BUDAPEST UNIVERSITY OF TECHNOLOGY AND ECONOMICS
FACULTY OF MECHANICAL ENGINEERING
DEPARTMENT OF APPLIED MECHANICS

László Gácsi

TDK

Safety Critical Control of mechanical systems with input constraint

Consultant:
Ádám Kiss, PhD
Assistant research fellow

Supervisor:
Máté Zoltán Tabajdi
Consultant

Budapest, 2023

Contents

1 Introduction **1**

1.1 Problem formulation 1

1.2 Literature 2

1.3 Paper's objective 3

2 Safety Critical Control with input constraint **4**

2.1 Theoretical background of Safety Critical Control 4

2.2 Relationship with Lyapunov-function 6

2.3 Theory of Backup set method 8

3 Construction of backup set and backup controller **12**

3.1 Forward invariance 12

3.1.1 Stabilization of operation points 13

3.1.2 Ensuring input constraint 13

3.1.3 Backup set constraint 15

4 Applications **16**

4.1 Scalar equation 16

4.1.1 Backup set and backup controller construction 16

4.1.2 Forward prediction 18

4.1.3 Results 19

4.2 Inverted pendulum 21

4.2.1 Backup set and backup controller construction 22

4.2.2 Ellipse constraint 23

4.2.3 Results 24

4.3 Vehicle braking 28

4.3.1 Mechanical model 29

4.3.2 Tire model 31

4.3.3 Driver model 32

4.3.4 Linearized system 32

4.3.5 Backup set and backup controller construction 34

4.3.6 Results 35

5 Summary **40**

5.1 Results 40

5.2 Future plans 40

Reference **42**

Acknowledgments

Supported by the ÚNKP-23-2-I-BME-335 New National Excellence Program of the Ministry for Culture and Innovation from the source of the National Research, Development and Innovation Fund.

KULTURÁLIS ÉS INNOVÁCIÓS MINISZTÉRIUM

NATIONAL RESEARCH, DEVELOPMENT AND INNOVATION OFFICE HUNGARY

1 Introduction

Dynamic systems are primarily expected to operate safely. However, what exactly we mean by safety is generally not a well-defined concept, even in engineering. In mechanical engineering, safety is most often considered in the context of operating machines at their operation point: for example, keeping an inverted pendulum in vertical position, or path following of an autonomous vehicle. In these situations, the most commonly used control tool is stabilization around the equilibrium point via linearization. Another much more up-to-date and sophisticated method is the Safety Critical Control, one of the most popular in recent years, which provides a completely different approach to safety compared to stabilization methods. Its starting point is that safety is interpreted as forward invariance of sets, in other words, it tries to keep the state variables (or a function of them) of the dynamical system within a predefined subset in the phase space, namely in the safe set. When looking for a relationship between these two methods, it becomes clear that the latter one is actually an extension of the former one, since the operation point has been extended to a set. Recognising its potential benefits, it is worth making use of it in more and more areas, but without losing sight of the real-life circumstances.

1.1 Problem formulation

One of the greatest limiting factors in the practical implementation of theoretical calculations is the input constraint: we are frequently confronted with situations where the magnitude of the predetermined input signal cannot be reached in real life due to saturation of the actuators. By looking at the control law, we cannot affect the intervention because it is always an output value, since the control loop is closed by some sort of feedback. The problem of input constraint frequently arises with any control method, also in the case of Safety Critical Control, it has a significant impact specifically on safety, as the forward invariance property is violated and safety can no longer be guaranteed. The general solution to input constraint cannot be the simple statement "let's make bigger and stronger motors", because in the first place, this idea would always lead to even bigger actuators or the modification of the mechanical system itself, which might not be practical nor realizable. On the other hand, we also have to consider the physical limitations, because for example, it is physically impossible to go down below zero (or even close to it) for a controller that operates based on absolute temperature or pressure.

A real-life case where this problem comes from is vehicle braking on a surface with asymmetric friction, when the vehicle can then easily spin due to the yaw torque resulting from the difference in braking forces on the two sides. But the driver must be assisted in order not to lose control of the vehicle, by an appropriate braking force, and the following dilemma must be addressed: if the difference in braking force is at its maximum, the braking distance is minimal, but the probability of a spin-out is high; on the other hand, if the difference in braking force is eliminated, there is no chance of a spin-out, but the

braking distance of the vehicle is drastically increased, thus exposing the driver to another danger and without a doubt braking distance should be the priority during emergency braking. The input constraint is present naturally, as braking forces due to traction have a maximal value which cannot be exceeded. In an earlier TDK and paper I have already studied vehicle braking on a surface with asymmetric friction but neglecting input constraint [1], [2]. It was found that, in most cases, Safety Critical Control required more braking force than was available. This is the motivation for the present paper, which aims to take into account the input constraint.

Overall, both finding a suitable controller that interacts always at the perfect time and a safe set under which the Safety Critical Control is feasible is quite challenging. All of these are propositions which are not taken into account by the "classical" Safety Critical Control, and thus need to be improved.

1.2 Literature

The notion of Control Barrier Function appeared first in [3] in 2014, and since then it has been growing enormously establishing the theory that has been evolving over the years and will be presented in Section 2. Along with the input constraint, there are other real-life problems that need to be taken into consideration when developing Safety Critical Control. Typically, one of these are the robustness [4], where the robustness of control barrier functions under model perturbation is investigated. Another real-life problem to explain is the time delay that can appear both in the control input or in the state. To handle this, the Control Barrier Functionals [5] has been introduced.

The issue of input constraints has been a concern for researchers since the origin of safety-critical control and there is still no universal solution. The literature is still relatively rich in attempts that either rely on their own ideas or on other control methods'. The idea from [6] realizes the fact that the size of the original safe set must be decreased in general and provides its own method for preparing even smaller subsets. However, there is no advice for how many subsets are needed until reaching the final, control invariant subset, and for how to figure out new *class* $\mathcal{K}_\infty$ functions which have significant effect on the successive safe sets. Another approach from [7] is again investigating the reduction of the safe set using sum-of-squares (SOS) programming but according to the authors, this method can be only used in polynomial systems. The next example [8] demonstrates the so-called Intergal Control Barrier Function which uses forward integration and is able to define safe sets for states and inputs separately. With that, unfortunately, instead of nominal (desired) control, we can only provide its derivative with respect to time. The reason why this is a problem is that, returning back to the inverted pendulum problem, if we specify a constant intervention as a nominal control, its derivative is obviously zero, and from that integrating back we get some constant that is unlikely to be the same as our preferred value.

The method from which this paper is inspired is the backup set method; the backup set itself was first mentioned in paper [9]. The idea behind the method is to examine how the future state of the dynamical system evolves under a controller that obeys an input constraint, and then determine the actual intervention based on this. In [10] can be found the case of an inverted pendulum, where the viability kernel (which is the largest control invariant subset of the safety set) and the prediction time are examined. However, it does not deal with such basic things as how to choose backup set and controller or what to do in the case of an input constraint with non-symmetric bounds.

1.3 Paper's objective

In this contribution, the main goal is to explore the backup set method in as much depth as possible and to give a generalized solution for implementing the backup set and backup controller using the well-known Lyapunov function. Although a similar Lyapunov-based backup sets can be found in the previously mentioned literature [10], it is not yet stated why this is an efficient method and how it can be algorithmically applied to any mechanical system.

In what follows, In Sect. 2, the theoretical background of safety and its relation to the Lyapunov function is considered in detail, which is necessary because the construction of the backup set method presented later is based on it. Then, the Backup set method is discussed. After that, in Section 3, a custom method will be presented to help construct backup sets and backup controllers. In addition, the steps of the methodology are provided in detail. In Section 4, after an overview of the Safety Critical Control and Backup set methods, three different application examples are presented: a simple scalar equation to help understand more easily the backup set method. Then, an inverted pendulum is demonstrated to show the high potential of the new method. Finally, the previously mentioned vehicle braking, with its even more complex dynamics, will be discussed.

2 Safety Critical Control with input constraint

With the Safety Critical Control, developers are able to provide a state-of-the-art solution to a wide range of mechanical problems that were before unimaginable. The method has proven its applicability in obstacle avoidance tasks for robots, drones, road vehicles [ref] and lots of other controllable moving objects. As I mentioned in the introduction, the present controller introduces a completely different approach to the notion of safety as opposed to stabilization, namely forward invariance of sets. Moreover, it has the enormous advantage that linearization is not required, it can be applied to any nonlinear dynamical system. This is due to the fact that its theoretical background is based on Lyapunov functions, which will be discussed in detail in Section 2.2.

2.1 Theoretical background of Safety Critical Control

Before proceeding to the controller, it is necessary to first interpret the dynamical system itself, which is assumed to be in affine form:

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})\mathbf{u}\,, \tag{2.1}$$

where $\mathbf{x}(t) \in \mathbb{R}^n$ is the state vector, $\mathbf{f}$: $\mathbb{R}^n \to \mathbb{R}^n$ and $\mathbf{g}$: $\mathbb{R}^n \to \mathbb{R}^{n\times m}$, which are the vector and matrix functions of the dynamical model, and $\mathbf{u} \in \mathbb{R}^m \in \mathcal{U}$ is the input signals. With the system being affine, the intervention signal (or its vector) can be linearly detached from the right-hand side of equation (2.1). This is key for the subsequent design of the control signal, and on the other hand, it is also typical for real dynamical systems, where most of the time the force or torque input appears linearly in the equation of motion. We give a mathematical formulation of the safety by an superlevel set of a function h, denoted by $\mathcal{S}$ and assumed to be continuously differentiable. With that:

$$\mathcal{S} = \{\mathbf{x} \in \mathbb{R}^n : h \geq 0\}\,, \tag{2.2}$$

$$\partial\mathcal{S} = \{\mathbf{x} \in \mathbb{R}^n : h = 0\}\,, \tag{2.3}$$

and hence we refer to $\mathcal{S}$ as the safe set, the bound of safety is $\partial\mathcal{S}$ and h is called *Control Barrier Function* (CBF) can be seen in Fig. 2.1. As we can see, it is relatively easy to find safe sets, and in fact a given region can usually be described by more than one safe set. This leads to the key of this method, namely to forward invariance, is shown by the following inequality:

$$\dot{h}(\mathbf{x}, \mathbf{u}) + \alpha(h(\mathbf{x})) \geq 0\,, \tag{2.4}$$

where α is a so-called *class* $\mathcal{K}_\infty$ function, which has two properties: $\alpha(0) = 0$ and is a strictly monotone increasing function. With these properties, it is evident that if we are on the boundary of safety ($\partial\mathcal{S}$), the previous inequality is:

$$\underbrace{\dot{h}(\mathbf{x}, \mathbf{u})}_{\geq 0} + \underbrace{\alpha(h(\mathbf{x}))}_{=0} \geq 0\,, \tag{2.5}$$

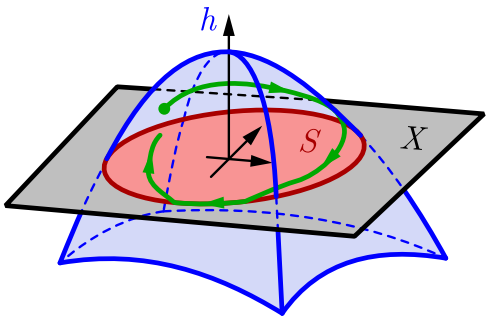

Figure 2.1: Graphical meaning of the safe set [11]

so if the time derivative of the Control Barrier Function is non-negative, then h increases from 0 to positive, i.e., from its boundary the trajectory of the system heads towards the center of the safe set, thus forward invariance is satisfied. The essence of the control is to find a suitable intervention $\mathbf{u}$ for this condition (2.4), which is generally possible by an optimization problem, also known as Quadratic Programming (QP):

$$\begin{aligned} \mathbf{u}(\mathbf{x}) = &\arg\min_{\mathbf{u}} \quad \frac{1}{2}||\mathbf{u} - \mathbf{k}_\mathrm{d}(\mathbf{x})||^2\,, \\ &\text{s.t.} \quad \dot{h}(\mathbf{x},\mathbf{u}) + \alpha(h(\mathbf{x})) \geq 0\,. \end{aligned} \tag{2.6}$$

We merely seek the minimum deviation of the intervention $\mathbf{u}$ from a known desired controller $\mathbf{k}_\mathrm{d}(\mathbf{x})$ such that forward invariance is satisfied. In the optimization problem, the time derivative of the CBF is determined by the chain rule similar to the Lyapunov function, and then substituting the dynamical system from (2.1):

$$\dot{h}(\mathbf{x},\mathbf{u}) = \frac{\partial h}{\partial \mathbf{x}}\dot{\mathbf{x}} = \underbrace{\frac{\partial h}{\partial \mathbf{x}}\mathbf{f}(\mathbf{x})}_{L_\mathbf{f}h(\mathbf{x})} + \underbrace{\frac{\partial h}{\partial \mathbf{x}}\mathbf{g}(\mathbf{x})}_{L_\mathbf{g}h(\mathbf{x})}\mathbf{u}\,, \tag{2.7}$$

where the corresponding terms $L_\mathbf{f}h(\mathbf{x})$ and $L_\mathbf{g}h(\mathbf{x})$ are called *Lie-derivatives*. If no further conditions are included in the QP in (2.6), then the explicit analytic solution of the intervention is known by the KKT-conditions (Karush-Kuhn-Tucker) :

$$\mathbf{u}(\mathbf{x}) = \begin{cases} \mathbf{k}_\mathrm{d}(\mathbf{x}) & \text{if } \varphi > 0\,, \\ \mathbf{k}_\mathrm{d}(\mathbf{x}) - \frac{\varphi\varphi_0^\top}{\varphi_0\varphi_0^\top} & \text{otherwise}\ , \end{cases} \tag{2.8}$$

where:

$$\varphi = L_\mathbf{f}h(\mathbf{x}) + L_\mathbf{g}h(\mathbf{x})\mathbf{k}_\mathrm{d}(\mathbf{x}) + \alpha(h(\mathbf{x}))\,, \tag{2.9}$$

$$\varphi_0 = L_\mathbf{g}h(\mathbf{x})\,. \tag{2.10}$$

It is an important observation that the quantity denoted by φ is a switching function, since its sign determines whether the known desired controller is able to keep the system trajectory within the safe set. If it fails to do so, then intervention according to (2.8) is required to guarantee safety. It is also a property of φ that it involves the α *class* $\mathcal{K}_\infty$ function, which can be appropriately chosen to manipulate whether the safety-critical

controller intervenes closer to or further from the boundary of the safe set. However, what can be eye-catching is the fractional term, since we can encounter a singularity as soon as the denominator approaches zero. This is equivalent to the statement $\varphi_0 = L_{\mathbf{g}}h(\mathbf{x}) \to 0$ which implies that theoretically we would need an infinitely large intervention signal for safe control. This is not to be confused with the case where $\varphi_0 = L_{\mathbf{g}}h(\mathbf{x}) \equiv 0$, which is related to the so-called relative degree, for which the High order Control Barrier Function (HOCBF) was introduced [12], but since this will not appear in the paper, this methodology will not be covered.

Walking through the framework of Safety-Critical Control, we notice that the input constraint is not part of it, so it is no coincidence that we are not able to satisfy it in many cases. On the other hand, it is also not a sufficient answer to expand the QP in (2.6) with an interval for $\mathbf{u}$:

$$\begin{aligned} \mathbf{u}(\mathbf{x}) = \arg\min_{\mathbf{u}} \quad & \frac{1}{2}||\mathbf{u} - \mathbf{k}_{\mathrm{d}}(\mathbf{x})||^2 , \\ \text{s.t.} \quad & \dot{h}(\mathbf{x}, \mathbf{u}) + \alpha(h(\mathbf{x})) \geq 0 , \\ & \mathbf{u}_{\min} \leq \mathbf{u} \leq \mathbf{u}_{\max} , \end{aligned} \tag{2.11}$$

because this would lead to the fact that there is no minimum of the objective function (quadratic term next to argmin) under such constraints.

2.2 Relationship with Lyapunov-function

The path to the solution (discussed in Section 3) is motivated by the Lyapunov function, the predecessor of the CBF, so it may be useful for us to recall it. To help understand the Lyapunov function, we refer to Fig. 2.2, which can be used not only for stability analysis of the equilibrium points of an arbitrary nonlinear dynamical system, but also for designing a stabilizing control. Furthermore, the similarity between the CBF shown in Fig. 2.1 and the Lyapunov function denoted by V in Fig. 2.2 is not an accident, and understanding the relationship between them is essential for us.

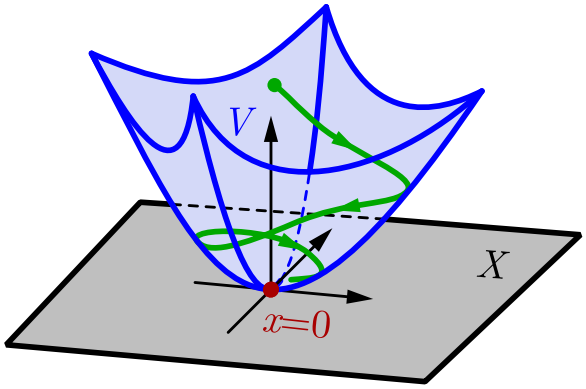

Figure 2.2: Graphical meaning of the Lyapunov function [11]

Based on the precedents, the first thing we need is an equilibrium point from which we want to verify its stability: let this point be $(\mathbf{x}_0, \mathbf{u}_0)$ and satisfy equation (2.1) if $\dot{\mathbf{x}}(t) = 0$:

$$\mathbf{f}(\mathbf{x}_0) + \mathbf{g}(\mathbf{x}_0)\mathbf{u}_0 = 0\,, \tag{2.12}$$

which is a system of algebraic equations that does not always have an explicit solution, but can be solved numerically by the equation $\mathbf{x}_0 \triangleq \mathbf{x}_0(\mathbf{u}_0)$. Around this operating point, the Lyapunov function V is written, which just happens to be at the origin in Fig. 2.2. Lyapunov's theorem is stated as follows [13]: assume a scalar function V: $\mathbb{R}^n \to \mathbb{R}$, which is zero only at the equilibrium point: $V(\mathbf{x}_0) = 0$. If

- $V(\mathbf{x}) > 0$ for all $\mathbf{x} \neq \mathbf{x}_0$, and
- $\dot{V}(\mathbf{x}) \leq 0$ for all $\mathbf{x}$, then

the $\mathbf{x}_0$ equilibrium point is stable, V is positive definite and $\dot{V}$ is negative semi-definite. Moreover, if the latter term is negative definite, then $\mathbf{x}_0$ is asymptotically stable. $\dot{V}$ is defined along the solution of the dynamical system by the chain rule:

$$\dot{V}(\mathbf{x}) = \frac{\partial V}{\partial \mathbf{x}}\dot{\mathbf{x}} = \underbrace{\frac{\partial V}{\partial \mathbf{x}}\mathbf{f}(\mathbf{x})}_{L_{\mathbf{f}}V(\mathbf{x})} + \underbrace{\frac{\partial V}{\partial \mathbf{x}}\mathbf{g}(\mathbf{x})}_{L_{\mathbf{g}}V(\mathbf{x})}\mathbf{u}(\mathbf{x})\,, \tag{2.13}$$

so that we can also interpret the *Lie*-derivatives, and here we have already assumed some feedback for the input. However, if we want to use V for stabilizing control rather than for verifying the stability of an equilibrium point, we can use QP as in (2.6):

$$\begin{aligned} \mathbf{u}(\mathbf{x}) = &\arg\min_{\mathbf{u}} \quad \frac{1}{2}||\mathbf{u} - \mathbf{k}_{\mathrm{d}}(\mathbf{x})||^2\,, \\ &\text{s.t.} \quad \dot{V}(\mathbf{x}, \mathbf{u}) + \gamma(V(\mathbf{x})) \leq 0\,, \end{aligned} \tag{2.14}$$

where γ is again a *class* $\mathcal{K}_\infty$ function. So far so good, but constructing a proper V is perhaps one step harder than constructing a safe sate. In general, quadratic functions are ideal for choosing a Lypunov function because of their positive definiteness, for example $V(x) = x^2$ for one-variable or $V(x, y) = x^2 + y^2$ for two-variable systems, but the criterion of negative (or negative semi) definiteness of the derivative along the system dynamics is much harder to satisfy, so there is no general methodology for choosing V. Nevertheless, by linearization, we are able to define effective Lyapunov functions using the so-called CTLE (*Continuous-time Lyapunov Equation*), by firstly transforming the nonlinear dynamical system (2.1) into the following form:

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})\mathbf{u} \equiv \mathbf{F}(\mathbf{x}, \mathbf{u}) \longrightarrow \dot{\mathbf{x}}(t) = \mathbf{A}(\mathbf{x} - \mathbf{x_0}) + \mathbf{B}(\mathbf{u} - \mathbf{u_0})\,, \tag{2.15}$$

where

$$\mathbf{A} = \left.\frac{\partial \mathbf{F}}{\partial \mathbf{x}}\right|_{\substack{\mathbf{x}=\mathbf{x}_0 \\ \mathbf{u}=\mathbf{u}_0}}, \qquad \mathbf{B} = \left.\frac{\partial \mathbf{F}}{\partial \mathbf{u}}\right|_{\substack{\mathbf{x}=\mathbf{x}_0 \\ \mathbf{u}=\mathbf{u}_0}}. \tag{2.16}$$

After shifting the state and the input with the operation point, the (2.15) linearized system simplifies as:

$$\dot{\tilde{\mathbf{x}}} = \mathbf{A}\tilde{\mathbf{x}} + \mathbf{B}\tilde{\mathbf{u}}\,, \tag{2.17}$$

where $\tilde{\mathbf{x}} = \mathbf{x} - \mathbf{x}_0$ and $\tilde{\mathbf{u}} = \mathbf{u} - \mathbf{u}_0$. According to Lyapunov's direct method full-state feedback controller ($\tilde{\mathbf{u}} = -\mathbf{K}\tilde{\mathbf{x}}$) can be used to obtain the closed-loop system:

$$\dot{\tilde{\mathbf{x}}} = \mathbf{A}\tilde{\mathbf{x}} + \mathbf{B}(-\mathbf{K}\tilde{\mathbf{x}}) = \underbrace{(\mathbf{A} - \mathbf{B}\mathbf{K})}_{\triangleq \mathbf{A}_{\mathrm{cl}}}\tilde{\mathbf{x}}\,, \tag{2.18}$$

where $\mathbf{K} \in \mathbb{R}^{n\times m}$ is the control gain matrix (or vector if $m = 1$), which does affect the behaviour of the closed-loop system: if all the real parts of the eigenvalues of $\mathbf{A}_{\mathrm{cl}}$ are negative, then $\mathbf{x}_0$ is a stable equilibrium point. Since this is the target, the choice of $\mathbf{K}$ is a crucial step, which can be done conventionally using the Routh-Hurwitz criterion, where the coefficients in the characteristic equation of the closed-loop matrix are investigated. It was mentioned earlier that quadratic functions are the most ideal Lyapunov functions, which in n-dimension have the following form:

$$V(\tilde{\mathbf{x}}) = \tilde{\mathbf{x}}^\top \mathbf{P}\tilde{\mathbf{x}}\,, \tag{2.19}$$

where $\mathbf{P} \in \mathbb{R}^{n\times n}$ is a symmetric, positive definite matrix. If this condition holds, $V(\tilde{\mathbf{x}})$ is a positive definite function and $V(\tilde{\mathbf{x}}) = 0$ if and only if $\tilde{\mathbf{x}} = \mathbf{0}$. To determine $\mathbf{P}$, it is useful to take the derivative of $V(\tilde{\mathbf{x}})$ with respect to time and substitute (2.18):

$$\dot{V}(\tilde{\mathbf{x}}) = \dot{\tilde{\mathbf{x}}}^\top \mathbf{P}\tilde{\mathbf{x}} + \tilde{\mathbf{x}}^\top \mathbf{P}\dot{\tilde{\mathbf{x}}} = \tilde{\mathbf{x}}^\top \underbrace{(\mathbf{A}_{\mathrm{cl}}^\top \mathbf{P} + \mathbf{P}\mathbf{A}_{\mathrm{cl}})}_{\triangleq -\mathbf{Q}}\tilde{\mathbf{x}}\,, \tag{2.20}$$

where if $\mathbf{Q} \in \mathbb{R}^{n\times n}$ is a positive definite matrix, which we have to specify and in the most convenient case is a unit matrix, then $\dot{V}(\tilde{\mathbf{x}})$ is also negative definite, and (2.20) is called the CTLE equation itself. At this point, two important observations must be added:

- if $\mathbf{x}_0$ is a stable equilibrium point, then $\mathbf{P}$ exists and is unique,
- a $V(\tilde{\mathbf{x}}) = c$ level curve (where $c \in \mathbb{R}^+$) is always an $n-$dimensional ellipse.

The level curves of Lyapunov functions already entail the forward invariance property, since – as shown in Fig. 2.2 – if the trajectory starts from the range $V < c$, it stays within this range for $\forall t$. This is exactly what we will exploit in the next section to define backup sets.

2.3 Theory of Backup set method

In recent years, the so-called backup set method has received the most attention, regarding the input constraint problem, since this method is general enough to be applicable to arbitrary dynamical systems, and even has a mathematical proof behind it. All this make

it outstanding among the other methods in the literature, although the first step in its use is the construction of valid backup sets, which this work aims to facilitate. The backup set method has the following core idea: to predict into the future dynamics so that being able to see whether the control task is feasible or not with the given input bounds. From today's point of view, prediction of dynamical systems is not a new idea, it is also used by *Model Predictive Control* (MPC), which has been known for more than half a century. Although there are similarities between MPC and CBF (and its later extension with Backup set), as both are optimization-based and can handle nonlinear systems, while the former is an "optimal control problem" [14] meaning that the controller is required to minimize a cost function subject to predefined constraints, the latter is set invariance-based.

As mentioned in the previous section, in the presence of input constraint, the CBF h we choose will not necessarily control in a safe way, in other words, it will not be control invariant. The reason for this is that the intervention is a function of the mechanical system and the defined safe set, i.e., $\mathbf{u} = \mathbf{u}(\mathbf{f}(\mathbf{x}), \mathbf{g}(\mathbf{x}), h(\mathbf{x}))$ according to QP (2.8). However, we do not have to adopt this at all costs; we can also have our own ideas for choosing $\mathbf{u}$ that makes h control invariant. This could also mean that if we are not satisfied with a particular $\mathbf{u}$ because it violates the input constraint, we can simply change h. In short, it is a much easier task to find a safe set that can be invariant under given input constraints than to prove one afterwards that it is indeed control invariant [15].

This is where the central idea of the method comes from, namely define a control invariant subset of the safe set $\mathcal{S}$, which is also the name of the method, the backup set $h_{\mathrm{b}}(\mathbf{x}) \geq 0$, with the domain notation $\mathcal{S}_{\mathrm{b}}$. Assume that it has the same properties as the safe set, only $\mathcal{S}_{\mathrm{b}} \subseteq \mathcal{S}$ must be satisfied. To do this, we need to find a controller $\mathbf{u}$ under which $h_{\mathrm{b}}(\mathbf{x})$ is control invariant, this will be the backup controller satisfying the input constraint, so $\mathbf{k}_{\mathrm{b}}(\mathbf{x}) \in \mathcal{U}$.

If it was just that straightforward to find for any set a controller that satisfies the input constraint and under which invariance is satisfied, there would be no need to use QP at all. On the contrary, it is a struggle to find backup sets and their backup controllers, and general guidelines do not exists yet. If we do manage to find (or rather discover) them, the backup set might be quite small. We would like to resolve this conservatism via forward prediction, which has been referred to several times before. To do this, we need the future value of the dynamics under the backup controller, obtained by integrating the following differential equation:

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})\mathbf{k}_{\mathrm{b}}(\mathbf{x}) \equiv \mathbf{f}_{\mathrm{b}}(\mathbf{x}) \,, \tag{2.21}$$

and the future state shall be $\varphi_{\mathrm{b}}(t, \mathbf{x}(0))$, which as we can see depends on the future moment we want to know (t) and the initial point from which the simulation starts $(\mathbf{x}(t = 0))$. This helps us to extend the original $\mathcal{S}_{\mathrm{b}}$, ideally up to $\mathcal{S}$. Instead, we extend it to a maximally reachable invariant set $\mathcal{S}_{\mathrm{b}} \to \mathcal{S}_{\mathrm{I}} \subseteq \mathcal{S}$, defined as follows [16]:

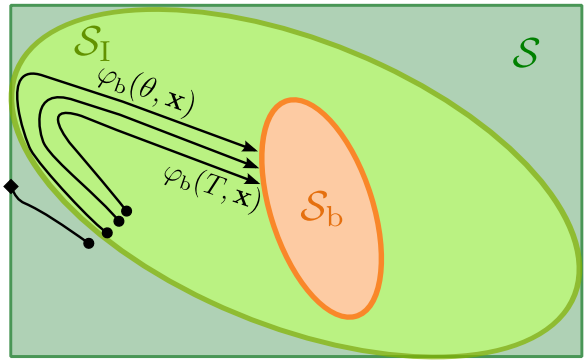

Figure 2.3: Graphical meaning of the $\mathcal{S}_\mathrm{I}$

$$\mathcal{S}_\mathrm{I} = \left\{ \mathbf{x} : \left(\varphi_\mathrm{b}(\theta, \mathbf{x}) \in \mathcal{S} \,,\, \forall \theta \in [0, T] \right) \wedge \left(\varphi_\mathrm{b}(T, \mathbf{x}) \in \mathcal{S}_\mathrm{b} \right) \right\}. \tag{2.22}$$

In words, $\mathcal{S}_\mathrm{I}$ is the set of points $\mathbf{x}$ where its future state is within the safe set at all instants of the interval $[0, T]$ and arrives at $\mathcal{S}_\mathrm{b}$ at time T, the idea of which is shown in Fig. 2.3, where we can see the so-called flows started from different starting points, which were solutions of the dynamics under the backup controller, and we can see that the flow started from outside $\mathcal{S}_\mathrm{I}$ fails to stay within the safe set. Another remarkable fact about $\mathcal{S}_\mathrm{I}$ is that we do not see any insight into its boundary during simulation, it cannot be determined explicitly [16]. The T may be called the integration time [thesis] or the time-horizon, the choice of which is that if $T = 0$, then $\mathcal{S}_\mathrm{b} = \mathcal{S}_\mathrm{I}$ and increasing it expands the backup set, as visually illustrated in Fig. 2.4. After a certain time, it is not worth increasing it because it doesn't increase $\mathcal{S}_\mathrm{I}$, but the computational need of the simulation grows and it should always be chosen efficiently for the given dynamical system. For example, for an inverted pendulum shorter than 1 m and lighter than 1 kg, it makes no sense to predict by hundreds of seconds, it may be enough to predict by only a few seconds.

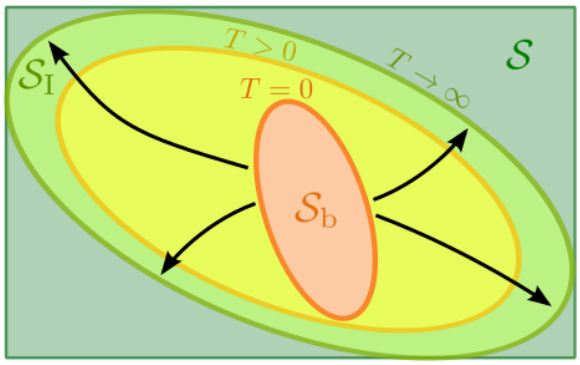

Figure 2.4: Graphical meaning of the backup set extension

In Fig. 2.4, we can observe the effect of this time T as we increase the backup set up to the invariant set. To understand how this concept is implemented in practice, we can translate the definition of invariant set in (2.22) into the QP optimization in (2.6), which

is then given by the following form:

$$\begin{aligned}\mathbf{u}(\mathbf{x}) = \arg\min_{\mathbf{u}\in\mathcal{U}} \quad & \frac{1}{2}||\mathbf{u}-\mathbf{k}_\mathrm{d}(\mathbf{x})||^2\,, \\ \text{s.t.} \quad & \dot{h}(\varphi_\mathrm{b}(\theta,\mathbf{x}),\mathbf{u}) + \alpha(h(\varphi_\mathrm{b}(\theta,\mathbf{x})) \geq 0\,,\ \forall\theta\in[0,T]\,, \\ & \dot{h}_\mathrm{b}(\varphi_\mathrm{b}(T,\mathbf{x}),\mathbf{u}) + \alpha_\mathrm{b}(h_\mathrm{b}(\varphi_\mathrm{b}(T,\mathbf{x})) \geq 0\,.\end{aligned} \tag{2.23}$$

To be applicable, we need to make two observations:

- in the first inequality, θ applies to every time instant between 0 and T, so this is actually an infinite amount of inequality. In practice, however, we are forced to discretize and work with only N_c (number of constraints),
- only in the simplest examples it is possible to define the flow analytically, so in all other cases we have to solve an "inner" IVP (Initial Value Problem).

This occurs for time derivatives of h and h_b, which can be computed using double chain rule as follows:

$$\dot{h}(\varphi_\mathrm{b}(\theta,\mathbf{x}),\mathbf{u}) = \frac{\partial h(\varphi_\mathrm{b}(\theta,\mathbf{x}))}{\partial\varphi_\mathrm{b}(\theta,\mathbf{x})}\underbrace{\frac{\partial\varphi_\mathrm{b}(\theta,\mathbf{x})}{\partial\mathbf{x}}}_{\hat{\mathbf{Q}}(\theta,\mathbf{x})}(\underbrace{\mathbf{f}(\mathbf{x})+\mathbf{g}(\mathbf{x})\mathbf{u}}_{\dot{\mathbf{x}}(t)})\,, \tag{2.24}$$

$$\dot{h}_\mathrm{b}(\varphi_\mathrm{b}(T,\mathbf{x}),\mathbf{u}) = \frac{\partial h_\mathrm{b}(\varphi_\mathrm{b}(T,\mathbf{x}))}{\partial\varphi_\mathrm{b}(T,\mathbf{x})}\underbrace{\frac{\partial\varphi_\mathrm{b}(T,\mathbf{x})}{\partial\mathbf{x}}}_{\hat{\mathbf{Q}}(T,\mathbf{x})}(\underbrace{\mathbf{f}(\mathbf{x})+\mathbf{g}(\mathbf{x})\mathbf{u}}_{\dot{\mathbf{x}}(t)})\,, \tag{2.25}$$

where $\hat{\mathbf{Q}}(\theta,\mathbf{x})$ (and obviously $\hat{\mathbf{Q}}(T,\mathbf{x})$) is the so-called sensitivity matrix [15]. It is another unknown function besides the flow, so it will be included in the IVP mentioned above, so:

$$\frac{\mathrm{d}\varphi_\mathrm{b}(\theta,\mathbf{x})}{\mathrm{d}\theta} = \mathbf{f}_\mathrm{b}(\varphi_\mathrm{b}(\theta,\mathbf{x}))\,, \tag{2.26}$$

$$\frac{\mathrm{d}\hat{\mathbf{Q}}(\theta,\mathbf{x})}{\mathrm{d}\theta} = \frac{\partial\mathbf{f}_\mathrm{b}(\theta,\mathbf{x})}{\partial\mathbf{x}}\hat{\mathbf{Q}}(\theta,\mathbf{x})\,, \tag{2.27}$$

and the associated initial conditions $\varphi_\mathrm{b}(0,\mathbf{x}) = \mathbf{x}$ and $\hat{\mathbf{Q}}(0,\mathbf{x}) = \mathbf{I}$, where $\mathbf{I}$ is an $n\times n$ unit matrix. Thus, finally, all terms in QP (2.23) are known, and only the backup set and backup controller suitable for the given control task need to be defined.

3 Construction of backup set and backup controller

As mentioned earlier, there is no exact algorithm for the construction of a valid backup set and a corresponding backup controller, and the existing examples [15], [16], [17] mainly capture the underlying physical meaning of the particular problem, which rather requires individual intuition (and talent). In this chapter, we propose an analytical way to overcome this problem using the Continuous-time Lyapunov Equation (CTLE). Most importantly, with that one can specify not only the backup set, but also the backup controller. Four main aspects have to be fulfilled for the proper construction of backup sets, we can divide the problem into four parts in the following: ensuring the forward invariance of the backup set, stabilizing the operation point, solving the input constraint of the backup controller and forcing the backup set into the safe set.

3.1 Forward invariance

The motivation behind it comes from the forward invariance property of the Lyapunov function. Let us introduce the backup set the following way:

$$h_{\mathrm{b}}(\mathbf{x}) = c - V(\mathbf{x})\,, \tag{3.1}$$

where let V be a Lyapunov function and c is one of its level curves and its magnitude is proportional to the size of the backup set. Substituting (3.1) into the inequality (2.4) that handles forward invariance:

$$\dot{h}_{\mathrm{b}}(\mathbf{x}) + \alpha_{\mathrm{b}}(h_{\mathrm{b}}(\mathbf{x})) = -\dot{V}(\mathbf{x}) + \alpha_{\mathrm{b}}(h_{\mathrm{b}}(\mathbf{x})) \geq 0\,. \tag{3.2}$$

The latter term is always non-negative by the *class* $\mathcal{K}_\infty$ function definition, and if $-\dot{V}$ is also non-negative, then the backup set in (3.1) is always rendered forward invariant. This means that $\dot{V}$ must be smaller or equal to zero which combined with the $V > 0$ assumption makes it clear that if a $\mathbf{x}_0$ operation point exists inside the backup set ($h_{\mathrm{b}}(\mathbf{x}_0) \geq 0$), then it is stable. To sum up, if a backup set is defined with a sub-level of a Lyapunov function, its forward invariance is satisfied only if the backup controller can stabilize the operation point $\mathbf{x}_0$ of the closed-loop system, then we are done with the forward invariance part. However, it is necessary not to ignore the fact that all this requires an operation point within the backup set. Previously, we obtained the relation $\mathbf{x}_0 = \mathbf{x}_0(\mathbf{u}_0)$, i.e., the following conditions for static intervention must be satisfied:

$$h(\mathbf{x}_0(\mathbf{u}_0)) \geq 0\,, \tag{3.3}$$

$$\mathbf{u}_{\min} \leq \mathbf{u}_0 \leq \mathbf{u}_{\max}\,. \tag{3.4}$$

If the intersection of these two sets determined by these inequalities is empty, then the control task will not be feasible. Thus, all application examples should also start by checking this, a kind of zero-step task.

3.1.1 Stabilization of operation points

Primarily, the stability of the working points needs to be guaranteed, otherwise, as discussed in the previous section, forward invariance cannot be guaranteed. In fact, this is where the CTLE will be actually utilised. For this reason, the final form of the backup set (2.21) will be written around the operation point:

$$h_{\mathrm{b}}(\mathbf{x}) = c - (\mathbf{x} - \mathbf{x}_0)^{\top}\mathbf{P}(\mathbf{x} - \mathbf{x}_0)\,, \tag{3.5}$$

where $\mathbf{P}$ is determined from the CTLE:

$$\mathbf{A}_{\mathrm{cl}}^{\top}\mathbf{P} + \mathbf{P}\mathbf{A}_{\mathrm{cl}} = -\mathbf{Q}\,. \tag{3.6}$$

Here we have to provide the matrix $\mathbf{Q}$, the main requirement is that it must be a positive definite matrix. The closed-loop matrix also contains the backup controller as in (2.18), which must be full-state feedback in order to apply the CTLE:

$$\mathbf{k}_{\mathrm{b}}(\mathbf{x}) = -\mathbf{K}(\mathbf{x} - \mathbf{x}_0) + \mathbf{u}_0\,. \tag{3.7}$$

For $\mathbf{x}_0$ to be stable, all the real parts of the eigenvalues of the closed-loop matrix must be negative. The most convenient method to ensure this is the Routh-Hurwitz criterion, from which we obtain various conditions for the choice of elements of $\mathbf{K}$.

3.1.2 Ensuring input constraint

Obviously, the backup controller defined by the full-state feedback seen in (3.13) cannot satisfy any input constraint, since as $||\mathbf{x}||_2 \to \pm\infty$, so $||\mathbf{k}_{\mathrm{b}}||_2 \to \mp\infty$, where $||.||_2$ denotes the norm two. To resolve this, saturation must be introduced:

$$\mathbf{k}_{\mathrm{b}}(\mathbf{x}) = \begin{cases} \mathbf{u}_{\min} & \text{if } -\mathbf{K}(\mathbf{x} - \mathbf{x}_0) + \mathbf{u}_0 < \mathbf{u}_{\min}\,, \\ \mathbf{u}_{\max} & \text{if } -\mathbf{K}(\mathbf{x} - \mathbf{x}_0) + \mathbf{u}_0 > \mathbf{u}_{\max}\,, \\ -\mathbf{K}(\mathbf{x} - \mathbf{x}_0) + \mathbf{u}_0 & \text{otherwise}\,. \end{cases} \tag{3.8}$$

The harmful side effect of this saturation is that it ruins the forward invariance of the backup set at the section where full-state feedback is no longer in operation. Thus, we are forced to make the backup set "not hang out of the linear section of the backup controller". What we mean by this can be seen visually in the example of the inverted pendulum later. But the point is that both the backup set and the backup controller are written around the operation point $(\mathbf{x}_0, \mathbf{u}_0)$. However, with a choice of certain c, there can exist a $\bar{\mathbf{x}}$ for which $h_{\mathrm{b}}(\bar{\mathbf{x}}) \geq 0$, but $\mathbf{k}_{\mathrm{b}}(\bar{\mathbf{x}}) = \mathbf{u}_{\min}$ or $\mathbf{k}_{\mathrm{b}}(\bar{\mathbf{x}}) = \mathbf{u}_{\max}$, meaning that despite being inside the backup set, the backup controller is saturated. This is absolutely not allowed, we must choose c well to prevent the backup set from "hanging out".

For this we use Lagrange-multipliers, a multivariable method, to adjust c such that the backup set reaches precisely the saturated point getting the backup set with maximal

size. Its fundamental formula is as follows [18]:

$$\nabla \tilde{f}(\mathbf{x}) = \Lambda \nabla \tilde{g}(\mathbf{x})\,, \tag{3.9}$$

$$\tilde{g}(\mathbf{x}) = 0\,. \tag{3.10}$$

Here, $\tilde{f}$: $\mathbb{R}^n \to \mathbb{R}$ is a multidimensional function whose extrema we seek under constraint $\tilde{g}$: $\mathbb{R}^n \to \mathbb{R}$ and $\Lambda \in \mathbb{R}$ is the Lagrange multiplier. The analogy is $\tilde{f}(\mathbf{x}) = k_\mathrm{b}(\mathbf{x})$ and $\tilde{g}(\mathbf{x}) = h_\mathrm{b}(\mathbf{x}) - c$ and it is not accidental that the backup controller is not now in bold, since for this analysis the backup controller must be a scalar function, and the following test must be performed separately for each component of $\mathbf{k}_\mathrm{b}(\mathbf{x})$, since the coordinate functions of the backup controller can be considered as independent variables. Thus, in this case $k_{\mathrm{b},i}(\mathbf{x})$ is the $i-$th element of $\mathbf{k}_\mathrm{b}(\mathbf{x})$, and is a scalar function of the form:

$$k_{\mathrm{b},i}(\mathbf{x}) = -\mathbf{K}_i^\top (\mathbf{x} - \mathbf{x}_0) + u_{0,i}\,, \tag{3.11}$$

where $\mathbf{K}_i$ is the $i-$th row vector of $\mathbf{K}$.

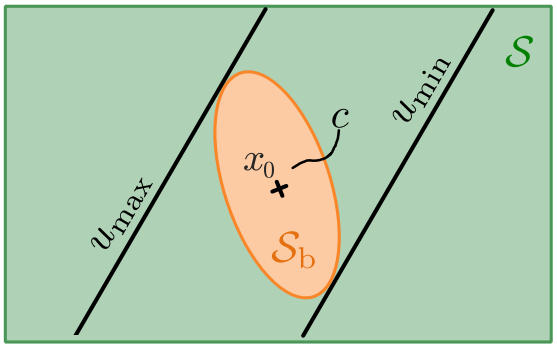

Figure 3.1: Graphical meaning of the Lagrange-multiplier

To demonstrate the method, see Fig. 3.1, which shows not only the orange backup set written around x_0 operation point and the green safe set, but also the saturated boundaries of the backup controller. Our goal is to set c, which characterizes the size of the backup set intersects at least one of the u_max and u_min boundaries. Consider that in order for the backup set to intersect both boundaries, $u_0 = 0.5(u_\mathrm{max} + u_\mathrm{min})$ would be required. Performing the operation in (3.10), we obtain:

$$-\mathbf{K}_i = -2\Lambda \mathbf{P}(\mathbf{x} - \mathbf{x}_0)\,, \tag{3.12}$$

and here we have exploited the fact that $\mathbf{P}$ is a symmetric matrix. Our goal is to eliminate Λ, since we do not need it further. To do this, we express the $\mathbf{x} - \mathbf{x}_0$ term from (3.12):

$$\mathbf{x} - \mathbf{x}_0 = \frac{1}{2\Lambda}\mathbf{P}^{-1}\mathbf{K}_i\,. \tag{3.13}$$

We need to substitute this back into (3.10):

$$\underbrace{\frac{1}{2\Lambda}\mathbf{K}_i^\top \mathbf{P}^{-1}}_{(\mathbf{x}-\mathbf{x}_0)^\top}\,\mathbf{P}\,\underbrace{\frac{1}{2\Lambda}\mathbf{P}^{-1}\mathbf{K}_i}_{(\mathbf{x}-\mathbf{x}_0)} = c\,. \tag{3.14}$$

After simplifications we get:

$$\Lambda = \pm\sqrt{\frac{\mathbf{K}_i^\top \mathbf{P}^{-1}\mathbf{K}_i}{4c}}\,. \tag{3.15}$$

Substituting this into (3.13), we obtain:

$$\mathbf{x} - \mathbf{x}_0 = \pm\frac{\sqrt{c}}{\sqrt{\mathbf{K}_i^\top \mathbf{P}^{-1}\mathbf{K}_i}}\mathbf{P}^{-1}\mathbf{K}_i\,, \tag{3.16}$$

so we know where the extreme values are, because the backup set is quadratic form, so we are not surprised that the $\pm$ sign gives us two solutions. The backup controller must saturate at this point because of the input constraint, so (3.16) should be substituted into $k_{\mathrm{b},i}(\mathbf{x}) = -\mathbf{K}_i^\top(\mathbf{x} - \mathbf{x}_0) + u_{0,i} \triangleq u_{\min,i}$ and $k_{\mathrm{b},i}(\mathbf{x}) = -\mathbf{K}_i^\top(\mathbf{x} - \mathbf{x}_0) + u_{0,i} \triangleq u_{\max,i}$. From here we obtain two solutions for c_i:

$$c_{1,i} = \frac{(u_{\max,i} - u_{0,i})^2}{\mathbf{K}_i^\top \mathbf{P}^{-1}\mathbf{K}_i}\,, \tag{3.17}$$

$$c_{2,i} = \frac{(u_{\min,i} - u_{0,i})^2}{\mathbf{K}_i^\top \mathbf{P}^{-1}\mathbf{K}_i}\,. \tag{3.18}$$

Of these, the appropriate choice will be the smaller one, i.e. $c_i = \min(c_{1,i}, c_{2,i})$. We must not forget that we have to perform this with all the element functions of the backup controller and choose the true minimum.

3.1.3 Backup set constraint

There is another aspect to consider for the correct choice of c in addition to the previous invariance condition, namely that the backup set should not only be within the linear section of the backup controller, but also within the safe set. This also requires the parameter c to be well tuned, and may even require a reduction on c calculated in the previous section. In general, this part of the problem can be solved via [19] by finding critical c_{cr} values for which the backup set and the safe set intersect each other in only one point. However, we will use a more conservative method to constrain the backup set, which will require less computational resource, but is customized to the application examples.

In the next section, we will demonstrate the construction of backup sets and backup controllers in practice through three application examples, together with a proper solution ensuring $\mathcal{S}_{\mathrm{b}} \subseteq \mathcal{S}$.

4 Applications

In this chapter, three dynamic systems are described in detail presenting how powerful tool the backup set method can be with the proposed method. Firstly, a simple scalar equation will be demonstrated so as to witness how to calculate the forward prediction part of the controller analytically. Then we see the inverted pendulum avoiding the danger of limited intervention. The chapter ends with an even more complex model, namely vehicle braking on a split$-\mu$ surface, where the driver poses a greater difficulty as an external disturbance and the problem of linearizability of the mechanical system.

4.1 Scalar equation

We are dealing with probably one of the simplest differential equation:

$$\dot{x} = x + u\,. \tag{4.1}$$

Without any intervention ($u \equiv 0$), the solution would be:

$$x(t) = Ce^t\,, \tag{4.2}$$

meaning that the state x goes to infinity (or minus infinity), depending on the initial condition. Let us say that the state x shall vary only between $-x_{\max}$ and $x_{\max}$, and the input u between $u_{\min}$ and $u_{\max}$, where $x_{\max} \in \mathbb{R}^+$ and $(u_{\min}, u_{\max}) \in \mathbb{R}$. The corresponding safe set can be the following:

$$h = x_{\max}^2 - x^2\,. \tag{4.3}$$

With the help of the well-known explicit formula from KKT-conditions (2.8), one can calculate the input signal if input constraint is neglected:

$$u(x) = \begin{cases} k_{\mathrm{d}}(x) & \text{if } -2x^2 - 2xk_{\mathrm{d}}(x) + \alpha(h) > 0\,, \\ -x + \frac{\alpha(h)}{2x} & \text{otherwise }, \end{cases} \tag{4.4}$$

where α is a *class* $\mathcal{K}_\infty$ function and $k_{\mathrm{d}}(x)$ is the desired controller. However, no matter how cleverly they are chosen, it can be proven that the extreme values of the input are $|u_{\max}| = |u_{\min}| = x_{\max}$ inside the safe set. This is shown in Fig. 4.1 in blue, where no matter how α and $k_{\mathrm{d}}(x)$ changes, the extreme values of $u(x)$ will always be on the boundary of the safe set.

4.1.1 Backup set and backup controller construction

If we are interested in regulating them, we shall use the Backup set method. Its first step is to find the equilibrium point by setting (4.1) equal to zero according to (2.12):

$$x_0 + u_0 = 0 \longrightarrow x_0 = -u_0\,. \tag{4.5}$$

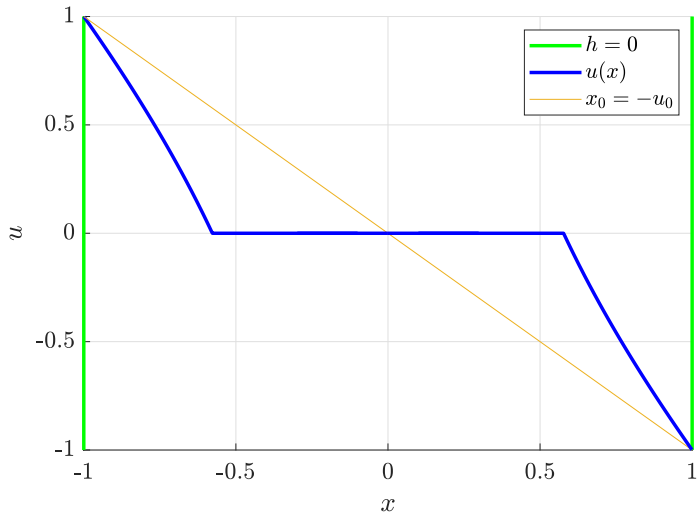

Figure 4.1: Input for scalar equation based on (4.1).

From that the backup controller is defined as based on (3.8):

$$k_{\mathrm{b}}(x) = \begin{cases} u_{\min} & \text{if } -K(x-x_0)+u_0 < u_{\min}\,, \\ u_{\max} & \text{if } -K(x-x_0)+u_0 > u_{\max}\,, \\ -K(x-x_0)+u_0 & \text{otherwise}\,, \end{cases} \tag{4.6}$$

where this time $\mathbf{K}$ is a scalar as $\mathbf{P}$, $\mathbf{A}_{\mathrm{cl}}$ and $\mathbf{Q}$ will be, moreover the eigenvalue of $\mathbf{A}_{\mathrm{cl}}$ is itself. Next, the linearized system is identical with the origin one resulting the closed-loop "matrix" and must fulfil the condition below:

$$A_{\mathrm{cl}} = 1 - K < 0 \longrightarrow \boxed{K > 1}\,. \tag{4.7}$$

The CTLE transforms into a scalar equation where the transpose of a scalar is simply itself and setting $\mathbf{Q} = Q \triangleq 1$:

$$(1-K)P + P(1-K) = -1 \longrightarrow \boxed{P = \frac{1}{2(K-1)}}\,. \tag{4.8}$$

The last unknown is c, which is according to (3.18):

$$c_1 = \frac{(u_{\max}-u_0)^2 P}{K^2}\,, \tag{4.9}$$

$$c_2 = \frac{(u_{\min}-u_0)^2 P}{K^2}\,, \tag{4.10}$$

and based on (3.5) the backup set is

$$h_{\mathrm{b}} = c - P(x-x_0)^2\,. \tag{4.11}$$

It shall not be forgotten that the backup set cannot exceed the safe set. For that in this simple case the explicit c values can be expressed:

$$c_3 = P(x_{\max}+x_0)^2\,, \tag{4.12}$$

$$c_4 = P(x_{\max}-x_0)^2\,. \tag{4.13}$$

Visually it means that for c_3 or c_4 the backup set and the safe set are tangent at only one point and the final value is $c = \min(c_1, c_2, c_3, c_4)$. Last step is to pick such u_0 static input values which retain every resulting x_0's in h according to (3.9) and simultaneously abide the input constraint:

$$x_{\max}^2 - (x_0(u_0))^2 \geq 0 \longrightarrow \boxed{-x_{\max} \leq u_0 \leq x_{\max}} \text{ AND } \boxed{u_{\min} \leq u_0 \leq u_{\max}}. \tag{4.14}$$

It is crucial to check before the simulation that these previous two sets have an intersection, otherwise neither safety nor the input constraint will work.

4.1.2 Forward prediction

This part of the simulation can be solved numerically via the (2.27) IVP, but since we are faced with such a simple equation, we can do it manually. Finally, we need the future state under the backup controller φ_{b}, where

$$\dot{x} = x + k_{\mathrm{b}}(x) \tag{4.15}$$

differential equation needs to be solved. Let us first consider the linear part of the backup controller:

$$\dot{x} = (1 - K)x + Kx_0 + u_0\,, \tag{4.16}$$

which can be solved analytically:

$$x_{\mathrm{lin}}(t) = Ce^{(1-K)t} - \frac{Kx_0 + u_0}{1 - K}\,. \tag{4.17}$$

Using this we can determine the flow:

$$\varphi_{\mathrm{b,lin}}(\theta, x) = Ce^{(1-K)\theta} - \frac{Kx_0 + u_0}{1 - K}\,, \tag{4.18}$$

and in order to find C, we can utilise the initial condition $\varphi_{\mathrm{b}}(0, x) = x$:

$$\varphi_{\mathrm{b,lin}}(\theta, x) = \left(x + \frac{Kx_0 + u_0}{1 - K}\right) e^{(1-K)\theta} - \frac{Kx_0 + u_0}{1 - K}\,, \tag{4.19}$$

and the sensitivity "matrix" is just its partial derivative with respect to x:

$$Q_{\mathrm{lin}}(\theta, x) = e^{(1-K)\theta}\,, \tag{4.20}$$

which automatically satisfies the other initial condition $Q(0, x) = 1$. To get the flow of the saturated section we solve the next differential equation:

$$\dot{x} = x + \hat{u}\,, \tag{4.21}$$

where let $\hat{u} = u_{\max}$ or $\hat{u} = u_{\min}$. Its solution is:

$$x_{\mathrm{sat}}(t) = Ce^t - \hat{u}\,. \tag{4.22}$$

Writing it in the sense of flow:

$$\varphi_{\mathrm{b,sat}}(\theta, x) = Ce^{\theta} - \hat{u}\,, \tag{4.23}$$

and determining C again with the initial condition:

$$\varphi_{\mathrm{b,sat}}(\theta, x) = (x + \hat{u})e^{\theta} - \hat{u}\,. \tag{4.24}$$

The final form of the flow under the backup controller is:

$$\varphi_{\mathrm{b}}(\theta, x) = \begin{cases} (x + u_{\min})e^{\theta} - u_{\min} & \text{if } -K(x - x_0) + u_0 < u_{\min}\,, \\ (x + u_{\max})e^{\theta} - u_{\max} & \text{if } -K(x - x_0) + u_0 > u_{\max}\,, \\ \left(x + \frac{Kx_0 + u_0}{1-K}\right) e^{(1-K)\theta} - \frac{Kx_0 + u_0}{1-K} & \text{otherwise}\,. \end{cases} \tag{4.25}$$

With that, everything is finally ready to solve the (2.23) optimization problem.

4.1.3 Results

The following parameters in Table (4.1) are fixed throughout the entire presentation of the results. The input bounds are purposefully chosen to produce a non-symmetric input constraint, which is one of the greatest benefits of this method. Of course, since we are not talking about an example with a concrete physical connotation, unitless quantities are used.

Table 4.1: Fixed parameters

Parameter	marking	value
Bound of the safe set	$x_{\max}$	1 [−]
Lower bound of the input	$u_{\min}$	−0.5 [−]
Upper bound of the input	$u_{\max}$	0.75 [−]
Gain of the backup controller	K	5 [−]
class $\mathcal{K}_\infty$ for safe set	α	3 [−]
Number of constraints	N_{c}	80 [−]
class $\mathcal{K}_\infty$ for backup set	α_{b}	1 [−]
Static input	u_0	0 [−]

Furthermore, the nominal (desired) controller is considered $k_{\mathrm{d}}(x) = 0$. The analysis starts with a comparison, namely with the Safety Critical Control, which is saturated at its intervention boundaries. Started from different starting points from $x(t = 0) = -0.6$ to $x(t = 0) = 0.6$. In Fig. 4.2, it can be seen that in none of the cases could the saturated CBF keep the trajectory within the safe set. For the state x to remain inside the safe set, $\dot{x} \geq 0$ must be satisfied for negative x, which in the light of (4.1) corresponds to $u \geq -x$, while the opposite condition is true for negative x. This is the reason why in Fig. (4.1) you can see a line of $-45°$, because this is the boundary. Since $u \geq -x$ ($\forall x \geq 0$) and $u \leq -x$ ($\forall x \leq 0$) are nowhere satisfied except the case without input constraint. It is not surprising that the saturated $u(x)$ (4.4) cannot keep the trajectory inside the safe set for any initial condition.

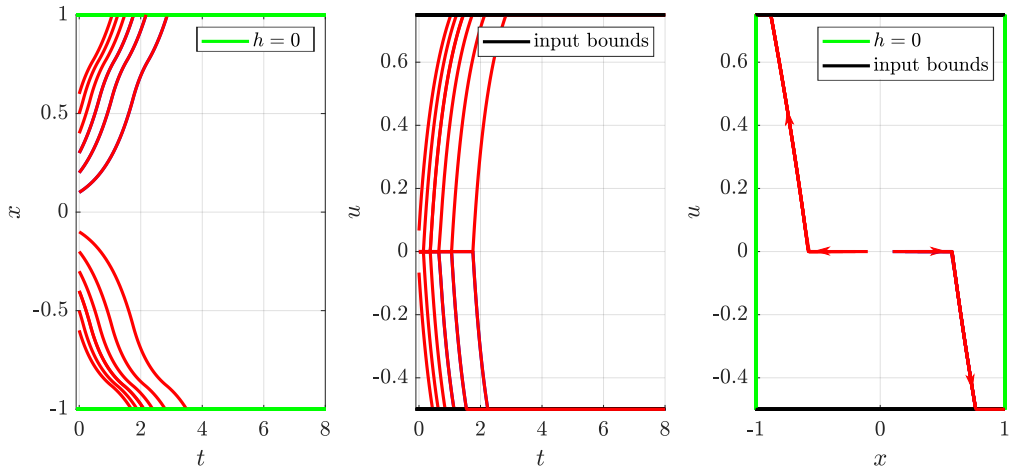

Figure 4.2: Scalar equation from different initial points with saturate CBF.

Next we can see what can be achieved by using the backup set method on Fig. 4.3. Starting the simulation from the same starting points as in Fig. 4.2, it can be seen that the trajectories were kept within the safe set (blue curves) from all but one (red curve) starting point. Moreover, it can be observed that, due to the non-symmetric boundaries, the directions in which larger interference was possible the backup controller intervened later. This time, T was set to 10. Furthermore, the invariant set, which is the interval $x = [-0.75, 0.5]$, is also explored, and by comparing it with the size of the backup set, we can see how much the latter can be increased. In addition, what is worth showing is precisely the effect of this T on the controller. Fig. 4.4 shows this for 3 different prediction times. If $T = 0$, as described in Section 2.3, the invariant set is identical with the backup set and as a result the trajectory comes back to the backup set. With increasing T we can see how the trajectories wander wider regions.

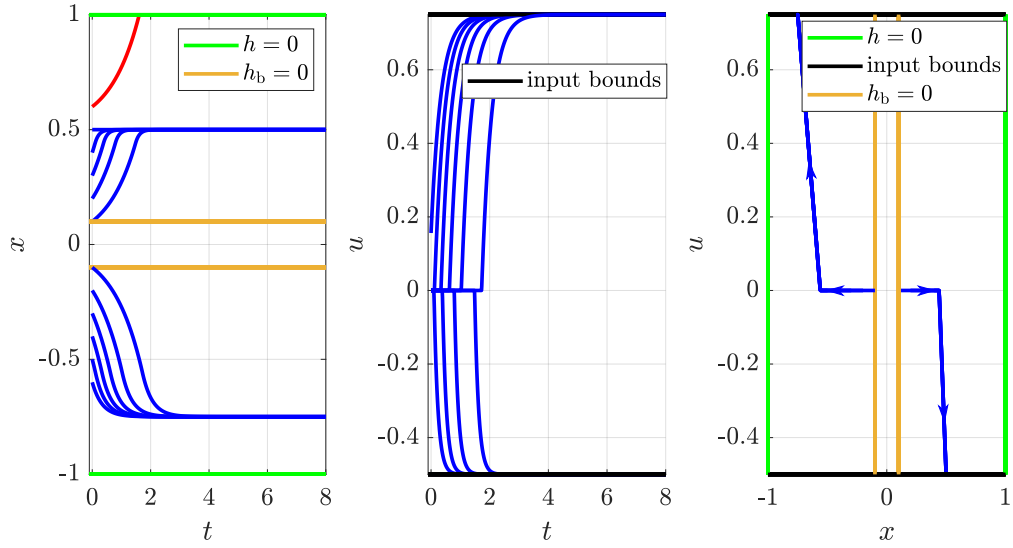

Figure 4.3: Scalar equation from different initial points with Backup-CBF and $T = 10$.

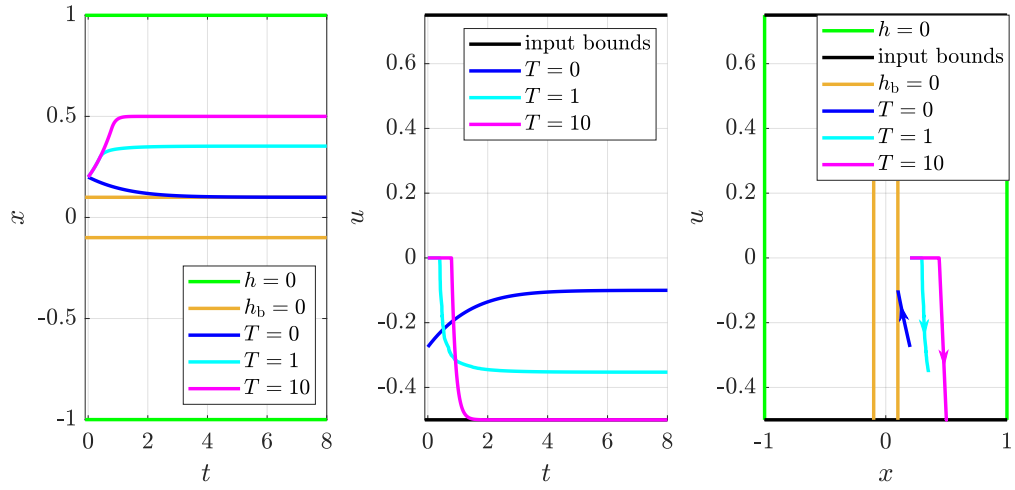

Figure 4.4: Scalar equation for 3 different T values.

4.2 Inverted pendulum

Presenting the backup method via the pendulum has two new features compared to the previous 1D model: first, it has physical meaning, so it is easier to imagine how the parameters used in the backup set method affect the mechanical system in reality, and second, there is no longer a closed form formula for the flow, so the prediction part of the task has to be performed numerically. The equation of motion of an inverted pendulum is:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} y \\ \frac{3g}{2L}\sin(x) \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{3}{mL^2} \end{bmatrix} u\,, \tag{4.26}$$

which can be derived from for example the Lagrange's equations of the second kind. In equation (4.26) the two states are the angle measured from the vertical axes ($x \triangleq \vartheta$) and the angular velocity ($y \triangleq \dot{\vartheta}$), u is the input proportional to some torque, g is the gravitational acceleration, m is the pendulum's mass and L is the pendulum's length. For further purpose of simplification let $L \triangleq 3g/2$ and $m \triangleq 4/3g^2$ to get unit coefficients:

$$\underbrace{\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix}}_{\dot{\mathbf{x}}} = \underbrace{\begin{bmatrix} y \\ \sin(x) \end{bmatrix}}_{\mathbf{f}(\mathbf{x})} + \underbrace{\begin{bmatrix} 0 \\ 1 \end{bmatrix}}_{\mathbf{g}(\mathbf{x})} u\,. \tag{4.27}$$

The initial objective could be for an inverted pendulum to stop before hitting the table after releasing it from an upper position, but not from the vertical one. It would imply the following safe set:

$$h(\mathbf{x}) = 1 - \left(\frac{x}{\hat{x}}\right)^2\,, \tag{4.28}$$

where $\hat{x} = \pi/2$. But, taking into account the above-mentioned fact about $L_{\mathbf{g}}h(\mathbf{x}) \equiv 0$, it is evident that with this safe set the exactly this situation would occur. Not to complicate the task by wandering into the theory of High order Control Barrier Functions, let's modify

this safe set a bit:

$$h(\mathbf{x}) = 1 - \left(\frac{x}{\hat{x}}\right)^2 - \left(\frac{y}{\hat{y}}\right)^2 , \tag{4.29}$$

where the zero-level subset of this particular safe set is an origin-centered ellipse with $(\hat{x}, \hat{y})$ semi-axis. In other word, beyond the angle, the angular velocity is also restricted though the $L_{\mathbf{g}}h(\mathbf{x}) \equiv 0$ case is avoided successfully.

4.2.1 Backup set and backup controller construction

The analysis begins with the search for the equilibrium point again by setting (4.26) equal to zero. Solving the (2.12) algebraic equation we get:

$$x_0 = \arcsin(-u_0) , \tag{4.30}$$

$$y_0 = 0 . \tag{4.31}$$

With these point we select which u_0 can keep (x_0, y_0) within the safe set:

$$h(\mathbf{x}_0(u_0)) = 1 - \left(\frac{x_0}{x_{\max}}\right)^2 - \left(\frac{y_0}{\hat{y}}\right)^2 = 1 - \left(\frac{\arcsin(-u_0)}{x_{\max}}\right)^2 \geq 0 . \tag{4.32}$$

From (4.31) u_0 can be expressed but also has to obey the input constraint:

$$\boxed{-\sin(x_{\max}) \leq u_0 \leq \sin(x_{\max})} \text{ AND } \boxed{u_{\min} \leq u_0 \leq u_{\max}} . \tag{4.33}$$

Next, we can write the backup controller (3.8) for two variables this time:

$$k_{\mathrm{b}}(\mathbf{x}) = \begin{cases} u_{\min} & \text{if } -K_1(x - x_0) - K_2(y - y_0) + u_0 < u_{\min} , \\ u_{\max} & \text{if } -K_1(x - x_0) - K_2(y - y_0) + u_0 > u_{\max} , \\ -K_1(x - x_0) - K_2(y - y_0) + u_0 & \text{otherwise} , \end{cases} \tag{4.34}$$

where $\mathbf{K} = (K_1, K_2)^\top$ gain vector since $u \in \mathbb{R}$. What we need to do now is to linearize around (x_0, y_0, u_0), where the matrix $\mathbf{A}$ and vector $\mathbf{B}$ will look like according to equations in (2.16):

$$\mathbf{A} = \begin{bmatrix} 0 & 1 \\ \cos(x_0) & 0 \end{bmatrix} , \tag{4.35}$$

$$\mathbf{B} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} , \tag{4.36}$$

thus we can now write the closed-loop matrix:

$$\mathbf{A}_{\mathrm{cl}} = \begin{bmatrix} 0 & 1 \\ \cos(x_0) - K_1 & -K_2 \end{bmatrix} . \tag{4.37}$$

Given that we have now a matrix, we will now apply the Routh-Hurwitz criterion to check which $\mathbf{K}$ gain vector can stabilize the linearized system around the equilibrium point. To do so, we derive the characteristic equation of the closed-loop matrix:

$$\lambda^2 + K_2\lambda + K_1 - \cos(x_0) = 0 , \tag{4.38}$$

and set all of the coefficients to be positive (or negavitve, the point is that they all must have the same sign):

$$K_1 > \cos(x_0)\,, \tag{4.39}$$

$$K_2 > 0\,. \tag{4.40}$$

Next the CTLE in (2.20) has to be solved to get the $\mathbf{P}$ matrix which is always unique and exists if $\mathbf{x}_0$ equilibrium point is stable. Fortunately, MATLAB has a built-in function (called lyap) that calculates it easily. The zero-superlevel of the backup set for two variables is an ellipse and the c constant can be computed with the Lagrange-multipliers method via equations (3.18). The other essential condition for the choice of c is to ensure the backup set does not exceed the safe set with that. Though, this problem is more of a geometrical one rather than a controlling one, because it is all about to constrain an ellipse (backup set) inside an other (safe set). The pleasant scenario would be same as it was in the 1D example, to intersect each other at only one point. For that a system of two quadratic equations need to be solved (because they are ellipses) and make sure that out of the four solutions (as two ellipses can intersect each other at maximum four points) there is only one real solution. As this would lead to a considerably more complicated algebraic problem, let us instead provide a more conservative, own method to address it.

4.2.2 Ellipse constraint

The idea behind this comes from the fact that while finding the intersection of two ellipses and giving a condition on it is tough, it is easier for an ellipse and a line. For the latter case we can make use of the quadratic formula and by setting its discriminant to zero we are able to ensure that the line is tangent to the ellipse, i.e. they intersect each other at only one point. It should be noted that this method works only for this particular example, so we are talking about a constraint of two n-dimensional quadratic sets. If the shape of the safe set is different, a new method has to be established, as discussed in Section 3.2.4. In this case, it is advantageous, because we work with similar sets in the pendulum problem here and in the vehicle example afterwards. The algorithm is shown in Fig 4.1 - Fig 4.3. and can be divided into the following steps:

1. Compute the eigenvectors is $\mathbf{P}$ which describes the slopes of the major-axis of the backup set.
2. Find intersection points of the backup set's major-axis and the safe set (Fig. 4.5).
3. Connect these intersection points with lines (Fig. 4.5).
4. For all of the four lines solve the quadratic equation that comes from the intersection of each line with the backup set and set the discriminant to zero (Fig. 4.6).

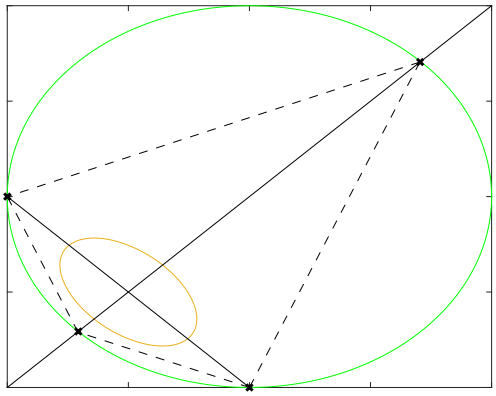

Figure 4.5: Eigen-axis of the backup set (solid lines) and connections between intersection points (dotted lines).

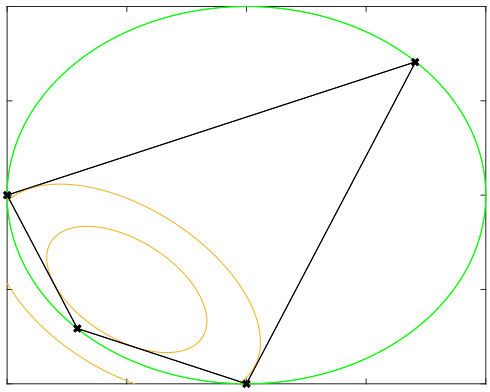

Figure 4.6: Inner ellipse constraint with the four lines.

5. Choose out of the four options the smallest c value, thus ensuring that the backup set always remains inside the safe set (in Fig. 4.6) the first two and the last two c values are the same, that is why only two backup set ellipses can be seen).

However, recall that this is only one possible solution among many, there are other ways to check that the backup set is actually inside the safe set. As mentioned, [19] provides a general approach for the multidimensional case.

4.2.3 Results

In the example of the inverse pendulum, again non-symmetric intervention boundaries were selected to demonstrate how powerful the proposed method is. For that, if assumed that the torque needed to keep the pendulum statically horizontal, that would lead to $u = \pm 1$, which comes from equation (4.26). Therefore let one of the boundaries be a little bit higher and the other one a little bit lower to test the Backup set method but ensuring that the double condition in (4.33) is satisfied. The constant parameters of the simulations for the inverted pendulum can be found in Table (4.2)

Fig. 4.7 shows all preliminary functions visualized in three dimension: with green stands the safe set including the orange backup set and both of them are ellipse based cylinders. With gray the saturated backup controller is plotted and with the perfect c parameter selection the backup set cylinder remains not just inside the safe set, but also inside the linear section of the backup controller. What is expected from the simulation is that the trajectories (the two states and the one input) should be within the green, top and bottom locked safe set cylinder, i.e. not crossing the saturated section of the backup controller. Also, at the end of each simulation, an equilibrium point is reached, so the pendulum is (pleasantly) stopped inside the safe set, so the angular velocity is zero, and the trajectories lie on the horizontal axis in the graphs.

Table 4.2: Fixed parameters for inverted pendulum

Parameter	marking	value
Maximal angle	$\hat{x}$	$\pi/2$ [rad]
Maximal angular velocity	$\hat{y}$	$4/3$ [rad/s]
Lower bound of the input	$u_{\min}$	-0.75 [SI]
Upper bound of the input	$u_{\max}$	1.25 [SI]
Gain vector of the backup controller	$\mathbf{K}$	$(3,3)^\top$ [SI]
class $\mathcal{K}_\infty$ for safe set	α	1 [$-$]
Number of constraints	N_c	80 [$-$]
class $\mathcal{K}_\infty$ for backup set	α_b	1 [$-$]
Static input	u_0	0 [$-$]

The nominal (desired) controller is once again considered $k_d(\mathbf{x}) = 0$ and the elements of $\mathbf{K}$ gain vector fulfil the stability criterion in (4.40). The most fundamental point of comparison is the emphasis on the backup set method control over the classical, but saturated CBF from (2.11). The colour codes are identical to the ones used so far. Fig. 4.8 shows the trajectories of the dynamics under the Safety Critical Control corresponding to the QP in (2.11), i.e. the intervention is saturated. Using the so-called brute force method, starting from nearly 300 initial points, the trajectories of the system are shown, which were separated into two groups by appropriate coloring: those that are able to stay inside the safe set are blue, those that are not are red. It is noticeable how relatively few of these trajectories are able to be retained, and it is obvious that this needs to be improved. In addition, a faded yellow area indicates the invariant set defined in $\mathcal{S}_I$ (2.22), which should be enlarged using the backup set method.

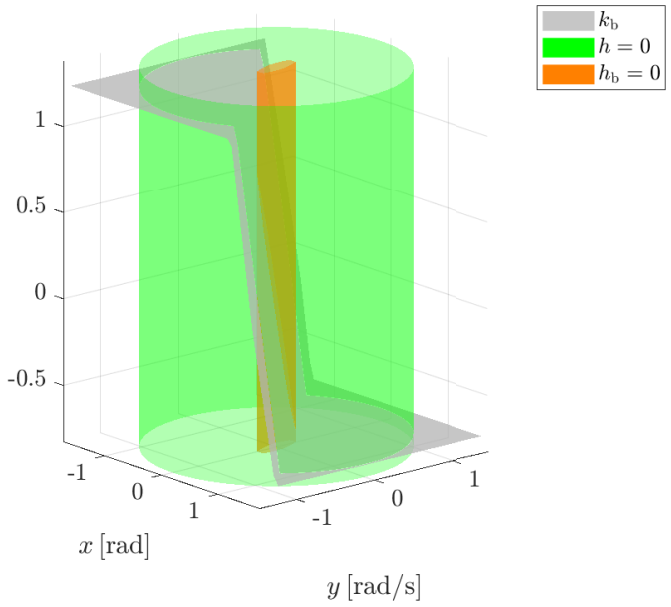

Figure 4.7: 3-dimensional plot of all the functions of Backup set method

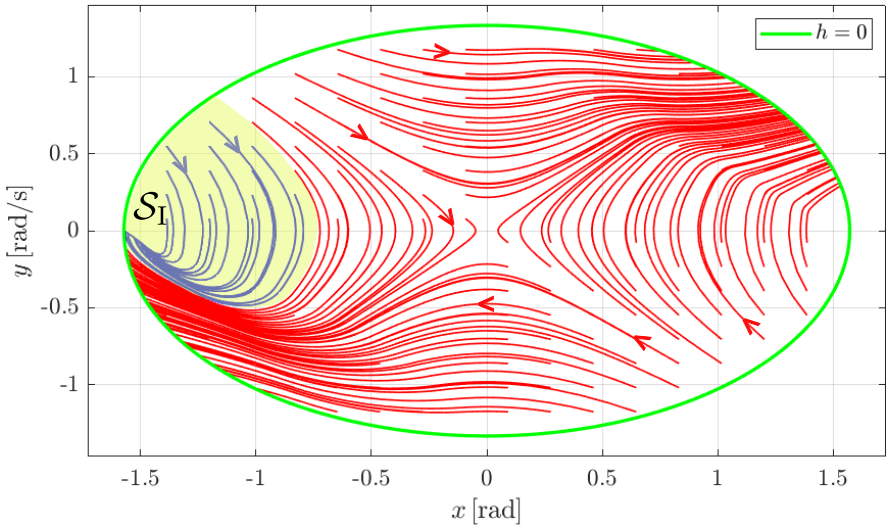

Figure 4.8: Trajectories under saturated CBF with $\mathcal{S}_\mathrm{I}$ indicated.

Fig. 4.9 and Fig. 4.10 represent the trajectories obtained with the backup set method, the only difference between them being the prediction time, which is $T = 1\,\mathrm{s}$ for the former one and $T = 10\,\mathrm{s}$ for the latter one. Still, the huge improvement over the saturated Safety Critical Control is how much we could increase the region that trajectories – launched from inside the safe set – eventually stay inside safe set throughout the whole simulation. In other words, the $\mathcal{S}_\mathrm{I}$ was successfully expanded. The difference, though, is the prediction time, which, while not significantly enlarging the region spanned by the blue curves between 1 and 10 seconds, is worth observing how they affect the dynamics of the controlled system. In both cases we notice the appearance of new equilibrium points, which are actually the equilibrium points of the closed-loop dynamics under the (2.23) controller. These are marked by $\times$ signs, closer to the origin at $T = 1\,\mathrm{s}$ and further away at $T = 10\,\mathrm{s}$.

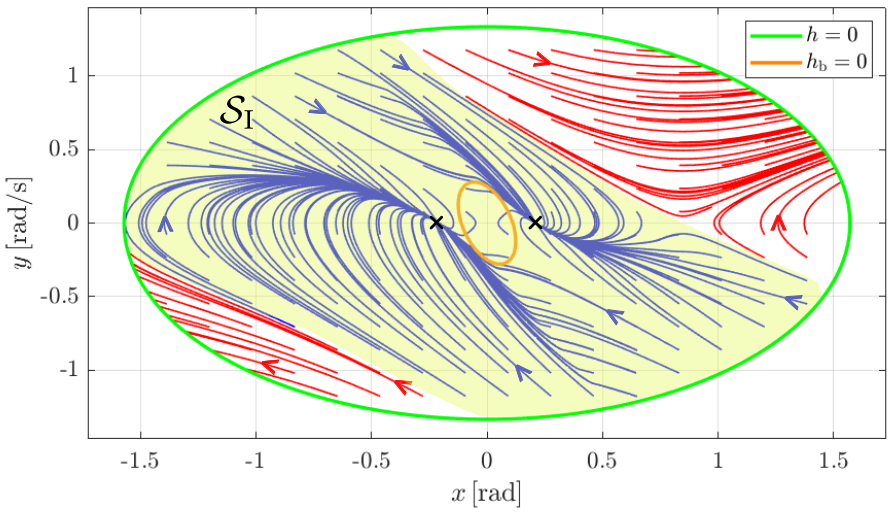

Figure 4.9: Trajectories under Backup set method with $T = 1$ s and with $\mathcal{S}_\mathrm{I}$ indicated.

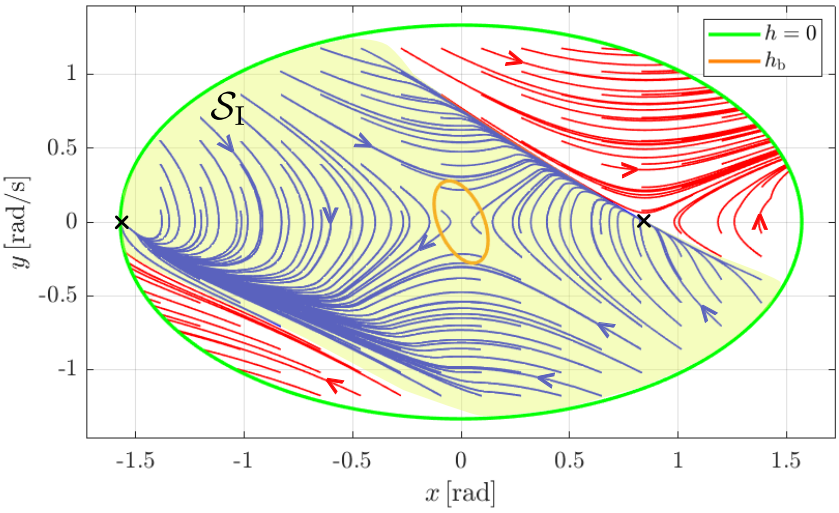

Figure 4.10: Trajectories under Backup set method with $T = 10\,\mathrm{s}$ and with $\mathcal{S}_\mathrm{I}$ indicated.

Returning to Fig. 4.7, to visualize the method in space, Fig. 4.11 illustrates two trajectories, in 3 dimensions on the left, and a top view on the right. It can be claimed that both the safety and the input constraint are satisfied. Based on this example, we intend to use a similar approach to control an even more complex mechanical system in the next section.

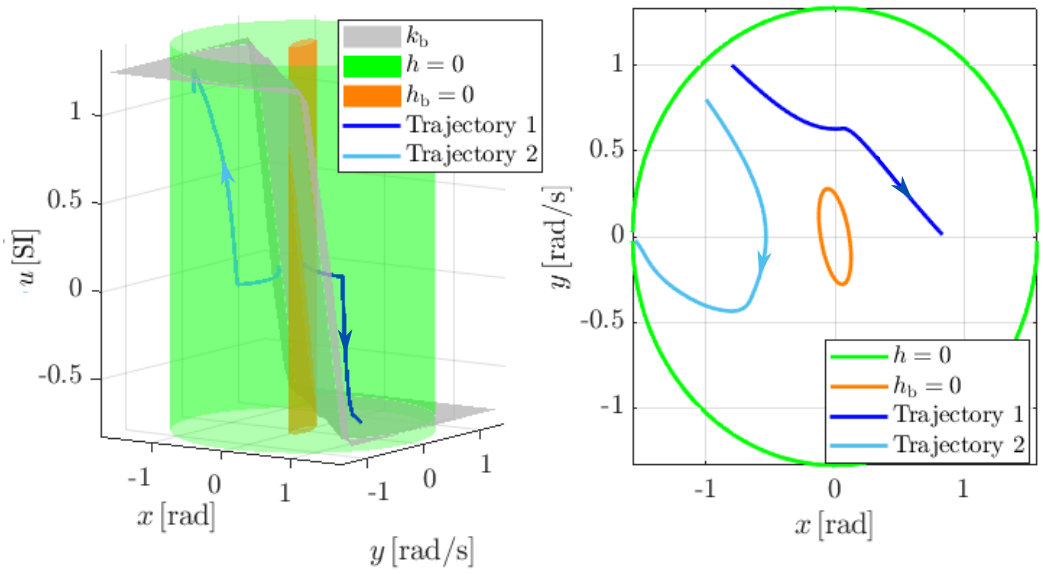

Figure 4.11: 3-dimensional plot of the whole mechanical system with two trajectories under Backup set method with $T = 10\,\mathrm{s}$ (right) and top view (left)

4.3 Vehicle braking

The third and most challenging application is vehicle braking on a split−μ surface. The marking μ refers here to the coefficient of traction, while the word "split" refers to the separation of the road surface, to sum up meaning that braking on a road surface with an asymmetric coefficient of traction [20]. In the conventional case, the longitudinal (forward) dynamics of the vehicle can be controlled by pressing the gas or brake pedal, while the lateral dynamics can be controlled by the steering wheel. A different and surprising phenomenon for the driver is braking on a surface with different traction. This is not caused by the steering movement but by the split−μ braking, which can result in critical dynamic values leading to a spin-out, for which ordinary drivers are less, if not not at all, prepared due to the uncommon nature of the situation.

Due to the different friction and adhesion conditions, the difference of braking forces causes a torque around the vertical axis (yaw moment), which can lead to the loss of stability and therefore loss of vehicle control. Based on measurement results, the loss of stability due to split−μ braking can lead to a critical increase in the vehicle's angular velocity about the vertical axis (*yaw rate*) $\dot{\psi}$ and side-slip angle β, as shown in Fig. 4.12. This can basically be avoided by not allowing the opposite sides of the vehicle to brake with different forces − i.e. both sides braking with the lower traction side's braking force − but this will result in a drastically increased stopping distance. It is indisputable that the driver must be assisted through the intervention in the braking forces in order to be able to brake safely and avoid spinning out. The question is, however, what kind of control should be chosen to deal with this problem. It would be a trivial solution to reduce the yaw rate and side-slip angle to 0, so that the vehicle has zero lateral dynamics. To do this, it is necessary eliminate the torque, so that the wheels on both sides brake with equal force. This method is called Select-Low (SL) because both sides brake with the braking force of the low traction side. Its opposite is Select-High (SH), where both wheels brake with their own maximal force possible [21].

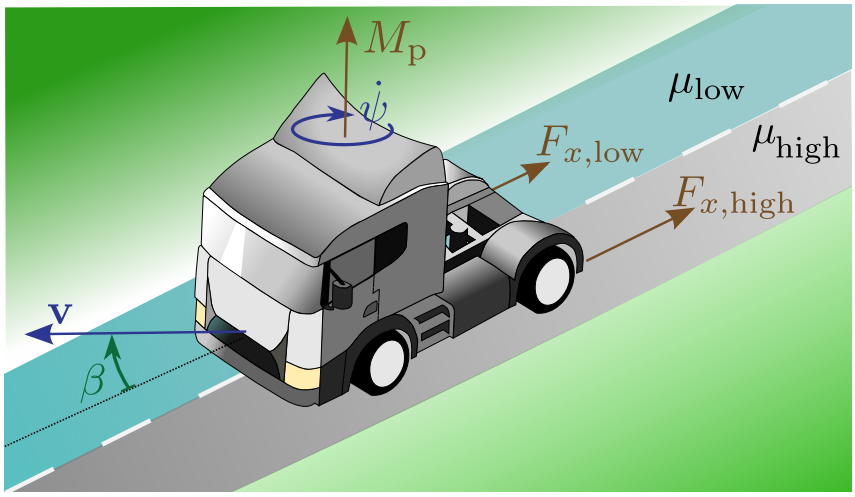

Figure 4.12: Vehicle dynamics during split−μ braking

The solution is between the two methods outlined: a certain yaw rate and side-slip angle is allowed, to which the driver can adapt and react to appropriately. This is why Safety Critical Control is perfectly applicable to this example, to prevent the vehicle's yaw rate and side slip angle from increasing significantly. The first step is to define the vehicle model itself, which includes three unknown parameters (independent of the particular vehicle model): the longitudinal and lateral forces on the wheels and the angle of wheel position relative to the longitudinal direction. The first two are obtained from the selected wheel model, while the latter is obtained from the driver model.

As mentioned in the introduction, the input constraint is present naturally, and both of its boundaries have physical meaning. One is the maximum braking force, which cannot be exceeded due to the traction conditions, it is not possible to brake with any force. The other limit is the minimum, which, if exceeded, would physically mean that the wheel would accelerate rather than decelerate. However, in the case of emergency braking, it is not allowed to accelerate wheels. In the example outlined, first the models that constitute the mechanical system will be reviewed, and then the application of the backup set method.

4.3.1 Mechanical model

To model braking on a split mu surface, it is necessary to separate the vehicle in the model into right and left sides for different braking forces, and front and rear axles for front axle steerability. This means that at least a four-wheel vehicle model is needed, and its free body diagram is shown in Fig. (4.13).

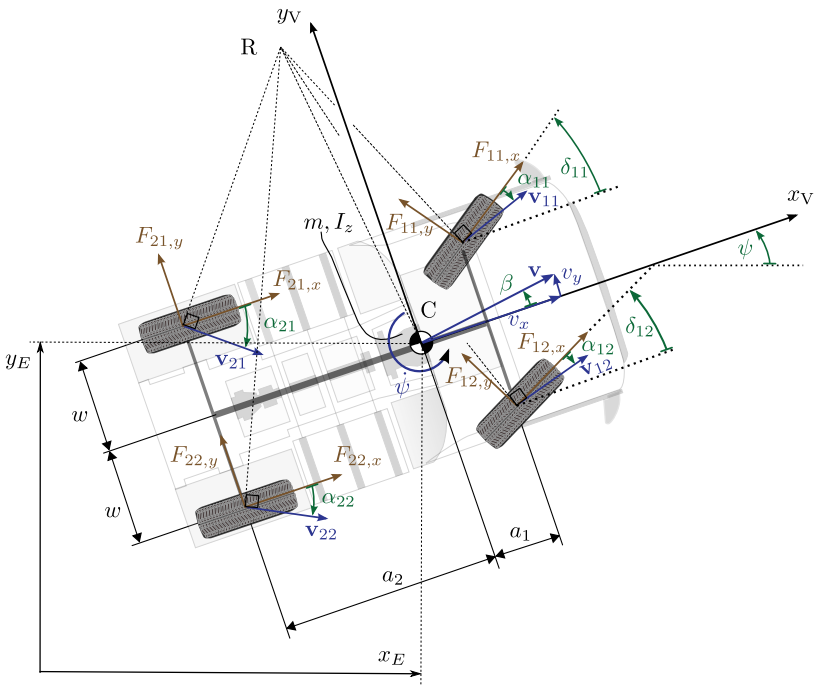

Figure 4.13: Mechanical model of a vehicle [2]

Using the Lagrange's equations of the second kind can be derived of the equation of motion of the planar vehicle model (also known as bicycle model) according to [22]:

$$\underbrace{\begin{bmatrix}\dot{v}_x\\ \dot{v}_y\\ \dot{\omega}\\ \dot{x}_{\mathrm{E}}\\ \dot{y}_{\mathrm{E}}\\ \dot{\psi}\end{bmatrix}}_{\dot{\mathbf{x}}} = \underbrace{\begin{bmatrix}\frac{1}{m}(F_{12,x}\cos(\delta)-(F_{11,y}+F_{12,y})\sin(\delta)+F_{22,x})+\omega v_y\\ \frac{1}{m}((F_{11,y}+F_{12,y})\cos(\delta)+F_{12,x}\sin(\delta)+F_{21,y}+F_{22,y})-\omega v_x\\ \frac{1}{I_z}(w\tilde{F}_w+a_1\tilde{F}_{a1}+a_2\tilde{F}_{a_2})\\ v_x\cos(\delta)-v_y\sin(\delta)\\ v_x\sin(\delta)+v_y\cos(\delta)\\ \omega\end{bmatrix}}_{\mathbf{f}(\mathbf{x})} + \underbrace{\begin{bmatrix}\frac{\cos(\delta)}{m} & \frac{1}{m}\\ \frac{\sin(\delta)}{m} & 0\\ \frac{a_1\sin(\delta)-w\cos(\delta)}{I_z} & -\frac{w}{I_z}\\ 0 & 0\\ 0 & 0\\ 0 & 0\end{bmatrix}}_{\mathbf{g}(\mathbf{x})}\underbrace{\begin{bmatrix}F_{11,x}\\ F_{21,x}\end{bmatrix}}_{\mathbf{u}}, \tag{4.41}$$

where:

$$\tilde{F}_w = F_{22,x}+F_{12,x}\cos(\delta)-(F_{12,y}-F_{11,y})\sin(\delta)\,, \tag{4.42}$$

$$\tilde{F}_{a_1} = (F_{11,y}+F_{12,y})\cos(\delta)+F_{12,x}\sin(\delta)\,, \tag{4.43}$$

$$\tilde{F}_{a_2} = -(F_{21,y}+F_{22,y})\,. \tag{4.44}$$

Here we assume that from Fig. (4.13) $\delta_{11}=\delta_{12}\triangleq\delta$ and the input will be the longitudinal wheel forces on the high traction side, with the following input constraint:

$$\underbrace{\begin{bmatrix}F_{11,x,\max}\\ F_{21,x,\max}\end{bmatrix}}_{\mathbf{u}_{\max}} \geq \begin{bmatrix}F_{11,x}\\ F_{21,x}\end{bmatrix} \geq \underbrace{\begin{bmatrix}F_{12,x,\max}\\ F_{22,x,\max}\end{bmatrix}}_{\mathbf{u}_{\min}}, \tag{4.45}$$

which makes sense if the braking forces are assumed to be essentially negative, as they decelerate the given wheels. Furthermore longitudinal forces play a key role in μ-split braking as we have seen. However, they can not only occur due to braking in reality, but also due to the so-called longitudinal slip phenomenon, which is neglected in this model. Thus, the longitudinal force in the model can only be the braking force applied by the brake controller. Finally to get the full picture, the safe set is interpreted as described above:

$$h(\mathbf{x}) = 1-\left(\frac{\beta}{\beta_{\mathrm{cr}}}\right)^2-\left(\frac{\omega}{\omega_{\mathrm{cr}}}\right)^2, \tag{4.46}$$

where β can be approximated by $\beta=v_y/v_x$ according to Fig. (4.12) as these values during the simulations will be much smaller, than 1 radian.

4.3.2 Tire model

The purpose of the wheel model is to determine the forces and moments acting on the wheel, based on the dynamic and geometric quantities. Since the vehicle model is planar, it is also useful to consider the wheel in plane too. The forces in the x and y directions in the vehicle model (4.41) are the forces acting at the wheel's contact point, and the longitudinal force in the x direction being the one already described, since it takes the role of the input signal. For the lateral force there are many models in the literature, from the simplest to the most complex. Pacejka's magic formula [23] is considered to be the most sophisticated one, but it employs an empirical formula, parameter tuning is required. However, a much simpler wheel model may be suitable for us, as the more complex the mechanical model, the more difficult the forward integration becomes in backup control. On the other hand, in order to obtain an affine dynamical system (2.1) by setting F_x to be the input, the linear wheel model will be used instead of the combined wheel model – which takes into account the mutual influence of F_x and F_y . Its formula is the following:

$$F_{ij,y} = C_{\alpha_{ij}} \alpha_{ij} \,, \tag{4.47}$$

where ij refers to the j-th wheel from left to right of the i-th axle and $C_{\alpha_{ij}}$ is the wheel's cornering stiffness, α_{ij} is the wheel's side-slip angle, which can be calculated via vehicle kinematics [22] for the first and second axle:

$$\alpha_{1j} = \tan^{-1}\left(\frac{v_y + x_{\mathrm{V},1j}\omega}{v_x - y_{\mathrm{V},1j}\omega}\right) - \delta \,, \tag{4.48}$$

$$\alpha_{2j} = \tan^{-1}\left(\frac{v_y + x_{\mathrm{V},2j}\omega}{v_x - y_{\mathrm{V},2j}\omega}\right) . \tag{4.49}$$

The linear wheel model uses the initial linear section of the slip curve shown in Fig. (4.14), and the cornering stiffness is its slope.

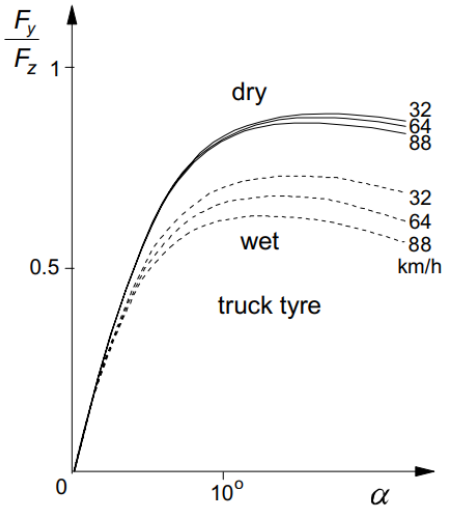

Figure 4.14: Lateral force – slip-angle curve [23].

4.3.3 Driver model

There are many different driver models in the literature, but we need to choose a model of a certain complexity for the task. As the emphasis is not on the driver, but on the braking controller, a simple driver model can be sufficient, the point is to keep the vehicle in lane under braking. What will be used, is the so-called Look-ahead method [24]. It takes into account the expected lateral displacement of the vehicle (in the global coordinate system) with respect to a given reference path, L distance ahead of the vehicle's current position and assuming that the vehicle's direction remains unchanged. For a better understanding helps Fig. 4.15.

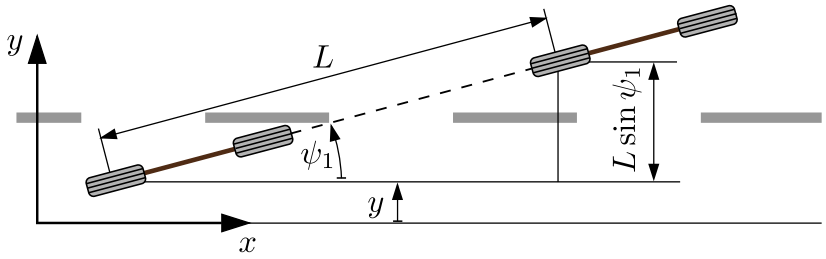

Figure 4.15: Presentation of the Look-ahead method [24].

To determine the steering angle, a proportional feedback controller is assumed for lateral displacement:

$$\delta = -P_y(y + L\sin(\psi))\,, \tag{4.50}$$

where if we consider small angles and use the notation $P_y L \triangleq P_\psi$, then:

$$\delta = -P_y y - P_\psi \psi\,. \tag{4.51}$$

Therefore, we feedback both the current lateral position of the vehicle and the yaw angle, which are part of the state space. However, the parameters for which the driver will react in a stable or unstable way can be found in this article [2]. Although it contains a driver model with time delay, but assuming zero delay, it provides the case without time delay, which will be used in the simulation later on.

4.3.4 Linearized system

A separate subsection should be devoted to linearization, which is an essential aspect of backup set and backup controller design. If equation (2.12) were to used to find the operation points of the vehicle model when braking, it would not provide a single solution that is even numerically valid. The only solution would be the rectilinear motion, which is contradictory under braking. In order to resolve the issue, the so-called frozen-time method can be used. This will give a time invariant system where the transient signals have enough time to decay [25]. In practice, this means that, for a constant velocity v_x, the operation points for $v_{y,0}$ and ω_0 can be determined by detaching the first and last three

rows of the vehicle model's differential equation (4.41). The significance of this will be that, unlike the two simpler examples shown, a new backup controller and a new backup set must be interpreted for each new v_x longitudinal velocity due to the frozen-time. Also, each time the backup set parameter c must be re-checked to see if it is correctly adjusted.

The linearized system is also strongly affected by another factor, namely the driver, which is considered as an external phenomenon with its own control laws. The driver is indeed capable of "pushing" the operation point in the linearized system out of the safe set with his steering angle, thus violating condition (3.4). Until now $\mathbf{u}_0$, even if it was within bounds in conditions (3.4), could be chosen arbitrarily, but this time it must be used to compensate the driver. Moreover, by using frozen-time, this must also be recalculated at each time instant. This means that $\mathbf{u}_0$ must depend on both δ and v_x. To simplify the analysis, let us introduce the following parameter p:

$$\mathbf{u}_0 \triangleq \begin{bmatrix} F_{11,x,0} \\ F_{21,x,0} \end{bmatrix} = \begin{bmatrix} F_{11,x,\max} \\ F_{21,x,\max} \end{bmatrix} - p \begin{bmatrix} F_{11,x,\max} - F_{12,x,\max} \\ F_{21,x,\max} - F_{22,x,\max} \end{bmatrix}, \tag{4.52}$$

which means if $p = 0$, then $\mathbf{u}_0 = [F_{11,x,\max}, F_{21,x,\max}]^\top$ i.e. $\mathbf{u}_0$ is actually the Select-High braking, and if $p = 1$, then $\mathbf{u}_0 = [F_{12,x,\max}, F_{22,x,\max}]^\top$ i.e. $\mathbf{u}_0$ is the Select-Low braking. Since $\mathbf{u}_0$ must obey the input constraint, a bound must be added to p as well, namely $p \in [0, 1]$. However, for a given δ and v_x, there might be no $p \in [0, 1]$, which means that there is no $\mathbf{u}_0$ within the input bounds, thus the operation point is out of the safe set. The Fig. 4.16 shows the surfaces that separate the points in the space (δ, v_x, p) for which the vehicle's operation points $(v_{y,0}, \omega_0)$ fall inside or outside the safe set (4.46).

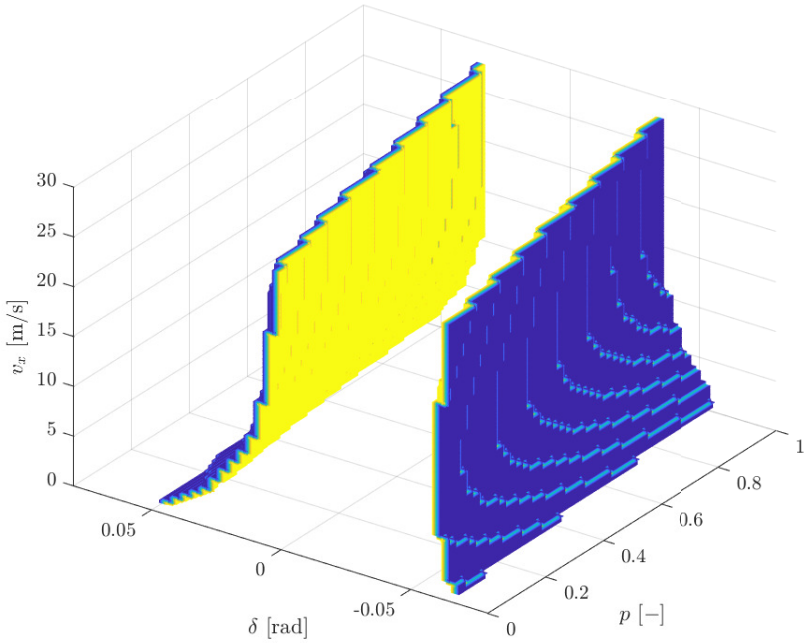

Figure 4.16: Space-separating surface, yellow sides indicate the inner space where $h > 0$.

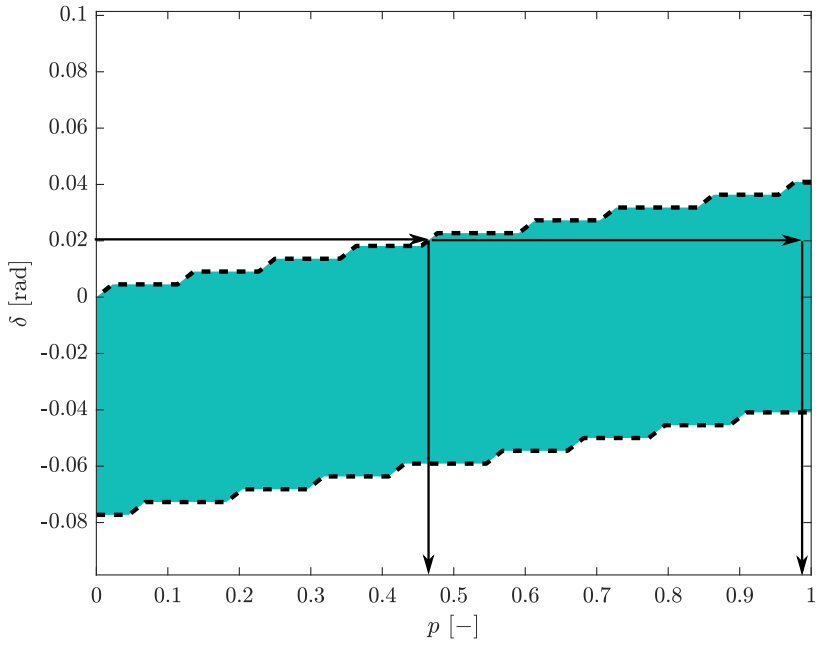

Figure 4.17: Choosing algorithm for p.

The smoothness of these curves obviously depends on how fine refinement is applied, since these surfaces can only be defined numerically due to the model's nonlinearity. To make the control feasible, assume that for each (δ, v_x) there exists $p \in [0, 1]$, then the operation point $(v_{y,0}, \omega_0)$ is located in the inner space in Fig. 4.16, which is covered by yellow surfaces. Then, there is a wide range of choice for choosing p illustrated in Fig. 4.17, which depicts a cross section of the surface in Fig. 4.16 for an arbitrary v_x. According to Fig. 4.17 and if the previous statements hold about p, there always exist a $p_{\min}$ and a $p_{\max}\, \forall(\delta, v_x)$, which can be computed also numerically. Since the purpose of the simulation is to reduce the stopping distance as much as possible while maintaining safety, it is beneficial to keep p as small as possible, since if $p = 0$, then $\mathbf{u}_0 = [F_{11,x,\max}, F_{21,x,\max}]^\top$. However, if $\mathbf{u}_0$ equals to one of its input bounds, it is unfortunate because the backup set would be defined at saturation. This time, according to (3.18) the smallest c would be zero causing the backup set to shrink to a single point. Therefore, a values between $p_{\max}$ and $p_{\min}$ should be selected, which is closer to $p_{\min}$. Let it be one tenth of the distance between them and closer to $p_{\min}$:

$$p(\delta, v_x) = p_{\min}(\delta, v_x) + 0.1 \cdot (p_{\max}(\delta, v_x) - p_{\min}(\delta, v_x)) \,. \tag{4.53}$$

4.3.5 Backup set and backup controller construction

As discussed, due to frozen-time, new operation points have to be calculated at each time instant, which are already known from the previous method. What this causes is the dependency of the backup set and backup controller parameters on v_x, hence the shape

of the backup controller:

$$\mathbf{k}_{\mathrm{b}}(\mathbf{x}_{\mathrm{r}}, v_x) = \begin{cases} \mathbf{u}_{\min} & \text{if } -\mathbf{K}(\mathbf{x}_{\mathrm{r}} - \mathbf{x}_0(v_x)) + \mathbf{u}_0(v_x) < \mathbf{u}_{\min}\,, \\ \mathbf{u}_{\max} & \text{if } -\mathbf{K}(\mathbf{x}_{\mathrm{r}} - \mathbf{x}_0(v_x)) + \mathbf{u}_0(v_x) > \mathbf{u}_{\max}\,, \\ -\mathbf{K}(\mathbf{x}_{\mathrm{r}} - \mathbf{x}_0(v_x)) + \mathbf{u}_0(v_x) & \text{otherwise}\,, \end{cases} \tag{4.54}$$

where $\mathbf{x}_0 = [v_{y,0}, \omega_0]^\top$ and $\mathbf{x}_{\mathrm{r}} = [v_y, \omega]^\top$. The matrix $\mathbf{K}$ is therefore a 2-by-2 matrix of the form:

$$\mathbf{K}(v_x) = \begin{bmatrix} K_1(v_x) & K_2(v_x) \\ K_3(v_x) & K_4(v_x) \end{bmatrix}, \tag{4.55}$$

and the subscript r refers to the reduced state, since in our case the equilibrium space is characterized by only 2 states. The choice of $\mathbf{K}$ is no longer straightforward, since it has to be updated for all v_x, but for simplicity we choose such a $\mathbf{K}$ that will be appropriate for all v_x during the simulation. The backup set will also be affected by the frozen-time method:

$$h_{\mathrm{b}}(\mathbf{x}_{\mathrm{r}}, v_x) = c(v_x) - (\mathbf{x}_{\mathrm{r}} - \mathbf{x}_0(v_x))^\top \mathbf{P}(v_x)(\mathbf{x}_{\mathrm{r}} - \mathbf{x}_0(v_x))\,, \tag{4.56}$$

causing the CTLE to construct new backup sets for every new v_x and determine new c's as well.

4.3.6 Results

Before getting to the results, we shall discuss firstly what circumstances do we assume during the following, $\mu-$split braking situation:

- modelling an emergency braking, when the nominal (desired) control should be the Select-High option: $\mathbf{k}_{\mathrm{d}}(\mathbf{x}) \triangleq [F_{11,x,\max}, F_{21,x,\max}]^\top$, so apply the maximum possible braking force,
- both sides of the road have homogeneous surface but with different coefficient of traction and the wheels do not change between the lanes,
- the driver parameters P_ψ and P_y are chosen in such way that at the beginning of the simulation the driver's behaviour is unstable modelling its unpreparedness for the $\mu-$split braking, but becomes stable at low v_x as the vehicle slows down.

Table 4.3 contains all the parameters used during the simulation. This time, the comparison will be carried out in contrast to the saturated CBF, the weakness of which we would like to improve by using the backup set method. The initial condition of each simulation was a $v_x(t = 0) = 100\,[\mathrm{km/h}]$ and everything else was zero. T integration time should not be set to a large value, because the due to the frozen-time method the operation points and thus the backup set are updated at every instant. So choosing a large T time may also predict inaccurate future dynamics for the backup controller.

Table 4.3: Fixed parameters for vehicle braking

Parameter	marking	value
Maximal yaw rate	ω_{cr}	0.1 [rad/s]
Maximal side-slip angle	β_{cr}	0.03 [rad]
Maximal braking force of wheel 11	$F_{11,x,\max}$	-12000 [N]
Maximal braking force of wheel 12	$F_{12,x,\max}$	-3600 [N]
Maximal braking force of wheel 21	$F_{21,x,\max}$	-5600 [N]
Maximal braking force of wheel 22	$F_{22,x,\max}$	-1400 [N]
Gain matrix of the backup controller	$\mathbf{K}$	$10^3 \cdot (10, 10; 4.6, 4.6)$ [SI]
class $\mathcal{K}_\infty$ for safe set	α	10 [–]
Number of constraints	N_{c}	80 [–]
class $\mathcal{K}_\infty$ for backup set	α_{b}	1 [–]
Prediction time	T	0.125 [s]
Driver parameters	(P_y, P_ψ)	(0.25, 0.6) [SI]
Mass of the vehicle	m	8850 [kg]
Moment of inertia around the vertical axes	I_z	36952 [kgm^2]
Distance between center of mass and first axle	a_1	1.142 [m]
Distance between center of mass and rear axle	a_1	2.458 [m]
Track width	w	1 [m]
Cornering stiffness of first axle wheels	$C_{\alpha 11}, C_{\alpha 12}$	-125000 [N/rad]
Cornering stiffness of rear axle wheels	$C_{\alpha 21}, C_{\alpha 22}$	-115000 [N/rad]

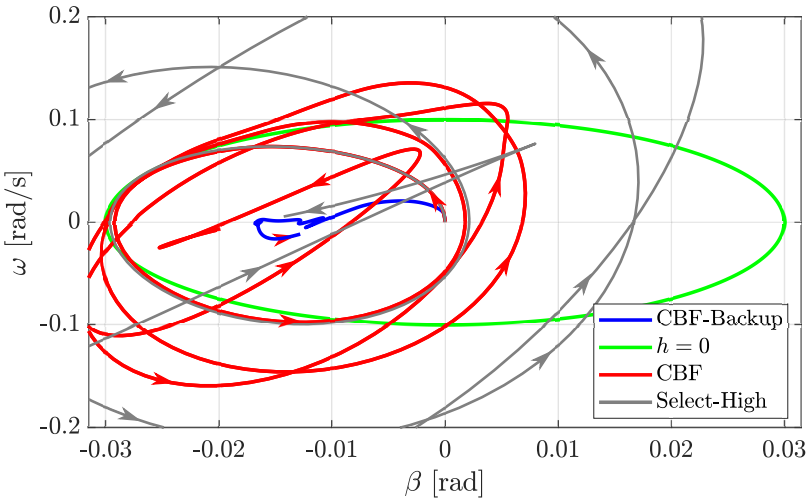

Figure 4.18: Safe set plane with comparison between CBF-Backup (blue) CBF (red) and Select-High (gray).

First of all, let us look at how the trajectories of the system evolve in the phase plane $(\beta - \omega)$, shown in Fig, 4.18. It is understandable that the Select-High braking, marked in gray, has the highest lateral dynamics, since without any assistance from the vehicle side, the driver cannot react properly on his own. With the saturated CBF, which is shown

in red, lower velocities were obtained, but the safety was not fulfilled here either, the trajectory left the safe set. In contrast, the CBF-Backup was able to keep the trajectory inside while ensuring input constraint, which is depicted in Fig. (4.20).

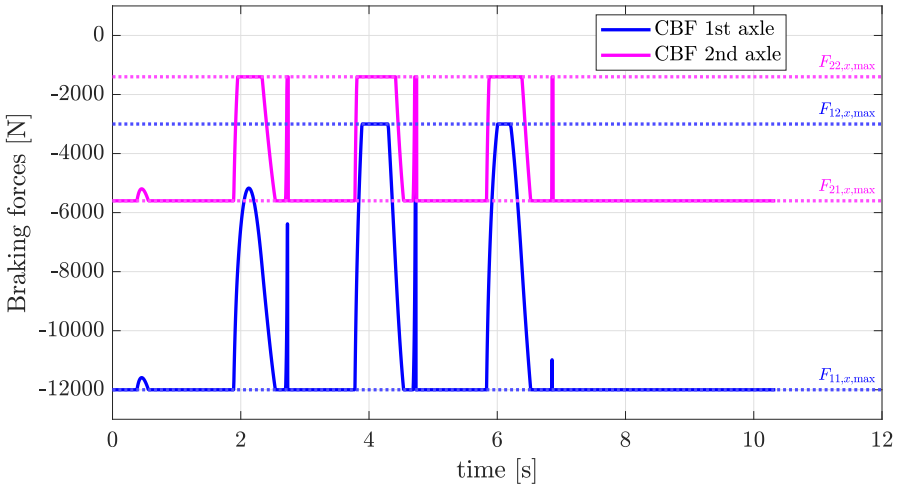

Figure 4.19: Braking forces applied during saturated CBF control.

Fig. 4.19 shows the inputs of the saturated CBF, with the front axle braking force in blue and the rear axle in magenta. Slightly chaotic interventions are observed when they oscillate between their minimum and maximum. Dotted lines indicate the limits of each intervention with corresponding colours. The input signals with Backup set method are shown in Fig. 4.20, which now successfully keeps the trajectory inside the safe set and is also smoother. The difference is that the intervention calculated using the backup set method reduces the braking forces earlier and by a larger amount compared to the saturated CBF. This is no coincidence, since more information about the dynamical system and its evolution is available in the latter case thanks to forward prediction.

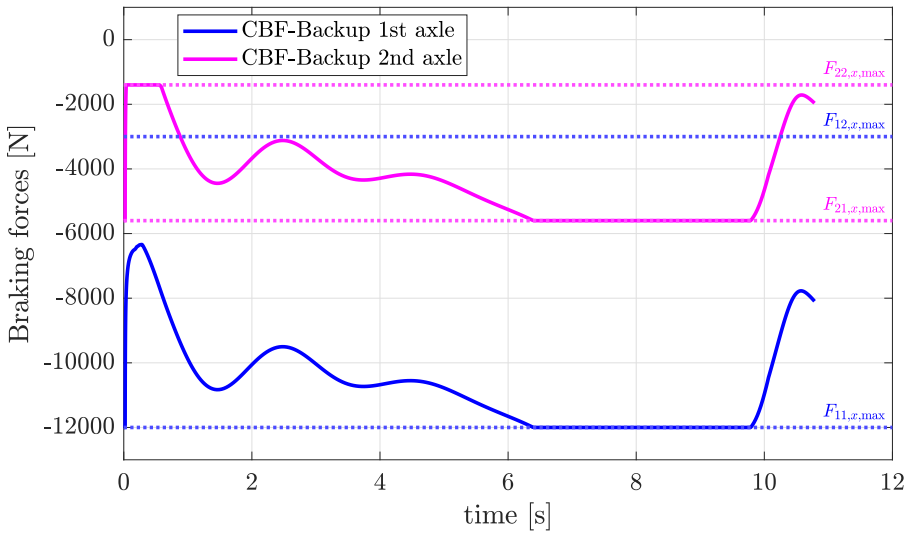

Figure 4.20: Braking forces applied during CBF-Backup control.

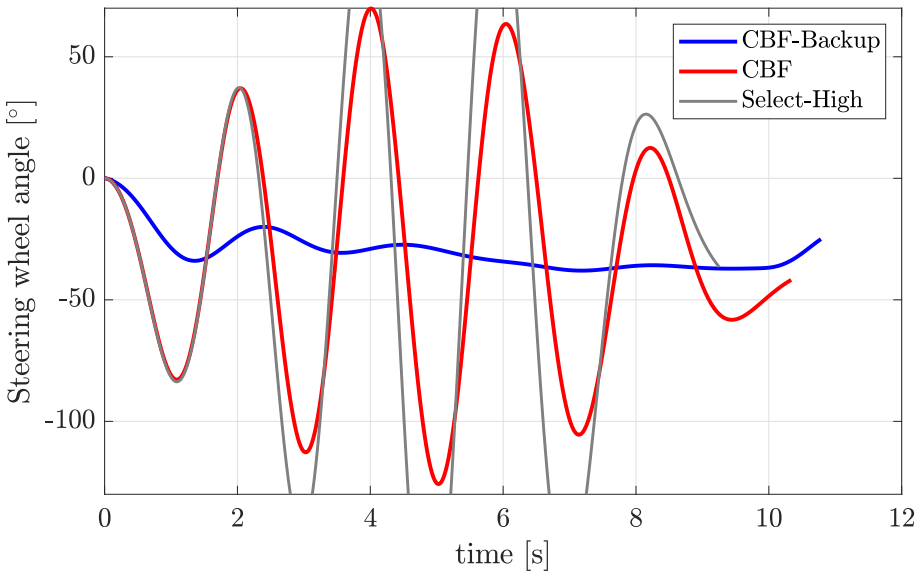

Figure 4.21: Driver's steering angle for CBF (red) and CBF-Backup (red) and Select-High (gray).

What is also important is the effect of brake control on the driver, as shown in Fig. 4.21. A noticeable difference, the saturated CBF and Select-High allowed the driver to steer up to 120 and 200 degrees, whereas with the same parameters in case of CBF-Backup, the driver had to steer 40 degrees at maximum, and did it even more smoothly and gently. Fig. 4.21 also shows how long it took the different methods to stop the vehicle assuming the same initial velocity. Obviously, Select-High was the quickest, as the braking forces are the highest, but in this case the probability of loss of stability is enormous. The same can be said for the saturated CBF, as it also violated the safety. Although it took more time for the Backup-CBF to stop the vehicle, but this is the only simulation that is feasible in reality and still safe. It can be declared that Safety Critical Control combined with the backup set method, can be considered to be a driving assistant system. This can be clearly identified from the results, as the brake control is able to brake the system in a way that the driver can react in an appropriate way. Furthermore, no oscillation occurred in this case, as occurred without the assistance.

What is really worth looking at is the three-dimensional shape of the safe set, where the β is split into two state variables v_x and v_y:

$$h(\mathbf{x}) = 1 - \left(\frac{v_y}{v_x \beta_{\text{cr}}}\right)^2 - \left(\frac{\omega}{\omega_{\text{cr}}}\right)^2, \tag{4.57}$$

In Fig. 4.22 these ellipses are defined around the operation points marked by the red curve, the blue curve shows the real trajectory which starts from $(v_x(t=0), 0, 0)$ and goes down approaching zero longitudinal velocity. We can see that not only the blue trajectory remains in the safe set (as it did in Fig. 4.18), but also the constraint of the backup set ellipses works, with a new parameter c recalculated at each update. In fact, in Fig. 4.22 we can now represent backup sets, which will display planar-ellipses for each v_x in space.

It can be concluded that the method presented in Section 3 can be applied to vehicle braking on an asymmetric surface using frozen-time method.

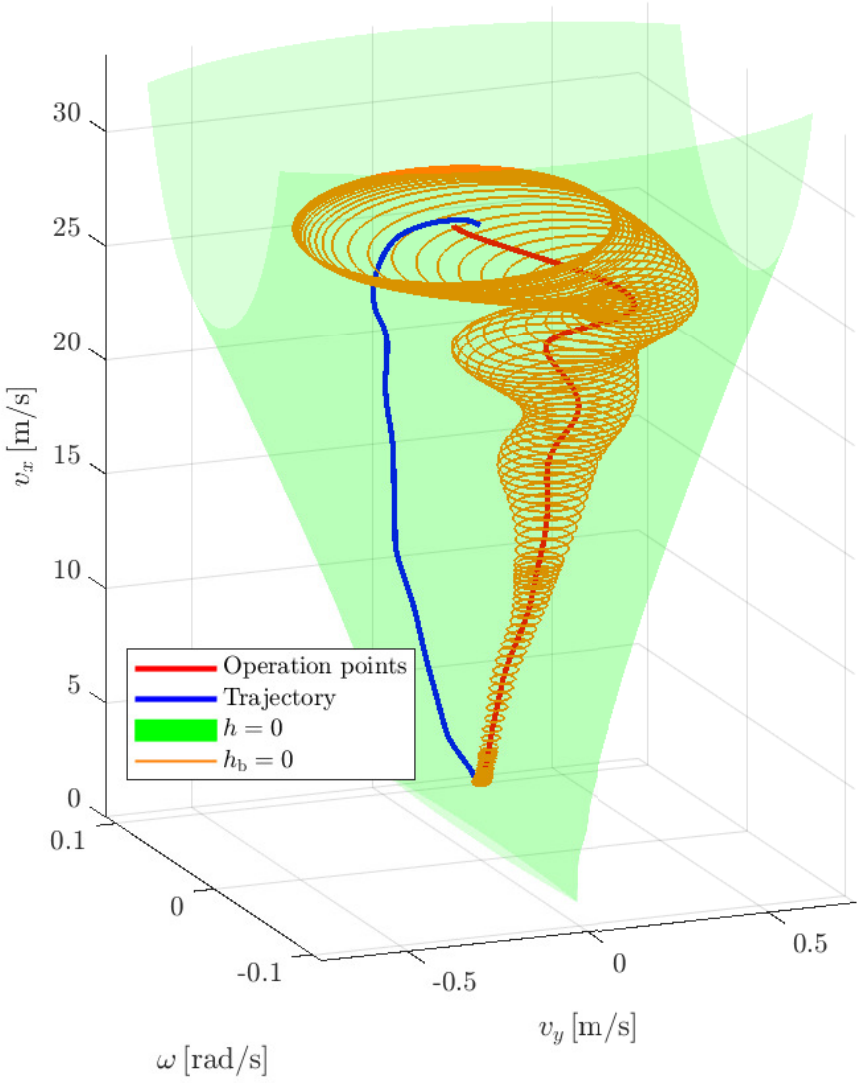

Figure 4.22: Three-dimensional representation of the safe set (green), backup sets (orange), trajectory (blue) and equilibium points (red).

5 Summary

5.1 Results

We have shown how safety-critical control of mechanical systems under input constraints can be performed using the backup set method. For this purpose, we designed our own backup set and backup controller construction method and applied it to three mechanical systems of different complexity. The Lyapunov function was a key element in the structure of the method and was used to prove the invariance of the backup set. In addition, we had to solve the constraint of the backup set: firstly, it had to be constrained into the safe set, and secondly, it had to be constrained into the linear segment of the backup controller. For the former we used our own geometric method in the two-dimensional case, for the latter we used Lagrange-multipliers.

The peculiarities of the three application examples are that in the first, scalar equation example, the future states were analytically determinable and could be substituted in the optimization. After that, we could not do this for the inverted pendulum, but needed a numerical simulation. We also saw in the phase space of the inverted pendulum how much we could improve the controller compared to Safety-Critical Control. In the last example, we had one even more difficult problem, namely that we had no suitable operation point around which the backup set and backup controller could be written during vehicle braking. The way to resolve the problem was to use the so-called frozen-time method, where we calculated the operation points for the lateral dynamics assuming a constant forward speed. This fact implied that all parameters calculated during the backup set method had to be updated for each new velocity. The other complicating factor was the driver, who can interfere with and even deteriorate the vehicle dynamics through steering angle. Against this, we had to tune the static braking force intervention in order to perform the control with the backup set method.

5.2 Future plans

The presented method for constructing backup set and backup controller relies essentially on the operation points of the linearized system, and if we do not have them, or if we are not able to keep them within the safe set with the available control, we are no longer able to control them effectively. Therefore, the relationship between the feasible control signal, the defined safe set and the operation point requires further investigation, as they are strongly interdependent factors. Besides, the appropriate handling of external disturbances, such as the case of the driver, where we cannot control the vehicle safely for certain steering angles.

Reference

[1] L. Gácsi, "Haszonjárművek biztonságkritikus szabályzása osztott tapadású felületen." TDK, 2022.

[2] L. GÁCSI and K. Ádám, "Aszimmetrikus felületen fékező haszonjárművek biztonságkritikus szabályzása: Safety critical control is commercial vehicles braking on asymmetric surfaces," *Nemzetközi Gépészeti Konferencia–OGÉT*, pp. 152–157, 2023.

[3] A. D. Ames, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs with application to adaptive cruise control," in *53rd IEEE Conference on Decision and Control*. IEEE, 2014, pp. 6271–6278.

[4] X. Xu, P. Tabuada, J. W. Grizzle, and A. D. Ames, "Robustness of control barrier functions for safety critical control," *IFAC-PapersOnLine*, vol. 48, no. 27, pp. 54–61, 2015.

[5] A. K. Kiss, T. G. Molnar, A. D. Ames, and G. Orosz, "Control barrier functionals: Safety-critical control for time delay systems," *International Journal of Robust and Nonlinear Control*.

[6] D. R. Agrawal and D. Panagou, "Safe control synthesis via input constrained control barrier functions," in *2021 60th IEEE Conference on Decision and Control (CDC)*. IEEE, 2021, pp. 6113–6118.

[7] H. Wang, K. Margellos, and A. Papachristodoulou, "Safety verification and controller synthesis for systems with input constraints," *arXiv preprint arXiv:2204.09386*, 2022.

[8] A. D. Ames, G. Notomista, Y. Wardi, and M. Egerstedt, "Integral control barrier functions for dynamically defined control laws," *IEEE control systems letters*, vol. 5, no. 3, pp. 887–892, 2020.

[9] T. Gurriet, M. Mote, A. D. Ames, and E. Feron, "An online approach to active set invariance," in *2018 IEEE Conference on Decision and Control (CDC)*. IEEE, 2018, pp. 3592–3599.

[10] T. Gurriet, M. Mote, A. Singletary, P. Nilsson, E. Feron, and A. D. Ames, "A scalable safety critical control framework for nonlinear systems," *IEEE Access*, vol. 8, pp. 187 249–187 275, 2020.

[11] T. G. Molnar, R. K. Cosner, A. W. Singletary, W. Ubellacker, and A. D. Ames, "Model-free safety-critical control for robotic systems," *IEEE robotics and automation letters*, vol. 7, no. 2, pp. 944–951, 2021.

[12] W. Xiao and C. Belta, "Control barrier functions for systems with high relative degree," in *2019 IEEE 58th conference on decision and control (CDC)*. IEEE, 2019, pp. 474–479.

[13] H. K. Khalil, "Lyapunov stability," *Control systems, robotics and automation*, vol. 12, p. 115, 2009.

[14] B. Kouvaritakis and M. Cannon, "Model predictive control," *Switzerland: Springer International Publishing*, vol. 38, 2016.

[15] T. G. Molnar and A. D. Ames, "Safety-critical control with bounded inputs via reduced order models," *arXiv preprint arXiv:2303.03247*, 2023.

[16] A. Singletary, A. Swann, Y. Chen, and A. D. Ames, "Onboard safety guarantees for racing drones: High-speed geofencing with control barrier functions," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 2897–2904, 2022.

[17] Y. Chen, M. Jankovic, M. Santillo, and A. D. Ames, "Backup control barrier functions: Formulation and comparative study," in *2021 60th IEEE Conference on Decision and Control (CDC)*. IEEE, 2021, pp. 6835–6841.

[18] D. P. Bertsekas, *Constrained optimization and Lagrange multiplier methods*. Academic press, 2014.

[19] A. K. Kiss, T. G. Molnar, D. Bachrathy, A. D. Ames, and G. Orosz, "Certifying safety for nonlinear time delay systems via safety functionals: a discretization based approach," in *2021 American Control Conference (ACC)*. IEEE, 2021, pp. 1058–1063.

[20] C. Ahn, B. Kim, and M. Lee, "Modeling and control of an anti-lock brake and steering system for cooperative control on split-mu surfaces," *International Journal of Automotive Technology*, vol. 13, pp. 571–581, 2012.

[21] S. B. Zagorski, "Compatibility of abs disc/drum brakes on class viii vehicles with multiple trailers and their effects on jackknife stability," Ph.D. dissertation, The Ohio State University, 2004.

[22] R. N. Jazar, *Vehicle dynamics*. Springer, 2008, vol. 1.

[23] H. Pacejka, *Tire and vehicle dynamics*. Elsevier, 2005.

[24] I. Vörös and D. Takács, "The effects of trailer towing on the dynamics of a lane-keeping controller," in *Dynamic Systems and Control Conference*, vol. 84270. American Society of Mechanical Engineers, 2020, p. V001T02A004.

[25] B. Varszegi, D. Takacs, and T. Insperger, "Acceleration helps in skateboarding at high speeds," *International Journal of Dynamics and Control*, vol. 6, no. 3, pp. 982–989, 2018.