set is considered as an a-priori safe set[8][9].

The problem with this method of calculating the safe flight envelope for higher dimensions is that, the amount of calculations required increases dramatically with higher number of dimensions. This phenomenon is often called the curse of dimensionality in literature, and is what makes this method unfeasible for higher dimensional problems[5].

One method to reduce the computational complexity of the problem is to reduce the problem into several smaller dimensional problems with the use of time scale separation methods[8], where the fast and slow dynamics of the aircraft are separated. However as was shown in [5], this is only feasible for the safe flight envelope calculation for the slow dynamics.

Another method to reduce the computational complexity is by system decomposition[10][11]. The method works by separating the full system into several subsystems by means of state decomposition, for which the reachability sets are separately calculated. These subsystems have their own subsystem state space χ_i with their own backwards reachable set $\mathcal{L}_i$ defined in only that subspace. When for instance a full system is decomposed into two subsystems, with two

backwards reachable sets $\mathcal{L}_{x_1} \in \chi_{x_1}$ and $\mathcal{L}_{x_2} \in \chi_{x_2}$, the full system backwards reachable set can be found by intersecting the backwards projections of the subsystems backwards reachable sets:

$$\mathcal{L} = proj^{-1}(\mathcal{L}_{x_1}) \cap proj^{-1}(\mathcal{L}_{x_2}) \tag{2.1}$$

Where $proj^{-1}$ is the back projection operator, that projects the subsystem state space back into the full state space $\mathcal{Z}$:

$$proj^{-1}(x_i) = \{z \in \mathcal{Z} \mid proj_{\chi_{x_i}}(z) = x_i\} \tag{2.2}$$

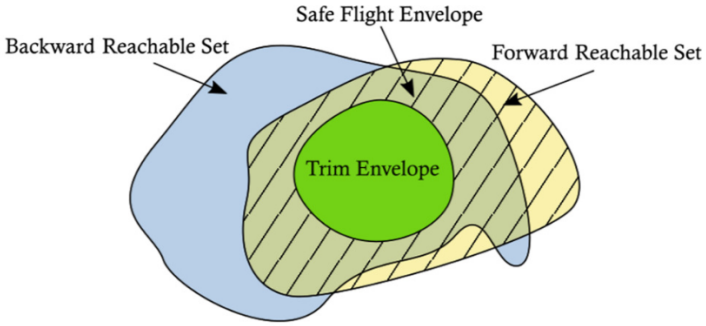


Figure 2.1: Safe Flight Envelope Definition from Backward and Forward Reachable Set[5]

Control Barrier Functions and Set Invariance

Once a safe set of a dynamical system has been determined, the invariance of this set can be guaranteed using CBFs. This subsection will be for the most part a summary of A.D. Ames et al. [4].

First the concept of Lyapunov functions and Control Lyapunov Functions (CLFs) will be discussed, after which CBFs will be further explained. Finally an example application of CBF implemented with a MPC controller will be given.

Throughout this document a general non-linear control affine system, given in Equation 2.3, will be assumed. With $x \in D \subset \mathbb{R}^n$ and $u \in U \subset \mathbb{R}^m$.

$$\dot{x} = f(x) + g(x)u \tag{2.3}$$

2.1.2 Lyapunov Functions

Lyapunov functions ensure asymptotic stability towards an equilibrium point in the system, which makes the system stable for a given control law or input, but not necessarily safe. This is done with the aid of a positive definite function that is equal to zero in the equilibrium point $V : D \subset \mathbb{R}^n \to \mathbb{R}_{\geq 0}$.

To ensure stability for a control law $k(x)$, the Lyapunov function must be driven to zero.

$$\exists u = k(x) s.t. \dot{V}(x, k(x)) \leq -\gamma(V(x)) \tag{2.4}$$

Where the derivative of the Lyapunov functions can be obtained by using the chain rule on Equation 2.3, resulting in:

$$\dot{V}(x, k(x)) = \nabla V(x) f(x) + \nabla V(x) g(x) k(x) \tag{2.5}$$

In Equation 2.5, γ is a class kappa function, that ensures asymptotic stability towards the equilibrium point. This is because the kappa function becomes greater when its argument is greater, resulting in a more negative derivative of the Lyapunov function.

The Control Lyapunov Function (CLF) is defined in the following way:

$$\inf_{u \in U} [\nabla V(x) f(x) + \nabla V(x) g(x) u] \leq -\gamma(V(x)) \tag{2.6}$$

Which allows the definition of the set of all stabilizing controllers:

$$K_{clf} = \{u \in U : \nabla V(x) f(x) + \nabla V(x) g(x) u \leq -\gamma(V(x))\} \tag{2.7}$$

2.1.3 Control Barrier Functions

The CBFs provide set invariance, and thus safety to the system for a given control law or input if the invariant set is also a safe set.

The invariant set $\mathcal{C}$ can be defined by the superlevel set of a function of a continuously differentiable function $h(x) : D \subset \mathbb{R}^n \to \mathbb{R}$. The interior of this invariant set is $\{x \subset D : h(x) > 0\}$ and the boundary of the set $\partial\mathcal{C}$ is $\{x \subset D : h(x) = 0\}$.

Similar as with the Lyapunov function, a control function can be described[12], which makes sure that the function $h(x)$ does not go below zero, and thus makes sure that the trajectory does not leave the invariant set $\mathcal{C}$:

$$\sup_{u \in U}[\nabla h(x)f(x) + \nabla h(x)g(x)u] \geq -\alpha(h(x)) \tag{2.8}$$

In the above equation α is a function of the extended class $_\infty$, which makes the derivative of the barrier function increase as the trajectory nears the boundary of the invariant set. At the boundary $\alpha(0) = 0$, and thus the derivative of the barrier function may only be greater or equal then zero.

Equation 2.8 allows for the definition of all controllers that render $\mathcal{C}$ to be invariant:

$$K_{cbf} = \{u \in U : \nabla h(x)f(x) + \nabla h(x)g(x)u \geq -\alpha(h(x)\} \tag{2.9}$$

So far for the CLF and the CBF the relative-degree has been assumed to be one, but for systems with a higher relative-degree, the above method will not work as the first order derivative of the function will not contain the input. To enforce arbitrarily high relative-degree safety constraints a type of CBF could be used called the Exponential Control Barrier Function[13], which uses an approach very similar to Non-linear dynamic inversion.

The Exponential Control Barrier Function (ECBF) works by reapplying the chain rule and finding higher order derivatives of $h(x)$ until, the input comes up. The r^{th} derivative of $h(x)$ will then be:

$$h^{(r)}(x, u) = L_f^r h(x) + L_g L_f^{r-1} h(x)u \tag{2.10}$$

In the above equation, L_f^r is the r^{th} Lie derivative (also called directional derivative) of the function f. Similar to Non Linear Dynamic Inversion $h^{(r)}(x, u)$ is then set equal to a virtual control input, $h^{(r)} = \mu$, which can be driven by a linear controller to control $h(x)$ and its derivatives.

A system can be defined with a control law to ensure that h(x) and its derivatives satisfy the safety constraints.

The system has the following state vector:

$$\eta_b(x) = \begin{bmatrix} h(x) \\ \dot{h}(x) \\ \vdots \\ h^{(r-1)} \end{bmatrix} = \begin{bmatrix} h(x) \\ L_f h(x) \\ \vdots \\ L_f^{(r-1)} h(x) \end{bmatrix} \tag{2.11}$$

Which results in the following system:

$$\dot{\eta}_b(x) = F\eta_b(x) + G\mu \tag{2.12}$$

With:

$$F = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}, G = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix} \tag{2.13}$$

If for the virtual control input μ, a control law $\mu \geq -K_\alpha \eta_b$ is chosen, the solution of the system will be:

$$h(x(t)) \geq Ce^{(F - GK_\alpha)t}\eta_b(x_0) \tag{2.14}$$

With:

$$C = \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix} \tag{2.15}$$

Then the function $h(x)$ is an ECBF if the equation below holds:

$$\sup_{u \in U} [L_f^r h(x) + L_g L_f^{r-1} h(x) u] \geq -K_\alpha \eta_b(x) \tag{2.16}$$

This only makes the set $\mathcal{C}$ invariant if K_α has certain properties. The properties of this row vector fall outside of the scope of this report, but the interested reader is advised to read A. D. Ames et al. [4] for more information about these properties.

2.1.4 Applications CBF

One possible application of CBF is using it in combination with a Model Predictive Control (MPC) controller, and Lyapunov Control Functions[3].

Model Predictive Control (MPC), predicts future states of a model, and assesses multiple of these predictions against cost functions. It tries to minimize this cost function by looking what input results in the minimal cost. There are multiple cost functions possible in MPC, but a common example is given in Equation 2.17[3]. In this equation, p is the terminal cost, and q is the stage cost, which together form the cost function. After an input has been found that results in the minimal value for the cost function, only the first step of this input is applied (this algorithm is set up in discrete time), after which the algorithm is performed again to find the input that results in the minimum of the cost function of the new current state.

$$J_t^* = \min_{\vec{u}_{t:t+N|t}} p(\vec{x}_{t+N|t}) + \sum_{k=0}^{N-1} q(\vec{x}_{t+k|t}, \vec{u}_{t+k|t}) \tag{2.17}$$

Discrete Control Barrier Functions (DCBF) can be used together with MPC, for the use in safety critical control. The discrete version of the CBF is given below in Equation 2.18.

$$\exists \vec{u} \mid h(\vec{x}_{k+1}) \geq (1-\gamma) h(\vec{x}_k), 0 < \gamma \leq 1 \tag{2.18}$$

DCBF can be combined with Discrete Lyapunov Control Functions (DLCF), to not only ensure system safety but also be able to stabilize the system with a feedback control law $\vec{u}$. The discrete version of the CLF is given in Equation 2.19.

$$\exists \vec{u} \mid V(\vec{x}_k + 1) \leq (1-\alpha) V(\vec{x}_k), 0 < \alpha \leq 1 \tag{2.19}$$

The DCBF and DLCF can be combined into one optimization program, which achieves the control objectives and guaranties safety[3]. The program was first introduced in [14], and is given in Equation 2.20 to Equation 2.22. In the formulation δ is a slack variable, that can be increased by the program if the Lyapunov function conflicts with the control barrier function.

$$\vec{u}_k^* =_{(\vec{u}_k, \delta)} \vec{u}_k^\intercal H(x) \vec{u}_k + l \cdot \delta^2 \tag{2.20}$$

$$\Delta V(\vec{x}_k, \vec{u}_k) + \alpha V(\vec{x}_k) \leq \delta \tag{2.21}$$

$$\Delta h(\vec{x}_k, \vec{u}_k) + \gamma h(\vec{x}_k) \geq 0 \tag{2.22}$$

Chapter 3

Research Question(s)

Throughout the research, the following main research question will be the central focus:

What will be the required properties of a CBF to guarantee set invariance of a (known) damage event based time-varying safe set?

With the research also the following sub-questions will be investigated:

- *Is it possible to make a time varying set invariant by using a time parametrization of a CBF that meets the constraints at each time instant?*
- *How to construct a continuous CBF that ensures set invariance for a known set that is changing over time?*
- *How to construct a time varying Exponential CBF, for systems with a relative degree higher then one?*
- *How to construct a time varying CBF of a system with actuation constraints?*
- *How to construct Discrete CBFs online from an online time-varying safe set, while ensuring set invariance between updates?*
- *How could a time varying CBF be applied in a safety critical MPC?*

Chapter 4

Theoretical Content/Methodology

The theoretical basis that will be used in the research is best summarized by A.D. Ames et al. [4], as it contains an extensive overview of theory and application of CBFs. In this section, the specific theorems that are presented in the paper that will be used as the basis of this research proposal will be given. Afterwards, the general Methodology of the research will be discussed.

4.1 Theoretical Content

The below theorems are copied form A.D. Ames et al. [4]. These theorems are the most important for the research and will be used as the basis of the research. Other theorems that will be used in the research will not be stated here, as providing the context of these theorems will be outside of the scope of this thesis proposal. The interested reader is advised to read the paper of A.D. Ames et al. [4] for the additional theorems, as well as the context wherein they are given. The context of the theorems provided below is given in Figure 2.1.1.

Theorem 1. Let $\mathcal{C} \subset \mathbb{R}^n$ a set defined as the superlevel set of a continuously differentiable function $h : D\mathbb{R}^n \to \mathbb{R}$. If h is a control barrier function on D and $\frac{\partial h}{\partial x}(x) \neq 0$ for all $x \in \partial\mathcal{C}$, then any Lipschitz continuous controller $u(x) \in K_{cbf}(x)$ for the system (3) renders the set $\mathcal{C}$ safe. Additionally, the set $\mathcal{C}$ is asymptotically stable in D.

Theorem 2. Let $\mathcal{C}$ be a compact set that is the super level set of a continuously differentiable function $h : D \to \mathbb{R}$ with the property that $\frac{\partial h}{\partial x} \neq 0$ for all $x \in \partial\mathcal{C}$. If there exists a control law $u = k(x)$ that renders $\mathcal{C}$ safe, i.e., $\mathcal{C}$ is forward invariant with respect to (3), then $h|_{\mathcal{C}} : \mathcal{C} \to \mathbb{R}$ is a control barrier function on $\mathcal{C}$.

4.2 Methodology

In this subsection the general methodology of the research will be explained. The research will first start with gaining intuition of constructing CBFs from safe sets. This will be done by using some toy example problem, e.g. simple double integrator dynamics, to find the safe set of the system and generating a CBF from that set.

From there two safe sets will be calculated before and after a failure mode for a toy problem, and two CBFs will be constructed from these safe sets, to gain insight how the construction of the CBF might change after the safe set changes. This will then eventually be done for more instances to investigate the transient behavior of going from the safe set before the fault event, to the safe set after the fault has occurred.

Then a continuous parameterization of a CBF for a simple toy problem is sought that satisfies the constraints for each time instant. After such a parametrization is found, a generalization is sought that will provide constraints for the definition of a time varying CBFs, that guarantees set invariance of a time varying set, and a more general algorithm is sought that can find these CBFs for a (class of) time varying safe sets. This step, will try to answer the hypothesis "There exist for every time varying safe set that can be made invariant, a continuous time varying CBF

that paired with a controller $u \in K_{cbf}$, will ensure that the set is invariant", and if achieved will be the main result of the research. This step in the research will most likely take the most amount of time, and will form the bulk of the research process.

After this step, the generalization may be extended to Exponential CBFs, such that it can also be applied to systems with a relative degree higher than one. Also this generalization may be extended to a system with actuation constraints.

If time allows it, the rest of the research will focus on the required properties of Discrete CBFs, such that it may applied with for instance a MPC controller.

The last phase will be the verification and validation of the theoretical results, if achieved. The process of this will be described in more detail in chapter 5.

Chapter 5

Set-up

To test the theoretic results, simulations will be performed to check if the CBFs will indeed provide set invariance for the fault event time varying safe sets. This will be done by using four example problems that will be simulated in Python and Matlab, because a lot of code and modules are already written in these two programming languages.

The simulation will be performed in the following way:

- The system is set up with a control law that is theoretically able to keep the system within the safe set, even after a scripted failure has occurred.
- The system is simulated with the scripted fault event.
- The states of the system are recorded throughout the simulation and compared with the safe set over time, to check whether or not it stays within the interior of the safe set.

The set-up will not answer the research question directly, but will aim verify the theoretical results of the research. Below the different example problems that will be used, will be described.

5.1 Double Integrator Dynamics

The double integrator dynamics, will be modeled as a simple linear system, where the input controls the second derivative of the system.

The system will be of the generic form:

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \end{bmatrix} = A \begin{bmatrix} x \\ \dot{x} \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u \tag{5.1}$$

5.2 Dubins Car

Dubins car is a simple non-linear system where the input controls the turn rate control. The equations of the system are given below:

$$\dot{x} = V cos(\theta) \tag{5.2}$$

$$\dot{y} = V sin(\theta) \tag{5.3}$$

$$\dot{\theta} = u \tag{5.4}$$

5.3 Inverted Pendulum

This is a slightly more complex non-linear system, that involves an inverted pendulum that is situated on a cart, as can be seen in Figure 5.1.

The system is described in Equation 5.5, with state vector $\eta = \begin{bmatrix} x & \dot{x} & \theta & \dot{\theta} \end{bmatrix}^T$.

$$\dot{\eta} = \begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{\theta} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} \dot{x} \\ -\frac{ml}{M+m}(\ddot{\theta}cos(\theta) - \dot{\theta}^2 sin(\theta)) + \frac{F}{m+M} \\ \dot{\theta} \\ \frac{ml}{I_p+ml^2}(gsin(\theta) - \ddot{x}cos(\theta)) \end{bmatrix} \tag{5.5}$$

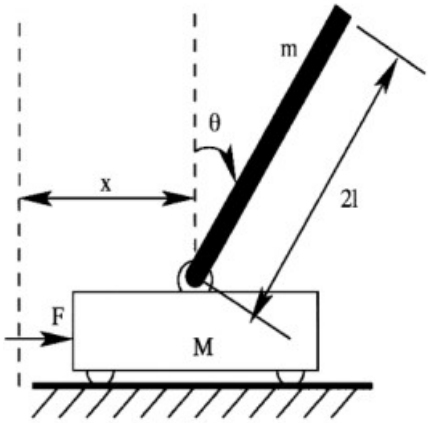


Figure 5.1: Cart with inverted pendulum[15]

Chapter 6

Results and Outcomes

The results of the set-up will mostly have a verification purpose for the theoretical results that may come from the research described in chapter 4. As the research is very theoretical in nature, it is difficult to exactly define which variables and parameters will be used.

In terms of data management, most of the code that is made available for use in the research group is distributed via GitHub, which has version control to keep data safe. For text, overleaf will be used, which stores its data on servers across the globe. For extra redundancy, from time to time a local backup will be made on a local PC. For other data, e.g. images, personal cloud services will be used to safely store the data.

Chapter 7

Project Planning

In this section the planning of the project is discussed. For an overview of the planning, see Figure 7.1.

In the planning the major milestones of the thesis project are:

- Midterm Review (will be on 5th of June)
- Green Light Review (will be on October 30th)
- Thesis Defence (will be around November 27th)

The milestones do not include the Kick-off, as this milestone has happened already. The bulk of the work will be from the midterm review to the green light review, where the work will be divided into two large chunks. These being, the theoretical research (approximately 3 months duration) and the verification (around 1 month in duration). Writing the draft before the green light review is estimated to be around 1 month of work. After the green light review, the preparation for the thesis defence will take approximately 1 month.

During this last phase of my studies I will not have any holidays planned, apart from a few days off here and there.

7.1 Gantt Chart

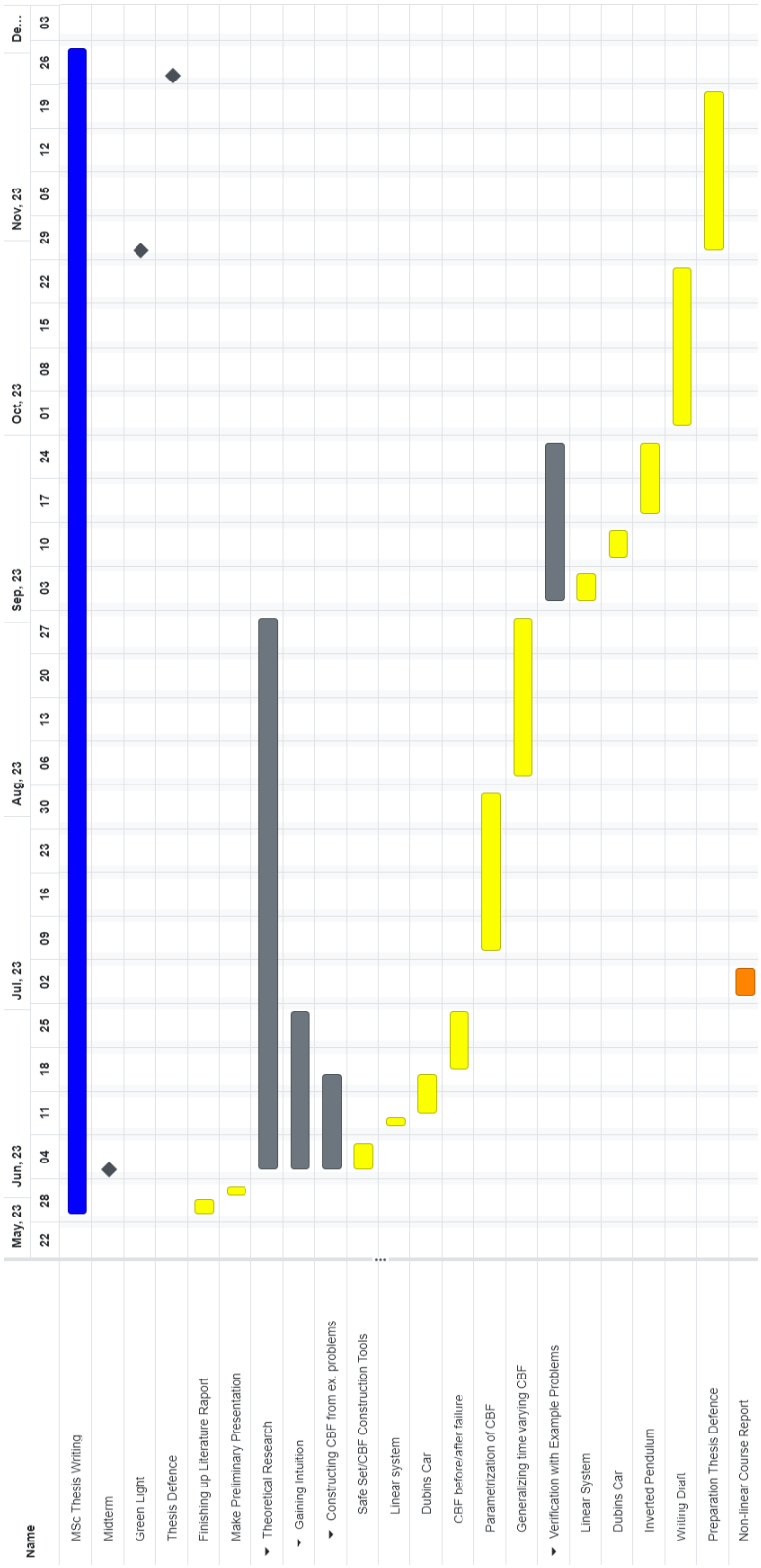


Figure 7.1: Gantt Chart of Thesis Project

Chapter 8

Conclusions

In this section the thesis proposal will be concluded. The main goal of the research is to contribute to the field of Control Barrier Functions, and aid in making dynamical systems more safe to operate, even when a failure happens to the system. To make a contribution, the thesis aims to find the required properties of Control Barrier Functions that can be used with time varying systems.

Part III

Appendices

Appendix A

Practical Application of the ftCBF for a Dubin's Car Simulation

This appendix chapter will go over some details concerning the practical application of the ftCBF on a Dubin's Car system.

A.1 Offline Computation of the ftCBF Constraint

The ftCBF constraint (from Part I (41)) is given below.

$$\begin{aligned}
&\min_{t\in I(x_0)|t_0\leq t\leq t^*} \gamma\cos(\theta_0+u_{min_0}(\frac{a^t}{ln(a)}-\frac{1}{ln(a)}))\\
&-(V_0+\gamma t)(u_{min_0}\cdot a^t)\sin(\theta_0+u_{min_0}(\frac{a^t}{ln(a)}-\frac{1}{ln(a)}))\\
&+2(V_0+\gamma t)\cos(\theta_s+u_{min_0}(\frac{a^t}{ln(a)}-\frac{1}{ln(a)}))\\
&+\int_0^t(V_0+\gamma\zeta)\cos(\theta_0+u_{min_0}(\frac{a^\zeta}{ln(a)}-\frac{1}{ln(a)}))d\zeta\\
&\geq -x_0
\end{aligned} \tag{A.1}$$

Before computing the value of the left hand side of the above constraint, it is practical to first calculate the value for t^* for which $\dot{\psi}_m \geq 0$, $\forall t \geq t^*$ (Part I Definition 10). To find this value, the equation for $\theta(t)$ for $u(t)=u^*$ could be used (Part I (37)):

$$\begin{aligned}
\theta(t) &= \theta_0+\int_0^t\dot{\theta}(\zeta)d\zeta\\
&= \theta_0+\int_0^t u(\zeta)d\zeta\\
&= \theta_0+u_{min_0}(\frac{a^t}{ln(a)}-\frac{1}{ln(a)})
\end{aligned} \tag{A.2}$$

For $t=t^*$ the angle θ should be $\theta(t^*)=\frac{\pi}{2}$:

$$\theta_0+u_{min_0}(\frac{a^{t^*}}{ln(a)}-\frac{1}{ln(a)})=\frac{\pi}{2} \tag{A.3}$$

and rearranging to above equation results in:

$$t^*=\frac{ln(\frac{ln(a)(\pi/2-\theta_0)}{u_{min_0}}+1)}{ln(a)} \tag{A.4}$$

For $\theta_0 = \pi$, $a = 0.9$ and $u_{min_0} = -0.5$, this results in $t^* \approx 3.815$, thus the left hand side of the constraint (A.1) for several θ_0 and v_0 was found in the interval $0 \leq t \leq 3.815$. u_{min_0} is kept constant, because it assumed that the failure will keep diminishing the control effectiveness until it is effectively zero, starting from the nominal value of u_{min}. If this was not the case, and the control effectiveness could fail within several time intervals, then for every u_{min_0} a new t^* should be computed. However, eventually for a value of u_{min_0} that is too low, the recovery function u^* will not be defined for the entire region $0 \leq \theta(t) \leq \pi$. This is because the recovery function is only defined for

$$\begin{aligned} \lim_{t\to\infty} \theta_0 + u_{min_0}(\frac{a^t}{ln(a)} - \frac{1}{ln(a)}) &\leq \frac{\pi}{2} \\ \theta_0 - \frac{u_{min_0}}{ln(a)} &\leq \frac{\pi}{2} \end{aligned} \tag{A.5}$$

and for $\theta_0 = \pi$ and $a = 0.9$, this results in $u_{min_0} \leq -0.165$. And thus if $u_{min_0} \geq -0.165$, t^* would be infinite for $\theta_0 = \pi$ and $a = 0.9$. A value less but close to -0.165 would also be impractical, as the recovery time t^* would be enormous. For this application the failure is assumed to be complete (i.e. until the control effectiveness is effectively zero), and occurring in only one time interval, and thus in the simulation $u_{min_0} = -0.5$ and is not allowed to change during the simulation.

The exact code that was used for the computation of the left hand side of $(A.1)$ can be seen in Figure A.1. In this code the minimum is found for $0 \leq t \leq t^*$.

```
%% Creating the ndgrid

dt=0.1; %Time step
theta_start=linspace(pi/2,pi,100); %Starting values of theta
V_start=linspace(Vc-0.1,120,100); %Starting values of V
umin_start=linspace(umin-0.01,0,100); %Starting values of umin
time = [0:dt:4]; %Time vector

[X1,X2,X3,X4]=ndgrid(theta_start,V_start,umin_start,time); %ND
grid including time

[XX1,XX2,XX3]=ndgrid(theta_start,V_start,umin_start); %ND grid not
including time


%% The actual calculation of the function

thetaf=X1+X3.*((a.^X4)/log(a)-1/log(a)); %Theta(t), rewritten in
terms of theta_start, time and umin_start
Vcf=X2+V_worst_slope*X4; %V(t), rewritten in V_start and time
uf=X3.*a.^X4; %Recovery function, which is a function of uminstart
and failure constant a

fun1=Vcf.*cos(thetaf); %Part of the psi2 constraint, for which the
integral has to be taken

testvar = dt*cumtrapz(fun1,4); %Integral of V(t)Cos(t)

f=V_worst_slope*cos(thetaf)-Vcf.*uf.*sin(thetaf)
+2*Vcf.*cos(thetaf)+testvar; %Minimum should be taken from this
equation

psim = min(f,[],4); %The LHS of constraint function
```

Figure A.1: The exact code that was used for generating the constraint function values offline, stored in the variable psim. In this figure V_worst_slope is the same as the constant γ, used in the thesis paper. In the offline calculation the most computationally expensive part of the calculation is the integral $\int_0^t V(t)\cos\theta(\zeta)d\zeta$.

A.2 Online Implementation of the ftCBF Constraint

For the online implementation, the constraint that was calculated for several values of θ_0, v_0 and stored in a variable called psim, was interpolated in real time using the Matlab[16] function interpn. To interpolate the current value of the constraint, only the current value for $\theta(t)$ and $V(t)$ should be given as an input to the function. The exact code of the real time interpolation of the ftCBF constraint is given in Figure A.2.

```
function [y, constraint, out] = fcn(theta, umin, Vc,
psim, XX1, XX2, XX3, x0, umax)

constraint = 0;
if theta<0
    theta =rem(theta, -2*pi)+2*pi;
else
    theta=rem(theta,2*pi);

end

if theta>pi
    theta = 2*pi-theta;
    out = umax;
else
    out = umin;
end


y = interpn(XX1, XX2, XX3, psim, theta, Vc, umin);

if x0<=-y
    constraint=1;
end
```

Figure A.2: This is the code of the ftCBF constraint that is used in real time in the simulation. This code works for the entire domain of $0 \leq \theta(t) \leq 2\pi$, because when $\theta > \pi$ the upper control bound is used instead of the lower control bound, but the constraint function remains the same. For the interpolation of the offline calculated constraint, the Matlab[16] function interpn was used. The variables XX1,XX2 and XX3 are ndgrid variables, for which the construction can be seen in Figure A.1.

Appendix B

ftCBF Application: Full State Drone Model

In this appendix chapter the preparatory steps required for the application of the ftCBF on a full state drone model will be discussed.

B.1 Equations of Motion

The equations of motion for the full state drone model are adopted from Sun et al. [17]. The equations of motion uses two coordinate frames, being the inertial frame $\mathcal{F}_I = \{O_I, x_I, y_I, z_I\}$ and the body frame $\mathcal{F}_B = \{O_B, x_B, y_B, z_B\}$. As is the convention, z_B will be pointed downwards and x_B will be pointed forwards. The direction of y_B can be derived from the right handed coordinate system. The equations of motion are given below:

$$\begin{aligned} \dot{\vec{P}}^I &= \vec{V}^I \\ m_v \dot{\vec{V}}^I &= m_v \vec{g}^I + \vec{R}\vec{F}^B \\ \dot{\vec{R}} &= \vec{R}\vec{\Omega}^B_\times \\ \vec{I}_v \dot{\vec{\Omega}}^B &= -\vec{\Omega}^B_\times \vec{I}_v \vec{\Omega}^B + \vec{M}^B \end{aligned} \tag{B.1}$$

In the above equations of motion, $\vec{P}$ is the position vector of the center of mass of the vehicle, and $\vec{V}$ is the velocity vector of the center of mass of the vehicle. $\vec{I}_v$ denotes the inertia matrix of the vehicle including the motors, and $\vec{g}$ is the gravity vector. $\vec{R}$ is the transformation matrix from $\mathcal{F}_B$ to $\mathcal{F}_I$. $\vec{F}$ and $\vec{M}$ are force and moment vectors respectively. The subscript $[\cdot]_\times$ denotes the skew symmetric matrix and is related to the cross product.

Furthermore, the resultant force $\vec{F}^B$ and the resultant moment $\vec{M}^B$ are given below:

$$\vec{F}^B = \begin{bmatrix} 0 \\ 0 \\ -\bar{\kappa}\sum_{i=1}^{4} \omega_i^2 \end{bmatrix} + \vec{F}_a \tag{B.2}$$

$$\begin{aligned} \vec{M}^B = \bar{\kappa} \begin{bmatrix} b\sin\beta & -b\sin\beta & -b\sin\beta & b\sin\beta \\ b\cos\beta & b\cos\beta & -b\cos\beta & -b\cos\beta \\ \sigma & -\sigma & \sigma & -\sigma \end{bmatrix} \begin{bmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{bmatrix} \\ + \begin{bmatrix} I_p q(\omega_1 - \omega_2 + \omega_3 - \omega_4) \\ -I_p p(\omega_1 - \omega_2 + \omega_3 - \omega_4) \\ I_p(\dot{\omega}_1 - \dot{\omega}_2 + \dot{\omega}_3 - \dot{\omega}_4) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ -\gamma r \end{bmatrix} + \vec{M}_a \end{aligned} \tag{B.3}$$

In the above equation, $\bar{\kappa}$ is a trust coefficient valid in the hovering condition, σ is the ratio between thrust and drag coefficient of the rotor, b and β are geometry parameters (shown in

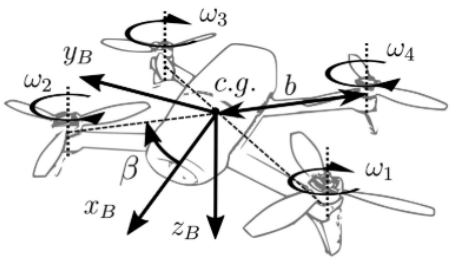


Figure B.1: A depiction of the drone parameters[17]

Figure B.1) and I_p is the moment of inertia of the rotor around the rotation axis and $\vec{F}_a$ and $\vec{M}_a$ are the additional aerodynamic resultant force and moment respectively. The variables p, q and r are the roll, pitch and yaw rates respectively.

B.2 Simulation Model

The model that was used for experimentation was adapted from the model made by Sun et al. [17] and is publicly available. The model is controlled by a Fault Tolerant Incremental Nonlinear Dynamic Inversion (INDI) controller, that allows for the control of a quadcopter with two opposing defect rotors.

To be able to use the model for the purposes of applying the ftCBF, a flaw in the system that caused instability was fixed. This was done by first prioritizing the commanded attitude before the commanded z was followed, as first the model would simultaneously try to recover the height of the vehicle and the attitude, which led to instability.

Another problem with the model was that the controller could not make a distinction between a commanded trust vector that was pointed up, and a commanded trust vector that was pointed down. This is due to the nature of the fault tolerant controller that only tracks the lateral components of the desired trust vector if two of the four rotors have failed. This is done because with two propellers only 2 outputs can be followed. This however results in the fact that the quadcopter will not be able to make a proper distinction between up and down, which makes it very difficult to develop a recovery function.

B.3 Recovery Function

For the application of the ftCBF, a recovery function must be found. This function does not have to be optimal, but must provide a way to increase the distance from the boundary eventually.

If the boundary in this case is assumed to be $\psi_0 := -z(t) \geq 0$ (direction of z^I axis is downwards), then a valid recovery strategy would be to first recover the attitude of the vehicle such that the thrust vector is pointed up, and afterwards recover the altitude by setting the propellers to an equal amount of maximum possible thrust.

On the current model however, such a recovery function is not possible as the attitude is not fully controllable. This is in one part due to the inability of the current INDI controller to differentiate up from down, and to another part due to what seems to be instabilities in the controller or system model.

Once the attitude can be controlled over the whole domain (meaning all possible starting attitudes and rotational rates), a recovery function as mentioned above would be easy to implement, which allows the ftCBF constraint to be calculated.

B.4 Discussion

The current way of application of the ftCBF on the full state drone model would not be practically usable, because the full ftCBF constraint cannot be computed accurately due to the curse of dimensionality.

Before a practical application of the ftCBF on a full state drone model can be made, first the curse of dimensionality should be mitigated and secondly the attitude of the vehicle must be made controllable over its entire domain.

Bibliography

[1] F. Blanchini. "Set invariance in control". In: *Automatica* 35 (11 Nov. 1999), pp. 1747–1767. ISSN: 0005-1098. DOI: 10.1016/S0005-1098(99)00113-2.

[2] Wei Xiao, Calin Belta, and Christos G. Cassandras. "Adaptive Control Barrier Functions". In: *IEEE Transactions on Automatic Control* 67 (5 May 2022), pp. 2267–2281. ISSN: 15582523. DOI: 10.1109/TAC.2021.3074895.

[3] Jun Zeng, Bike Zhang, and Koushil Sreenath. "Safety-Critical Model Predictive Control with Discrete-Time Control Barrier Function". In: *Proceedings of the American Control Conference* 2021-May (May 2021), pp. 3882–3889. ISSN: 07431619. DOI: 10.23919/ACC50511.2021.9483029.

[4] Aaron D. Ames et al. "Control barrier functions: Theory and applications". In: *2019 18th European Control Conference, ECC 2019* (June 2019), pp. 3420–3431. DOI: 10.23919/ECC.2019.8796030.

[5] H. N. Nabi et al. "Effects of structural failure on the safe flight envelope of aircraft". In: *Journal of Guidance, Control, and Dynamics* 41 (6 Feb. 2018), pp. 1257–1275. ISSN: 15333884. DOI: 10.2514/1.G003184/ASSET/IMAGES/LARGE/FIGURE29.JPEG. URL: https://arc.aiaa.org/doi/10.2514/1.G003184.

[6] John Lygeros. "On reachability and minimum cost optimal control". In: *Automatica* 40 (6 June 2004), pp. 917–927. ISSN: 0005-1098. DOI: 10.1016/J.AUTOMATICA.2004.01.012.

[7] Ian M. Mitchell. "The flexible, extensible and efficient toolbox of level set methods". In: *Journal of Scientific Computing* 35 (2-3 June 2008), pp. 300–329. ISSN: 08857474. DOI: 10.1007/S10915-007-9174-4/METRICS. URL: https://link.springer.com/article/10.1007/s10915-007-9174-4.

[8] Thomas J.J. Lombaerts et al. "Safe maneuvering envelope estimation based on a physical approach". In: *AIAA Guidance, Navigation, and Control (GNC) Conference* (2013). DOI: 10.2514/6.2013-4618. URL: http://arc.aiaa.org.

[9] Thomas Lombaerts et al. "On-Line Safe Flight Envelope Determination for Impaired Aircraft". In: *Advances in Aerospace Guidance, Navigation and Control* (2015), pp. 263–282. DOI: 10.1007/978-3-319-17518-8_16. URL: https://link.springer.com/chapter/10.1007/978-3-319-17518-8_16.

[10] Mo Chen, Sylvia Herbert, and Claire J. Tomlin. "Exact and Efficient Hamilton-Jacobi-based Guaranteed Safety Analysis via System Decomposition". In: *Proceedings - IEEE International Conference on Robotics and Automation* (Sept. 2016), pp. 87–92. ISSN: 10504729. DOI: 10.48550/arxiv.1609.05248. URL: https://arxiv.org/abs/1609.05248v1.

[11] Mo Chen et al. "Decomposition of Reachable Sets and Tubes for a Class of Nonlinear Systems". In: *IEEE Transactions on Automatic Control* 63 (11 Nov. 2018), pp. 3675–3688. ISSN: 15582523. DOI: 10.1109/TAC.2018.2797194.

[12] Aaron D. Ames et al. "Control Barrier Function Based Quadratic Programs for Safety Critical Systems". In: *IEEE Transactions on Automatic Control* 62.8 (2017), pp. 3861–3876. DOI: 10.1109/TAC.2016.2638961.

[13] Quan Nguyen and Koushil Sreenath. "Exponential Control Barrier Functions for enforcing high relative-degree safety-critical constraints; Exponential Control Barrier Functions for enforcing high relative-degree safety-critical constraints". In: (2016). DOI: 10.1109/ACC.2016.7524935.

[14] Ayush Agrawal and Koushil Sreenath. "Discrete control barrier functions for safety-critical control of discrete systems with application to bipedal robot navigation." In: *Robotics: Science and Systems*. Vol. 13. Cambridge, MA, USA. 2017.

[15] Xu Chen. *Inverted Pendulum Modeling*. 2016. URL: `https://faculty.washington.edu/chx/teaching/me547/pendulum_on_a_cart_prob.pdf`.

[16] MATLAB. *version R2021a*. Natick, Massachusetts: The MathWorks Inc., 2021.

[17] Sihao Sun et al. "Incremental Nonlinear Fault-Tolerant Control of a Quadrotor with Complete Loss of Two Opposing Rotors". In: *IEEE Transactions on Robotics* 37 (1 Feb. 2021), pp. 116–130. ISSN: 19410468. DOI: `10.1109/TRO.2020.3010626`.