# Formal Synthesis of Event-Triggered Controllers

T.L. van der Zijden

Master of Science Thesis

Delft University of Technology

Delft Center for Systems and Control

# Formal Synthesis of Event-Triggered Controllers

Master of Science Thesis

For the degree of Master of Science in Systems and Control at Delft University of Technology

T.L. van der Zijden

August 21, 2021

Faculty of Mechanical, Maritime and Materials Engineering (3mE) · Delft University of Technology

# Abstract


In networked control systems (NCS), the components of the control system communicate over a network, which poses challenges to the control synthesis procedure. Signal sampling is one of those challenges. A typical approach is to sample the signal periodically, but this can be inefficient. Event-triggered control (ETC) is a more efficient way to implement digital control for NCS because the control input is only recalculated when a triggering condition is violated. However, formal synthesis of event-triggered controllers is an open topic.

In this work, we propose different methods to synthesize event-triggered controllers. The event-triggered controller consists of a triggering function based on a certificate function and a stabilizing feedback law. Counterexample-Guided Inductive Synthesis (CEGIS) is used to synthesize formally correct controllers. The feedback law is synthesized along with the certificate function or using feedback linearization. This framework is also extended to the synthesis of periodic event-triggered controllers, which periodically evaluate the triggering condition. A method is provided for how the sampling time of the triggering condition should be chosen. The synthesis approach is tested on several systems, through which the effectiveness of the approaches are demonstrated. It is shown that the method can synthesize event-triggered controllers for general nonlinear systems.

# Table of Contents

# Acknowledgements

I would like to thank my daily supervisor Cees Verdier for the helpful guidance he has given me during this project. His enthusiasm for controller synthesis and theoretical background helped me to really enjoy doing research. Furthermore, I would like to thank my supervisor Manuel Mazo for the fruitful discussions we had on different topics.

Delft, University of Technology
August 21, 2021

T.L. van der Zijden

# Chapter 1

# Introduction

## 1-1 Motivation

Networked control systems (NCS) are spatially distributed control systems. Communication between plants, sensors, actuators, and controllers takes place over a digital communication network. NCS are, for example found in mobile sensor networks [36], remote surgery [34], automated highway systems [55] and energy networks [57]. Over the last decades, they have become popular because control systems have become more interconnected and complex. Using a network to connect spatially distributed components of the control system results in flexible architectures and generally reduces installation and maintenance costs [24].

New challenges are introduced due to imperfections of the network. One of those challenges is that signals in NCS need to be sampled before transmission through the network. Another challenge is the limited availability of bandwidth on the network. Therefore components of the network should communicate as little as possible. The sampling process and limited bandwidth have an impact on the design process of the controller. Besides limited bandwidth and sampling, a modern controller must be efficient in energy consumption, computational power, and memory. *Event-triggered controllers* cope with those challenges. These controllers do not recalculate the control input periodically, but only when some performance treshold is exceeded. As a result, they require fewer calculations than conventional controllers, resulting in energy, bandwidth, and computational power savings.

It must be possible to design controllers for NCS, in a way that a wide variety of tasks or specifications can be satisfied. Besides that, guarantees on the correctness of the controller are needed. Ideally, the controller is synthesized automatically, with little interference from a control engineer. *Formal controller synthesis* methods deal with these challenges for controller design.

Although formal synthesis methods have been developed for various systems, formal controller synthesis of event-triggered controllers is an open research topic. Therefore the research goal of this thesis is to develop formal controller synthesis methods for event-triggered control.

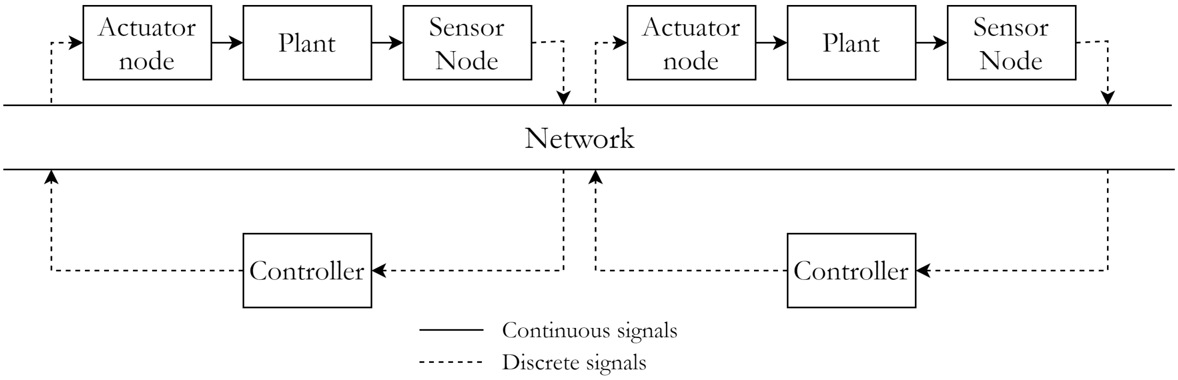



**Figure 1-1:** Multiple control loops that access the same network

## 1-2 Controller Synthesis

A NCS system with multiple control loops is depicted in Figure 1-1. A control loop consists of a plant with corresponding sensor nodes, actuator nodes, and a controller. The sensor node measures selected physical quantities of the plant, such as its position or speed. These quantities are referred to as the systems states. The states are sent through a shared digital communication network. Based on the states, the controller calculates the control signal and sends it to the actuator. An actuator physically controls the plant. NCS can consist of multiple control loops that can access the same network. A scheduler decides which control loop can access the network.

In this thesis we consider automatic synthesis of the controller in the control loop. For the class of linear systems a wide range of controller synthesis methods is available. However the field of controller synthesis for the more general class of nonlinear systems is less mature. Older methods include feedback linearization and backstepping, which are only applicable to a subclass of nonlinear systems [28]. Other methods are Model Predictive Control (MPC) [35] and abstraction and simulation [38, 74, 61]. However those methods need computationally expensive calculations online or they require much memory. Another method designs controllers from control certificate functions. These functions are design tools to modify the system behavior such that the closed-loop system satisfies the desired system properties. Examples of such control certificate functions are the Control Lyapunov Function (CLF) [6, 54], used as a design tool for stabilization and the Control Barrier Function (CBF) [73], used for safety specification.

The Counterexample-Guided Inductive Synthesis (CEGIS) methodology can be used to synthesize control certificate functions and/or control laws [58, 49, 47, 69, 68]. CEGIS is an iterative design method. Each iteration, a candidate solution is found based on previously added states. Subsequently, the candidate is verified, typically utilizing an Satisfiability Modulo Theory (SMT)-solver, a tool capable of verifying the validity of first-order logic formulas. If the candidate is not valid, a state (counter-example) is obtained at which the candidate does not satisfy the control specification. The counter-example is used to improve the candidate solution in the next iteration. The algorithm is iterated until a valid solution is found or until a prespecified termination time or iteration. The synthesis of control certificate functions with CEGIS is not restricted to a subclass of systems and the resulting controllers do not need expensive online computation or memory.

## 1-3 Event-Triggered Control

Modern controllers are implemented on digital devices; hence it is required to implement sampling of measured signals. The sampling of signals at the sensor side is needed to implement a controller on a digital device because the plant's output is continuous-time, and the controller signal is discrete-time. At the actuator side, the converse is done to convert the discrete control signal to a continuous-time input signal to the plant. For example, this can be done by keeping the control input constant between two sampling instants, referred to as zero-order hold.

Conventionally sampling takes place periodically at equally spaced sampling instants and is referred to as time-triggered sampling. Although guarantees on the stability and performance of the system exist and that reliable design methods are available, time-triggered sampling is conservative [5]. It is not always needed to sample and calculate the control input periodically in order to achieve the desired performance. Therefore, time-triggered sampling can be inefficient in terms of energy consumption and network occupation. Event-triggered sampling implementations were proposed to overcome these limitations [60, 40, 62, 23]. Event-triggered controllers only measure and recalculate the control input when some performance bound is exceeded. Event-triggered control (ETC) thus shows potential to incorporate sampling in NCS efficiently. Furthermore, it solves another issue of NCS by actively reducing communication between components. Event-triggered sampling also affects that the energy consumption of the control system is reduced.

A controller consists of two components in the event-triggered sampling framework: a feedback law and a triggering function. The feedback law maps the states to an input, and the triggering function is used to determine when the plant is sampled. Different implementations have been proposed for the triggering function. Motivated by the ability of certificate functions to design controllers, most triggering functions are also based on certificate functions. However, for most implementations, an *ISS-Lyapunov Function* is used that is not easy to find [60, 40]. Another challenge in the design of the triggering function is to avoid *Zeno behavior* [43]. In this context, Zeno behavior means that the event-triggered controller triggers infinitely many times in finite time. Not all ETC implementations can rule out Zeno behavior [56].

Thus, ETC is an efficient way to incorporate sampling and save bandwidth on a network. However, no general method exists to co-synthesize the feedback law and triggering function of the controller. While the CEGIS synthesis method is applied to a wide variety of systems [49, 50, 47, 69, 68], it is not yet applied to the synthesis of event-triggered controllers. This motivates the development of methods that will synthesize CLF for ETC automatically.

## 1-4 Related Work

Besides synthesis of controllers that satisfy classical stability specifications, the research direction of formal controller synthesis is focused on the synthesis of controllers that satisfy *temporal specifications* [7]. These types of specifications are incorporated by logic formulae qualified over time. Temporal specifications can, for example, include statements like: 'The system trajectories must finally reach region B while visiting region A once.' Different methods have been developed for automatic controller synthesis. The synthesis methods mainly fit into three categories: 1) abstraction and simulation, 2) optimization, 3) certificate functions [66].

Abstractions are finite-state models for control systems with an infinite number of states or inputs [38, 74, 61]. States are aggregated into partitions in such a way that the resulting abstraction is a finite-state model. Using the abstraction, one can synthesize controllers that satisfy temporal specifications. Efficient algorithms exist to synthesize controllers for the abstracted model that enforce a temporal specification. For example, fixed-point algorithms described in [61] can be used to construct the controller.

The key idea of the optimization approach is that the formal synthesis problem can be transformed into an optimization problem that is solved with a wide variety of existing solvers [9]. In these approaches, mainly the temporal logic variant Signal Temporal Logic (STL) is used, which directly reasons over continuous-time signals. STL provides quantitative semantics on how robustly the temporal logic formula is satisfied. As a result, the quantitative semantics provide a clear cost function for optimization-based methods, such as MPC [18, 45, 18, 52].

Another formal controller synthesis method is to design controllers based on certificate functions [49, 50, 19, 59, 67]. By their existence, these kinds of functions certify certain behavior. In control theory, Lyapunov functions are a familiar tool to verify the stability of a system. As a tool for the design of stabilizing controllers, the CLF was introduced in [6]. The existence of a CLF for the system ensures that the system can be asymptotically stabilized with an associated feedback law based on the CLF. Furthermore, CBFs can be used to design feedback laws with safety specifications. For control affine systems, a control law can be deduced from the CLF by Sontag's formula [59, 6]. Other types of controllers utilizing a CLF are optimization-based controllers [64, 42] or switching mode controllers [46, 67]. The synthesis of a certificate-based controller thus comes down to finding a valid certificate function.

Although abstraction and optimization approaches are applied to different kinds of systems, most ETC triggering conditions are based on certificate functions. Therefore, the certificate function paradigm is the most natural choice for event-triggered controller synthesis. In the following subsection, literature on the synthesis of certificate functions is described.

### 1-4-1 Certificate Function Synthesis

Synthesis methods for certificate functions can be divided in their versatility to solve the problem for a general class of system, their scalability, and the amount of expert knowledge needed. For the class of strict feedback systems, feedback linearizable systems, and passive or feedback-passive system, a CLF and stabilizing controller can be found by *backstepping* [19, 29, 27, 25, 72]. Another approach to finding certificate functions for control affine polynomial systems is the Sum-Of-Squares (SOS)-programming approach [63, 44, 70]. The certificate function is assumed to have a specific polynomial structure or template. By using a template of the CLF, the problem reduces to finding a suitable set of parameters for the template. The parameters of the template are found by solving an optimization problem. A tool implementing this approach is SOSTOOLS [41].

A more recent approach is the CEGIS algorithm, which is an iterative algorithm, originally proposed to find parameters for programs [58]. Each iteration of the algorithm consists of two steps. In the first step, a candidate function is found. In the second step, it is verified whether the candidate function satisfies the conditions put on it. If the conditions are not satisfied, a so-called witness state or counter-example is extracted, at which the conditions are violated. In [49, 50, 47], the procedure is used to synthesize a CLF for (disturbed) switched systems with reach-while-stay (RWS) specifications. A polynomial template is used for the CLF. Candidates, in the form of a set of template parameters, are found with the SMT-solver Z3. The verification step is implemented using the SMT-solver dReal. When the logic formula is satisfied, it also returns a counter-example at which the formula is satisfied. For polynomial systems and templates, the verification problem can also be written as a system of Linear Matrix Inequalitys (LMIs), for which efficient solvers exists [46]. The approach of [46] can also be extended for the synthesis of a CLF for the class of polynomial control affine systems. Finding a CLF candidate with an SMT solver is a bottleneck of the approaches, as mentioned above [46].

In [67] it is described how candidate functions can be found without assuming an explicit template for the CLF. Genetic Programming (GP) is used to evolve the structure of the candidate functions. A cost function is defined that captures how well the conditions on the certificate function are satisfied over a finite set of states. The parameters of the candidate functions are found by minimizing this cost function. In [67] this is done with the Covariance Matrix Adaption Evolution Strategy (CMA-ES) global optimization solver. CMA-ES is a differentiation-free optimization method, which is robust with respect to discontinuous and non-convex cost functions [22]. The approach of [67] is extended to simultaneous CLF and control law synthesis for hybrid systems [69], and sampled-data systems [68]. Using GP to evolve the structure of the candidate functions, the search space is significantly increased, which also results in increased synthesis time. However, it is not needed to specify a template. Therefore less expert knowledge is required in the synthesis process [68].

### 1-4-2 Event-Triggered Control

ETC was already shortly introduced in Section 1-1 as a method to incorporate sampling in NCS and to reduce communication between components of the network. In ETC, control tasks are executed when an event occurs based on an event-triggered mechanism. Different variants of ETC have been proposed in literature: early ETC implementations focus on Continuous Event-Triggered Control (CETC) for which the triggering condition is continuously evaluated [60, 15, 53, 12]. Consequently this raises the need of extra hardware to implement CETC on digital platforms, which might be impractical. Therefore Periodic Event-Triggered Control (PETC) and Self-Triggered Control (STC) were proposed [39, 62, 5, 33, 32]. In the PETC scheme, the triggering condition is only evaluated periodically. In STC, the controller decides its own next triggering instant, instead of checking a triggering condition.

Most of the ETC research focuses on the design of triggering conditions. Early CETC implementations consider a triggering condition based on state variation [1, 53]. However, these methods lack theoretical results on stability and convergence [30].

As in controller synthesis, control certificate functions can be used for the design of event-triggered controllers. A monotone decrease of the Lyapunov function is ensured by using triggering conditions based on control certificate functions. The most studied case is where the control certificate function is a so-called Input-to-State Stable (ISS) Lyapunov function [60], [40]. The ISS condition allows to prove Input-to-State Stable of the closed-loop system with respect to measurement errors. The ISS-Lyapunov function-based event-triggered controller guarantees the non-existence of Zeno behavior. The condition holds for linear systems [60], and some polynomial systems [5], but in general, it is restrictive. Furthermore, it is not easy to verify the conditions of the ISS Lyapunov function. An approach proposed in [30], [31] disregards the ISS condition and uses a triggering function and feedback law that is based on Sontag's theorem [59]. This theorem states that the existence of a smooth CLF implies smooth stabilizability. The controllers of [30], [31] ensure positivity of the dwell time. They can be applied to control affine systems. In [56] the ISS condition is substantially relaxed, but for some trajectories, Zeno behavior might occur. In [43] a triggering condition is designed for which a minimal convergence rate is known. This triggering condition is based on a normal CLF instead of an ISS Lyapunov function. Extra conditions are put on the CLF, in such a way that no Zeno behavior can occur.

Synthesis of the above described ETC implementations is not trivial, and for most implementations, it is an open question. In this thesis, methods to synthesize event-triggered controllers automatically are developed. The research goal and contributions are further discussed in the next section.

## 1-5 Research Goal and Contributions

The previous section described how event-triggered control implementations use control certificate functions to design triggering functions and feedback laws. In literature, it is assumed that the certificate function and feedback law are given. However, the synthesis of certificate functions and feedback laws is nontrivial for general nonlinear systems. Furthermore, existing methods for the synthesis of certificate functions are not applied to ETC yet. This problem leads to the following research goal:

**Problem**

Develop a framework for automated synthesis of event-triggered controllers for general nonlinear systems with stability and possibly safety specifications that can be readily implemented on digital devices.

To be able to solve this problem, it will be divided in the subproblems described below.

**Subproblem 1**

Synthesize an event-triggered controller for a nonlinear system that ensures that:

S1) the event-triggered controller does not trigger infinitely fast;

S2) the origin is exponentially stable and system trajectories cannot leave a known safe set.

It is needed to monitor the triggering condition continuously, hence it is not readily implementable on digital devices. Therefore the periodic event-triggered mechanism is used, as described in [43]. In this work, an upper bound on time between two measurements is given, recalled to as the maximum admissible sampling period (MASP).

**Subproblem 2**

Find an underapproximation of the maximum admissible sampling period, such that a periodic event-triggered controller provides the same control specification as the continuous event-triggered controller.

As described in the previous sections, the use of certificate functions to synthesize controllers and the triggering function of event-triggered controllers is promising. However, for some certificate-based triggering functions, issues arise. For example, an ISS Lyapunov function is challenging to synthesize [60]. Another implementation is not able to rule out Zeno behavior for some trajectories [56]. Moreover, the approaches of [30], [31] are only applicable to control affine systems. Furthermore, [30], [31] do not provide a known convergence rate. The problems are solved in [43]. In this work, a triggering function based on a CLF with extra conditions is proposed. This CLF excludes Zeno behavior, is not based on the ISS condition, has a known convergence rate, and applies to general nonlinear systems.

Therefore to solve the first problem, a CLF for ETC is proposed which is based on the CLF as proposed in [43]. The CEGIS framework is used to find a function that satisfy the conditions on the proposed CLF for ETC.

Two methods to design a feedback law are applied. For the first method, a feedback law is simultaneously found in the CEGIS framework. The second approach uses feedback linearization to find a control law before synthesis of the CLF. The first method offers the most flexibility in the synthesis process, whereas the second method reduces the number of parameters to be bound in the synthesis process. In [43] a non-closed form expression is given for the maximum time between two sampling instants. We propose to find an approximation of this intersample time by finding Lipschitz constants and optimization.

## 1-6 Outline

In this section, it is described how this thesis is structured. In Chapter 2, control theory results that are used in this work are described. In Chapter 3 it is described how both a CLF and control law can be found in the CEGIS framework. In Chapter 6, this research is concluded.

# Chapter 2

# Preliminaries

In this chapter, preliminary theory is described that is used in this thesis. In Section 2-2 stability theorems are described. Preliminaries on event-triggered control (ETC) are described in Section 2-3. Finally, feedback linearization is described in Section 2-4.

## 2-1 Notation

The sets of natural and real numbers are denoted by $\mathbb{N}, \mathbb{R}$, respectively. A non-negative subset is denoted using the subscript $._{\geq 0}$, e.g. $\mathbb{R}_{\geq 0} = \{x \in \mathbb{R} \mid x \geq 0\}$. Given a function $f : \mathbb{R}^n \to \mathbb{R}^m$ that maps $x \in \mathbb{R}^n$ into $f(x) = (G_1(x), \dots, G_m(x))^T \in \mathbb{R}^m$, $G'(x) = \left(\frac{\partial G_i(x)}{\partial x_j}\right) \in \mathbb{R}^{m \times n}$ is used to denote its Jacobian matrix. A function is said to be $C^l$ if it can be differentiated $l$ times. A function $f : \mathbb{R}^n \to \mathbb{R}^m$ is said to be Lipschitz continuous on compacts if for every compact set $S \subset \mathbb{R}^n$ there exists a constant $L > 0$ such that $\|f(x) - f(y)\| \leq L\|x - y\|$ for every $x, y \in S$. Given a set $S \subseteq \mathbb{R}^n$, the boundary set is denoted as $\partial S$ and the interior as $int(S)$. The image of set $A$ under $f$ is denoted by $f[A]$. An $n$-dimensional vector of zeros is denoted as 0.

## 2-2 Control Lyapunov Functions

Synthesis of controllers for continuous-time nonlinear systems is addressed, described by

$$\dot{x}(t) = f(x(t), u(t)) \quad t \geq 0, \tag{2-1}$$

where $x(t) \in D \subseteq \mathbb{R}^n$ denotes the state vector and $u(t) \in U \subseteq \mathbb{R}^m$ the input vector. It is assumed that the system $f(x(t), u(t))$ is locally Lipschitz over the domain $D$. Furthermore, $D$ contains the origin. The control law $\mathcal{U} : D \to U$ maps the states to the inputs:

$$u(t) = \mathcal{U}(x(t)). \tag{2-2}$$

Combining Equations (2-1) and (2-2) results in a closed-loop system

$$\dot{x}(t) = f(x(t), \mathcal{U}(x(t))). \tag{2-3}$$

Suppose that $\bar{x} \in D$ is an equilibrium point of (2-1); that is $f(\bar{x}, \mathcal{U}(\bar{x}) = 0$. It is assumed that the equilibrium point is at the origin of $\mathbb{R}^n$, i.e. $\bar{x} = 0$.

**Remark 2.1.** *Without loss of generality, it can be assumed that the equilibrium point is at $\bar{x} = 0$. Any equilibrium point can be shifted to the origin via a change of variables. Suppose $\bar{x} \neq 0$ and consider the change of variables $y = x - \bar{x}$, then in the new coordinate system, the system has an equilibrium point at the origin.*

An equilibrium point is stable if all solutions starting in a nearby point stay bounded. The formal definition is as follows:

**Definition 2.2** (Stability and Asymptotic Stability [27, Definition 4.1])). *An equilibrium point $\bar{x} = 0$ of the closed-loop system* (2-3) *is said to be stable if, for any $\varepsilon > 0$, there is $\delta(\epsilon) > 0$, such that*

$$\|x(0)\| < \delta \Rightarrow \|x(t)\| < \varepsilon \quad \forall t \geq 0. \tag{2-4}$$

*Furthermore, $\bar{x}$ is said to be asymptotically stable, if it is stable and $\delta$ can be chosen such that*

$$\|x(0)\| < \delta \Rightarrow \lim_{t \to \infty} x(t) = 0. \tag{2-5}$$

Lyapunov functions are used to access stability of an equilibrium point.

**Definition 2.3** (Lyapunov Function [27]). *Given the closed-loop system* (2-3)*, a continuously differentiable function $V(x)$ defined over the domain $D \subset \mathbb{R}^n$ is called a Lyapunov Function if $V(0) = 0$ and*

$$\begin{array}{ll} (\forall x \in D \backslash \{0\}) & V(x) > 0; \\ (\forall x \in D) & \dot{V}(x) \leq 0. \end{array} \tag{2-6}$$

**Theorem 2.4** ([27, Theorem 4.1], ). *Given the system* (2-3)*. If there exists a Lyapunov function as defined in Definition 2.3, then the origin is a stable equilibrium point. Moreover, if*

$$(\forall x \in D \backslash \{0\}) \; \dot{V}(x) < 0, \tag{2-7}$$

*then the origin is asymptotically stable.*

A stronger version of asymptotic stability is exponential stability.

**Definition 2.5** (Exponential stability [27, Definition 4.5]). *An equilibrium point $\bar{x} = 0$ of the closed-loop system 2-3 is said to be exponentially stable if there exists positive constants $c, k, \gamma_c$ such that*

$$\|x(t)\| \leq k \|x(0)\| e^{-\gamma_c(t)}, \forall \|x(0)\| < c \tag{2-8}$$

Inspired by the use of Lyapunov functions for stability verification of the closed-loop system (2-3), in [6] Control Lyapunov Functions (CLFs) were proposed for the design of stabilizing controllers. The definition of a CLF is as follows:

**Definition 2.6** (Control Lyapunov Function [6])**.** *A $C^1$ function $V : D \to \mathbb{R}$, is called a CLF if $V(0) = 0$, and*

$$(\forall x \in D\backslash\{0\})\ V(x) > 0; \tag{2-9a}$$
$$(\forall x \in D\backslash\{0\})\ \inf_{u\in U}[V'(x)f(x,u) < 0]. \tag{2-9b}$$

The second condition implies that for each $x \in D\backslash\{0\}$, a control input $u \in U$ can be chosen to ensure an instantaneous decrease in $V$. This is also referred to as Lyapunov decrease. Obviously, the condition (2-9b) can be reformulated as

$$(\forall x \in D\backslash\{0\})\ (\exists u \in U)\ V'(x)f(x,u) < 0. \tag{2-10}$$

Below, the theorem ensuring the existence of control law is stated.

**Theorem 2.7** (Existence of Stabilizing Control Law [59, Theorem 1])**.** *Let $U \subseteq \mathbb{R}^m$ be convex, then the existence of a CLF $V$ ensures that a smooth stabilizing controller $u = \mathcal{U}(x)$ exists, where $\mathcal{U} : \mathbb{R}^n \to U$ is continuous.*

The exponentially stabilizing CLF ensure exponential stability (Definition 2.5) of the closed-loop system.

**Definition 2.8** (Exponentially Stabilizing Control Lyapunov Function [4, Definition 1])**.** *Given the system* (2-1)*, a continuously differentiable function $V : D \to \mathbb{R}$ is an exponentially stabilizing control Lyapunov function if there exists positive constants $c_1, c_2, c_3 > 0$, such that*

$$(\forall x \in D)\ c_1\|x\|^2 \leq V(x) \leq c_2\|x\|^2; \tag{2-11a}$$
$$(\forall x \in D)\ \inf_{u\in U}[V'(x)f(x,u) \leq -c_3V(x)]. \tag{2-11b}$$

**Lemma 2.9** ([10, Theorem 2.21])**.** *Given an Exponentially Stabilizing Control Lyapunov Function defined in Definition 2.8, the origin of the system* (2-1) *is exponentially stable and the following holds*

$$\|x(t)\| \leq ce^{-\gamma_c t}\|x(0)\|. \tag{2-12}$$

## 2-3 Event-Triggered Control

In the context of ETC, the nonlinear system (2-1) is considered. As described earlier, ETC was proposed as an alternative to periodic sampling. Event-triggered controllers only recalculate the input when a condition is violated. An additional advantage is that event-triggered controllers can deal with constrained energy, communication, and computational resources [60]. Early ETC implementations focus on Continuous Event-Triggered Control (CETC) for which the triggering condition is continuously evaluated. Consequently, this raises the need for extra hardware to implement CETC on digital platforms, which might be impractical. Therefore Periodic Event-Triggered Control (PETC) was proposed [23]. In the PETC scheme, the triggering condition is only evaluated periodically.

In literature different approaches have been proposed for the design of the triggering function [60, 15, 53, 12]. Most CLF-based ETC implementations make use of an Input-to-State Stable (ISS)-Lyapunov function, which can be difficult to find. In the work of [43] a CETC and PETC implementation are proposed, which is based on a less restrictive CLF. Furthermore, the controller implementations have a known convergence rate, and they do not exhibit Zeno behavior.

The main results and resulting theorems of [43] are described in this section. For a more detailed description and proofs, the interested reader is referred to [43]. A general nonlinear system (2-1) is consdired, where $D = \mathbb{R}^n$. The ETC implementation is based on a $\gamma$*-stabilizing CLF*, which gives a known convergence rate.

**Definition 2.10** ($\gamma$-stabilizing Control Lyapunov Function [43, Definition 2])**.** *Given the system* (2-1)*. Consider a continuous function* $\gamma : [0;\infty) \to [0;\infty)$*, such that* $\gamma(v) > 0\ \forall v > 0$*. A* $C^1$*-smooth function* $V : \mathbb{R}^n \to \mathbb{R}$ *is called a* $\gamma$*-stabilizing CLF, if* $V(0) = 0, \lim_{|x|\to\infty} = \infty$*,* $f(0,\mathcal{U}(0)) = 0$*, and there exists a map* $\mathcal{U} : \mathbb{R}^n \to U$*, satisfying the conditions*

$$(\forall x \in \mathbb{R}^n \backslash \{0\})\ V(x) > 0; \tag{2-13a}$$
$$(\forall x \in \mathbb{R}^n)\ V'(x) f(x,\mathcal{U}(x)) \leq \gamma(V(x)). \tag{2-13b}$$

The problem as posed in [43] is to design an event-triggered controller that has a convergence rate arbitrarily close to convergence rate of the continuous-time controller $u = \mathcal{U}(x)$. The event-triggered controller should satisfy the condition

$$\dot{V}(x(t)) \leq -\sigma\gamma\left(V(x(t)\right) \quad \forall t \geq 0, \tag{2-14}$$

where $\sigma \in (0, 1)$ is a design parameter. The parameter $\sigma$ regulates how close the convergence rate of the event-triggered controller is to the convergence rate of the continuous-time controller.

**Continuous Event-Triggered Control** The continous event-triggered controller proposed in [43] is

$$\begin{aligned} &u(t) = \mathcal{U}\left(x\left(t_k\right)\right)\ \forall t \in \left[t_k, t_{k+1}\right), \quad t_0 = 0, \\ &t_{k+1} = \begin{cases} \inf\left\{t > t_k \mid \mathcal{C}(x(t), x(t_k)) = 0\right\}, & V\left(x\left(t_k\right)\right) > 0; \\ \infty, & V\left(x\left(t_k\right)\right) = 0, \end{cases} . \\ &\mathcal{C}(x(t), x(t_k)) = V'(x(t)) f(x(t),\mathcal{U}(x(t_k))) + \sigma\gamma(V(x(t))). \end{aligned} \tag{2-15}$$

The event-triggered controller gives a convergence rate as defined in the following lemma.

**Lemma 2.11** (CLF with known convergence [43, Proposition 1])**.** *Let the system* (2-1) *have a* $\gamma$*-stabilizing CLF* $V$*, corresponding to the controller* $\mathcal{U}$*. Let* $x(t)$ *be a solution to the closed-loop system* (2-3)*. Furthemore, define the following function* $\Gamma : (0,\infty) \to \mathbb{R}$

$$\Gamma(s) := \int_1^s \frac{dv}{\gamma(v)}, \quad s > 0. \tag{2-16}$$

*Then on the the interval of the solutions existence, the function* $V(x(t)$ *satisfies the following inequality*

$$0 \leq V(x(t)) \leq \Gamma^{-1}(\Gamma(V(0)) - t). \tag{2-17}$$

As shown in [43, Example 1]), for a function $\gamma(v) = \gamma_c v$, the $\gamma$-stabilizing CLF provides exponential decrease of the $\gamma$-stabilizing CLF.

**Proposition 2.12** ([43])**.** *For a function $\gamma(v) = -\gamma_c v$, Equation* (2-17) *reduces to*

$$0 \leq V(x(t)) \leq e^{-\gamma_c t} V(x(0)). \tag{2-18}$$

To assure practical ability of the proposed controller it is proved in [43] that a unique solution between two sampling instants exists, and furthermore that solutions to the closed-loop system are not Zeno.

**Definition 2.13** (Zeno Behavior, [43, Definition 3])**.** *A solution to the closed-loop system* (2-1) *is said to be Zeno, or exhibit Zeno behavior, if the sequence of sampling instants is infinite and has a limit $t_\infty = \lim_{k\to\infty} t_k = \sup_{k\geq 0} t_k < \infty$.*

Three assumptions are adopted to ensure uniqueness of solutions and to exclude Zeno behavior. For a given $x_* \in \mathbb{R}^n$ the sublevel set $B(x_*)$ is defined:

$$B(x_*) := \{x \in \mathbb{R}^n \mid V(x) \leq V(x_*)\}. \tag{2-19}$$

Given a set $S_0 \subset \mathbb{R}^n$, the set $B(S_0)$ is the union of the sets $B(x_*)$ for all $x_* \in S_0$:

$$B(S_0) := \bigcup_{x_* \in S_0} B(x_*). \tag{2-20}$$

**Assumption 2.14** ( [43, Assumption 2])**.** *Let the Lipschitz function $(x_*)$ on the compact set $B(x_*)$ be defined as*

$$L_f(x_*) := \sup_{\substack{x_1,x_2 \in B(x_*) \\ x_1 \neq x_2}} \frac{\|f(x_1, \mathcal{U}(x_*)) - f(x_2, \mathcal{U}(x_*))\|}{\|x_2 - x_1\|} < \infty \quad L_f(0) := 0. \tag{2-21}$$

*It is assumed that the Lipschitz constant $L_f(x_*)$ is a locally bounded function of $x_*$.*

This implies that the difference of the system dynamics for two points $x_1, x_2 \in B(x_*)$ for a last measured state $x_* \in B(x_*)$ can be bounded by a Lipschitz constant $L_f$ as $L_f \|x_1 - x_2\|$.

**Proposition 2.15.** *Assumption 2.14 holds if $\mathcal{U}(x)$ is locally bounded and the jacobian $f'(x, u)$ with respect to $x$ exists and is continuous in $x$ and $u$.*

**Assumption 2.16** ([43, Assumption 3])**.** *The Lipschitz constant of $V'$ on the compact set $B(x_*)$ is defined as*

$$L_{V'}(x_*) := \sup_{\substack{x_1,x_2 \in B(x_*)) \\ x_1 \neq x_2}} \frac{\|V'(x_1) - V'(x_2)\|}{\|x_2 - x_1\|}, \quad L_{V'}(0) := 0. \tag{2-22}$$

*It is assumed that the gradient $V'(x)$ is locally Lipschitz.*

This statement is a stronger of smoothness of the CLF. The following proposition states a sufficient condition to satisfy Assumption 2.14.

**Proposition 2.17** ([43])**.** *Assumption 2.14 is satisfied when* $V \in C^2$*.*

The third assumption allows to establish the relation between the convergence of the CLF and the solution $x(t)$. By assuming a relation between those two, Zeno behavior can be excluded.

**Assumption 2.18** ([43, Assumption 4])**.** *The* $\gamma$*-CLF* $V(x)$ *and continuous-time feedback map* $u = \mathcal{U}(x)$ *satisfy the following properties*

$$\begin{array}{ll} (\forall x \in \mathbb{R}^n) & \|f(x,\mathcal{U}(x)\| \leq M_1(x)|V'(x)|, \\ (\forall x \in \mathbb{R}^n \backslash \{0\}) & \cos\theta(x) \leq -M_2(x), \end{array} \tag{2-23}$$

*where* $M_1(x)$ *is uniformly bounded and* $M_2(x)$ *is uniformly strictly positive on any compact set.*

The intuition behind this assumption is that the conventional CLF decrease condition does not exclude fast changing system solutions. This can be seen by decomposing the closed-loop continuous time system $f(x,\mathcal{U}(x)$ into a sum of two vectors, one parallel to the gradient of the CLF $\nabla V(x) = V'(x)^T$ and the other orthogonal to it as

$$f(x,\mathcal{U}(x)) = -\alpha(x)\nabla V(x) + v_\perp, \tag{2-24}$$

with $\alpha \in \mathbb{R}$ and $\nabla V(x) \perp v_\perp(x) \in \mathbb{R}^n\ \forall x \neq 0$. The traverse component $v_\perp$ is not bounded in any way. This has as a result that the magnitude of the systems solutions can change much faster than the decrease of the CLF. The conditions of Assumption 2.18 ensure that the transverse component of the velocity $v_\perp$ is proportional to the gradient component $-\alpha(x)\nabla V(x)$, and, also that both components decay as $O(\|V'(x)\|)$ as $\|x\| \to 0$. In [43], the conditions in Assumption 2.18 are reformulated into a single condition

**Lemma 2.19** ([43, Lemma 1])**.** *For a* $\gamma$*-stabilizing CLF* $V$*, Assumption 2.18 holds if and only if a locally bounded function* $M(x) > 0$ *exists such that*

$$(\forall x \in D) \quad \|V'(x)\|\,\|f(x,\mathcal{U}(x)\| + \|f(x,\mathcal{U}(x))\|^2 \leq M(x)\,|V'(x)f(x,\mathcal{U}(x))|\,. \tag{2-25}$$

For the real-time implementation of event-triggered controllers, it is needed to have a minimal time between two triggering instants, referred to as the dwell time.

**Definition 2.20** (Dwell-time [43, Definition 4])**.** *The value* $\mathcal{T}(x_0) = \inf_{k\geq 0}(t_{k+1}(x_0) - t_k(x_0))$ *is called the dwell-time.*

To examine the behavior of the solutions between two sampling instant, the auxiliary Cauchy problem, see [43] is introduced:

$$\dot{\xi}(t) = F\left(\xi(t), u_*\right), \xi(0) = \xi_0, t \geq 0, \tag{2-26}$$

where $u_* \in U$. To estimate the time elapsed between consecutive events $t_{k+1} - t_k$ it suffices to study the behavior of the solution $\xi(t) = \xi\left(t \mid x_*, \mathfrak{U}\left(x_*\right)\right)$ to the Cauchy problem (2-26), with $\xi_0 = x_* \neq 0$ and $u_* = \mathcal{U}(x_*)$. Namely, the first instant $\bar{t}$ is found, such that, $V'(\xi(\bar{t}))f\left(\xi(\bar{t}), u_*\right) = -\sigma\gamma(V(\xi(\bar{t})))$. The following lemma implies that $\bar{t} \geq \tau\left(x_*\right)$, where $\tau(\cdot)$ is some function, uniformly strictly positive on any compact set.

**Lemma 2.21** (Solution uniqueness and dwell-time positivity [43, (Lemma 2]). *Let Assumptions 2.14, 2.16 and 2.18 be valid and $\gamma(\cdot)$ either non-decreasing or $C^1$. Then a function $\tau : \mathbb{R}^n \to (0, \infty)$ exists, that depending on $\sigma, \gamma, L_f, L_{V'}, M$ , satisfies the conditions:*

1. *$\tau(\cdot)$ is uniformly strictly positive on any compact set;*
2. *for any $x_*$ the solution $\xi(t)$ is well-defined on the closed interval $[0, \tau(x_*)]$ and*

$$V'(\xi(t))f(\xi(t), \mathcal{U}(x_*)) < -\sigma\gamma(V(\xi(t))) \quad \forall t \in [0, \tau(x_*)) . \tag{2-27}$$

The above theorem lemma leads to the following theorem

**Theorem 2.22.** *Let the assumptions of Lemma 2.21 hold, then the following estimate for the dwell-time of the controller* (2-15) *holds*

$$\mathcal{T}(x_0) \geq \inf_{x \in B(x_0)} \tau(x) > 0, \tag{2-28}$$

*where $B(x_0) := \{x \mid V(x) \leq V(x_0)\}$. The dwell time $\mathcal{T}$ is uniformly positive for all solutions starting in a compact set $S_0$, i.e.* $\inf_{x_0 \in S_0} \mathcal{T}(x_0) > 0$. *The explicit formula of $\tau$ is*

$$\tau(x_*) := \min \left\{ \frac{(1-\sigma)^2}{\mu(x_*)^2 M(x_*)^2}, \frac{1}{1 + 2L_f(x_*)} \right\} > 0, \tag{2-29}$$

*with*

$$\mu(x_*) := \sqrt{e} \max \left\{ L_f(x_*), L_{V'}(x_*) \left(1 + L_f(x_*)\sqrt{e}\right) \right\} . \tag{2-30}$$

**Periodic Event-Triggered Control** Fixing two constants $\tilde{\sigma} \in (\sigma, 1)$ and $K > 1$ the proposed PETC implementation in [43] is

$$\begin{aligned} u(t) = u_n :=\ & \mathcal{U}(x(k_n h)) \quad \forall t \in [k_n h, k_{n+1} h); \quad k_0 = 0, \\ k_{n+1} &= \min\{k > k_n \mid \neg\mathcal{P}(x(kh), u_n)\}, \\ \mathcal{P}(x, u) &:= V'(x)f(x, u) < -\tilde{\sigma}\gamma_c V(x) \wedge \frac{\|V'(x)\| \|f(x, u)\| + \|f(x, u)\|^2}{M(x)|V'(x)f(x, u)|} \leq K, \end{aligned} \tag{2-31}$$

where $h > 0$ is the sampling interval. The choice of $h$ is based on the following lemma. It deals with the solution $\xi(t) = \xi(t \mid \bar{x}, u_*)$ of the Cauchy problem (2-26).

**Lemma 2.23.** *Let the Assumptions 2.14, 2.16 and 2.18 be valid, $\gamma(\cdot)$ be either non-decreasing or $C^1$-smooth, $\tilde{\sigma} \in (\sigma, 1)$ and $K > 1$. Then there exists a function $\tau^0 : \mathbb{R}^n \to (0, \infty)$ such that*

1. *$\tau^0$ is uniformly positive on any compact set;*
2. *if $x_* \neq 0$, $\bar{x} \in B(x_*)$ and $P(\bar{x}, \mathcal{U}(x_*))$ is valid, then the solution $\xi(t)$ is well defined for $t \in [0, \tau^0(x_*)]$ and the following inequality holds*

$$V'(\xi(t))f(\xi, \mathcal{U}(x_*) < -\sigma\gamma(V(\xi(t)) \quad \forall t < \tau^0(x_*). \tag{2-32}$$

The following is the result of this lemma.

**Theorem 2.24.** *Let the assumptions of lemma 2.23 be valid. The periodic event-triggered controller* (2-31) *provides the inequality* (2-18) *for any $x(0) \in S_0$ if the sampling interval $h$ is chosen such that*

$$h \in (0, \tau_{MASP}), \tag{2-33}$$

*where $\tau_{MASP}$ is defined as*

$$\tau_{MASP} := \inf_{x_* \in B(S_0)} \tau^0(x_*) = \inf_{x_* \in B(S_0)} \min \left\{ \frac{(\tilde{\sigma} - \sigma)^2}{K^2 \mu(x_*)^2 M(x_*)^2 \tilde{\sigma}^2}, \frac{1}{1 + 2L_f(x_*)} \right\}, \tag{2-34}$$

*and $\mu(x_*)$ as in Equation* (2-30)

## 2-4 Feedback Linearization

In this section the technique of feedback linearization is described. In Chapter 3 feedback linearization is used to design the feedback law of event-triggered controllers. Feedback linearization is a well-known approach to nonlinear control design. The central idea is to transform the nonlinear system dynamics into linear dynamics, such that linear control techniques can be applied. It is described how a feedback linearizing controller can be designed. Consider the nonlinear system

$$\dot{x} = f(x) + g(x)u, \tag{2-35}$$

The matrices $f : D \to \mathbb{R}^n$ and $g : D \to \mathbb{R}^{n \times m}$ are sufficiently smooth on a domain $D \subset \mathbb{R}^n$. Furthermore, $f(0) = 0$. By a change of coordinates the dynamics can be mapped in a special form. This mapping is recalled to as a diffeomorphism, which is defined as follows:

**Definition 2.25** (Diffeomorphism, [27])**.** *The continuously differentiable map $z = \Phi(x)$ with a continuously differentiable inverse is called a diffeomorphism.*

A feedback linearizable system is defined as follows.

**Definition 2.26** (Feedback Linearizability, [27, Definition 13.1 ])**.** *Consider the system Equation* (2-35)*. The system is said to be feedback linearizable if there exists a diffeomorphism $\Phi : D \to \mathbb{R}^n$* (2.25)*, such that the image $\Phi[D]$ contains the origin and the change of variables $z = \Phi(x)$ transforms the system into the form*

$$\dot{z} = Az + B\gamma(x)[u - \alpha(x)], \tag{2-36}$$

*with $(A, B)$ controllable and $\gamma(x)$ nonsingular for all $x \in D$.*

There exists conditions that guarantee the existence of a transformed system of the form (2-36) exists. The interested reader is referred to Appendix A for a more detailed description on this.

Let $\gamma^{-1}(x)$ be the inverse matrix of $\gamma(x)$ in Equation (2-36). Then the feedback control

$$u = \mathcal{U}(x) = \gamma^{-1}(x)[\alpha(x) + v] \tag{2-37}$$

cancels the nonlinearities, which results in the system with linear dynamics

$$\dot{z} = Az + Bv. \tag{2-38}$$

As a result, the origin can be stabilized by $v = -Kz$, where $A-BK$ is Hurwitz (all eigenvalues negative) and the closed-loop system, in the $z$-coordinates is

$$\dot{z} = (A - BK)z. \tag{2-39}$$

The control in $x$-coordinates is

$$u = \mathcal{U}(x) = \gamma^{-1}(x)[\alpha(x) - K\Phi(x)], \tag{2-40}$$

and closed-loop system is

$$\dot{x} = f(x) + g(x)\gamma^{-1}(x)[\alpha(x) - K\Phi(x)] := f_c(x) \tag{2-41}$$

**Lemma 2.27.** *Given the system* (2-39)*, then the origin of the closed-loop system $f_c(x)$ inherits exponential stability of the origin in $z$-coordinates.*

*Proof.* Since,

$$\dot{z} = \frac{\partial \Phi}{\partial x}(x)\dot{x} = (A - BK)z, \tag{2-42}$$

the following holds:

$$\frac{\partial \Phi}{\partial x}(x) f_c(x) = (A - BK)\Phi(x). \tag{2-43}$$

Using the fact that $f_c(0) = 0$, the Jacobian matrix of each side of Equation (2-43), the following is obtained:

$$\frac{\partial f_c}{\partial x}(0) = J^{-1}(A - BK)J, \quad \text{where} \quad J = \frac{\partial \Phi}{\partial x}(0) \tag{2-44}$$

The matrix $J$ is nonsingular and the similarity transformation $J^{-1}(A - BK)J$ preserves the eigenvalues of $A - BK$. Hence, $J^{-1}(A - BK)J$ is Hurwitz, and $x = 0$ is exponentially stable by Theorem 3.2 of [28]. □

# Chapter 3

# CEGIS

In this chapter, the first subproblem as defined in Section 3-1 is considered. For the sake of convenience, the problem is recalled.

**Subproblem 1**

Synthesize an event-triggered controller for a nonlinear system that ensures that:
S1) the event-triggered controller does not trigger infinitely fast;
S2) the origin is exponentially stable and system trajectories cannot leave a known safe set.

As described in Section 2-3, an event-triggered controller consists of a feedback law and triggering condition. The triggering condition is based on a certificate function. In this chapter, two different certificate functions are used, inspired by [43], to certify the behavior as described in the above subproblem.

In this chapter, Counterexample-Guided Inductive Synthesis (CEGIS) is used to synthesize certificate functions [26, 46]. Two different methods are proposed to synthesize a feedback law. In the first approach, a closed-form feedback law for general nonlinear systems is co-synthesized with the certificate function in the CEGIS synthesis loop. The second approach uses feedback linearization to synthesize a feedback law, whereafter a certificate function is found. This approach is only suited for the class of feedback linearizable systems, see Section 2-4.

A predefined template assumes a particular structure of the certificate function and possibly other functions, reducing the synthesis procedure to finding parameters of the templates. Several studies have proposed to use CEGIS for the synthesis of different kinds of controllers for different systems. For example, CEGIS is applied to nonlinear switched systems [46], general nonlinear systems [67] and sampled-data systems [68]. The main contribution of this chapter is the extension of the CEGIS framework to the synthesis of certificate functions suited for event-triggered controllers.

This chapter is organized as follows. To begin, a formal problem definition for Subproblem 1 is provided in Section 3-1. The triggering condition is then introduced in Section 3-2, followed by the introduction of the certificate functions used in the triggering condition. In Section 3-3 the two methods to design a feedback law are presented, whereafter the CEGIS framework is outlined. Finally, this chapter is summarized in Section 3-4.

## 3-1 Problem Definition

In this section the problem is formalized. A control affine system is considered of the form

$$\dot{x}(t) = f(x(t)) + g(x(t))u(t) = F(x(t), u(t)), \quad t \geq 0, \tag{3-1}$$

where $x(t) \in D \subseteq \mathbb{R}^n$ and $u(t) \in \mathbb{R}^m$ denote the state and input respectively. The sets $D$ is the state space. The control law $\mathcal{U} : D \to \mathbb{R}^m$ maps the states to the inputs. The origin is contained in the domain $D$ and the system has an equilibrium point $\bar{x} = 0$, i.e. $F(0, 0) = 0$. Assuming the equilibrium point to be at the origin can be done without loss of generality (see Remark 2.1). Furthermore, the following assumption is adopted.

**Assumption 3.1.** *The jacobian $F'(x, u)$ exists and is continuous in $x$ and $u$ for all $x \in D$, $u \in \mathbb{R}^m$.*

In this chapter, the event-triggered controller is of the form

$$\begin{aligned} u(t) &= \mathcal{U}\left(x\left(t_k\right)\right) \;\; \forall t \in \left[t_k, t_{k+1}\right), \quad t_0 = 0, \\ t_{k+1} &= \begin{cases} \inf\{t > t_k \mid \mathcal{C}(x(t), x(t_k)) = 0\}, & V\left(x\left(t_k\right)\right) > 0 \\ \infty, & V\left(x\left(t_k\right)\right) = 0. \end{cases} \end{aligned} \tag{3-2}$$

with $\mathcal{U}$ and $\mathcal{C}$ the feedback law and triggering function, respectively.

It is wanted to synthesize controllers that meet the control specification in a compact set $D$. As in the synthesis process the controller is not formally verified, it must be made sure that trajectories cannot leave $D$. The first control specification is defined as follows. All trajectories starting in a set $R \subseteq int(D)$ stay in $D$ and the origin is exponentially stable with a desired convergence rate $\gamma_c > 0$ and gain $\rho > 0$:

$$\texttt{CS}_1 : \forall x\left(0\right) \in R, \forall t \in \left[0, \infty\right) : x(t) \in D \wedge \lim_{t \to \infty} x(t) = 0 \wedge \|x(t)\| \leq \rho e^{-\gamma_c t} \|x(0)\|. \tag{3-3}$$

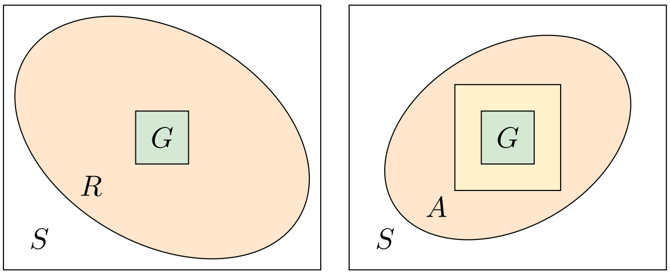
The set $R$ follows from the certificate functions defined later. The above definition allows to pose a formalized version of Subproblem 1.

**Subproblem 1 for control specification $\texttt{CS}_1$ (formalized)**

Given the compact sets $D$, the system (3-1) and a desired exponential convergence rate $\gamma_c > 0$, synthesize an event-triggered controller of the form (3-2) consisting of a control law $u = \mathcal{U}(x)$ and triggering condition $\mathcal{C}(x)$, such that if $x(0) \in R$, all solutions $x(t) \in R$ to the closed-loop system given by (3-1) and (3-2)

S1) do not exhibit Zeno behavior;
S2) satisfy control specification $\texttt{CS}_1$.

Note that for $\texttt{CS}_1$ the set $R$ is not specified by the user. Below a control specification is defined in which a set of initial states is explicitly defined by the user. It is given as follows.




**(a)** $\texttt{CS}_1$ **(b)** $\texttt{CS}_2$

**Figure 3-1:** Representation of the sets

All trajectories starting in a set $I \subset S$ stay in $S$ and the origin is exponentially stable with a desired convergence rate $\gamma_c > 0$ and gain $\rho > 0$:

$$\texttt{CS}_2 : \forall x(0) \in I, \forall t \in [0, \infty) : x(t) \in S \wedge \lim_{t \to \infty} x(t) = 0 \wedge \|x(t)\| \leq \rho e^{-\gamma_c t} \|x(0)\|. \qquad (3\text{-}4)$$

A representation of the sets used in the control specifications, projected onto a two-dimensional space, is shown in Figure 3-1. For $\texttt{CS}_2$ the formalized subproblem is as follows.

**Subproblem 1 for control specification $\texttt{CS}_2$ (formalized)**

Given the compact sets $(S, I)$, the system (3-1) and a desired exponential convergence rate $\gamma_c > 0$, synthesize an event-triggered controller of the form (3-2) consisting of a control law $u = \mathcal{U}(x)$ and triggering condition $\mathcal{C}(x)$, such that if $x(0) \in I$, all solutions $x(t) \in R$ to the closed-loop system given by (3-1) and (3-2)

S1) do not exhibit Zeno behavior;

S2) satisfy specification $\texttt{CS}_2$.

We propose to solve the problems by using the triggering condition of [43] as described in Section 2-3. The triggering condition is based on a certificate function, which varies depending on the specification.

## 3-2 Control Strategy

In Section 2-3 the event-triggered control (ETC) implementation of [43] was presented. The implementation is based on a $\gamma$-stabilizing Control Lyapunov Function (CLF), which achieves stabilization with a known convergence rate. Furthermore, by the additional assumption Assumptions 2.14, 2.16 and 2.18, the solutions of the closed-loop system are uniquely defined, and the controller does not exhibit Zeno behavior. However, the $\gamma$-stabilizing CLF can not directly be synthesized with CEGIS. The reason is that SMT-solvers can only verify propositional formulas over a compact domain.

### 3-2-1 Certificate Function for Control Specification 1

To enforce $\texttt{CS}_1$ the following certificate function is proposed.

**Definition 3.2** (Exponential Stabilizing Control Lyapunov Function for Event-Triggered Control)**.** *Given a convergence rate $\gamma_c > 0$, a $C^2$ smooth function $V : D \to \mathbb{R}$ is called an exponentially stabilizing Control Lyapunov Function for event-triggered control w.r.t. the compact sets $S$ and system* (3-1) *if $V(0) = 0$ and there exists a locally bounded function $M : D \to \mathbb{R}$, a locally bounded function $\mathcal{U} : D \to \mathbb{R}^m$ and scalar constants $c_1, c_2 > 0$, such that*

$$(\forall x \in D)\ M(x) > 0; \tag{3-5a}$$

$$(\forall x \in D)\ c_1\|x\|^2 \leq V(x) \leq c_2\|x\|^2; \tag{3-5b}$$

$$(\forall x \in D)\ V'(x)F(x,\mathcal{U}(x)) \leq -\gamma_c V(x); \tag{3-5c}$$

$$(\forall x \in D)\ \|V'(x)\|\,\|F(x,\mathcal{U}(x))\| + \|F(x,\mathcal{U}(x))\|^2 \leq M(x)\left|V'(x)F(x,\mathcal{U}(x))\right|. \tag{3-5d}$$

For simplicity, the Exponential Stabilizing Control Lyapunov Function for Event-Triggered Control is referred to as CLF in this chapter. Note that the CLF above is a variant of the $\gamma$-stabilizing CLF as defined in Definition 2.10, with $\gamma(v) = \gamma_c v$, such that exponential convergence is obtained.

The triggering function used, is based on the one described in Section 2-3. Given a constant $\sigma \in (0, 1)$, convergence rate $\gamma_c$ and certificate function as defined in Definition 3.2, the triggering function is defined as:

$$\mathcal{C}\left(x(t), x\left(t_k\right)\right) := V'(x(t))F\left(x(t),\mathcal{U}\left(x\left(t_k\right)\right)\right) + \sigma\gamma_c V(x(t)). \tag{3-6}$$

The parameter $\sigma$ regulates the convergence rate. By the following lemma, the controller (3-1) provides dwell-time positivity.

**Lemma 3.3.** *Given a system* (3-1) *satisfying Assumption 3.1, functions $V,\mathcal{U}, M$, such that $V$ is a CLF w.r.t $\mathcal{U}$, $M$, a desired convergence rate $\gamma_c > 0$, and a compact sublevel set $R := \{x \in D \mid V(x) \leq \beta\} \subset int(D)$, then if $x(0) \in R$, the minimal dwell-time $\mathcal{T}(x_0) := \inf_{k\geq 0}(t_{k+1}(x_0) - t_k(x_0))$ is uniformly positive for all $x(0) \in R$.*

*Proof.* First we show that the assumptions Assumptions 2.14, 2.16 and 2.18 hold on $R$. By Assumption 3.1, it follows from Proposition 2.15 that Assumption 2.14 is satisfied on $R$. By definition of the CLF, $V$ is $C^2$ smooth, therefore from Proposition 2.17 it follows that Assumption 2.16 is satisfied on $R$. By definition of the CLF, $M > 0$ is locally bounded for all $x \in R$. Furthermore as condition Equation (3-5b) holds, by Lemma 2.19 Assumption 2.18 is satisfied on $R$. Furthermore, $V$ is a $\gamma$-stabilizing CLF for $x_0 \in R$, with $\gamma(v) = \gamma_c v$, $\gamma_c > 0$. For Lemma 2.21 to hold on $R$, the assumptions need to hold over $B(R)$, as defined in Equation (2-20). As $R$ is a sublevel set, $B(R) = R$ and Lemma 2.21 holds. Therefore, for $x_0 \in R$ the minimal dwell time satisfies

$$\mathcal{T}(x_0) \geq \inf_{x\in R} \tau(x) > 0, \tag{3-7}$$

where $\tau$ is defined in Equation (2-29). Hence, for all $x_0 \in R$, $\mathcal{T}$ is uniformly positive on $R$: $\inf_{x_0\in R} \mathcal{T}\left(x_0\right) > 0$.

□

Using the CLF and triggering condition the result for Subproblem 1 for $\texttt{CS}_1$ is formulated:

**Theorem 3.4.** *Let $R$ be a sublevel set contained in $D$, i.e. $R := \{x \in D \mid V(x) \leq \beta\}$, with $\beta$ such that $R \subseteq int(D)$. Given a system* (3-1) *satisfying Assumption 3.1, an event-triggered controller of the form* (3-2)*, a CLF $V$, feedback law $\mathcal{U}$ and function $M$ as defined in Definition 3.2 w.r.t compact set $D$, a desired convergence rate $\gamma_c > 0$ and constant parameter $\sigma > 0$, then if $x(0) \in R$, solutions $x(t)$ to the closed-loop system given by* (3-1) *and* (3-2)

*S1) do not exhibit Zeno behavior;*
*S2) satisfy $\texttt{CS}_1$.*

*Proof.* Since $R := \{x \in D \mid V(x) \leq \beta\}$ and $D$ is compact, it follows that $R$ is compact. By Lemma 3.3 the minimal dwell time $\mathcal{T}$ is uniformly positive on $R$, hence no Zeno solutions can exist on $R$, and S1) is proven. From Equation (3-5c) and the triggering condition formed by (3-2) and (3-6), it follows that for all $x(t) \in R$, there exists a control input $u(t) = \mathcal{U}(x(t))$ such that for all $t > 0$, $x(t) \in R$, it holds that $V'(x(t))F(x(t),\mathcal{U}(x(t))) < -\gamma_c V(x(t))$. By Lemma 2.11 and Proposition 2.12 it holds that for all $x(t) \in R$

$$V(x(t)) \leq e^{-\gamma_c t} V(x(0)) \leq V(x(0)). \tag{3-8}$$

Hence, given $x(0) \in R$, we have $V(x(t)) < V(x(0))$, while $x(t) \in R$. Since $R \subset int(D)$ we have that $V(x(t)) > V(x(0))$ for all $x \in \partial D$. Hence, for all $x(0) \in R$, $x(t)$ remains in the interior of $D$ for all $t \geq 0$. Furthermore, by the conditions (3-5b), (3-5c) $V$ is an exponentially stabilizing CLF (see Definition 2.8) and the origin is exponentially stable by Lemma 2.9 and thus S2) is satisfied. □

### 3-2-2 Certificate Function for Control Specification 2

A set of initial states for which $\texttt{CS}_1$ hold does not directly follow from its definition. Therefore $\texttt{CS}_2$ was introduced. For this control specification, the set of initial states is defined by the user. In this section a control Lyapunov barrier functions (CLBF) is defined, which establishes sufficient conditions for specification $\texttt{CS}_2$.

**Definition 3.5** (Exponential Stabilizing Control Lyapunov Barrier Function for Event-Triggered Control)**.** *Given a convergence rate $\gamma_c$, a $C^2$ smooth function $V : D \to \mathbb{R}$ is called an exponentially stabilizing Control Lyapunov Barrier Function for event-triggered control w.r.t. the compact sets $S, I$ and system* (3-1) *if $V(0) = 0$ and there exists a locally bounded function $M : D \to \mathbb{R}$, a locally bounded function $\mathcal{U} : D \to \mathbb{R}^m$ and scalar constants $c_1, c_2, \beta > 0$, such that*

$$(\forall x \in S)\ M(x) > 0 \tag{3-9a}$$
$$(\forall x \in A)\ c_1\|x\|^2 \leq V(x) \leq c_2\|x\|^2 \tag{3-9b}$$
$$(\forall x \in \partial S)\ V(x) > \beta \tag{3-9c}$$
$$(\forall x \in I)\ V(x) \leq \beta \tag{3-9d}$$
$$(\forall x \in A)\ V'(x)F(x,\mathcal{U}(x)) \leq -\gamma_c V(x) \tag{3-9e}$$
$$(\forall x \in A)\ \|V'(x)\|\, \|F(x,\mathcal{U}(x)\| + \|F(x,\mathcal{U}(x))\|^2 \leq M(x)\, |V'(x)F(x,\mathcal{U}(x))|\,, \tag{3-9f}$$

*where $A := \{x \in S \mid V(x) \leq \beta\}$.*

By the following lemma, the controller (3-1), employing the CLBF provides dwell-time positivity.

**Lemma 3.6.** *Given a system* (3-1) *satisfying Assumption 3.1, functions* $V, \mathcal{U}, M$*, such that* $V$ *is a CLBF w.r.t* $\mathcal{U}$*,* $M$ *and a desired convergence rate* $\gamma_c > 0$*, then if* $x(0) \in I$*, the minimal dwell-time* $\mathcal{T}(x_0) := \inf_{k\geq 0}(t_{k+1}(x_0) - t_k(x_0))$ *is uniformly positive for all* $x(0) \in I$*.*

*Proof.* Analogous to the proof of Lemma 3.3 Assumptions 2.14, 2.16 and 2.18 hold on $A$. By definition, $A$ is also a sublevelset of $V$ and from Lemma 2.21 it holds that the minimal dwell time satisfies

$$\mathcal{T}(x_0) \geq \inf_{x\in A)} \tau(x) > 0, \tag{3-10}$$

where $\tau$ is defined in Equation (2-29). Hence, for all $x(0) \in I$, $\mathcal{T}$ is uniformly positive on $R$: $\inf_{x(0)\in R} \mathcal{T}(x_0) > 0$. □

Using the CLBF and triggering condition the result for Subproblem 1 for $\texttt{CS}_2$ is formulated:

**Theorem 3.7.** *Given a system* (3-1)*, satisfying Assumption 3.1, an event-triggered controller of the form* (3-2)*, a CLBF* $V$*, feedback law* $\mathcal{U}$ *and function* $M$ *as defined in Equation* (3-9) *w.r.t compact sets* $S, I$*, a desired convergence rate* $\gamma_c > 0$ *and constant parameter* $\sigma \in (0, 1)$*, then if* $x(0) \in I$ *solutions* $x(t)$ *to the closed-loop system given by* (3-1) *and* (3-2)

*S1) do not exhibit Zeno behavior;*
*S2) satisfy* $\texttt{CS}_2$*.*

*Proof.* As $A \subset S$, $A$ is a compact set. Let $R = A$. Then by Lemma 3.3 the dwell time $\mathcal{T}$ is uniformly positive on $A$, hence no Zeno solutions can exist on $A$, and S1) is proven. For $x(0) \in I$ it follows from Equation (3-9d) and the definition of $A$ that $x(0) \in A$. The proof of S2) is analogous to the proof of S2) in Theorem 3.4, from which it follows that for all $x(0) \in A$, $x(t)$ remains in the interior of $S$ for all $t \geq 0$. Furthermore, by the conditions (3-9b), (3-9e) $V$ is an exponentially stabilizing CLF (see Definition 2.8) and the origin is exponentially stable by Lemma 2.9 and S2) is satisfied. □

## 3-3 Synthesis

In the previous section, certificate functions were established to infer that the closed-loop system satisfies $\texttt{CS}_1$ or $\texttt{CS}_2$. In this section, a framework is proposed to synthesize a certificate function $V$, feedback law $\mathcal{U}$ and function $M$, such that the closed-loop system satisfy the control specifications. Candidate solutions for the tuple $(V, \mathcal{U}, M)$ are found, using different methods. The candidate solutions are subsequently formally verified, using an Satisfiability Modulo Theory (SMT)-solver. A counterexample at which the conditions are not met is extracted to improve the candidate solution in the next iteration. This framework is also referred to as CEGIS.

The SMT-solver dReal can not verify formulas with conditions that become tight for $x \to 0$ due to the $\delta$-complete decision procedure, dReal relies on. This is described in more detail in Section 3-3-4. For the CLF all conditions become tight for $x \to 0$, and for the CLBF the conditions (3-9a), (3-9b), (3-9d) and (3-9e). Therefore in the remaining of this thesis a compact goal set $G \subset S$ is excluded from verification. Verification of the goal set is subject of future research.

This section is organized as follows. In Section 3-3-1 the standard form of a CEGIS problem is introduced. In Section 3-3-2 and Section 3-3-3 it is described how a feedback law can be derived in different ways. In Section 3-3-4 it is described how candidate solutions are verified using an SMT-solver. In Section 3-3-5 it is described how candidate solutions are found. Finally, in Section 3-3-6 the total CEGIS algorithm is outlined.

### 3-3-1 Standard CEGIS Form

The conditions on the certificate functions in Definition 3.2 and Equation (3-9) can be expressed as a propositional formula $\varphi$ in the form:

$$\varphi := (\forall x \in X) \quad \left( \bigwedge_{i=1}^{k} \left( \bigvee_{j=1}^{l_i} \phi_{i,j}(x) \leq 0 \right) \right), \tag{3-11}$$

where $\phi_{i,j} : \mathbb{R}^n \to \mathbb{R}$ and $X$ is some compact set. The functions $(V, \mathcal{U}, M)$ are parameterized in templates. The template of $V(x)$, $\mathcal{U}(x)$ and $M(x)$ are recalled to as $V_T(m, x)$, $\mathcal{U}_T(n, x)$ and $M_T(p, x)$, where $m$, $n$, $p$ denote the parameters of the templates. In Chapter 5 it is described which templates are used. By using these templates, the problem is reduced to finding a parameter vector $c := [m, n, p]$, such that the conditions on the certificate functions are satisfied. As $\varphi$ is dependent on the templates, which are themselves a function of $c$, $\varphi$ is also a function of $c$:

$$\varphi(c) := (\forall x \in X) \quad \left( \bigwedge_{i=1}^{k} \left( \bigvee_{j=1}^{l_i} \phi_{i,j}(x, c) \leq 0 \right) \right). \tag{3-12}$$

Hence, the synthesis problems in this chapter are of the form:

$$(\exists c \in \mathbb{R}^f) \begin{cases} \varphi_1(c) \\ \varphi_2(c) \\ \vdots \\ \varphi_f(c), \end{cases} \tag{3-13}$$

where $f$ denotes the number of conditions. We propose two different approaches for the synthesis of the control law $\mathcal{U}(x)$, which are described in the next two subsections.

### 3-3-2 Template Feedback Law Synthesis by CEGIS

In the first approach a feedback law $\mathcal{U}$ is found in the CEGIS synthesis process, simultaneously with the functions $V$ and $M$. We refer to this as template-based feedback law synthesis. The parameters of the templates $V_T(m, x)$, $\mathcal{U}_T(n, x)$ and $M_T(p, x)$ are found with CEGIS. The parameter vector is $c = [m, n, p]$. For $\texttt{CS}_1$ this results in the following problem $\texttt{P}_1$ in standard form:

$$\texttt{P}_1 := (\exists c = [m,n,p] \in R^f) \begin{cases} (\forall x \in S \backslash G) & M_T(p,x) > 0; \\ (\forall x \in S \backslash G) & c_1\|x\|^2 \leq V_T(m,x) \leq c_2\|x\|^2; \\ (\forall x \in S \backslash G) & V_T'(m,x)F(x,\mathcal{U}_T(n,x)) < -\gamma_c V_T(m,x); \\ (\forall x \in S \backslash G) & \|V_T'(m,x)\| \, \|F(x,\mathcal{U}_T(n,x)\| + \|F(x,\mathcal{U}_T(n,x))\|^2 \\ & \leq M_T(p,x) \, |V_T'(m,x)| \, F(x,\mathcal{U}_T(n,x)). \end{cases} \tag{3-14}$$

The CLBF of Equation (3-9) used for $\texttt{CS}_2$ also involves the parameter $\beta$. This parameter is also found in the synthesis process. Hence, the parameter vector is $c = [m, n, p, \beta]^T$. Note that the conditions (3-9e), (3-9f) are required to hold over the sublevel set $A := \{x \in S \mid V(x) \leq \beta\}$. As shown in [65, Appendix B] a logic formula of the form $(\forall x \in A)\ f(x) \leq 0$ is equivalent to $(\forall x \in S)\ -V(x) + \beta + \eta \leq 0 \vee f(x) \leq 0$. The arbitrary constant $\eta > 0$ is used to cast non-strict inequalities to strict inequalities. The problem $\texttt{P}_2$ in standard form to be solved is:

$$\texttt{P}_2 := (\exists c = [m,n,p,\beta] \in \mathbb{R}^f) \begin{cases} (\forall x \in S \backslash G) & M_T(p,x) > 0; \\ (\forall x \in S \backslash G) & c_1\|x\|^2 \leq V_T(m,x) \leq c_2\|x\|^2; \\ (\forall x \in \partial S) & V_T(m,x) > \beta; \\ (\forall x \in I) & V_T(m,x) \leq \beta; \\ (\forall x \in S \backslash G) & V_T'(m,x)F(x,\mathcal{U}_T(n,x)) < -\gamma_c V_T(m,x) \vee \\ & -V_T(m,x) + \beta + \eta \leq 0; \\ (\forall x \in S \backslash G) & \|V_T'(m,x)\| \, \|F(x,\mathcal{U}_T(n,x)\| + \|F(x,\mathcal{U}_T(n,x))\|^2 \\ & \leq M_T(p,x) \, |V_T'(m,x)F(x,\mathcal{U}_T(n,x))| \ \vee \\ & -V_T(m,x) + \beta + \eta \leq 0 \end{cases} \tag{3-15}$$

An overview of the synthesis process for $\texttt{P}_1$ and $\texttt{P}_2$ can be found in Figure 3-2. The set $\hat{X}_{\varphi,s}$ is elaborated on in Section 3-3-6.

**Remark 3.8.** *For some applications it is needed to constrain the input to a compact set $U$. Assume that the set $U$ is a hyper-rectangle, i.e. $U = [\underline{u}_1, \bar{u}_1] \times [\underline{u}_2, \bar{u}_2] \times \cdots \times [\underline{u}_m, \bar{u}_m]$. By adding the following condition to the problems $\texttt{P}_1$ and $\texttt{P}_2$, it is ensured that $u \in U$:*

$$(\forall x \in X_u) \ \bigwedge_i^m \left(\mathcal{U}_{T,i}(n,x) \geq \underline{u}_i \ \wedge \ \mathcal{U}_{T,i}(n,x) \leq \bar{u}_i\right), \tag{3-16}$$

*where $X_u \subset D$ is some compact set and $\mathcal{U}_{T,i}(n,x)$ denotes the $i$-th element of $\mathcal{U}_T(n,x)$. For $\texttt{CS}_1$ it is $X_u = S \backslash G$ and for $\texttt{CS}_2$, $X_u = A \backslash G$.*

### 3-3-3 Feedback Law Synthesis by Feedback Linearization

Using the technique of feedback linearization as described in Section 2-4, a stabilizing controller can be found before employing the CEGIS algorithm. This way, the number of parameters to be found in the synthesis process descreases. However, this approach is only suited for feedback linearizable systems. The following assumption is adopted to be able to design a feedback linearizing control law.

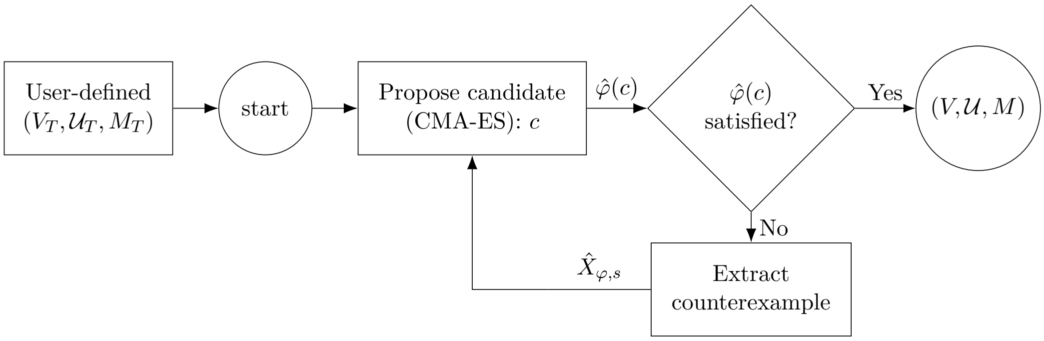



**Figure 3-2:** Algorithm to solve the problems $\mathtt{P}_1$ and $\mathtt{P}_2$. The tuple of all conditions $\varphi_i(c)$, $i \in \{1, \ldots, f\}$ is denoted as $\hat{\varphi}(c)$. The tuple sets of samples $X_{\varphi_i,s}$, $i \in \{1, \ldots, f\}$ is denoted as $\hat{X}_{\varphi,s}$

**Assumption 3.9.** *The system* (3-1) *is feedback linearizable (see Definition 2.26). Furthermore a diffeomorphism* $\Phi$ *over the domain* $D$ *is given such that the change of variables* $z = \Phi(x)$ *transforms the system into the form*

$$\dot{z} = Az + B\gamma(x)[u - \alpha(x)], \tag{3-17}$$

As described in Section 2-4, choosing the control law as

$$u = \mathcal{U}(x, z) = \gamma^{-1}(x)[\alpha(x) + v(z)], \tag{3-18}$$

results in linear closed loop dynamics in the $z$-coordinates, i.e. $\dot{z} = Az + Bv(z)$. The origin can be stabilized by choosing $v(z) = -Kz$, where $(A - BK)$ is Hurwitz. The poles of $(A - BK)$ are specified by the user. As a result, the control law in $x$-coordinates is given by

$$u = \mathcal{U}(x) = -\gamma^{-1}(x)[\alpha(x) - K\Phi(x)]. \tag{3-19}$$

Below, two methods are described how feedback linearization can be used in the synthesis process.

**Deriving $V_T$ and $\mathcal{U}$ from feedback linearization** For the linear system $\dot{z} = (A - BK)z$ a quadratic Lyapunov function $V(z) = z^T z$ always exists [27]. Therefore, the template $V_T(m, x)$ can be chosen as $V_T(m, x) = \Phi(x)^T P(m)\Phi(x)$, where $m$ are the elements of $P$. By Definition 2.25, the diffeomorphism $\Phi$ is continuously differentiable, hence $V_T(m, x)$ is continuously differentiable.

So compared to the previous approach, the template $V_T(m, x)$ follows from the feedback linearizing control law $\mathcal{U}$ (3-19). The template for $M_T(p, x)$ is user-defined. The parameters of the templates $V_T(m, x)$ and $M_T(p, x)$ are found with CEGIS. An overview of the synthesis process can be found in Figure 3-3. This approach leads to the following problem $\mathtt{P}_3$ to be

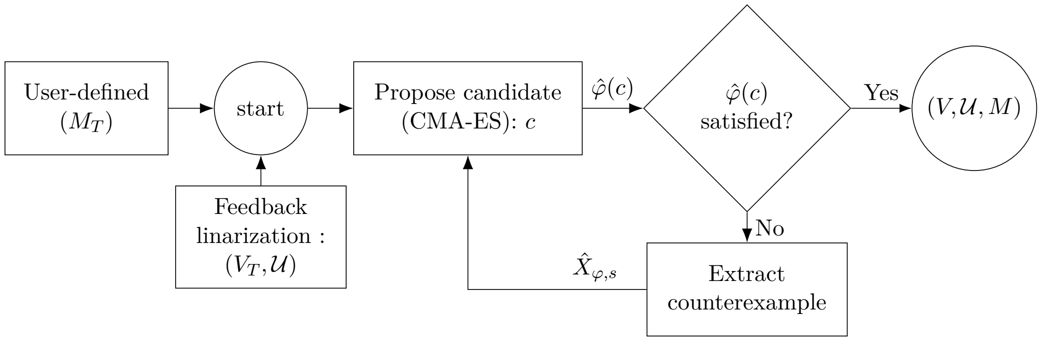



**Figure 3-3:** Algorithm to solve the problems $\mathtt{P}_3$ and $\mathtt{P}_4$

solved with CEGIS for $\mathtt{CS}_1$:

$$\mathtt{P}_3 := \quad (\exists c = [m,p] \in \mathbb{R}^f) \begin{cases} (\forall x \in S\backslash G) & M_T(p,x) > 0; \\ (\forall x \in S\backslash G) & c_1\|x\|^2 \le V_T(m,x) \le c_2\|x\|^2; \\ (\forall x \in S\backslash G) & V_T'(m,x)F(x,\mathcal{U}(x)) < -\gamma_c V_T(m,x); \\ (\forall x \in S\backslash G) & \|V_T'(m,x)\|\,\|F(x,\mathcal{U}(x)\| + \|F(x,\mathcal{U}(x))\|^2 \\ & \le M_T(p,x)\,|V_T'(m,x)F(x,\mathcal{U}(x))|\,. \end{cases} \tag{3-20}$$

For $\mathtt{CS}_2$ the problem $\mathtt{P}_4$ is defined as:

$$\mathtt{P}_4 := (\exists c = [m,p,\beta] \in \mathbb{R}^f) \begin{cases} (\forall x \in S\backslash G) & M_T(p,x) > 0; \\ (\forall x \in S\backslash G) & c_1\|x\|^2 \le V_T(m,x) \le c_2\|x\|^2; \\ (\forall x \in \partial S) & V_T(m,x) > \beta; \\ (\forall x \in I) & V_T(m,x) \le \beta; \\ (\forall x \in S\backslash G) & V_T'(m,x)F(x,\mathcal{U}(x)) < -\gamma_c V_T(m,x) \vee \\ & -V_T(m,x) + \beta + \eta \le 0; \\ (\forall x \in S\backslash G) & \|V_T'(m,x)\|\,\|F(x,\mathcal{U}(x)\| + \|F(x,\mathcal{U}(x))\|^2 \\ & \le M_T(p,x)\,|V_T'(m,x)F(x,\mathcal{U}(x))| \ \vee \\ & -V_T(m,x) + \beta + \eta \le 0 \end{cases} \tag{3-21}$$

**Remark 3.10.** *In the current approach the poles of the $z$-dynamics are defined by the user, which fully defines the controller $\mathcal{U}$* (3-18)*. One could also define a linear template $v(n,z) := K_T(n,x)z$, which results in a feedback law template $\mathcal{U}_T(n,x) := -\gamma^{-1}[-\alpha(x) - K_T(n,x)\Phi(x)]$. The search for the parameter vector $n$ is done simultaneoulsy in the CEGIS problems $\mathtt{P}_3$* (3-20)*and $\mathtt{P}_4$* (3-21)*. This way, it is also possible to bound the input as described in Remark 3.8.*

**Deriving $V$ and $\mathcal{U}$ from feedback linearization** The method described above derives a feedback law $\mathcal{U}$ an a template $V_T$ from feedback linearization. In the remaining of this subsection it is described how both $\mathcal{U}$ and $V$ can be derived from feedback linearization. Consider the Lyapnov Equation

$$P(A - BK) + (A - BK)^T P = -Q. \tag{3-22}$$

Because $(A - BK)$ is Hurwitz, for any $Q = Q^T > 0$ there exists $P = P^T > 0$ such that the Lyapunov equation is satisfied. By the following lemma $V$ can be found by solving an equation similar to the Lyapunov equation.

**Lemma 3.11.** *Given a convergence rate $\gamma_c$. Let $P = P^T > 0$ solve the equation*

$$(A - BK)^T P + P(A - BK) + \gamma_c P < 0, \tag{3-23}$$

*then $\dot{V}(z) \leq -\gamma_c V(z)$.*

*Proof.* This proof follows the line of reasoning of [4]. Because $(A - BK)$ is Hurwitz, for any $Q = Q^T > 0$ there exists $P = P^T > 0$ such that the Lyapunov equation

$$(A - BK)^T P + P(A - BK) = -Q \tag{3-24}$$

is satisfied. Let $\gamma := \frac{\lambda_{\min}(Q)}{\lambda_{\max}(P)} > 0$, where $\lambda_{\max}(\cdot)$ and $\lambda_{\min}(\cdot)$ denote the maximum and minimum eigenvalues of a symmetric matrix, respectively. Applying the Rayleigh-Ritz inequality, see e.g. [4] it is obtained that $Q \geq \gamma_c P$. Hence,

$$A^T P + PA + \gamma_c P \leq 0 \tag{3-25}$$

Defining the Lyapunov function $V(z) = z^T P z$ it follows from Equation (3-25) that along the trajectories of the system $\dot{z} = (A - BK)z$ it holds that $\dot{V}(z) \leq -\gamma_c V(z)$. □

By Lemma 2.27, the function $V$ can be defined as $V(x) = \Phi(x)^T P \Phi(x)$. From Lemma 2.27 the origin in $x$-coordinates inhertis exponential stability of the origin in $z$-coordinates. The matrix $P$ is found by solving the Linear Matrix Inequality (LMI) Equation (3-23). The equation can be solved by a convex optimization solver, such as CVX. By this approach the conditions (3-9b), (3-9c) are satisfied, and it is only needed to find a function $M$ with CEGIS. An overview of the overall synthesis process can be found in Figure 3-4. The CEGIS problem is:

$$\mathtt{P5} := (\exists c = [m] \in \mathbb{R}^f) \begin{cases} (\forall x \in S \backslash G) & M_T(p, x) > 0 \\ (\forall x \in S \backslash G) & \|V'(x)\| \, \|F(x, \mathcal{U}(x)\| + \|F(x, \mathcal{U}(x))\|^2 \\ & \leq M_T(p, x) \, |V'(x) F(x, \mathcal{U}(x))| \, . \end{cases} \tag{3-26}$$

Next, it is described how the problems $\mathtt{P}_1$, $\mathtt{P}_2$, $\mathtt{P}_3$, $\mathtt{P}_4$, $\mathtt{P}_5$ can be solved with CEGIS.

### 3-3-4 Verification

To determine whether the first-order propositional logic formula $\varphi$ of the form (3-11) is satisfied an SMT solvers is used. SMT-solvers use a combination of background theories to determine whether a first-order logic formula is satisfied over real numbers [8]. Proving that $\varphi$ is satisfied is done by proving that $\neg\varphi$ is unsatisfiable.

If $\phi_{i,j}$ is polynomial, the problem is decidable. In this case, the SMT solver Z3 [14] can be used to prove the formulae of the form (3-11). The output of the solver is satisfiable

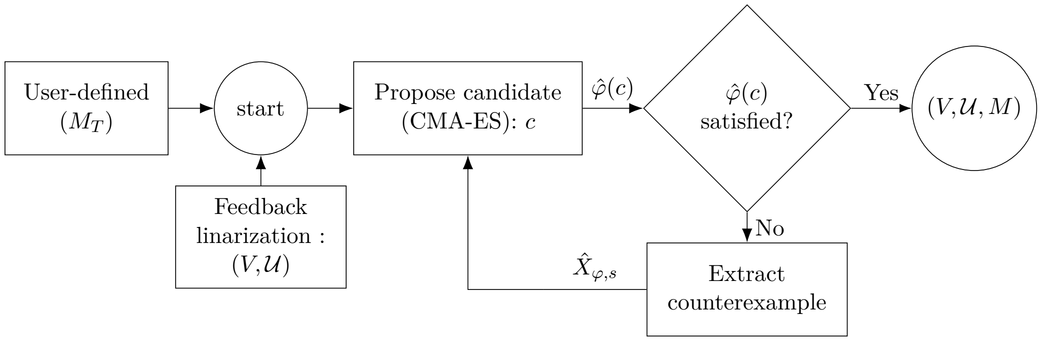



**Figure 3-4:** Algorithm to solve the problems $\mathtt{P}_5$

(sat), unsatisfiable (unsat), or unknown. When the solver returns that $\neg\varphi$ is unsat, it is proved that the original formula $\varphi$ is satisfied. The solver can return unknown when the logic formulas provided are too expensive in terms of calculations needed and/or when the procedure implemented in Z3 is incomplete, i.e., an algorithm is used that does not guarantee that a solution is found.

First-order logic formulas involving, for example, transcendentals, like the sine function, or exponentials are undecidable [51] and cannot be solved with Z3. However, a $\delta$-complete decision procedure [20] can be used in this case. This procedure determines whether a first-order logic formula is unsatisfiable (unsat) or if its $\delta$-weaking is satisfiable ($\delta$-sat). The $\delta$-weakening can be thought of as a perturbed version of the original inequality that makes the decision-making process decidable. The $\delta$-complete decision procedure is implemented SMT-solver dReal [21], which is used in this thesis. The perturbation $\delta$ is a user-defined parameter.

Note that in the conditions of the certificates in Definition 3.2, Equation (3-9) for some conditions the goal set $G$ is excluded from the domain of interest. This is because those inequalities become tight for $x \to 0$. It is inherent to the $\delta$-complete decision procedure that it is not possible to verify tight inequalities, see [65, Example 2.4.1]). However, for polynomial systems the problem is decidable and an SMT-solver, such as Z3 can be used. As Z3 does not rely on $\delta$-complete decision procedures the goal set can be taken as $G = \{0\}$ in this case. For polynomial systems and templates the conditions for which the goal set were excluded can be verified with Z3.

## 3-3-5 Candidate Proposal

Different methods exist to find candidate parameters of the certificate function. In [48] an SMT-solver is used to find candidate parameters. By assuming that the template functions in the candidate parameter vector $c$ are linear, the candidate generation problem becomes a linear arithmetic formula. These formulas are solved with SMT-solver Z3. However, in [48], it is concluded that the framework's primary shortcoming stems from the difficulty of proposing candidates using an SMT-solver. In [68], candidate generation is posed as an optimization problem. Covariance Matrix Adaption Evolution Strategy (CMA-ES) is used to solve the optimization problem. This optimization method is a derivative-free method for numerical optimization of nonlinear and non-convex continuous optimization problems [22]. Candidate generation by optimization is driven by a fitness function $\mathcal{F}(c)$, based on the finite set $X_\varphi$. The fitness function is based on a quantitative measure of how robustly the inequality in Equation (3-11) is satisfied. In this thesis, CMA-ES is used to solve an optimization for candidate generation.

The fitness function that drives candidate generation is discussed in more detail in this section. To find a candidate parameter $c$, a fitness function is defined based on [67]. The fitness function is used as the objective for an optimization that is solved with CMA-ES [22].

We wish to define a satisfaction measure that is negative if a formula of the form (3-12) is satisfied for a given $c$. Consider a logical disjunction $(\phi_1 \leq 0 \vee \phi_2 \leq 0)$. If $\phi_1 \leq 0$ or $\phi_2 \leq 0$, $\min(\phi_1, \phi_2)$ results in a negative value, because in this case $\phi_1$ or $\phi_2$ is negative. For a logical conjunction $(\phi_1 \leq 0 \wedge \phi_2 \leq 0)$, $\max(\phi_1, \phi_2)$ results in a negative value only if $\phi_1 \leq 0$ and $\phi_2 \leq 0$. Therefore, for a given a formula $\varphi(c)$ of the form (3-12), for a point $x \in X_\varphi,\ c \in \mathbb{R}^f$, a satisfaction measure $\rho_\varphi : D \to \mathbb{R}$ is defined as

$$\rho_\varphi(x, c) = \max_{i \in \{1,\dots,k\}} \left( \min_{j \in \{1,\dots,l_i\}} \phi_{ij}(x, c) \right). \tag{3-27}$$

In this satisfaction measure, if the inequality in Equation (3-11) is satisfied, $\phi_{i,j}(x, c)$ is negative. Therefore, if $\rho_\varphi$ is negative, $\varphi$ is true. Based on the measure $\rho_\varphi$, the following error metric is defined:

$$e_\varphi(x, c) := \max\left(\rho_\varphi(x, c), 0\right). \tag{3-28}$$

If $\varphi$ is true for a given point $x$ and parameter $c$, then $e_\varphi(x, c)$ equals zero; otherwise, it is positive. Then, for a given $c$, the fitness $\mathcal{F}_\varphi$ is defined over the set $X_\varphi$ as

$$\mathcal{F}_\varphi(X_{s,\varphi}, c) := \frac{1}{1 + \left\|\left[e_\varphi\left(x_1, c\right), \dots, e_\varphi\left(x_d, c\right)\right]\right\|}, \tag{3-29}$$

Note that a valid parameter vector $c$ for all $x \in X$ results in a value of $p$ for the fitness function. The CMA-ES optimization algorithm can be used to solve the optimization problem

$$\text{argmax}_c \frac{1}{p} \sum_{i=1}^{p} \mathcal{F}_{\varphi_i}. \tag{3-30}$$

to find a candidate $c$.

**Remark 3.12.** *An effect of the δ-satisfiability approach is that cases of 'δ-sat' or 'unsat' are not mutually exclusive [67]. To prevent this overlap, candidate solutions are generated that are robust with respect to the δ-perturbation. Instead of finding a solution that satisfies* (3-11)*, a solution is found that satisfies the following formula:*

$$\varphi' := (\forall x \in X) \quad \left( \bigwedge_{i=1}^{k} \left( \bigvee_{j=1}^{l_i} \phi_{i,j}(x) + \varepsilon \leq 0 \right) \right), \tag{3-31}$$

*where $\varepsilon > \delta$. For a more thorough description of this approach see [68].*

### 3-3-6 CEGIS Algorithm Outline

In this subsection it is described how the parameter vector $c$ of $\mathtt{P}_1$-$\mathtt{P}_5$ can be found. This is done in the CEGIS framework. The set $X_{\varphi_i}$ denotes the domain of interest $X$ for the $i$-th condition $\varphi_i$, $i \in \{1, \ldots, d\}$. The algorithm consists of the following steps:

1. For each condition $\varphi_i$, a finite set $X_{\varphi_i,s} \subset X_{\varphi_i}$ is initialized with $n_{\text{initial}}$ uniform randomly selected states.
2. A candidate parameter $c$ is found, solving the optimization (3-30) with CMA-ES.
3. All the conditions $\varphi_i(c)$ are verified with dReal.
4. If some $\varphi_i(c)$ is not satisfied, a counterexample $x_c$ is extracted that disproves $\varphi_i(c)$.
5. For all $\varphi_i$ it is checked whether $x_c \in X_{\varphi_i}$, if true, $x_c$ is added to the set $X_{\varphi_i,s}$.
6. Steps 2-5 are repeated until all $\varphi_i$ are true, or when a maximum number of iterations is met.

## 3-4 Summary

In this chapter, it was described how underlying functions of event-triggered controllers could be synthesized. The event-triggered controller is based on a control certificate function and a feedback law. Two different certificate functions were proposed, based on the work of [43] that can be used for two controller specifications. The second specification considers forward invariance of some safe set as opposed to the first specification. Two approaches were presented to find a feedback law in the synthesis process. The first approach uses CEGIS to find the certificate function and a feedback law simultaneously. The second approach first finds a feedback law using feedback linearization, whereafter a certificate function is found using CEGIS. It was also shown that for polynomial system equations, feedback law and certificate function asymptotic stability could be verified.

Chapter 4

# Periodic Event-Triggered Control

The event-triggered control implementation described in the previous chapter requires continuous evaluation of the triggering condition. Consequently, this raises the need for extra hardware to implement this type of implementation on digital platforms, which might be impractical. As described in Chapter 1, Periodic Event-Triggered Control (PETC) implementations only require checking the triggering condition periodically [23]. The periodic nature of the triggering condition leads to several benefits. First, PETC is better suited for practical implementation as it can be implemented on digital embedded systems. Secondly, scheduling simplifies as the number of possible triggering instants reduces to a finite set. Finally, using PETC, a minimum inter-event time of at least the event-triggering condition's sampling interval is ensured.

In this chapter we show that the PETC triggering condition described in Section 2-3 in combination with the certificate functions described in Chapter 3 can also satisfy control specification $\texttt{CS}_1$ or $\texttt{CS}_2$. We show that Subproblem 1 can also be solved by using a periodic event-triggered controller. In Section 2-3 it was described that the time between two samples of the triggering condition should be chosen lower than the maximum admissible sampling period (MASP) to ensure that the system is sampled before an event is triggered. This resulted in the following Subproblem:

**Subproblem 2**

Find an approximation of the maximum admissible sampling period, such that a periodic event-triggered controller provides the same control specification as the continuous event-triggered controller.

An expression for the MASP for the specific PETC implementation was provided in Section 2-3. This expressions involves Lipschitz functions, for which a closed-form expression is unknown. In this chapter, we describe how an approximation of the MASP can be found using optimization. This involves the calculation of Lipschitz constants from the Lipschitz functions of the closed-loop system and the derivative of the certificate function.

This chapter is organized as follows. First, in Section 4-1 we derive a formalized version of Subproblem 2. Second, in Section 4-2 it is described how the certificate function defined in the previous chapter are used in combination with a periodic event-triggered controller to enforce control specification $\texttt{CS}_1$ or $\texttt{CS}_2$. Then, in Section 4-3 it is described how an approximation of the MASP can be found. In Section 4-4 it is described how the approximations can be improved an how the MASP can be influenced in the synthesis process. Lastly, in Section 4-5 this chapter is summarized.

## 4-1 Problem Definition

In this chapter the same system (3-1) is considered as in Chapter 3. The periodic event-triggered controller considered in this chapter is of the form

$$\begin{aligned} u(t) = u_n &:= \mathcal{U}\left(x\left(k_n h\right)\right) \quad \forall t \in \left[k_n h, k_{n+1} h\right); \quad k_0 = 0, \\ k_{n+1} &= \begin{cases} \min\left\{k > k_n \mid \neg\mathcal{P}(x(kh), u_n)\right\}, & x\left(k_n h\right) \neq 0, \\ \infty, & x\left(k_n h\right) = 0. \end{cases} \end{aligned} \tag{4-1}$$

where $h > 0$ is the sampling interval and $\mathcal{P}$ is some boolean triggering function. For PETC Zeno behavior is automatically excluded, because $h > 0$. For PETC the problem formulation is similar to the problems of the previous chapter, except that the specification to exclude Zeno behavior is omitted. The formalized version of Subproblem 1 for PETC and $\texttt{CS}_1$ is formulated as follows.

**Subproblem 1, $\texttt{CS}_2$, PETC (formalized)**

Given the compact sets $D$, the system in Equation (3-1) and a desired exponential convergence rate $\gamma_c$, synthesize a periodic event-triggered controller of the form (4-1) consisting of a control law $u = \mathcal{U}(x)$ and triggering condition $\mathcal{P}(x, u)$ such, that if $x(0) \in R$, all solutions $x(t) \in R$ to the closed-loop system given by (3-1) and (4-1) satisfy specification $\texttt{CS}_1$.

The formalized version of Subproblem 1 for PETC and $\texttt{CS}_2$ is formulated as follows.

**Subproblem 1, $\texttt{CS}_2$, PETC (formalized)**

Given the compact sets $(S, I)$, the system in Equation (3-1) and a desired exponential convergence rate $\gamma_c$, synthesize a periodic event-triggered controller of the form (4-1) consisting of a control law $u = \mathcal{U}(x)$ and triggering condition $\mathcal{P}(x, u)$ such, that if $x(0) \in I$, all solutions $x(t)$ to the closed-loop system given by (3-1) and (4-1) satisfy specification $\texttt{CS}_2$.

The MASP determines the sampling interval $h$. So for the periodic event-triggered controller an estimation of the MASP is needed.

**Subproblem 2 (formalized)**

For the set $R$, find an estimation $\tau^*$ of the MASP $\tau_{\text{MASP}}$.

## 4-2 Control Strategy

We use the PETC triggering condition $\mathcal{P}$ as defined in Equation (2-31). For a given constant $\sigma \in (0,1)$, $\tilde{\sigma} \in (\sigma, 1)$ and $K > 1$, $\mathcal{P}$ is defined as:

$$\mathcal{P}(x,u) := V'(x)F(x,u) < -\tilde{\sigma}\gamma_c V(x) \wedge \frac{\|V'(x)\| \, \|F(x,u)\| + \|F(x,u)\|^2}{M(x)|V'(x)F(x,u)|} \leq K. \tag{4-2}$$

The parameter $\sigma$ regulates the convergence rate. The parameters $\tilde{\sigma}$ and $K$ influence the MASP. The certificate function $V$, feedback law $\mathcal{U}$ and function $M$ are assumed to be known. They can be found by approaches described in the previous chapter. For a given $x_* \in S$ the sublevel set $B(x_*)$ is defined:

$$B(x_*) := \{x \in S \mid V(x) \leq V(x_*)\}. \tag{4-3}$$

Given a set $S_0 \subset S$, the set $B(S_0)$ is the union of the sets $B(x_*)$ for all $x_* \in S_0$:

$$B(S_0) := \bigcup_{x_* \in S_0} B(x_*). \tag{4-4}$$

For sake of convenience the Lipschitz functions defined in Section 2-3 are recalled. The Lipschitz function of the closed-loop system given by (3-1) and (4-1) is defined as

$$L_F(x_*) := \sup_{\substack{x_1, x_2 \in B(x_*) \\ x_1 \neq x_2}} \frac{\|F(x_1, \mathcal{U}(x_*)) - F(x_2, \mathcal{U}(x_*))\|}{\|x_2 - x_1\|} \quad L_F(0) := 0. \tag{4-5}$$

The Lipschitz function of the Jacobian of $V$ is

$$L_{V'}(x_*) := \sup_{\substack{x_1, x_2 \in B(x_*) \\ x_1 \neq x_2}} \frac{\|V'(x_1) - V'(x_2)\|}{\|x_2 - x_1\|}, \quad L_{V'}(0) := 0. \tag{4-6}$$

Using the CLF and PETC triggering condition the result for Subproblem 1 for $\texttt{CS}_1$ is formulated:

**Theorem 4.1.** *Let $R$ be a sublevel set contained in $D$, i.e. $R := \{x \in D \mid V(x) \leq \beta\}$, with $\beta$ such that $R \subseteq int(D)$. Given a system* (3-1)*, satisfying Assumption 3.1, an event-triggered controller of the form* (4-1)*, a Control Lyapunov Function (CLF) $V$, feedback law $\mathcal{U}$ and function $M$ as defined in Definition 3.2 w.r.t compact sets $S$, a desired convergence rate $\gamma_c > 0$ and constant parameters $\sigma > 0$, $\tilde{\sigma} \in (\sigma, 1)$, $K > 1$, then if the sampling interval $h$ is chosen as $h \in (0, \tau_{MASP})$, with $\tau_{MASP}$ defined in Equation* (2-34) *and $x(0) \in R$, solutions $x(t) \in R$ to the closed-loop system given by* (3-1) *and* (3-2) *satisfy specification $\texttt{CS}_1$.*

*Proof.* Following the same line of reasoning as in the proof of Lemma 3.3 it is proved that Assumptions 2.14, 2.16 and 2.18 are satisfied on $R$. Therefore, by Theorem 2.24 it follows that the MASP on $R$ is

$$\tau_{\text{MASP}} = \inf_{x_* \in B(R)} \tau^0(x_*) = \inf_{x_* \in B(R)} \min\left\{\frac{(\tilde{\sigma} - \sigma)^2}{K^2 \mu(x_*)^2 M(x_*)^2 \tilde{\sigma}^2}, \frac{1}{1 + 2L_F(x_*)}\right\}, \tag{4-7}$$

with $\mu$ as in Equation (2-30), and the Lipschitz function $L_F$ as in Equation (4-5) and $L'_V$ as in Equation (4-6). By definition, $L_F$ and $L_{V'}$ are positive. Furthermore, by Assumptions 2.14 and 2.16, $L_F > 0$ and $L_{V'} > 0$ are locally bounded for all $x_* \in R$, and thus $\mu$ is a positive definite locally bounded function. By Definition 3.2 the function $M > 0$ is locally bounded. Therefore, the function $\tau^0(x_*)$ is uniformly positive on $R$. As $h$ is chosen as $h \in (0, \tau_{\text{MASP}})$, the triggering conditions is checked before an event occurs and by Theorem 2.24, $V'(x(t))F(x(t),\mathcal{U}(x(t))) < -\gamma_c V(x(t))$ holds for all $t > 0$, $x(t) \in R$. Analogous to the proof of Theorem 3.4, it follows that $\forall x(0) \in R$, $x(t) \in D$ and that the origin is exponentially stable and $\texttt{CS}_1$ is satisfied.

□

**Theorem 4.2.** *Given a system* (3-1)*, satisfying Assumption 3.1, an event-triggered controller of the form* (4-1)*, a CLF $V$, feedback law $\mathcal{U}$ and function $M$ as defined in Definition 3.2 w.r.t compact sets $S, I$, a desired convergence rate $\gamma_c > 0$ and constant parameters $\sigma > 0$, $\tilde{\sigma} \in (\sigma, 1)$, $K > 1$, then if the sampling interval $h$ is chosen as $h \in (0, \tau_{MASP})$, with $\tau_{MASP}$ defined in Equation* (2-34) *and $x(0) \in I$, solutions $x(t) \in A$ to the closed-loop system given by* (3-1) *and* (3-2) *satisfy specification $\texttt{CS}_2$.*

*Proof.* Following the same line of reasoning as in the proof of Lemma 3.6 it is proved that Assumptions 2.14, 2.16 and 2.18 are satisfied on $A$. Analogous to the proof of Theorem 4.1 it follows that $V'(x(t))F(x(t),\mathcal{U}(x(t))) < -\gamma_c V(x(t))$ holds for all $t > 0, x(t) \in A$. Analogous to the proof of Theorem 3.4, it follows that $\forall x(0) \in I$, $x(t) \in A \subset S$ and that the origin is exponentially stable and $\texttt{CS}_2$ is satisfied. □

By Theorem 4.1 the sampling interval $h$ is chosen as $h \in (0, \tau_{\text{MASP}})$. In the next section, it is described how an approximation of $\tau_{\text{MASP}}$ can be found.

## 4-3 Approximation of the MASP

The formula for $\tau_{\text{MASP}}$ in Equation (4-7) is the solution to the following optimization problem

$$\begin{aligned} \tau_{\text{MASP}} = \underset{x_*}{\text{minimize}} \quad & \min\left\{\frac{(\tilde{\sigma}-\sigma)^2}{K^2\mu(x_*)^2 M\left(x_*\right)^2\tilde{\sigma}^2}, \frac{1}{1+2L_F\left(x_*\right)}\right\}. \\ \text{subject to} \quad & x_* \in B(S_0). \end{aligned} \tag{4-8}$$

The goal is to find an intersampling time $\tau^*$ that is an approximation of $\tau_{\text{MASP}}$. The optimization problem above involves the Lipschitz functions $L_F$ and $L_{V'}$. Closed-form expressions for these functions are unknown. Therefore, in this section it is described how for a given $x_*$, an approximation of the values of $L_F(x_*)$ and $L_{V'}(x_*)$ can be found.

Note that an increase in $L_F$ and $L_{V'}$ cause a decrease of $\tau_{\text{MASP}}$. Hence, to make sure that $\tau^* \leq \tau_{\text{MASP}}$, the goal is to find overapproximations of $L_F$ and $L_{V'}$. These overapproximations are denoted as $L_{F,a}$ and $L_{V',a}$, that is $L_F(x_*) < L_{F,a}(x_*)$ and $L_{V'}(x_*) < L_{V',a}(x_*)$. For a given $x_*$ an overapproximation of the Lipschitz constant can be found by the following lemma.

**Lemma 4.3.** *Consider a function $\phi$ defined on $D \subset \mathbb{R}^n$, $\phi : D \to \mathbb{R}^n$. Let $\phi_i$ denote the i-th element of $\phi$. Assume that $\phi_i$ is differentiable with respect to $x$. Then a Lipschitz constant $L$ that satisfies*

$$\sup_{x_1,x_2 \in D, x_1 \neq x_2} \frac{\|\phi(x_1) - \phi(x_2)\|}{\|x_1 - x_2\|} \leq L, \tag{4-9}$$

*is*

$$L = \sqrt{\sum_i^n \left( \sup_{x \in D} \left\| \frac{\partial \phi_i}{\partial x} \right\| \right)^2}, \tag{4-10}$$

*Proof.* Consider the functions $\phi_i : D \to \mathbb{R}$, $i \in \{1, \ldots, n\}$. As $\phi_i$ is differentiable, by the mean value theorem, see e.g. [28], given any $x_1, x_2 \in D$, there exists a point $z \in [x_1, x_2]$, such that

$$\frac{\phi_i(x_1) - \phi_i(x_2)}{x_1 - x_2} = \frac{\partial \phi}{\partial x}(z). \tag{4-11}$$

On any compact set $D$, $\frac{\partial \phi_i}{\partial x}$ is bounded. Thus there exists a constant $L_i > 0$ such that $\frac{\partial \phi_i}{\partial x} < L_i \quad \forall x \in D$. Therefore, it holds that

$$|\phi_i(x_1) - \phi_i(x_2)| \leq L_i \|x_1 - x_2\| \quad i \in \{1, \ldots, n\}, \tag{4-12}$$

which by definition of the two norm implies

$$\|\phi(x_1) - \phi(x_2)\|^2 = \sum_{i=1}^n |\phi_i(x_1) - \phi_i(x_2)|^2 \leq \sum_{i=1}^n L_i^2 \|x_1 - x_2\|^2, \tag{4-13}$$

which shows that $\|\phi(x_1) - \phi(x_2)\| \leq L\|x_1 - x_2\|$ with $L = \sqrt{\sum_i^n L_i^2}$ [17].

□

By Assumption 3.1 $f'(x,u) = [\partial f(x,u)/\partial x]$ exists. Furhtermore, the certificate functions are $C^2$ smooth. Therefore, Lemma 4.3 can be applied. The approximations are:

$$L_{F,a}(x_*) = \sqrt{\sum_i^n \left( \sup_{x \in B(x_*)} \left\| \frac{\partial f(x, \mathcal{U}(x_*))}{\partial x} \right\| \right)^2}, \tag{4-14}$$

and

$$L_{V',a}(x_*) = \sqrt{\sum_i^n \left( \sup_{x \in B(x_*)} \left\| \frac{\partial^2 V(x)}{\partial x^2} \right\| \right)^2}. \tag{4-15}$$

Recall that the optimization variable $x_*$ in Equation (4-8) is constrained to the set $B(S_0)$ (2-20), with $S_0$ a compact set of initial states. By Theorem 4.1 all solutions to the closed-loop system stay in $R$. As $R$ is level sets of $V$, $B(R) = R$. Thus, for $\texttt{CS}_1$, $B(S_0) = R$. For $\texttt{CS}_2$ the optimization variable is constrained to $B(I)$, for which a closed form expression is difficult to find. As it is known that $I \subset A$ and $A$ is forwariant invariant from Theorem 4.2, the set $I$ can be replaced by the level set $A$. As a result for $\texttt{CS}_2$ the optimization is constrained to

$B(A) = A$. The cost function of the optimization (4-8) is also scaled with a parameter $N > 0$, to avoid numerical issues. Hence, we solve the following optimization:

$$\begin{aligned} \tau_N^* := \underset{x_*}{\text{minimize}} \quad & \min\left\{\frac{N(\tilde{\sigma}-\sigma)^2}{K^2\mu(L_{F,a}(x_*), L_{V',a}(x_*))^2 M\left(x_*\right)^2 \tilde{\sigma}^2}, \frac{N}{1+2L_{F,a}\left(x_*\right)}\right\}, \\ \text{subject to} \quad & x_* \in B(S_0). \end{aligned} \tag{4-16}$$

This is a non-convex nonlinear optimization problem, which can be solved with available solvers. As the optimization is non-convex, an approximation can be found that is higher than the true MASP. This is undesired as this can result in the sampling time $h$, being chosen to high. Subject of future research is to verify whether the approximated MASP is lower than the true MASP.

## 4-4 Improvement of Approximations of the MASP

In the previous section, it was described how an approximation of the MASP could be found. For some applications, it might be practical to be able to influence the MASP in the synthesis procedure, for example due to sample limit on hardware. Furthermore, when the MASP is high, it is advantageous, because the triggering condition should be checked less frequently, resulting in less communication and computation. In this section, two methods are described how the MASP can be influenced in the synthesis process.

### 4-4-1 Fitting Polynomial $M$-function

Note that an increase in the function $M$ cause a decrease of $\tau^*$. Therefore one would want the inequality to be tight:

$$(\forall x \in B(S_0)) \;\; \|V'(x)\| \, \|F(x,\mathcal{U}(x)\| + \|F(x,\mathcal{U}(x))\|^2 \leq M(x) \left|V'(x)F(x,\mathcal{U}(x))\right|. \tag{4-17}$$

In the synthesis process of Chapter 3, no conditions were put on this tightness. Therefore in this section it is described how a polynomial function $M$ can be fitted, such that the inequality becomes tight. A polynomial template function $M_T(p,x)$ for $M$ is defined as

$$M_T(p,x) := \sum_{i}^{n_{\text{mon}}} p_i s_i, \tag{4-18}$$

where $p$ is a parameter vector, and $p_i$ is the $i$-th element of $p$. Furthermore, $s_i$ are monomials and $n_{\text{monomial}}$ is the number of monomials. The target data the function $M$ should be fitted to is defined as:

$$M_{\text{target}}(x) := \frac{\|V'(x)\| \, \|f(x,\mathcal{U}(x)\| + \|f(x,\mathcal{U}(x))\|^2}{|V'(x)F(x,\mathcal{U}(x))|} \quad \forall x \in \{x \in B(S_0) \mid x \neq 0\}. \tag{4-19}$$

A finite set of $n_s$ states $X_s := \{x_1, \ldots, x_{n_s}\}$ can be sampled to find values of $M_{\text{target}}$. The loss function must take into account that the fitted polynomial must be as close as possible to the target data, but the inequality Equation (4-17) should always be satisfied. Hence, it is required that the residuals $\epsilon(p, x_s) := M_T(p, x_s) - M_{\text{target}}(x_s) \geq 0 \ \forall x_s \in X_s$. The loss function for a set $X_s$ is given as

$$\mathcal{L}(p, X_s) = \sum_{i}^{n_s} l(p, x_i)^2, \tag{4-20}$$

where the loss the function $l$ is defined as

$$l(p, x_s) := \begin{cases} \epsilon(p, x_s), & \text{if } \epsilon(p, x_s) \geq 0; \\ P_{\text{penalty}}\epsilon(p, x_s), & \text{if } \epsilon(p, x_s) < 0. \end{cases} \tag{4-21}$$

where $P_{\text{penalty}} > 0$ is some arbitrarily large constant to penalize negative residuals. The following problem is solved to find the optimal parameters $p_{\text{opt}}$:

$$p_{opt} := \text{argmin}_p \ \mathcal{L}(p, x_s). \tag{4-22}$$

Note that this is a nonlinear least-squares problem, which can be solved with solvers implemented in Scipy [37]. Even if $\epsilon(p_{\text{opt}}, x_s) \geq 0$ for all $x_s$ there is no formal guarantee that $M(x) = M_T(p_{\text{opt}}, x)$ satisfies Equation (4-17) as the function is only fitted to a finite set of samples. Therefore we verify it with an Satisfiability Modulo Theory (SMT) solver. If Equation (4-17) does not hold, the constant $P_{\text{penalty}}$ is increased to increase penalization of negative residuals.

### 4-4-2 Bounding Template Parameters

Consider the expression for the MASP $\tau_{\text{MASP}}$ in Equation (4-16), which is dependent on the Lipschitz functions $L_F$ and $L_{V'}$. The function $L_F$ is dependent on $\mathcal{U}$ and $L_{V'}$ on $V$. For higher values of $L_f$ and $L_V$, the MASP, is lower. Thus, by bounding the absolute value of the parameters of the templates, the MASP can be raised. This is accomplished by adding an additional term to the fitness function (3-29). Assume that the parameter vector $c$ is bounded, that is $c \in C \subset \mathbb{R}^f$. The domain $C$ is a hyper-rectangle, i.e. $C = [\underline{c}_1, \bar{c}_1] \times [\underline{c}_2, \bar{c}_2] \times \cdots \times [\underline{c}_f, \bar{c}_f]$. Let $c_i$ denote the $i$-th element of parameter vector $c$, a logic formula representing the bounds on $c$ is given as

$$\psi(c) := \bigwedge_{i}^{f} (-c_i + \underline{c}_i \leq 0 \ \wedge \ c_i - \bar{c}_i \leq 0) \tag{4-23}$$

Following the same line of reasoning to define the satisfaction measure in Section 3-3-5, the following satisfaction measure is defined:

$$\rho_c(c) := \max_{i \in \{1,\ldots,f\}} (\max(-c_i + \underline{c}_i, c_i - \bar{c}_i)) \tag{4-24}$$

Note that if for a given parameter vector $c$, $\psi(c)$ is true, then $\rho_c(c)$ is negative. Based on the measure $\rho_c$, the following error measure is constructed:

$$e_c(c) := \max(\rho_c(c), 0), \tag{4-25}$$

which for a given vector $c$ is equal to zero if $\psi(c)$ is true and positive if not. The parameter fitness is defined as

$$\mathcal{F}_c(c) := \frac{1}{1 + e_c(c)} \tag{4-26}$$

By definition, $\mathcal{F}_c \in [0, 1]$, and if for a given $c$, $\psi(c)$ is true $\mathcal{F}_c(c)$ equals 1. The overall fitness function $\mathcal{F}$ is composed of the sample-based fitness Equation (3-29) and the parameter fitness:

$$\mathcal{F}(x, c) := \frac{1}{2}(\mathcal{F}_\varphi(X_{s,\varphi}, c) + \mathcal{F}_c(c)), \tag{4-27}$$

## 4-5 Summary

In this chapter, it was described how an approximation of the MASP can be found. An approximation can be found by solving an optimization problem. This optimization problem involves the computation of Lipschitz constants of the system and the derivative of the certificate function. It was described how these Lipschitz constants could be found. It was also described how a polynomial function $M$ can be fitted, such that the inequalities (4-17) is tight. This results in a lower approximation of the MASP.

# Chapter 5

# Results

In this section, the effectiveness of the described methods will be demonstrated on benchmark systems. Event-triggered controllers will be synthesized as described in Chapter 3 for several systems. An approximation of the maximum admissible sampling period (MASP) is found, using the methods described in Chapter 4.

## 5-1 CEGIS Synthesis

In this section, event-triggered controllers are synthesized for the synthesis problems in $\mathtt{P}_1$ and $\mathtt{P}_2$ of Section 3-3-2 and $\mathtt{P}_3$ and $\mathtt{P}_4$ of Section 3-3-3. For the problems in $\mathtt{P}_1$ and $\mathtt{P}_2$ a feedback law is found simultaneously with Counterexample-Guided Inductive Synthesis (CEGIS) and for the problems $\mathtt{P}_3$ and $\mathtt{P}_4$ a feedback law is found with feedback linearization. The systems in Table 5-1 are considered for synthesis of event-triggered controllers. The systems (1) - (4) are two-dimensional and the system (8) is three-dimensional. Diffeomorphisms are found, using the approach as described in Appendix A. First it is verified whether the system is feedback linearizable. Then a 'virtual output' is selected, by looking at the system dynamics. Using this approach a diffeomorphism is found for the systems (1) - (4). The system (8) is feedback linearizable, but a suitable 'virtual output' was not found to transform the system is feedback linearized form (2-36).

The synthesis procedure is implemented in Python, which runs on an Intel Core i7-8750H 2.20Ghz using 6 CPU cores. The (private) code is published on the SYNC-LAB repository `https://gitlab.tudelft.nl/sync-lab/cadusy/formal-etc`. For all experiments, 5000 initial points are used for the CMA optimization, i.e. $n_{\text{initial}} = 5000$. The robustness parameter $\varepsilon$ (see Remark 3.12) is taken as 0.05. The perturbation parameter of dReal $\delta$ is taken as 0.01. The synthesis algorithm is run five times for the systems in Table 5-1. First, simulations are shown for an inverted pendulum on a cart with the synthesized controllers. Then the synthesis procedure is benchmarked on multiple systems, including the inverted pendulum on a cart. In the next subsection, the choice of templates is motivated.

### 5-1-1 Templates

Consider synthesis problems $\mathtt{P}_1$ (3-14) and $\mathtt{P}_2$ (3-20). For these problems it is needed to choose the templates $V_T$,$\mathcal{U}_T$ and $M_T$. Note that by definition of the Control Lyapunov Function (CLF) and control Lyapunov barrier functions (CLBF), the resulting certificate functions needs to be $C^2$-smooth. Therefore the template $V_T$ is chosen to be polynomial with degree 2, i.e.

$$V_T(m,x) = \sum_{i}^{n_{\text{mon}}} m_i s_i(x), \tag{5-1}$$

where $m$ are the parameters, and $m_i$ is the $i$-th element of $m$. Furthermore, $s_i$ are monomials and $n_{\text{mon}}$ is the number of monomials. In the next section an example of a polynomial template is given. The template of $\mathcal{U}_T(x)$ is also chosen to be polynomial with degree 2. The template can also be linear (polynomial with degree 1), but a linear controller may not exist for some systems. Choosing $\mathcal{U}_T(x)$ polynomial, results in a locally bounded feedback law $\mathcal{U}(x)$. To keep the number of parameters to a minimum, the template $M_T(p,x)$ is chosen to be a constant. This is possible as by definition a locally bounded function $M > 0$ is bounded by some constant, i.e. $M(x) \leq \sup_{x_*} M(x)$ for all $x_*$. When needed, a polynomial function $M$ can be found after synthesis, using the approach described in Section 4-4.

For the problems $\mathtt{P}_3$ (3-15), $\mathtt{P}_4$ (3-21) and $\mathtt{P}_5$ (3-26), it is only needed to define the template $M_T(p,x)$ and to define the poles of the $z$-dynamics. The template $M_T(p,x)$ is also chosen constant. For demonstration purposes, the poles for two-dimensional systems are placed at -1 and -0.5 and for the three dimensional system at -1.5, -1 and -0.5.

### 5-1-2 Case Study - Inverted Pendulum on a Cart

In this case study, the non-polynomial nonlinear inverted pendulum (3) is considered. Simulations are shown for controllers that are synthesized according to $\mathtt{P}_2$ (3-15) and $\mathtt{P}_4$ (3-21). Remember that these problems correspond to synthesis of the CLBF for $\mathtt{CS}_2$ for a template-based and feedback linearizing control law, respectively. Again, the desired convergence rate is $\gamma_c = 0.1$. The simulation step size is taken as $h = 0.01$. After a simulation, it is checked that the step size is smaller than the minimal time between two triggering times. The parameter $\sigma$ is set to $\sigma = 0.9$, $\tilde{\sigma} = 0.9$ and $K = 1.1$.

The following templates are used for the template-based feedback law synthesis:

$$\begin{aligned} V_T(m,x) &= m_1 x_1^2 + m_2 x_1 x_2 + m_3 x_2^2, \\ \mathcal{U}_T(n,x) &= n_1 x_1 + n_2 x_2 + n_3 x_1^2 + n_4 x_2^2, \\ M_T(p,x) &= p_1. \end{aligned} \tag{5-2}$$

In the synthesis procedure the parameters $m_i$, $i \in \{1,2,3\}$, $n_j$, $j \in \{1,2,3,4\}$, $p_1$ and $\beta$ are found. A solution found is:

$$\begin{aligned} V(x) &= 1.94x_1^2 + 1.39x1x2 + 1.80x2^2, \\ \mathcal{U}(x) &= -0.31x_1^2 + 0.41x_2^2 - 8.47x_1 - 5.45x_2, \\ M &= 7.2, \ \beta = 0.33. \end{aligned} \tag{5-3}$$

| System | $F(x,u)$ | $(S,I,G)$ | $\Phi(x)$ |
|---|---|---|---|
| (1) | $\begin{pmatrix} x_2 \\ -x_1+u \end{pmatrix}$ | $([-1,1]^2,$ $[-0.5,0.5]^2,$ $[-0.1,0.1]^2)$ | $\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$ |
| (2) | $\begin{pmatrix} x_2-x_1^3 \\ u \end{pmatrix}$ | $([-1,1]^2,$ $[-0.5,0.5]^2,$ $[-0.1,0.1]^2)$ | $\begin{bmatrix} x_1 \\ x_2-x_1^3 \end{bmatrix}$ |
| (3) | $\begin{pmatrix} x_2 \\ 4.9\sin(x_1)-0.5x_2+0.5\cos(x_1)u \end{pmatrix}$ | $([-1,1],\times[-3,3],$ $[-0.5,0.5]^2,$ $[-0.25,0.25]^2)$ | $\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$ |
| (4) | $\begin{pmatrix} \sin(x_2) \\ -x_1^2+u \end{pmatrix}$ | $([-1,1]\times[-\pi/2,\pi/2],$ $[-0.5,0.5]^2,$ $[-0.1,0.1]^2)$ | $\begin{bmatrix} x_1 \\ \sin x_2 \end{bmatrix}$ |
| (5) | $\begin{pmatrix} x_2 \\ -2x_1+3x_2+u \end{pmatrix}$ | $([-1,1]^2,$ $[-0.5,0.5]^2,$ $[-0.1,0.1]^2)$ | $\begin{bmatrix} x_1 \\ \sin x_2 \end{bmatrix}$ |
| (6) | $\begin{pmatrix} x_2 \\ -\sin(x_1)+u \end{pmatrix}$ | $([-1,1]^2,$ $[-0.5,0.5]^2,$ $[-0.25,0.25]^2)$ | $\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$ |
| (7) | $\begin{pmatrix} x_2-1.01x_1 \\ x_3-1.01x_2 \\ -2.37x_1+3.67x_2-1.31x_3-3.33u \end{pmatrix}$ | $\Big([-3,3]^3$ $[-1,1]^3$ $[-0.2,0.2]^3\Big)$ | |
| (8) | $\begin{pmatrix} -10x_1+10x_2+u \\ 28x_1-x_2-x_1x_3 \\ x_1x_2-2.6667x_3 \end{pmatrix}$ | $\Big([-5,5]^3$ $[-0.5,0.5]^3$ $[-0.3,0.3]^3\Big)$ | - |

**Table 5-1:** (1): linear system, [47]. (2): 2nd-order polynomial system, [47], (3): 2nd-order pendulum system, [68], (4): 2nd-order non-polynomial system [47], (5): linear system [60], (6) single-link robot arm [3], [47], (7): adaptive cruise control system [43], (8): Lorentz chaotic system. The function $F(x,u)$ are the system dynamics, $S$, the safe set, $I$, initial set and $G$ goal set, $\Phi(x)$ is the diffeomorphism.

For the feedback linearizing controller it is only needed to specify the template $M_T$, which is again chosen to be a constant. The feedback law $\mathcal{U}$ follows from the feedback linearization. As described in Section 3-3-3, the template $V_T$ also follows from feedback linearization. The template in $z$-coordinates is $V_T(m,z) = z^T P(m) z$, where $P(m)$ is a symmetric matrix. In $x$-coordinates it is $V_T(m,x) = \Phi(x)^T P(m)\Phi(x)$. Hence, the templates are:

$$\begin{aligned} V_T(m,x) &= m_1 x_1^2 + m_2 x_1 x_2 + m_3 x_2^2, \\ M_T(x) &= p_1. \end{aligned} \tag{5-4}$$

In the synthesis procedure the parameters $m_i$, $i \in \{1,2,3\}$, $p_1$ and $\beta$ are found. A solution found is

$$\begin{aligned} V(x) &= 4.28x_1^2 + 3.48x_1x2 + 4.61x2^2, \\ \mathcal{U}(x) &= \frac{-0.5x_1 - 2.5x_2 - 4.9\sin(x_1)}{\cos(x_1)}, \\ M &= 5.2, \beta = 1.33. \end{aligned} \tag{5-5}$$

A simulation with continuous control and event-triggered control (ETC) with the same control law $\mathcal{U}(x)$ for a template-based controller and feedback linearized controller can be found in Section 5-1-2. For both controllers the trajectory of the system under time-triggered and event-triggered control resemble. The control input is a 'staircased' version of the continuous controller. The inter-event time under the feedback linearized controller is higher than under the template-based controller.

Consider the simulations with the template-based controller and feedback linearizing controller. It can be seen that the convergence of the feedback linearizing controller is less fast than that of the template-based controller. They are the same in the number of triggering times.

A simulation with three different synthesized controllers can be found in Section 5-1-2. Considering the three controllers of the template-based controller, the trajectories are somewhat different. However, note that the input values of controller $C_3$ are much higher than that of $C_1$ and $C_2$. The values of $C_2$ are between those of $C_1$ and $C_3$. However, note that the number of triggering instants of $C_1$ and $C_3$ is almost the same, whereas the number of triggering instants of $C_2$ is around three times higher. Controller $C_1$ and $C_2$ have approximately the same convergence, whereas $C_3$ converges faster.

For the feedback linearizing controller, it can be seen that the trajectories almost exactly resemble. Also, the control values are approximately the same. However, note that they differ in the number of triggering times. $C_2$ triggers less than $C_1$ and $C_3$. The convergence of $V$ is almost the same. Comparing both controllers most notable is that the three feedback linearizing controllers resemble in trajectory and input, whereas the template-based controllers have different behavior.

### 5-1-3 Synthesis Results

The synthesis algorithm is run 50 times for the systems in Table 5-1. The templates are chosen as described in the previous subsection. The desired convergence rate is taken as $\gamma_c = 0.1$.

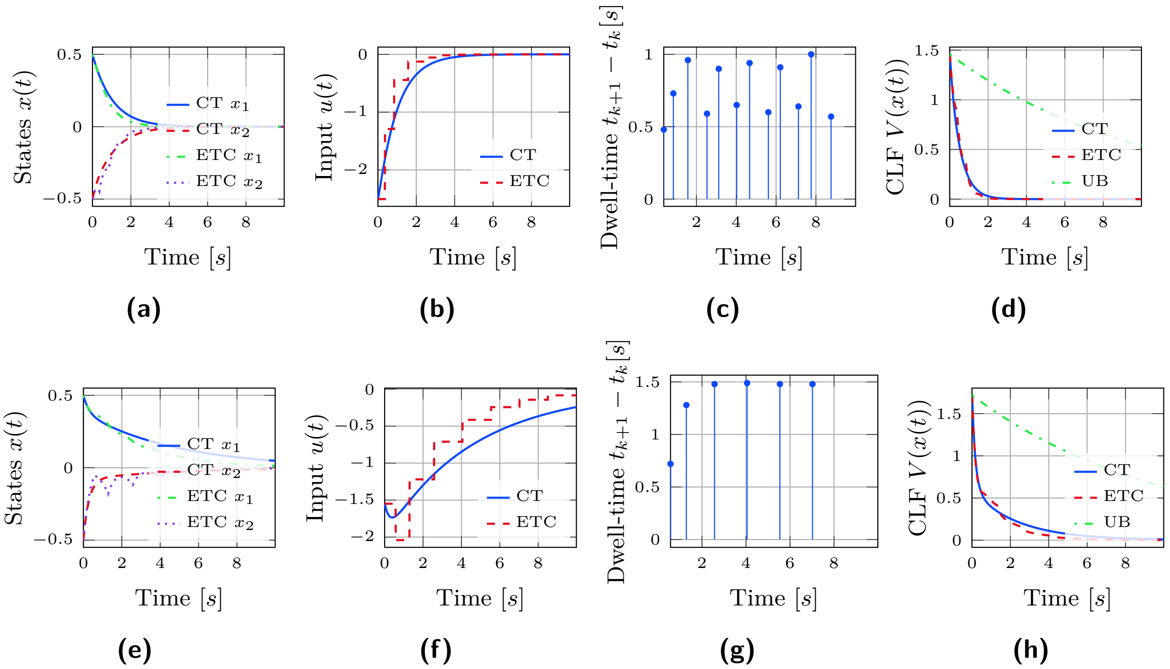



**Figure 5-1:** Continuous control (CT) and event-triggered control (ETC) for pendulum system, UB denotes the upper bound on the CLBF. Upper row: template-based controller, lower row: feedback linearized controller.

.

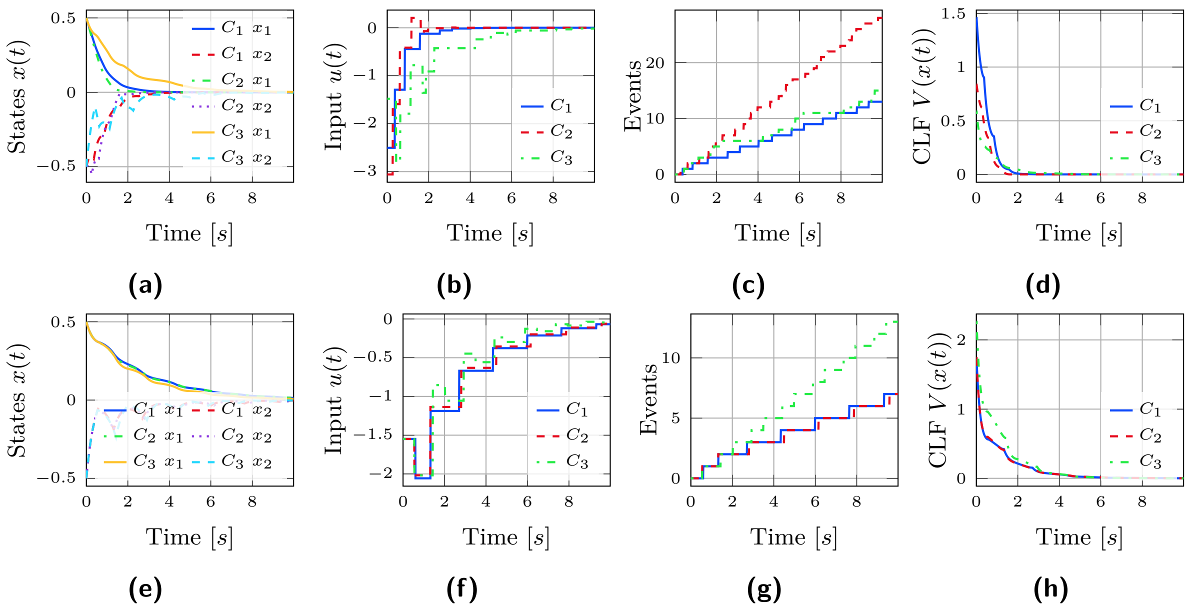
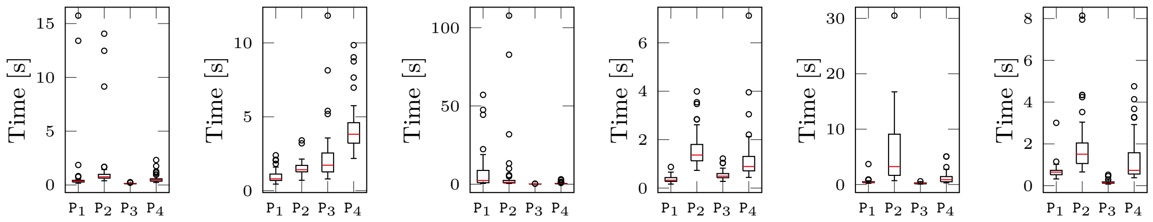


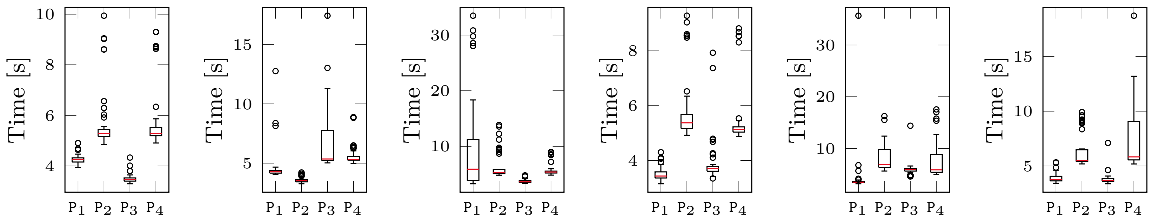

**Figure 5-2:** Pendulum system controlled by three different controllers ($C_1$, $C_2$ and $C_3$)

.




**(a)** System (1) **(b)** System (2) **(c)** System (3) **(d)** System (4) **(e)** System (5) **(f)** System (6)

**Figure 5-3:** Statistics over 50 synthesis runs on the total time per synthesis run to find a candidate parameter vector




**(a)** System (1) **(b)** System (2) **(c)** System (3) **(d)** System (4) **(e)** System (5) **(f)** System (6)

**Figure 5-4:** Statistics over 50 synthesis runs on the total time per synthesis run to verify candidates.

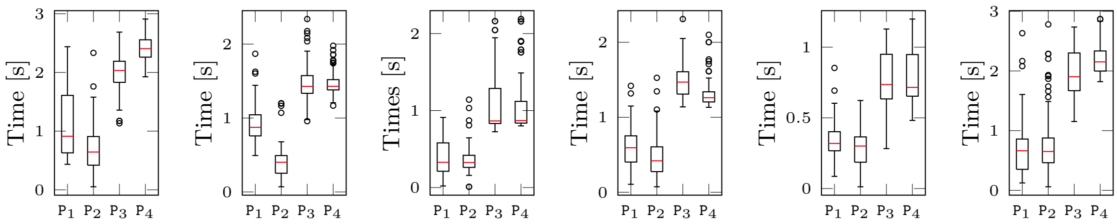
The maximum number of CEGIS iterations until a solution is found is below 4 in all synthesis runs. For most of the synthesis runs a solution is found in 1 iteration. In Figure 5-3 statistics can be found on the total time per synthesis run to find a candidate parameter vector for the synthesis problems $\mathtt{P}_1$-$\mathtt{P}_4$. Most of the time a solution can be found within 10 seconds. It can be seen in the figure that on average the total time to find a candidate is higher for $\mathtt{CS}_2$ than for $\mathtt{CS}_1$. In Figure 5-4 a figure can be found with statistics on the total time per synthesis run to verify candidate parameters. The verification time for $\mathtt{CS}_1$ is on average higher than the verification time of $\mathtt{CS}_2$. The time to find candidate parameters is of the same magnitude of the time to verify candidates.

For the synthesis problem $\mathtt{P}_5$ the total time for one synthesis run to find candidates is within 0.1 second for all the systems in Table 5-1. Also the total verification time is on average lower than two seconds for all the systems.

Statistics on the average dwell-time for a simulation of 20 seconds starting from the initial condition $x(0) = [-0.5, 0.5]$ can be found in Figure 5-5. From the figure, it can be concluded that for all the systems the average dwell-time for the feedback linearized controllers $\mathtt{P}_3$ and $\mathtt{P}_4$ is higher than the for the feedback linearized controllers $\mathtt{P}_1$ and $\mathtt{P}_2$.

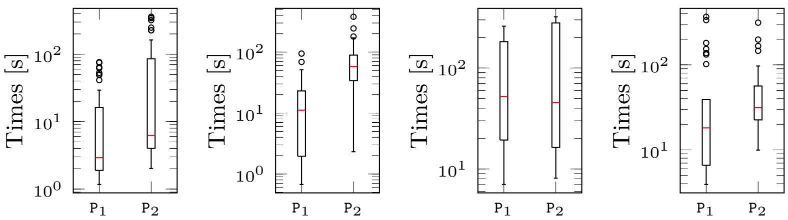
For the three-dimensional systems (7), (8) in Table 5-1, 50 controllers were synthesized following the approaches of $\mathtt{P}_1$ and $\mathtt{P}_2$ to be able to determine how the method scales to higher dimensions. In Figure 5-6 the time to find candidate parameters and to verify them is found. Whereas for two-dimensional systems a solution can be found within 10 seconds, for the system (7) a solution is found in on average 60 seconds and for the system (8) in 30 seconds. In some cases the synthesis takes 600 seconds.

The goal set of the polynomial systems in Table 5-1 can be verified with Z3 as described in




**(a)** System (1) **(b)** System (2) **(c)** System (3) **(d)** System (4) **(e)** System (5) **(f)** System (6)

**Figure 5-5:** Statistics over 50 synthesis runs on the average dwell-time for a simulation of 20 seconds starting at $x(0) = [-0.5, 0.5]$ for synthesis problems $\mathtt{P}_1$-$\mathtt{P}_4$




**(a)** System (7), candidate time **(b)** System (8), candidate time **(c)** System (7), verification time **(d)** System (8), verification time

**Figure 5-6:** Statistics on synthesis of event-triggered controller for three dimensional systems

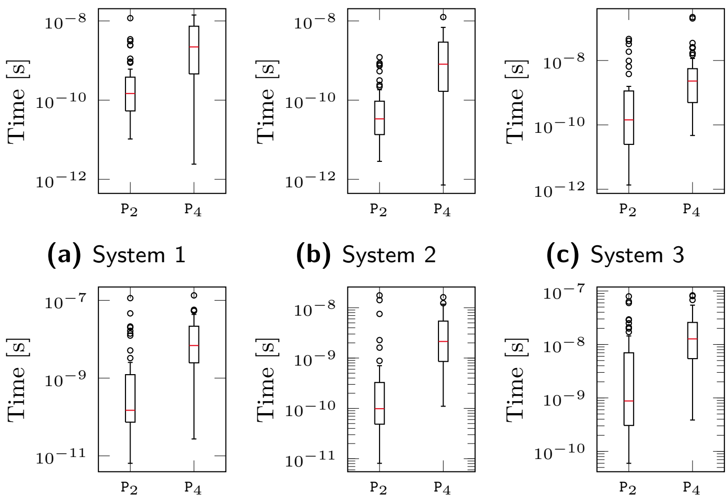
Section 3-3-4. The conditions of the CLF (3.2) and CLBF (3-9) that exclude the goal set are now verified in the goal set. For the two-dimensional systems it is verified that the conditions are satisfied in the goal set. Verification in the goal set for the three-dimensional systems was not successful. It returns unknown for last two conditions of Equation (3-9). As described in Subsection 3-3-4, this can happen due to the conditions being too expensive for Z3 or the procedure implemented in Z3 being incomplete.

## 5-2 Finding Maximum Allowable Sampling Time

The approach as described in Section 4-3 is used to find an approximation $\tau^*$ of the MASP $\tau_{\text{MASP}}$. As $I$ as explicitly known, it is easier to find the MASP for $\mathtt{CS}_2$. Therefore, in this section the MASP for the problems $\mathtt{P}_2$ and $\mathtt{P}_4$ are found, involving $\mathtt{CS}_2$. For $\mathtt{CS}_2$ the set $S_0$ is $S_0 = I$.

In section 5-2 statistics on the approximations of the MASP can be found. It can be seen that for all the systems the approximated MASP is higher for the feedback linearizing controllers ($\mathtt{P}_4$) compared to the template-based controllers ($\mathtt{P}_2$). It is also noted that the approximated MASP is much lower than the minimal dwell time in simulations.

A function $M(x)$ is fitted following the approach as described in Section 4-4 to 50 event-triggered controllers synthesized for the system (2). The approximations of the MASP are compared with the approximations of the MASP which are found using a constant function $M(x) = M_c$, with $M_c$ a constant. For 40 of the controllers, the fitted $M$ function results in a lower approximation of the MASP. For the controller with improvement, the approximation of the MASP is on average 38.1 times higher when a tight function $M$ is fitted. For the




**(a)** System 1 **(b)** System 2 **(c)** System 3

**(d)** System 4 **(e)** System 5 **(f)** System 6

**Figure 5-7:** Statistics on the MASP for the different systems for 5 synthesis runs

controllers where no improvement is obtained, the fitted tight $M$ function lies above the constant $M_c$ at the state where the dwell time is minimized.

In Section 4-4 it was also described that by bounding the parameters of the templates, the Lipschitz functions $L_f$ and $L_V$ can be bounded and thus the MASP can be increased. In this experiment the lower bound on the parameters of the templates $V_T$ and $M_T$ is set to -2 and the upper bound on 2. A total of 50 new event-triggered controllers is synthesized with bounded parameters for the system (2) and for the problem $\mathtt{P}_4$. Comparing the average MASP over 50 synthesis runs with the newly synthesized controllers with average MASP of System (2), $\mathtt{P}_4$ it is concluded that the average MASP with bounded parameters over 50 synthesis runs is 43.9 times higher compared to the average MASP for the unbounded parameters. In another experiment a lower bound of -1 and upper bound of 1 is set to the parameters. However in this case a solution with optimal fitness cannot be found by CMA-ES. This indicates that a solution does not exists.

## 5-3 Literature Comparison

In this section the behavior of synthesized event-triggered controllers is compared with other ETC implementations. First, system (5) is considered. The same system was considered for the design of an event-triggered controller in for example [60, 12, 23, 15]. The triggering mechanisms in [60, 12, 23, 15] is based on a ISS-CLF.

A template-based controller is synthesized for $\mathtt{CS}_1$ with $(S, G) = ([-60, 60]^2, S = [-0.1, 0.1]^2)$ and $\gamma_c = 0.01$. The initial state $x(0)$ is chosen as $x(0) = [-10, 10]$. It is verified that $R = \{x \in S \mid V(x) \leq V(x(0))\} \subset int(S)$. Simulations can be found in Figure 5-8. The trajectories converge to the origin within 5 seconds, whereas the event-triggered controller of [16, Fig. 3] converges in 10 seconds. For the synthesized controller the dwell-time is constant, whereas the dwell-time in [16] increases when the states approach a vicinity of the origin. It can also be observed that the the synthesized controller, has on average a 25 times higher

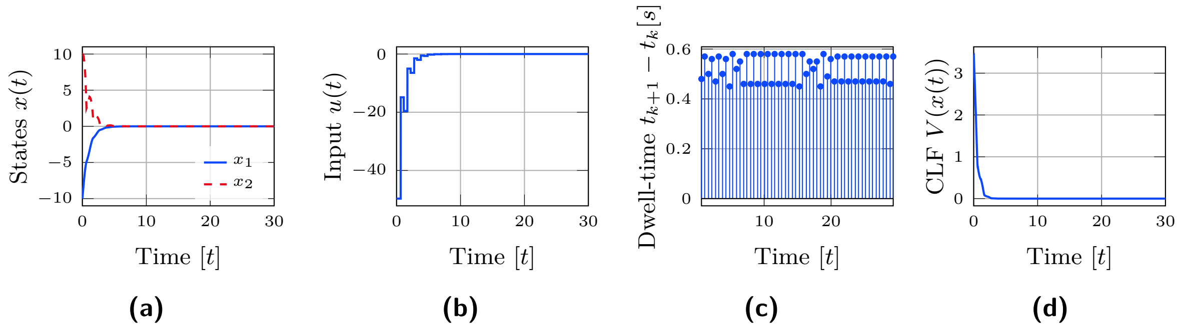



**Figure 5-8:** Simulation of the system (5)

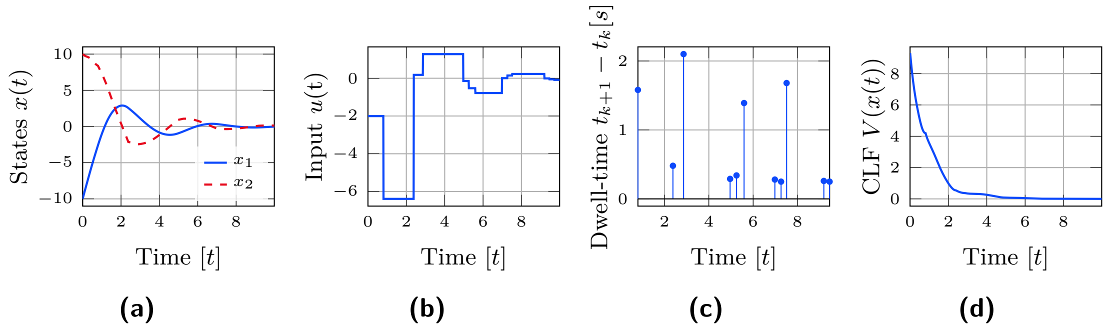



**Figure 5-9:** Simulation of the system (6)

dwell-time than the one of [15]. In [15] a MASP of $6.5 \times 10^{-9}$ is reported, which is similar to our approximation.

Next, system (6) is considered. This system represent a single-link robot arm. The design of an event-triggered controller for this system was also considered in [3, 71]. In this work they also use a triggering mechanism that is based on a ISS CLF. We synthesize a template-based controller for $\texttt{CS}_1$ with $(S, G) = ([-30, 30]^2, S = [-0.5, 0.5]^2)$ and $\gamma_c = 0.01$. As in [3], the initial state is $x(0) = [-10, 10]$. The level set $R$ is contained in $S$. A simulation can be found in Figure 5-9. Again, the dwell-time is much higher, compared to [3, Fig 4.] and no clear drop in dwell time is observed in the vicinity of the origin, whereas this is the case in [3]. In [3] a MASP is reported of $5.6 \times 10^{-3}$, which is significantly higher than our approximation.

The synthesis of controller for the systems (1), (2), (4), using CEGIS was also considered in [65, 47]. In [65] a switched controller is synthesized for sampled-data systems and in [47] for a general nonlinear system for a control specification similar to $\texttt{CS}_2$. In Table 5-2 a table can be found with the total synthesis time of the [66, 47] and the synthesis of this work. The synthesis time for all methods is of the same order of magnitude.

## 5-4 Summary and Discussion

In this chapter we applied the results of Chapter 3 and Chapter 4 to synthesize (periodic) event-triggered controllers. We have tested the approaches on multiple systems. A control law is found by either simultaneously synthesizing it with a certificate function or by feedback

| | [65] | [47] | $\mathtt{CS}_2$ templated-based | $\mathtt{CS}_2$ feedback linearization |
|---|---|---|---|---|
| System (1) | 3.6 | 5.2 | 7.1 | 6.3 |
| System (2) | 7.9 | 2.6 | 12.8 | 6.2 |
| System (3) | 12.3 | 3.0 | 7.4 | 6.7 |

**Table 5-2:** Comparison of synthesis time with [65, 48]

linearization. In Section 5-1 an event-triggered controller were synthesized for five different systems according to $\mathtt{P}_1$-$\mathtt{P}_4$.

In Section 5-1-2 simulations were shown of synthesized controllers for the system (2).The primary conclusion is that when a template controller ($\mathtt{P}_1$, $\mathtt{P}_2$) is used, the difference between synthesized and feedback linearized controllers is more substantial than for feedback linearized controllers ($\mathtt{P}_3$, $\mathtt{P}_4$). On the one hand, this offers flexibility to find controllers with the desired behavior; on the other hand, the synthesis is less predictable. This result is as expected, as for the feedback linearizing controller, the controller is fixed, and the different behavior is only caused by the CLBF that is synthesized. For the template-based controller, it is also dependent on the control law that is synthesized.

In Section 5-1-3 synthesis results for multiple systems were outlined. For all two-dimensional systems, solutions are found. It was observed that the time to find and verify candidate parameters is less for $\mathtt{CS}_1$ compared to $\mathtt{CS}_2$. This is as expected, as $\mathtt{CS}_1$ involves less conditions and parameters, resulting in an easier-to-solve optimization problem in the search for a candidate parameter and less conditions to verify in the verification step of the synthesis algorithm. A disadvantage of $\mathtt{CS}_1$ is that it is not possible by the user to enforce a specific set of initial conditions, whereas this is possible for $\mathtt{CS}_2$. As a result, the sublevel set $R$ can only be a small subset of the domain $D$.

It was also observed that for synthesis of a CLF to enforce $\mathtt{CS}_1$ ($\mathtt{P}_5$) the synthesis time is much lower than for the problems $\mathtt{P}_1$-$\mathtt{P}_4$. This is because convex optimization is used to find a function $V$, which is more efficient in computational time. Also, only two conditions need to be verified. However, it must be noted that this approach is only applicable to feedback linearizable systems to enforce $\mathtt{CS}_1$.

For the feedback linearizing controller, less parameters need to be found. Therefore, it was expected that a candidate can be found faster. However, note that this cannot be concluded from the synthesis results. This can be explained by the fact that by fixing a feedback linearized controller, the candidate space decreases and thus the number of possible solutions, which makes it more difficult to find a solution.

Event-triggered controllers were also synthesized for a three-dimensional linear and polynomial system. As expected, the synthesis time is higher, because more parameters need to be found. As a diffeomorphism is unknown, $\mathtt{P}_3$ and $\mathtt{P}_4$ cannot be solved for the three-dimensional system (8). This illustrates that feedback linearization involves more expert knowledge because a diffeomorphism need to be provided, which might be difficult to find.

Another disadvantage of a feedback linearizing controller is that it is needed to specify the poles of the closed-loop $z$-dynamics. It was illustrated that for some poles, no solution can be found. The synthesis framework can also be adopted to be able to place the poles automatically, as described in Remark 3.10, but this increases the number of parameters to be found.

In Section 5-2 an approximation of the MASP is found for the synthesized controllers of Section 5-1-3. It was noted that for the feedback linearized controllers, the MASP is higher compared to the template-based controllers. This can be caused by the chosen poles of the $z$-dynamics for the feedback linearizing controller. By fitting a function $M$ following the approach of Section 4-4 the approximations of the MASP can be improved; that is the approximations are 38.1 times higher. By bounding the parameters of the templates controllers are found with a MASP that is on average 43.9 times higher.

Another observation is that the dwell-time in simulations is much larger than the approximations of the MASP; it appears that estimates on the MASP are conservative. This possibly can be caused by the estimations of Lipschitz constants being conservative or the optimization, used to find approximations getting stuck in local minima. Finding better approximations is a topic of further research. Another possible cause is that the provided lower bound on the MASP is conservative.

In Section 5-3 the behavior of synthesized controllers was compared with other ETC implementations. The main observation is that, in the vicinity of the origin the dwell-time does not drop for our implementation, whereas it does for the implementations [60, 12, 23, 15]. Intuitively, the stabilization of the system needs less control attention in a vicinity of the origin, and hence the dwell time can be high. A topic of future research is to influence the dwell time, especially in a neighbourhood of the origin. For the system (6) a MASP was provided in [3] that is much higher than the approximation found in this work for this particular system. This again motivates for further research to find better approximations of the MASP and to be able to influence the MASP.

The synthesis results were also compared with existing CEGIS synthesis approaches. It was concluded that the synthesis time for event-triggered controllers is comparable with other CEGIS approaches to synthesize for example switched controllers for general nonlinear systems.

Chapter 6

# Conclusion and Recommendations

## 6-1 Conclusions

The goal of this thesis was to develop a framework for automated synthesis of event-triggered controllers for general nonlinear systems with stability and possibly safety specifications that can be readily implemented on digital devices. The triggering function of the event-triggered controllers in this work are based on certificate functions. In existing event-triggered control (ETC) literature the certificate function and feedback law are assumed to be known. However, synthesis of a certificate function and feedback law is non-trivial for general nonlinear systems. This is mainly due to the fact that Zeno behavior must be excluded. In Chapter 3 the following subproblem was considered.

**Subproblem 1**

Synthesize an event-triggered controller for a nonlinear system that ensures that:

- S1) the event-triggered controller does not trigger infinitely fast;
- S2) the origin is exponentially stable and system trajectories cannot leave a known safe set.

We proposed two methods to synthesize event-triggered controllers, such that Subproblem 1 was solved. A Counterexample-Guided Inductive Synthesis (CEGIS) approach was used to synthesize certificate function for event-triggered control. The methods differ in how the feedback law is found, i.e. a feedback law is simultaneously found with CEGIS, or feedback linearization is used. Using the approaches, it is also possible to synthesize event-triggered controllers that consider safety. Simultaneously finding a controller has the disadvantage the search space becomes bigger, compared to the feedback linearization approach. However, the feedback linearization approach is not applicable to all nonlinear systems. In the synthesis framework candidate solutions are verified with the Satisfiability Modulo Theory (SMT) solver dReal. For general nonlinear systems it is not possible to verify the conditions in a region around the origin. Verification around the origin is only possible for polynomial systems with polynomial template functions.

In Chapter 4, a periodic event-triggered controller counterpart of the event-triggered controllers in Chapter 3 was introduced. Furthermore, in this chapter we considered the following subproblem.

> **Subproblem 2**
>
> Find an approximation of the maximum admissible sampling period, such that a periodic event-triggered controller provides the same control specification as the continuous event-triggered controller.

We proposed a method to be able to find an approximation of the maximum admissible sampling period (MASP), which solves Subproblem 2. The approximations found with this method are used to choose the sampling interval. This allows the event-triggered controller to be implemented digitally. A disadvantage of the proposed method is that the approximation is not guaranteed to be lower than the real MASP. This can lead to the sampling interval being chosen to low.

In Chapter 5, it was proved through experiments that the method is able to find event-triggered controllers for systems up to three dimensions. It was expected that using a feedback linearizing controller the time to find candidate parameters would decrease, as the number of parameters to be found with CEGIS decreases. However, it was concluded that the candidate time for feedback linearization was comparable with the candidate time of template-based controllers. Other notable differences are that the average inter-event time and estimated MASP for feedback linearized controllers are higher. In terms of expert knowledge needed, for feedback linearization, less templates need to be provided, compared with template-based synthesis. However, for feedback linearization it is assumed that a diffeomorphism is known. In general, this is not the case and expert knowledge is needed to derive a diffeomorphism.

In Chapter 5 it was also concluded that the time to find candidates and the verification time grows quick for increasing dimension. To keep the increase in synthesis time limited, the number of parameters to be found in the synthesis must be kept low. The use of SMT-solvers for verification can be a bottleneck.

To summarize, it is possible to synthesize (periodic) event-triggered controllers with stability and safety specifications for general nonlinear systems using the approaches described in this thesis. However, the main bottleneck is that the search space increases significantly for higher dimensions. As a result, the synthesis time increases significantly. In the next section, recommendations are given for future research.

## 6-2 Recommendations

Altough the approaches have shown to work well, some issues arise in practice. In this section we propose other synthesis approaches and future research directions. Below we will describe some recommendations to solve some of the issues described in this chapter.

**Sontags Controller** Instead of using a template controller or a feedback linearizing feedback law one could use Sontag's universal control law. It is possible to reduce the number of parameters needed to find in the synthesis process by using this control law. Sontag's universal control law was described in Section 2-2. It was described that it provides an explicit formula to derive a stabilizing controller from the Control Lyapunov Function (CLF). The main disadvantage of using Sontag's law is that its form does not allow direct verification by the SMT-solvers used in this work. It is recommended to research whether it is possible to make verification in some sort possible to use Sontag's control law.

**Verification of Non-Polynomial Systems Around the Origin** Inherent to $\delta$-satisfiability approaches for SMT verification is that it is not possible to verify an inequality that becomes equality for some limit value. Therefore for nonpolynomial systems, it was not possible to verify the conditions in a region around the origin. However, by taking a polynomial abstraction of the original nonpolynomial system the resulting polynomial system can be verified with a solver that does not rely on a $\delta$-satisfiability approach, such as Z3.

**Guiding the MASP** In the current approach, we can not directly influence the MASP. This has as a result that the functions found in the CEGIS approach lead to a MASP that is lower than needed. We recommend researching a method to be able to guide the MASP. This can be done, for example, by bounding individual functions that involve computation of the MASP, which are the function $L_f$, $L_{V'}$ and $M$ or by directly optimizing the MASP in the synthesis process.

**Verification of the MASP** A disadvantage of the proposed method to find an approximation of the MASP is that the approximation is not guaranteed to be lower than the real MASP. This can lead to the sampling interval being chosen to low. It is recommended to research whether the approximations can be verified with an SMT-solver. When this verification can be done in reasonable time, SMT-solvers can also be used to find better approximations, by using line search methods for example. This way, iteratively an approximation can be found and verified, such that the approximation is close to the real value of the MASP.

**Genetic Programming** In the CEGIS approach used in this thesis, it is needed to specify a template of the to-be-found functions. For the certificate function and feedback law we used polynomial templates. Although a solution is found for the templates that are used, it can be that using a polynomial template, a solution can not be found. Furthermore, some expert knowledge is needed to define the templates. A solution to this is to automatically evolve the structure of the function along with its parameters. As a result, it is not needed to define templates. In [68] for example this is done with Genetic Programming (GP). Although this approach has shown great potential for different types of systems, the search space is significantly increased [67], [68], which increases synthesis time.

Furthermore, other learning approaches recently developed can be considered for synthesis of certificate functions for event-triggered control. For example, in [2] Lyapunov Neural Networks are constructed to automatically find Lyapunov functions, without defining a template.

**Sample-Based Verification** A bottleneck for scalability to higher dimension is the verification time needed by the SMT-solver. Ideally, one would be able to parallelize computation, such that the verification is speed up. Algorithms that are of decentralized nature are suitable for parralelization, i.e. the algorithms allow for multiple separate calculations at the same time. However, to the best of our knowledge, the SMT-solver dReal and Z3 used in this work does not allow for parralelization. Opposed to SMT-solvers, sample-based algorithms allow parralelization as they are distributed [13], [11]. These algorithms exploit local continuity to extend validity of an inequality in a finite number of sampling points to an infinite bounded set of points. However the reported verification times are much higher than for the SMT-solver we use in this work. More research is needed to find whether this type of verification is useful.

# Appendix A

# Conditions on Feedback Linearizability

Consider the system as defined in Section 3-1. In this section the conditions on feedback linearizability of a nonlinear system is outlined. To be able to describe the results on feedback linearizability of the nonlinear system Equation (3-1) we introduce the following input-output system

$$\begin{aligned} \dot{x} &= f(x) + g(x)u \\ y &= h(x). \end{aligned} \tag{A-1}$$

The relative degree is defined as the number of times the output $y$ has to be differentiated until the input $u$ appears. The formal definition is given below.

**Definition A.1** (Relative Degree, ([27], Definition 8.1))**.** *The nonlinear system Equation* (3-1) *has relative degree $\rho$, $1 \leq \rho \leq n$, in an open set $\mathcal{R} \subset D$ if, for all $x \in \mathcal{R}$, $L_g L_f^{i-1} h(x) = 0$, for $i = 1, 2, \dots, \rho - 1$; $\quad L_g L_f^{\rho-1} h(x) \neq 0$.*

For the case that $\rho = n$, a simple expression is available for the control law. Hence the goal is to select a 'virtual output' $h(x)$, such that with respect to this output, the system has relative degree $\rho = n$ in a domain $D_x \subset D$, with $x(0) \in D_x$. This comes down to finding $h(x)$, such that partial differential equations

$$L_g L_f^{i-1} h(x) = 0, \quad i = 1, 2, \dots, n-1 \tag{A-2}$$

are satisfied, subject to the condition

$$L_g L_f^{n-1} h(x) \neq 0, \tag{A-3}$$

for all $x \in D_x$. Conditions exist for the existence of a function $h$ such that a solution exists of the problem above. We introduce some mathematical notations and definitions before stating a result on feedback linearizability.

**Definition A.2** (Lie Derivative)**.** *Let $h : \mathrm{D} \to \mathbb{R}$ be a smooth function, and $f : \mathbb{D} \to \mathbb{R}^n$ be a smooth vector field on the domain $\mathbb{D} \in \mathbb{R}^n$. Then, the Lie derivative of $h(x)$ with respect to $f(x)$ is a scalar function defined by*

$$L_f h(x) = \frac{\partial h}{\partial x}(x) f(x). \tag{A-4}$$

**Definition A.3** (Lie Bracket)**.** *Let $f$ and $g$ be two vector fields on the domain $\mathbb{D} \in \mathbb{R}^n$. The Lie bracket of $f(x)$ and $g(x)$ is a vector field defined by*

$$[f,g](x) = \frac{\partial g}{\partial x}(x)f(x) - \frac{\partial f}{\partial x}(x)g(x). \tag{A-5}$$

**Definition A.4** (Distributions)**.** *For vector fiels $f_1, f_2, \ldots, f_k$ on $D \subset \mathbb{R}^n$, let*

$$\Delta(x) = \text{span}\,\{f_1(x), f_2(x), \ldots, f_k(x)\} \tag{A-6}$$

*be the subspace of $\mathbb{R}^n$ spanned by the vectors $f_1(x), f_2(x), \ldots, f_k(x)$ at any fixed $x \in D$. The collection of all vector spaces $\Delta(x)$ for $x \in D$ is called a distribution and referred to by*

$$\Delta = \text{span}\,\{f_1, f_2, \ldots, f_k\} \tag{A-7}$$

**Definition A.5** (Involutive Distributions)**.** *A distribution $\Delta$ is involutive if, whenever $f, g \in \Delta$, also $[f,g] \in \Delta$*

The above definitions allow to introduce the conditions of feedback linearizability.

**Theorem A.6** (Conditions on Feedback Linearizability, [27, Theorem 8.2])**.** *The system Equation* (3-1) *is feedback linearizable in a neighborhood of $x_0 \in D$ if and only if there is a domain $D_x \subset D$, with $x_0 \in D_x$, such that*

1. *The matrix $\left[g(x)\ ad_f g(x) \cdots ad_f^{n-1} g(x)\right]$ has rank $n$ for all $x \in D_x$;*
2. *the distribution $\mathcal{D} = \text{span}\left\{g, ad_f g, \ldots, ad_f^{n-2} g\right\}$ is involutive in $D_x$.*

# Bibliography

[1] Karl-Erik Åarzén. A simple event-based PID controller. *IFAC Proceedings Volumes*, 32(2):8687–8692, 1999.

[2] Alessandro Abate, Daniele Ahmed, Mirco Giacobbe, and Andrea Peruffo. Formal Synthesis of Lyapunov Neural Networks. *IEEE Control Systems Letters*, 5(3):773–778, 2021.

[3] Mahmoud Abdelrahim, Romain Postoyan, Jamal Daafouz, and Dragan Nešić. Robust event-triggered output feedback controllers for nonlinear systems. *Automatica*, 75:96–108, 2017.

[4] Aaron D. Ames, Kevin Galloway, Koushil Sreenath, and Jessy W. Grizzle. Rapidly exponentially stabilizing control lyapunov functions and hybrid zero dynamics. *IEEE Transactions on Automatic Control*, 59(4):876–891, 2014.

[5] Adolfo Anta and Paulo Tabuada. To sample or not to sample: Self-triggered control for nonlinear systems. *IEEE Transactions on Automatic Control*, 55(9):2030–2042, 2010.

[6] Zvi Artstein. Stabilization with relaxed controls. *Nonlinear Analysis*, 7(11):1163–1173, 1983.

[7] Christel Baier and Joost-Pieter Katoen. *Principles of Model Checking*, volume 1. 2008.

[8] Clark Barrett, Roberto Sebastiani, Sanjit A. Seshia, and Cesare Tinelli. Satisfiability modulo theories. In *Handbook of Model Checking*, pages 305–343. 2009.

[9] Calin Belta and Sadra Sadraddini. Formal Methods for Control Synthesis: An Optimization Perspective. *Annual Review of Control, Robotics, and Autonomous Systems*, 2(1):115–140, 2019.

[10] Franco Blanchini and Stefano Miani. *Set-theoretic methods in control*. Birkhäuser, 2015.

[11] Ruxandra Bobiti and Mircea Lazar. Automated-Sampling-Based Stability Verification and DOA Estimation for Nonlinear Systems. *IEEE Transactions on Automatic Control*, 63(11):3659–3674, 2018.

[12] Dominicus Paulus Borgers and W. P.M.H. Heemels. Event-separation properties of event-triggered control systems. *IEEE Transactions on Automatic Control*, 59(10):2644–2656, 2014.

[13] Peng Cheng and Vijay Kumar. Sampling-based falsification and verification of controllers for continuous dynamic systems. *International Journal of Robotics Research*, 27(11-12):1232–1245, 2008.

[14] Leonardo De Moura and Nikolaj Bjørner. Z3: An efficient SMT Solver. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 4963 LNCS(April 2008):337–340, 2008.

[15] M. C.F. Donkers and W. P.M.H. Heemels. Output-based event-triggered control with guaranteed L - infinity gain and improved and decentralized event-triggering. *IEEE Transactions on Automatic Control*, 57(6):1362–1376, 2012.

[16] M. C.F. Donkers, P. Tabuada, and W. P.M.H. Heemels. Minimum attention control for linear systems: A linear programming approach. *Discrete Event Dynamic Systems: Theory and Applications*, 24(2):199–218, 2014.

[17] Kenneth Eriksson, Claes Johnson, and Donald Estep. *Vector-Valued Functions of Several Real Variables*, pages 789–814. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004.

[18] Samira S. Farahani, Vasumathi Raman, and Richard M. Murray. Robust Model Predictive Control for Signal Temporal Logic Synthesis. *IFAC-PapersOnLine*, 48(27):323–328, 2015.

[19] R. Freeman. *Robust Non Linear Control Design*. Birkhäuser, 1996.

[20] Sicun Gao, Jeremy Avigad, and Edmund M. Clarke. $\delta$-Complete decision procedures for satisfiability over the reals. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 7364 LNAI(1041377):286–300, 2012.

[21] Sicun Gao, Soonho Kong, and Edmund M. Clarke. dReal: An SMT solver for nonlinear theories over the reals. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 7898 LNAI(1041377):208–214, 2013.

[22] N. Hansen and A. Ostermeier. Completely derandomized self-adaptation in evolution strategies. *Evolutionary computation*, 9(2):159–195, 2001.

[23] W. P.M.H. Heemels, M. C.F. Donkers, and Andrew R. Teel. Periodic event-triggered control for linear systems. *IEEE Transactions on Automatic Control*, 58(4):847–861, 2013.

[24] Joao P. Hespanha, Payam Naghshtabrizi, and Yonggang Xu. A survey of recent results in networked control systems. *Proceedings of the IEEE*, 95(1):138–172, 2007.

[25] Shao Chen Hsu, Xiangru Xu, and Aaron D. Ames. Control barrier function based quadratic programs with application to bipedal robotic walking. *Proceedings of the American Control Conference*, 2015-July:4542–4548, 2015.

[26] James Kapinski, Sriram Sankaranarayanan, Jyotirmoy V. Deshmukh, and Nikos Aréchiga. Simulation-guided Lyapunov analysis for hybrid dynamical systems. *HSCC 2014 - Proceedings of the 17th International Conference on Hybrid Systems: Computation and Control (Part of CPS Week)*, pages 133–142, 2014.

[27] Hassan K Khalil. Nonlinear Systems, 1996.

[28] H.K. Khalil. *Nonlinear Control.* Number 1. 2015.

[29] M Krstić, I Kanellakopoulos, and P Kokotovic. Nonlinear and Adaptive Control Design, 1995.

[30] Nicolas Marchand, Sylvain Durand, and Jose Fermi Guerrero Castellanos. A general formula for event-based stabilization of nonlinear systems. *IEEE Transactions on Automatic Control*, 58(5):1332–1337, 2013.

[31] Nicolas Marchand, J. J. Martinez, S. Durand, and J. F. Guerrero-Castellanos. *Lyapunov event-triggered control: A new event strategy based on the control*, volume 46. IFAC, 2013.

[32] Mazo and Paulo Tabuada. Input-to-state stability of self-triggered control systems. pages 928–933, 2009.

[33] Manuel Mazo, Adolfo Anta, and Paulo Tabuada. An ISS self-triggered implementation of linear controllers. *Automatica*, 46(8):1310–1314, 2010.

[34] Cai Meng, Tianmiao Wang, Wusheng Chou, Sheng Luan, Yuru Zhang, and Zengmin Tian. Remote surgery case: Robot-assisted teleneurosurgery. *Proceedings - IEEE International Conference on Robotics and Automation*, 2004(1):819–823, 2004.

[35] Manfred Morari, Carlos E. Garcia, and David M. Prett. Model predictive control: Theory and practice. *IFAC Proceedings Volumes*, 21(4):1–12, 1988.

[36] Petter Ögren, Edward Fiorelli, and Naomi Ehrich Leonard. Cooperative control of mobile sensor networks: Adaptive gradient climbing in a distributed environment. *IEEE Transactions on Automatic Control*, 49(8):1292–1302, 2004.

[37] TRAVIS E. OLIPHANT. Python for Scientific Computing Python Overview. *Computing in Science and Engineering*, pages 10–20, 2007.

[38] Giordano Pola, Antoine Girard, and Paulo Tabuada. Approximately bisimilar symbolic models for nonlinear control systems. *Automatica*, 44(10):2508–2516, 2008.

[39] R. Postoyan, Adolfo Anta, W. P.M.H. Heemels, Paulo Tabuada, and Dragan Nesic. Periodic event-triggered control for nonlinear systems. *IEEE Conference on Decision and Control*, 2013.

[40] R. Postoyan, Paulo Tabuada, Dragan Nesic, and Adolfo Anta. A Framework for the Event-Triggered Stabilization of Nonlinear Systems. *IEEE Transactions on Automatic Control*, 60(4):982–996, 2015.

[41] Stephen Prajna, Antonis Papachristodoulou, and Pablo A. Parrilo. Introducing SOSTOOLS: A general purpose sum of squares programming solver. *Proceedings of the IEEE Conference on Decision and Control*, 1(2):741–746, 2002.

[42] James A. Primbs, Vesna Nevistić, and John C. Doyle. Nonlinear Optimal Control: a Control Lyapunov Function and Receding Horizon Perspective. *Asian Journal of Control*, 1(1):14–24, 1999.

[43] Anton V. Proskurnikov and Manuel Mazo. Lyapunov event-triggered stabilization with a known convergence rate. *IEEE Transactions on Automatic Control*, 65(2):507–521, 2020.

[44] Dimitrios Pylorof and Efstathios Bakolas. Safe nonlinear control design for input constrained polynomial systems using sum-of-squares programming. *International Journal of Control*, 0(0):1–11, 2020.

[45] Vasumathi Raman, Mehdi Maasoumy, Alexandre Donze, Richard M. Murray, Alberto Sangiovanni-Vincentelli, and Sanjit A. Seshia. Model predictive control with signal temporal logic specifications. *Proceedings of the IEEE Conference on Decision and Control*, 2015-Febru(February):81–87, 2014.

[46] Hadi Ravanbakhsh and Sriram Sankaranarayanan. Counter-Example Guided Synthesis of control Lyapunov functions for switched systems. In *Proceedings of the IEEE Conference on Decision and Control*, volume 54rd IEEE, pages 4232–4239. IEEE, 2015.

[47] Hadi Ravanbakhsh and Sriram Sankaranarayanan. Counter-Example Guided Synthesis of control Lyapunov functions for switched systems – extended. *Proceedings of the IEEE Conference on Decision and Control*, 54rd IEEE:4232–4239, 2015.

[48] Hadi Ravanbakhsh and Sriram Sankaranarayanan. Counterexample Guided Synthesis of Switched Controllers for Reach-While-Stay Properties. *arXiv preprint arXiv:1505.01180*, arXiv:1505, 2015.

[49] Hadi Ravanbakhsh and Sriram Sankaranarayanan. Robust controller synthesis of switched systems using counterexample guided framework. *Proceedings of the 13th International Conference on Embedded Software, EMSOFT 2016*, 2016.

[50] Hadi Ravanbakhsh and Sriram Sankaranarayanan. A class of control certificates to ensure reach-while-stay for switched systems. *Electronic Proceedings in Theoretical Computer Science, EPTCS*, 260(Synt):44–61, 2017.

[51] Hadi Ravanbakhsh and Sriram Sankaranarayanan. Learning control lyapunov functions from counterexamples and demonstrations. *Autonomous Robots*, 43(2):275–307, 2019.

[52] Sadra; Sadraddini and Calin Belta. Robust Model Predictive Control for Signal Temporal Logic Synthesis. *IFAC-PapersOnLine*, 48(27):323–328, 2015.

[53] J. H. Sandee, W. P.M.H. Heemels, and P. P.J. Van Den Bosch. Event-driven control as an opportunity in the multidisciplinary development of embedded controllers. *Proceedings of the American Control Conference*, 3:1776–1781, 2005.

[54] Ricardo G. Sanfelice. On the existence of control Lyapunov functions and state-feedback laws for hybrid systems. *IEEE Transactions on Automatic Control*, 58(12):3242–3248, 2013.

[55] P. Seiler and R. Sengupta. Analysis of communication losses in vehicle control problems. *Proceedings of the American Control Conference*, 2:1491–1496, 2001.

[56] Alexandre Seuret, Christophe Prieur, and Nicolas Marchand. Stability of nonlinear systems by means of event-triggered sampling algorithms. *arXiv preprint arXiv: . . .* , pages 1–40, 2014.

[57] Abhinav Kumar Singh, Ravindra Singh, and Bikash C. Pal. Stability analysis of networked control in smart grids. *IEEE Transactions on Smart Grid*, 6(1):381–390, 2015.

[58] Armando Solar-Lezama, Liviu Tancau, Rastislav Bodik, Sanjit Seshia, and Vijay Saraswat. Combinatorial sketching for finite programs. *International Conference on Architectural Support for Programming Languages and Operating Systems - ASPLOS*, pages 404–415, 2006.

[59] Eduardo D. Sontag. A 'universal' construction of Artstein's theorem on nonlinear stabilization. *Systems and Control Letters*, 13(2):117–123, 1989.

[60] Paulo Tabuada. Event-Triggered Real-Time Scheduling of Stabilizing Control Tasks. *IEEE Transactions on Automatic Control*, 52(9):1680–1685, 2007.

[61] Paulo Tabuada. *Verification and Control of Hybrid Systems: A Symbolic Approach.* Springer International Publishing, 2009.

[62] Paulo Tabuada, K.H. Johansson, and W. P.M.H. Heemels. An Introduction to Event-triggered and Self-triggered Control. *IEEE Conference on Decision and Control*, pages 3270–3285, 2012.

[63] Weehong Tan and Andrew Packard. Searching for Control Lyapunov Functions using Sums of Squares Programming. *Allerton Conference on Communication, Control, and Computing*, pages 210–219, 2004.

[64] Andrew J. Taylor, Victor D. Dorobantu, Yisong Yue, Paulo Tabuada, and Aaron D. Ames. Sampled-Data Stabilization with Control Lyapunov Functions via Quadratically Constrained Quadratic Programs. (1), 2021.

[65] Cees Verdier. *Formal synthesis of analytic controllers An evolutionary approach.* PhD thesis, 2019.

[66] Cees Verdier, N Kochdumper, Matthias Althoff, and Manuel Mazo. Formal synthesis of closed-form sampled-data controllers for nonlinear continuous-time systems under STL specifications. *arXiv preprint arXiv: . . .* , 1(March):1–25, 2020.

[67] Cees Verdier and Manuel Mazo. Formal Controller Synthesis via Genetic Progamming. In *IFAC-PapersOnLine*, volume 50, pages 7205–7210. Elsevier B.V., 2017.

[68] Cees Verdier and Manuel Mazo. Formal Synthesis of Analytic Controllers for Sampled-Data Systems via Genetic Programming. *Proceedings of the IEEE Conference on Decision and Control*, 2018-Decem:4896–4901, 2019.

[69] Cees Verdier and Manuel Mazo. Formal controller synthesis for hybrid systems using genetic programming. *arXiv*, pages 1–28, 2020.

[70] Li Wang, Dongkun Han, and Magnus Egerstedt. Permissive Barrier Certificates for Safe Stabilization Using Sum-of-squares. *Proceedings of the American Control Conference*, 2018-June:585–590, 2018.

[71] W. Wang, R. Postoyan, Dragan Nesic, and W. P.M.H. Heemels. Periodic event-triggered output feedback control of nonlinear systems. *Proceedings of the IEEE Conference on Decision and Control*, 2018-Decem(Cdc):957–962, 2019.

[72] Zheng Wang, Xiaoping Liu, Kefu Liu, Shuai Li, and Huanqing Wang. Backstepping-Based Lyapunov Function Construction Using Approximate Dynamic Programming and Sum of Square Techniques. *IEEE Transactions on Cybernetics*, 47(10):3393–3403, 2017.

[73] Peter Wieland and Frank Allgower. Constructive safety using control barrier functions. *IFAC Proceedings Volumes (IFAC-PapersOnline)*, 7(PART 1):462–467, 2007.

[74] Majid Zamani, Manuel Mazo, Mahmoud Khaled, and Alessandro Abate. Symbolic Abstractions of Networked Control Systems. *IEEE Transactions on Control of Network Systems*, 5(4):1622–1634, 2018.

# Glossary

## List of Acronyms

| | |
|---|---|
| **ETC** | event-triggered control |
| **CLF** | control-lyapunov function |
| **STL** | signal-temporal logic |
| **CBF** | control barrier functions |
| **CEGIS** | Counterexample-Guided Inductive Synthesis |
| **CETC** | Continuous Event-Triggered Control |
| **CBF** | Control Barrier Function |
| **CLBF** | control Lyapunov barrier functions |
| **CLF** | Control Lyapunov Function |
| **CMA-ES** | Covariance Matrix Adaption Evolution Strategy |
| **GP** | Genetic Programming |
| **ISS** | Input-to-State Stable |
| **MASP** | maximum admissible sampling period |
| **MPC** | Model Predictive Control |
| **NCS** | networked control systems |
| **PETC** | Periodic Event-Triggered Control |
| **RWS** | reach-while-stay |
| **SMT** | Satisfiability Modulo Theory |
| **SOS** | Sum-Of-Squares |
| **STC** | Self-Triggered Control |
| **STL** | Signal Temporal Logic |
| **LMI** | Linear Matrix Inequality |