THESIS FOR THE DEGREE OF LICENTIATE OF ENGINEERING

Vehicle-in-the-loop validation of autonomous cars

A framework for modelling, analysis, and control of test-scenarios

Angel Molina Acosta

Department of Electrical Engineering
Chalmers University of Technology
Gothenburg, Sweden, 2022

Vehicle-in-the-loop validation of autonomous cars

A framework for modelling, analysis, and control of test-scenarios

Angel Molina Acosta

Department of Electrical Engineering
Chalmers University of Technology
SE-412 96 Gothenburg, Sweden
Phone: +46 (0)31 772 1000
www.chalmers.se

Printed by Chalmers Reproservice
Gothenburg, Sweden, May 2022

To my family and friends.

Vehicle-in-the-loop validation of autonomous cars

A framework for modelling, analysis, and control of test-scenarios

Angel Molina Acosta
Department of Electrical Engineering
Chalmers University of Technology

Abstract

Validation of autonomous cars is a difficult task because of the complexity that results from the integration of multiple systems and the variety of operating conditions. To this end, testing with real vehicles is crucial to ensure a thorough validation of AD cars. However, testing AD vehicles in public roads is not viable in early stages of the development cycle. An alternative is to conduct tests in controlled environments, such as proving grounds.

This thesis proposes a framework for modelling, analysis, and control of tests-scenarios for validation of autonomous cars by exposing the vehicle-under-test to a traffic scenario at a test track, where mobile test-targets represent other road users. The framework is suitable for leader-follower, multi-agent systems where the motion of the followers should be coordinated with the motion of an externally controlled leader. Scenarios are modelled as switched systems. The feasibility of the scenario is investigated using backward reachability analysis. A constrained optimal control problem is formulated to control the state of the multi-agent system through a sequence of goal sets. Simulation results illustrate the usefulness of the framework.

A second contribution in this thesis is a novel method for decentralized computation of backward reachable sets and robust control invariant sets. The method is applicable to large-scale systems arising from the interconnection of multiple subsystems with linear dynamics. Polyhedral constraints and additive disturbances are considered. Compared to the standard centralized procedure for computation of control invariant sets, the proposed method is more efficient for large-scale systems where the coupling among the subsystems is sparse.

Keywords: scenario-based testing, multi-agent systems, backward reachability, Model Predictive Control

List of Papers

This thesis is based on the following papers:

[A] **Angel Molina Acosta**, Paolo Falcone, "Modelling and Control of Test-Scenarios for Validation of Autonomous Driving Functions". Published in *Proceedings of the American Control Conference*, May 2021.

[B] Davide Liuzza, **Angel Molina Acosta**, Paolo Falcone, Luigi Glielmo, "Distributed control invariant set: the linear affine case". In preparation for submission to *Automatica*.

Other publications by the author, not included in this thesis, are:

[C] T. Johansson, **A. Molina Acosta**, A. Schliep, P. Falcone, "Reinforcement Learning as an Alternative to Reachability Analysis for Falsification of AD Functions". *Machine Learning for Autonomous Driving Workshop at the 35th Conference on Neural Information Processing Systems*, Dec. 2021.

Acknowledgments

This thesis has been possible thanks to the support that I received from various persons. I would like to thanks prof. Paolo Falcone for his guidance during this project. Thanks to Davide Liuzza for the useful discussions and interesting ideas. Thanks to Francesco Costagliogla for setting up this project and to Fredrik Björkqvist for his support. Thanks to my colleagues at Chalmers and Volvo Cars for making this a fun journey. Thanks to my wife and family for their love and support. This project was funded by Volvo Cars through their Industrial PhD Programme.

Acronyms

AD:	autonomous driving
VUT:	vehicle-under-test
LTI:	linear time invariant
MPC:	model predictive control
RCI:	robust control invariant
PLP:	parametric linear programming

Contents

Abstract i

List of Papers iii

Acknowledgements v

Acronyms vi

I Overview 1

1 Introduction 3
1.1 Thesis contributions 5
1.2 Thesis outline 6

2 Modelling test-scenarios 7
2.1 Switched systems 7

3 Backward reachability 13
3.1 Robust controllable sets 13
3.2 Robust control invariant sets 16
3.3 Projection of polyhedra 17

4 Model Predictive Control **21**
4.1 MPC for linear systems 21
MPC for finite-time control into a goal set 24

5 Summary of included papers **27**
5.1 Paper A 27
5.2 Paper B 28

6 Concluding Remarks and Future Work **29**

References **31**

II Papers **35**

A Modelling and Control of Test-Scenarios **A1**
1 Introduction A3
2 Preliminaries A4
2.1 Systems with state-dependent switching A4
2.2 Robust controllable sets A5
3 Problem description A6
3.1 Example scenario A6
3.2 Modelling scenarios as switched systems A7
4 Proposed control framework A9
4.1 Problem decomposition A10
4.2 Feasibility analysis A11
4.3 Robust constrained optimal control formulation A13
5 Simulations A15
6 Conclusions A17
References A19

B Distributed Control Invariant Set **B1**
1 Introduction B3
1.1 Paper contribution B5
2 Background B7
3 Distributed extrusion intersection B10
4 Distributed linear affine reachability B14
4.1 Linear matrix operators B14

4.2 Local reachability B17
4.3 Paper goal B23
4.4 LP form for linear local reachability B24
4.5 Distributed reachability algorithm B28
5 Distributed invariant set B35
6 Numerical Example B40
7 Conclusions B44
References B46

Part I

Overview

CHAPTER 1

Introduction

In recent years the automotive industry and scientific community have invested significant efforts in the development of autonomous driving (AD). Validation of AD vehicles is a difficult task because of the complexity that results from the integration of multiple systems and the variety of operating conditions. Computer simulations can be an efficient approach to validate functions for AD. However, simulations have a limited fidelity and thus can not be the only approach to validate AD vehicles. Testing with real vehicles is crucial to ensure a thorough validation of AD cars.

Due to safety concerns, testing AD vehicles in public roads is not viable in early stages of the development cycle. An alternative is to conduct tests in controlled environments, such as proving grounds. Vehicle testing on proving grounds is currently used to validate Advanced Driver Assistance Systems (ADAS) such as Autonomous Emergency Braking. The European New Car Assessment Programme (EuroNCAP) describes in its protocols a series of tests to evaluate ADAS functions in proving grounds [1]. In such tests, automated target carriers are used to represent cars, cyclists, or pedestrians moving around the vehicle under test (VUT). The test-targets are designed to allow for collisions with the VUT while minimizing damage to the equipment.

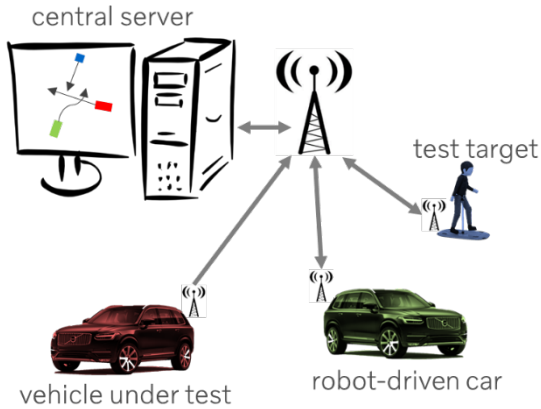

Figure 1.1: Schematic of a vehicle-in-the-loop testing system.

This testing approach falls within the category of vehicle-in-the-loop (VIL) testing.

In [2] results from surveys among automotive companies reveal that proving grounds should be prepared to support automated VIL testing for validation of AD vehicles. EuroNCAP has recently proposed more complex VIL tests to evaluate the next generation of ADAS functions [3]. Such scenarios require precise coordination of the test-targets and the vehicle under test (VUT). Testing systems for such purposes have been developed to a certain extent by some vehicle manufacturers [4]. Yet some aspects of VIL testing remain to be solved. For validation of AD functions, the VUT should be allowed to drive autonomously for it to showcase the AD function being validated. This poses challenges in terms of safety, test feasibility, and reproducibility.

To address the aforementioned challenges, various companies in the Swedish automotive industry joined forces in a research project aiming to develop a novel testing system for proving grounds [5]. The system consists of a central server that receives measurements of the states of the test-targets and the VUT (see Fig. 1.1). The server plans and communicates trajectories to the test-targets. The targets follow the prescribed trajectories, which can be adjusted by the central server during execution. The VUT drives partially or fully controlled by its AD functions which introduce uncertainty because the central server does not have full control over the VUT.

The VIL testing framework described previously motivates the problem addressed in this thesis. The problem can be stated as a leader-follower coordination task, where a set of followers (the test-targets) need to be coordinated with an autonomous or externally controlled leader (the vehicle-under-test). The followers should be controlled throughout the stages composing the test-scenario while satisfying a set of requirements.

To solve such a problem, a systematic methodology is needed to translate a test scenario into an abstraction that allows for analysis and control synthesis. Then, the feasibility of the scenario needs to be assessed, i.e. determining if the control objectives can be achieved despite of limitations on the motion of test-targets and the uncertainty due to the unknown motion of the VUT. Finally, if the scenario is deemed feasible, a control strategy should be designed to execute the scenario and ensure the requirements are satisfied.

This thesis focuses on the development of such analysis and control synthesis framework. The thesis aims at answering following research questions:

- How test-scenarios consisting of various stages can be modelled?
- How the feasibility of a scenario can be analyzed considering the physical limits of the test-targets and a partially unknown motion of the VUT?
- How to design a control strategy that coordinates the motion of the test-targets with the VUT as specified by a test-scenario?

1.1 Thesis contributions

The contributions of this thesis build upon two papers. The first contribution, presented in Paper A, is a framework for controlling a number of test-targets in VIL-testing applications. The framework proposes to model a test-scenario as a switched system. Feasibility of the scenario is analyzed by means of backward reachability analysis. A Model Predictive Control scheme is proposed to control the test-targets and ensure a correct execution of the scenario despite the uncertainty introduced by the externally controlled VUT.

The second contribution, presented in detail in Paper B, is a decentralized method for computation of backward reachable sets and control invariant sets. The method is intended for large-scale systems that arise from the interconnection of subsystems with linear affine dynamics. Polyhedral constraints and additive disturbances are considered. For sparsely coupled systems, the

proposed method can lead to more efficient computations than a centralized approach. The methods described in Paper B can be integrated into the framework proposed in Paper A. This integration is beyond the scope of the thesis.

1.2 Thesis outline

This thesis is structured as follows. Chapter 2 describes an approach to model test-scenarios in a systematic way using switched systems. Chapter 3 discusses backward reachability analysis, controllable sets, and control invariant sets. Chapter 4 provides fundamental concepts on Model Predictive Control and discusses its application to a finite-time control problem. Chapter 5 summarizes the articles upon which the thesis is based. Chapter 6 contains conclusions and suggestions for future work. The papers that contain the main contributions of this thesis are included in Part II.

CHAPTER 2

Modelling test-scenarios

This chapter describes the modelling approach chosen to describe a test-scenario. The adopted models belong to the class of hybrid systems. Hybrid systems combine discrete and continuous-states and are well-suited to describe cyber-physical systems: physical systems controlled by digital computers. A wide variety of hybrid systems have been proposed [6]. This chapter focuses on switched systems, a type of hybrid system whose discrete-state evolves based on a switching signal. An example is provided to illustrate how a test-scenario can be modelled as a switched system.

2.1 Switched systems

The switched systems discussed in this section are based on concepts from [7]. The discussion focuses on switched systems with linear dynamics in discrete-time. This section provides insight into the modelling approach discussed in Paper A.

A switched system has a discrete-state $q(t)$ which takes values in a set $\mathcal{Q} := \{1, \cdots, N_q\} \subset \mathbb{N}$. Here $t \in \mathbb{N}$ denotes the sampling instant. The different values that $q(t)$ can take are referred to as the operating modes of the system.

The time evolution of $q(t)$ is clarified later in this section.

The continuous-state of the switched system is $\xi(t) \in \mathbb{R}^n$ and it has dynamics

$$\xi(t+1) = A^i\xi(t) + B^iu(t) + E^iw(t) \tag{2.1a}$$
$$i = q(t), \tag{2.1b}$$

where $u(t) \in \mathbb{R}^m$ is the continuous control input and $w(t) \in \mathbb{R}^p$ is an external disturbance. In (2.1a) the matrices $A^i \in \mathbb{R}^{n\times n}$, $B^i \in \mathbb{R}^{n\times m}$, and $E^i \in \mathbb{R}^{n\times p}$ have superscript i to indicate their dependency on the discrete-state $q(t)$, as clarified by (2.1b). According to (2.1), the continuous dynamics are described by a collection of linear systems and the discrete-state $q(t)$ indicates which linear system is active at a certain sampling instant t. Although examples with mode-dependent dynamics are not discussed in this thesis the methods in Paper A are suitable for such dynamics.

Mode-dependent constraints are considered: $\xi(t) \in \Xi^i$ and $u(t) \in \mathcal{U}^i$ should be satisfied if $q(t) = i$. The disturbance is assumed to be bounded as $w(t) \in \mathcal{W}^i$. Only linear, convex constraints are considered in this work. Thus, for all $i \in \mathcal{Q}$, sets Ξ^i, $\mathcal{U}^i$, and $\mathcal{W}^i$ are polyhedral sets defined as the intersection of a finite number of half-spaces.

The evolution of the discrete-state $q(t)$ is governed by the occurrence of events. In general, events can be triggered in different ways, for example by discrete external inputs, timers, or the continuous-state ξ entering a region of the state-space. This work considers only the case where $q(t)$ switches based on events associated to the state $\xi(t)$. The dynamics of the state $q(t)$ are described by the following state-update function

$$\begin{aligned} q(t+1) &= f_q(q(t), \delta^i(t)) \\ i &= q(t), \end{aligned} \tag{2.2}$$

with binary variables $\delta^i(t) \in \{0,1\},\ \forall i \in \mathcal{Q}$, and a function $f_q : \mathcal{Q}\times\{0,1\} \to \mathcal{Q}$. In this work the variables $\delta^i(t)$ keep track of when the continuous-state $\xi(t)$ is inside a specified set, as follows

$$\delta^i(t) = \begin{cases} 1, & \text{if } \xi(t) \in \Xi^i_G \\ 0, & \text{otherwise,} \end{cases} \tag{2.3}$$

where the set $\Xi^i_G \in \Xi^i$ is a convex polyhedron referred to as the goal set for mode i.

For the modelling purposes considered here, the function f_q in (2.2) is chosen as follows:

$$q(t+1) = f_q(q, \delta^i) = \begin{cases} q(t) + \delta^i(t), & \text{if } q(t) \in \mathcal{Q} \setminus N_q \\ q(t), & \text{if } q(t) = N_q \end{cases} \qquad (2.4)$$
$$\text{with } i = q(t)$$

Expressions (2.3)-(2.4) imply that the discrete-state $q(t)$ increases by 1 when the continuous-state $\xi(t)$ enters the set Ξ_G^i. Then $q(t)$ remains constant once it reaches N_q.

The switched system described previously is suitable to model test-scenarios consisting of a fixed sequence of stages, a sequence that is known before test execution. The discrete-state q encodes the current stage of the scenario while the continuous state ξ collects variables related to the motion of the scenario-participants. Transitions between stages occur when the state ξ satisfies a collection of linear inequalities describing the sets Ξ_G^i which represent a desired configuration for the participants, for example specific positions and velocities. The scenario ends when the state ξ enters the set $\Xi_G^{N_q}$ during stage N_q.

More general hybrid systems have been described in the literature (see [6], [7]). However, the class of switched systems described in this section is suitable to represent a wide variety of test-scenarios. As discussed in following chapters, the structure of the switched systems considered here can be exploited to develop tractable methods for analysis and control.

Illustrative example

This section illustrates how a test-scenario can be modelled as a switched system. The example is inspired by a test contained in Euro NCAP's protocol for assessing an Autonomous Emergency Braking function [1]. Such a function is designed to slow down or stop the car when a potential collision is detected. For this test Euro NCAP's protocol specifies a constant speed for the vehicle-under-test but the example described next assumes that the VUT is driving autonomously and thus it autonomously selects its speed profile.

The scenario takes place at a T-junction where the VUT intends to turn left while a pedestrian (a dummy carried by a moving platform) is crossing the road (see Fig. 2.1). The overall goal of the scenario is to bring the participants close to a collision and evaluate how the VUT reacts. The VUT is assumed to follow a fixed path when turning at the intersection (red-dashed

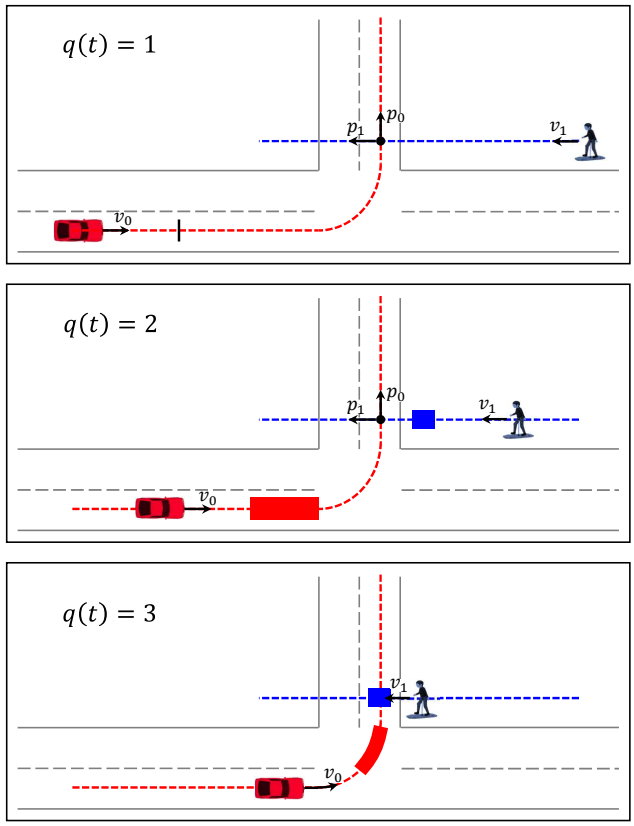

Figure 2.1: Schematic of a test-scenario consisting of 3 stages (not to scale).

line). A straight path is assigned to the pedestrian (blue-dashed line). The position of the VUT and pedestrian along their path are denoted by p_0 and p_1, respectively; the velocities along the path are v_0 and v_1; the accelerations are a_0 and a_1. The origin of the reference system is placed at the intersection of both paths, as shown in the figure.

As depicted in Fig. 2.1, the scenario is divided into three stages, hence a switched system with discrete-state $q(t) \in \mathcal{Q} = \{1, 2, 3\}$ is used. In the first stage, the VUT and the pedestrian start from standstill and they accelerate up to a certain speed. In the second stage the VUT approaches the intersection

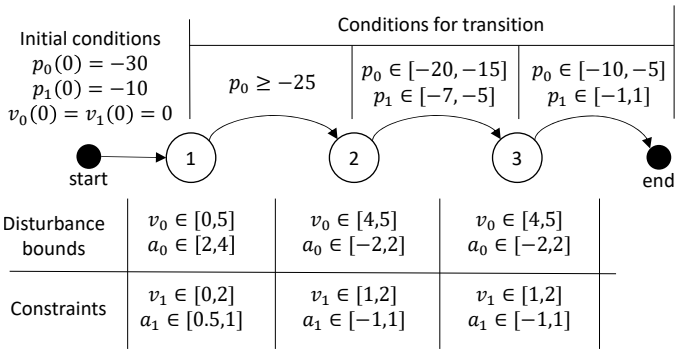

Figure 2.2: Specifications for the test-scenario shown in Fig. 2.1. Position, velocity, and acceleration are given in m, m/s, and m/s^2 respectively.

and the pedestrian approaches the pedestrian cross. In the third stage the pedestrian crosses the road while the VUT drives through the intersection. It is assumed that a central server monitors the scenario and controls the motion of the pedestrian while the VUT drives autonomously.

The scenario specifies desired values for the position, velocity, and acceleration of the participants during the three stages. Initial conditions and conditions for transition between stages are also specified. An example of such specifications is shows in Figure 2.2, where the numerical values were chosen somewhat arbitrarily, just for illustration purposes.

Transition from $q(t) = 1$ to $q(t) = 2$ occurs when the position of the VUT satisfies $p_0 \geq -25$, illustrated at the top of Fig. 2.1 with a black line over the VUT's path. Then, the transition from $q(t) = 2$ to $q(t) = 3$ occurs when $p_0 \in [-20, -15]$ and $p_1 \in [-7, -5]$ are satisfied, which is shown in the middle of Fig. 2.1 with the red and blue rectangles over the paths. At the end of stage 3, the positions should satisfy $p_0 \in [-10, -5]$ and $p_0 \in [-1, 1]$, which represents a potential collision. No further stages are specified in this example since the main goal is to bring the participants close to a collision.

The central server must ensure that the stage-dependent constraints over v_1 and a_1 are satisfied. Also the conditions for transition should be met to ensure a correct evolution of the scenario. The values specified for v_0 and a_0 are regarded as bounds over a disturbance because these variables are controlled externally, i.e. by the autonomous functions of the VUT. The

a priori unknown speed profile of the VUT makes it challenging (possibly infeasible) to control the pedestrian and satisfy the conditions for transitions subject to the constraints over v_0 and a_0.

To complete the model of the scenario as a switched system, the continuous-state is chosen as $\xi = [p_0, p_1, v_1]^T$, the control input is $u = a_1$, and the disturbance $w = [v_0, a_0]^T$. Adopting double-integrator dynamics to describe the evolution of $p_j(t)$ and $v_j(t)$ as a function of $a_j(t)$, with $j \in \{0, 1\}$, the dynamics for $\xi(t)$ are:

$$\xi(t+1) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & T_s \\ 0 & 0 & 1 \end{bmatrix} \xi(t) + \begin{bmatrix} 0 \\ 0.5T_s^2 \\ T_s \end{bmatrix} u(t) + \begin{bmatrix} T_s & 0.5T_s^2 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} w(t), \tag{2.5}$$

with T_s the sampling time. Expression (2.5) is a particular case of (2.1) where matrices A^i, B^i, and E^i remain constant for $i = 1, \cdots, N_q$.

Sets Ξ^i, $\mathcal{U}^i$, $\mathcal{W}^i$, Ξ^i_G can be easily derived from the specifications in Fig. 2.2. For example, for mode 1 the admissible state and input sets are respectively $\Xi^1 = \{\xi : v_1 \in [0, 2]\}$ and $\mathcal{U}^1 = \{u : a_1 \in [1.5, 2]\}$; the disturbance set is $\mathcal{W}^1 = \{w : v_0 \in [0, 5], a_0 \in [2, 4]\}$; the goal set is $\Xi^i_G = \{\xi : p_1 >= -25\}$.

To ensure a correct execution of a test-scenario modelled as discussed before, one should solve a sequence of finite-time entry problems. The state ξ should be controlled to visit the goal sets $\Xi^1_G, \cdots, \Xi^{N_q}_G$, ensuring robustness with respect to the disturbance, and satisfying mode-dependent state and input constraints. The problem at hand is stated more formally as follows:

Problem 1: *Given a switched system specified by* (2.1) *and* (2.4)*, design a control input* $u(t)$ *such that the continuous state* $\xi(t)$ *is guaranteed to visit the goal sets* $\Xi^1_G, \Xi^2_G, \cdots, \Xi^{N_q}_G$ *while satisfying the mode-dependent constraints* $\xi(t) \in \Xi^{q(t)}$ *and* $u(t) \in U^{q(t)}$*, for all possible sequences of the disturbance* $w(t) \in W^{q(t)}$*.*

It might be the case that the problem above cannot be solved starting from certain initial conditions, due to the constraints on the inputs (e.g., the limited acceleration of the test-targets) and the external disturbance (e.g., the uncertainty in the speed of the VUT). Hence, investigating the feasibility of the problem is important before attempting to design a controller. Paper A describes a method to determine feasible initial conditions using backward reachability analysis, a topic that is introduced in Chapter 3. To solve Problem 1, a Model Predictive Control scheme is proposed in Paper A, for which Chapter 4 provides additional background.

CHAPTER 3

Backward reachability

This chapter discusses backward reachability for systems with LTI dynamics in discrete-time subject to linear, convex constraints and bounded additive disturbances. Systems with inputs are considered here rather than autonomous systems. Backward reachability refers to a property of dynamical systems that indicates from which initial states a system can be controlled to reach a desired terminal state. The concepts presented here are the basis for the method described in Paper A to analyze the feasibility of a test-scenario when modelled as described in Chapter 2. Backward reachable sets are discussed in Paper B for the case of interconnected LTI systems.

3.1 Robust controllable sets

Backward reachability for LTI systems is usually discussed in the context of sets. For a goal set specified in the state-space, backward reachable sets contain initial states from which the system can reach the goal set. In the literature, backward reachable sets are also referred to as controllable sets [7]. Controllable sets are useful because they allow to identify states for which a control objective can be achieved, in finite-time, without violating constraints.

Although the definitions presented here can be extended to systems with more general dynamics, only systems with the the following LTI dynamics are considered in this chapter:

$$x(t+1) = Ax(t) + Bu(t) + Ew(t), \tag{3.1}$$

where $t \in \mathbb{N}$ is the sampling instant, $x(t) \in \mathbb{R}^n$ is the state, $u(t) \in \mathbb{R}^m$ is the control input, and $w(t) \in \mathbb{R}^p$ is the disturbance. State and input constraints are specified via polyhedral sets defined as the intersection of a finite-number of half-spaces. The constraints are assumed to be provided in H-representation, short for half-space representation, as follows:

$$x(t) \in \mathcal{X} = \{x \in \mathbb{R}^n : H^x x \leq h^x\} \tag{3.2a}$$

$$u(t) \in \mathcal{U} = \{u \in \mathbb{R}^m : H^u u \leq h^u\}, \tag{3.2b}$$

where H^x, H^u are matrices and h^x, h^u are vectors. The sign $\leq$ should be interpreted element-wise. The size of the matrices and vectors in (3.2) depends on the number of half-spaces defining the polyhedron and should be compatible with the corresponding state or input vector. Similarly, the disturbance is assumed to be bounded by a polyhedron:

$$w(t) \in \mathcal{W} = \{w \in \mathbb{R}^p : H^w w \leq h^w\}. \tag{3.3}$$

Having presented the system dynamics and applicable constraints, the notion of robust controllable set is introduced in the following definition.

Definition 1: *For a system with dynamics* (3.1) *subject to* (3.2)-(3.3)*, the one-step robust controllable set to a goal set* $\mathcal{S} \subset \mathbb{R}^n$ *is defined as [7]:*

$$\text{Pre}(S) = \{x \in \mathcal{X} : \exists u \in \mathcal{U} \text{ s.t. } Ax + Bu + Ew \in \mathcal{S}, \forall w \in \mathcal{W}\}. \tag{3.4}$$

The notation $\text{Pre}(\cdot)$ is commonly used in the literature on backward reachable sets. It denotes the *predecessor operator* which returns a set containing all admissible states that can be robustly controlled into the goal set in one time step by means of an admissible input and for all possible disturbances.

In Paper A, the notion of *k-step controllable sets* is used. These sets are obtained by recursive application of the $\text{Pre}(\cdot)$ operator, which results in sets that contain admissible states which can be robustly controlled into the goal set in k steps or less. In Paper B the $\text{Pre}(\cdot)$ operator is treated in the context of interconnected LTI systems. The coming section describes a method to compute robust controllable sets.

Computation of robust controllable sets

For the class of systems described previously, controllable sets can be computed exactly. The method described next has been adapted from [7]. First the projection operation is introduced, which is used later in this section and discussed further in Section 3.3.

Definition 2: *For a set* $\mathcal{P} = \left\{[x^T \ u^T]^T : x \in \mathbb{R}^n, \ u \in \mathbb{R}^m\right\}$ *the projection onto the* x*-space is*

$$\mathrm{Proj}_x(\mathcal{P}) = \left\{x \in \mathbb{R}^n : \exists u \in \mathbb{R}^m \text{ s.t. } [x^T \ u^T]^T \in \mathcal{P}\right\}. \tag{3.5}$$

For a system (3.1) subject to (3.2)-(3.3) and a goal set $\mathcal{S} \subset \mathbb{R}^n$, let the set $\Phi \subset \mathbb{R}^{n+m}$ be

$$\Phi = \left\{\begin{bmatrix} x \\ u \end{bmatrix} \in \mathcal{X} \times \mathcal{U} : Ax + Bu + Ew \in \mathcal{S}, \ \forall w \in \mathcal{W}\right\}, \tag{3.6}$$

where $\mathcal{X} \times \mathcal{U}$ denotes the Cartesian product. The set Φ contains state-input pairs for which the system (3.1) evolves into the set $\mathcal{S}$ for all possible disturbances. The controllable set (3.4) can be obtained from the set Φ via projection: $\mathrm{Pre}(\mathcal{S}) = \mathrm{Proj}_x(\Phi)$.

When the sets $\mathcal{X}$, $\mathcal{U}$, $\mathcal{W}$ are $\mathcal{S}$ are given in H-representation, the set Φ in (3.6) can be expressed in H-representation too. Let the goal set $\mathcal{S}$ be

$$\mathcal{S} = \{x \in \mathbb{R}^n : H^s x \leq h^s\} \tag{3.7}$$

with matrix $H^s \in \mathbb{R}^{l \times n}$ and vector $h^s \in \mathbb{R}^l$, where $l \in \mathbb{N}$ is the number of half-spaces defining $\mathcal{S}$. An H-representation for Φ in (3.6) is

$$\Phi = \left\{\begin{bmatrix} x \\ u \end{bmatrix} \in \mathbb{R}^{n+m} : H^\Phi \begin{bmatrix} x \\ u \end{bmatrix} \leq h^\Phi\right\}, \tag{3.8}$$

with matrix H^Φ and vector h^Φ as follows:

$$H^\Phi = \begin{bmatrix} H^x & 0 \\ 0 & H^u \\ H^s A & H^s B \end{bmatrix}, \ h^\Phi = \begin{bmatrix} h^x \\ h^u \\ \tilde{h}^s \end{bmatrix}. \tag{3.9}$$

In (3.9) the zero entry denotes a matrix of zeroes with suitable dimensions, and $\tilde{h}^s \in \mathbb{R}^l$ is a vector whose i-th entry is computed as

$$\tilde{h}_i^s = \min_{w \in \mathcal{W}} h_i^s - H_i^s E w, \tag{3.10}$$

where h_i^s is the i-th element of h^s and H_i^s the i-th row of H^s. The purpose of $\tilde{h}^s$ is to account for the disturbance w by tightening the goal set $\mathcal{S}$.

Computing the robust controllable set $\text{Pre}(\mathcal{S})$ consists of two steps. First one obtains the H-representation of the set Φ, which requires linear programming to solve (3.10). Then one computes the projection $\text{Proj}_x(\Phi)$. Methods to compute projection of polyhedra are described in Section 3.3.

3.2 Robust control invariant sets

Invariant sets are a well-studied topic in the control community [8], [9]. This section focuses on robust control invariant (RCI) sets for the class of systems considered in Section 3.1. The main feature of RCI sets is their infinite-time constraint satisfaction property, which is stated formally in the following definition [7].

Definition 3: *For a system* (3.1) *subject to* (3.2)-(3.3)*, a robust control invariant set* Ω *is a set that satisfies* $x(t) \in \Omega \implies \exists u \in \mathcal{U}$ *such that* $x(t+1) \in \Omega, \forall w \in \mathcal{W}$ *and* $\forall t \in \mathbb{N}$.

For a set of admissible states $\mathcal{X}$, the maximal RCI set $\Omega_\infty \subseteq \mathcal{X}$ is the RCI set that contains all possible RCI sets $\Omega \subset \mathcal{X}$. Hence, Ω_∞ is the largest RCI set in $\mathcal{X}$. Algorithm 1, adapted from [7], describes a recursive routine to compute the set $\Omega_\infty \subseteq \mathcal{X}$ using the one-step robust controllable set.

Algorithm 1 Computation of the maximal robust control invariant set.

Input: sets $\mathcal{X}$, $\mathcal{U}$, $\mathcal{W}$ and dynamics (3.1)
Output: maximal RCI set $\Omega_\infty \subseteq \mathcal{X}$

```
Set Ω_0 ← X
Set Ω_1 ← Pre(Ω_0) ∩ Ω_0
Set k ← 1
while Ω_k ≠ Ω_{k−1} do
    Ω_{k+1} ← Pre(Ω_k) ∩ Ω_k
    k ← k + 1
end while
Set Ω_∞ ← Ω_k
```

For systems with linear dynamics and bounded sets $\mathcal{X}$, $\mathcal{U}$, $\mathcal{W}$, Algorithm 1 is guaranteed to converge to the maximal RCI set as $k \to \infty$ [10]. However,

finite-time termination of the algorithm is not guaranteed in general. It can be the case that Algorithm 1 returns an empty set Ω_∞, which indicates that the set $\mathcal{X}$ does not contain a RCI set.

Paper B discusses the computation of the maximal RCI set for the case of interconnected systems with LTI dynamics. The paper discusses how Algorithm 1 can be computed in a distributed fashion. A numerical example is provided there to illustrate the advantage of distributed computation over centralized computation.

3.3 Projection of polyhedra

As discussed in Section 3.1, the projection of polyhedra, also known as orthogonal projection, is an essential operation in the computation of controllable sets. A dedicated section on the projection operation is deemed necessary because it has been identified as the main bottleneck in the methods described in papers A and B. This section describes briefly some methods for projection and discusses which gave the best results in the context of computation of controllable sets.

Let the polyhedron $\mathcal{P} \subset \mathbb{R}^{n+m}$ be

$$\mathcal{P} = \left\{ \begin{bmatrix} x \\ y \end{bmatrix} \in \mathbb{R}^{n+m} : x \in \mathbb{R}^n,\ y \in \mathbb{R}^m \text{ s.t. } H \begin{bmatrix} x \\ y \end{bmatrix} \leq h \right\}. \tag{3.11}$$

The matrix $H \in \mathbb{R}^{l \times n}$ and vector $h \in \mathbb{R}^l$ specify a system of l linear inequalities. The H-representation of a polyhedron might contain redundant inequalities which are those that can be removed from the description without changing the polyhedron $\mathcal{P}$.

Turning now to the main topic of this section, the projection of a polyhedron $\mathcal{P}$ onto the x-space is defined as [7]:

$$\text{Proj}_x(\mathcal{P}) = \left\{ x \in \mathbb{R}^n : \exists y \in \mathbb{R}^m \text{ s.t. } [x^T\ y^T]^T \in \mathcal{P} \right\}. \tag{3.12}$$

For a polyhedron in H-representation, the projection operation is equivalent to eliminating variables from the system of inequalities. In (3.12), the last m variables are the ones eliminated, but any subset from the $n+m$ variables could be considered for elimination via projection. The projection operation has a simple geometric interpretation. Figure 3.1 illustrates the projection of a polyhedron in $\mathbb{R}^3$ onto a space in $\mathbb{R}^2$.

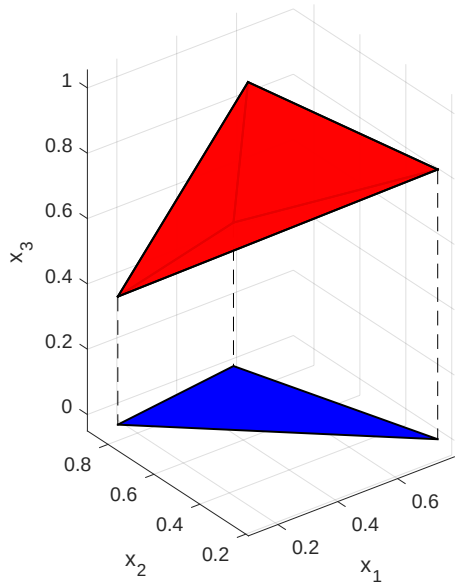

Figure 3.1: Geometric interpretation of the projection operation. A polyhedron $\mathcal{P} \subset \mathbb{R}^3$ described by 4 inequalities is shown in red. The projection of $\mathcal{P}$ onto the $x_1 - x_2$ space is the polyhedron show in blue.

Projection of polyhedra has been studied for several years and various methods have been developed for such purpose. The oldest method known is Fourier-Motzkin elimination [11]. It consists of eliminating one variable at a time from the system of inequalities. The method requires only algebraic manipulation of the inequalities. However, the method has, in the worst case, a double-exponential time-complexity dependent on the number of inequalities and the number of variables to eliminate [12]. Moreover, the method introduces several redundant inequalities which need to be identified and removed, a process that can also be computationally expensive. Hence, the application of Fourier-Motzkin elimination is limited to polyhedra in lower dimensions described by a small number of inequalities.

Another approach to project a polyhedron is based on vertex enumeration, that is computing all the vertices $v \in \mathbb{R}^{n+m}$ on the border of $\mathcal{P}$. The projection of a point $v \in \mathbb{R}^{n+m}$ onto the space $\mathbb{R}^n$ is simply given by the first n components of v (Fig. 3.1 can help to clarify this concept). Hence, the projection of $\mathcal{P}$ can

be easily determined from its V-representation but the projection will also be given in a V-representation. An additional procedure is needed to convert to H-representation. Algorithms for conversion between H and V representations and vice-versa have an exponential complexity [7], [13]. Hence, projection via vertex computation is only suitable for polyhedra in lower dimensions having a small number of vertices.

In [14] the authors show that the projection of a polyhedron can be obtained from the solution of a parametric linear programming (PLP) problem. The advantage of their method is that it computes the projection without redundant inequalities. Other projection methods have been reported in the literature. For further details, the reader is referred to [12], [14], [15].

The Multi-Parametric Toolbox for MATLAB [16] provides functions for projection based on the methods described before. The toolbox was used for the computations involved in the numerical examples in papers A and B, where polyhedra in dimensions ranging from 6 to 24 were projected onto lower dimensions. Fourier-Motzkin elimination had an acceptable performance when projecting polyhedra described with few inequalities, e.g. up to a hundred. However, the method based on PLP proved to be faster when dealing with polyhedra in higher dimensions with hundreds of facets.

The methods proposed in papers A and B, which require projecting polyhedra, are intended for off-line use. Nevertheless, the computational complexity of the projection operation needs to be addressed before the methods can be applied to larger problems.

CHAPTER 4

Model Predictive Control

This chapter presents fundamental concepts on Constrained Optimal Receding Horizon Control, also known as Model Predictive Control (MPC). MPC is a well-established control technique based on solving online an optimization problem with a receding prediction horizon. MPC is ideal for control tasks that require constraint satisfaction. The purpose of this chapter is to complement the description of the control approach presented in Paper A, where MPC is used for control of test-scenarios as those described in Chapter 2.

4.1 MPC for linear systems

As indicated in previous chapters, the scope of this thesis is limited to systems with LTI dynamics subject to linear, convex constraints and additive disturbances. MPC is discussed here in the context of systems with the following dynamics:

$$x(t+1) = Ax(t) + Bu(t) + Ew(t) \tag{4.1}$$

with state and input constraints $x(t) \in \mathcal{X} \subset \mathbb{R}^n$, $u(t) \in \mathcal{U} \subset \mathbb{R}^m$; a bounded disturbance $w(t) \in \mathcal{W} \subset \mathbb{R}^p$; and sampling instant $t \in \mathbb{N}$. Sets $\mathcal{X}, \mathcal{U}$, and $\mathcal{W}$ are

assumed to be polyhedral. When MPC is applied to control the system (4.1), the input $u(t)$ is determined by solving a constrained optimization problem with a finite prediction horizon of length $N \in \mathbb{N}$. The optimization problem is as follows [7]:

$$\min_{u_0,\cdots,u_{N-1}} J_0(x_0,\cdots,x_N,u_0,\cdots,u_{N-1}) \tag{4.2a}$$

$$\text{subject to } x_{k+1} = Ax_k + Bu_k + Ew_k,\ k = 0,\cdots,N-1 \tag{4.2b}$$

$$x_k \in \mathcal{X},\ u_k \in \mathcal{U},\ k = 0,\cdots,N-1 \tag{4.2c}$$

$$x_N \in \mathcal{X}_f \tag{4.2d}$$

$$x_0 = x(t) \tag{4.2e}$$

$$w_k = \bar{w}(t+k)\ k = 0,\cdots,N-1. \tag{4.2f}$$

The expressions in (4.2) are clarified in the following paragraphs. First, an important distinction is made between predicted state and the current state of the system. Following the system dynamics in (4.1), $x(t)$ denotes the state of the system at sampling instant t. On the other hand, in (4.2) x_k is a variable in the optimization problem that corresponds to the k-th predicted state of the system. Predictions of x_k for $k = 0,\cdots,N-1$ are obtained via the system dynamics (4.2b), starting from $x_0 = x(t)$, applying the input sequence $u_0,\cdots,u_{N-1}$ and predicted disturbances $w_0,\cdots,w_{N-1}$. The variables $u_0,\cdots,u_{N-1}$ are optimization variables. The state $x(t)$ is assumed to be known at sampling instant t.

The function J_0 in (4.2a) is the cost to be optimized. A convex function is assumed for J_0 which can be a linear or quadratic function of $x_0,\cdots,x_N$ and $u_0,\cdots,u_{N-1}$. With such a choice of cost function, (4.2) is a linear or quadratic optimization problem that can be solved with standard algorithms for convex optimization [17]. The function J_0 is typically designed to penalize deviations of the state and input from a desired value or reference trajectory.

State and input constraints are specified in (4.2c). The constraint (4.2d) requires the last predicted state x_N to belong to a terminal set $\mathcal{X}_f$. As discussed later in this chapter, the terminal set $\mathcal{X}_f$ can have properties that contribute to the feasibility of problem (4.2).

According to (4.2f), the predicted disturbance w_k is obtained from $\bar{w}(t+k)$, the latter being a nominal disturbance. It is therefore assumed that a nominal disturbance sequence $\bar{w}(t),\cdots,\bar{w}(t+N-1) \in \mathcal{W}$ is available at every sampling instant $t \in \mathbb{N}$. It could be the case that the actual disturbance

$w(t)$ does not match the predicted value, i.e. $w(t) \neq w_0$. This will cause a mismatch between the predicted state x_1 and the actual state $x(t+1)$, thus compromising constraint satisfaction. As described later in this chapter, an additional constraint can be introduced to address this issue.

In receding horizon control, the optimization problem (4.2) is solved at every sampling instant t, which gives an optimal sequence of inputs: $u_0^*, \cdots, u_{N-1}^*$. The control input applied at instant t is $u(t) = u_0^*$ and the input is held constant until the next sampling instant. At sampling instant $t+1$, the optimization problem is solved again using a measurement of the state $x(t+1)$ and shifting the prediction horizon.

The MPC scheme described previously is commonly referred to as robust MPC with open-loop predictions and nominal cost. Other approaches for robust MPC have been proposed which may provide better performance but result in more complex optimization problems (see [7] for details).

The stability of MPC is a well-studied topic [18]. Roughly speaking, stability refers to asymptotically controlling the state of the system towards an equilibrium point. In Paper A the control problem is not to stabilize the system towards an equilibrium point, but rather to control the state to enter a goal set in a finite number of steps. Hence, a discussion on stability is not deemed necessary in this thesis. Instead, an aspect of MPC that is more relevant for the problem in Paper A is discussed next.

Feasibility of MPC

For the optimization problem in (4.2), feasibility refers to the existence of an input sequence $u_0, \cdots, u_{N-1}$ that allows to satisfy the specified state and input constraints. If at some instant t the problem is infeasible, then the MPC fails to assign an input $u(t)$. It is therefore important to investigate for which states the optimization problem is feasible.

For the problem (4.2) to be feasible at sampling instant t, the state $x(t)$ should belong to a set of feasible states $\mathcal{X}_0$ having the following properties:

$$\begin{aligned} \mathcal{X}_0 = \{ & x_0 \in \mathbb{R}^n : \exists u_0, \cdots, u_{N-1} \in \mathcal{U} \text{ s.t. } \forall w_0, \cdots, w_{N-1} \in \mathcal{W}, \\ & x_k \in \mathcal{X} \text{ for } k = 0, \cdots, N-1, \; x_N \in \mathcal{X}_f, \\ & \text{where } x_{k+1} = Ax_k + Bu_k + Ew_k \}. \end{aligned} \tag{4.3}$$

Expression (4.3) was borrowed from [7] and modified to account for the disturbance. The states in $\mathcal{X}_0$ can be controlled into $\mathcal{X}_f$ in N steps by means of

an admissible sequence of inputs, while satisfying state constraints, and for all possible disturbance sequences. The properties of $\mathcal{X}_0$ are similar to those of robust controllable sets discussed in Chapter 3. In fact, the set $\mathcal{X}_0$ corresponds to $\mathcal{K}_N(\mathcal{X}_f)$, the N-step robust controllable set with $\mathcal{X}_f$ as goal set. The set $\mathcal{K}_N(\mathcal{X}_f)$ is defined recursively as follows [7]:

$$\mathcal{K}_i(\mathcal{X}_f) = \mathrm{Pre}\left(\mathcal{K}_{i-1}(\mathcal{X}_f)\right), \ i = 1, \cdots, N \tag{4.4}$$

$$\mathcal{K}_0(\mathcal{X}_f) = \mathcal{X}_f \tag{4.5}$$

with $\mathrm{Pre}(\cdot)$ the predecessor operator defined in (3.4).

At sampling instant t, problem (4.2) is feasible if the state satisfies $x(t) \in \mathcal{X}_0 = \mathcal{K}_N(\mathcal{X}_f)$. If the problem should remain feasible at the next instant $t+1$ and for all future instants, then a robust control invariant set $\mathcal{X}_f$ is needed [7]. Feasibility of the optimization problem for arbitrarily many sampling instants is not required for the application discussed in Paper A. The next section describes a finite-time MPC scheme that is the basis for the control approach proposed in Paper A.

MPC for finite-time control into a goal set

When solving (4.2) in a receding horizon fashion, a fixed prediction horizon of length N is used at every iteration. On the other hand, this section describes a MPC scheme with a varying horizon where the control objective is to enter a goal set in a finite number of steps rather than stabilization around an equilibrium. Paper A deals with such a kind of control objective. A control approach that is similar to the one described here has been proposed in [19].

Robust controllable sets $\mathcal{K}_i(\mathcal{X}_f)$ as defined in (4.4) are assumed to be known (for example, computed off-line as described in Chapter 3). The dependency of the sets $\mathcal{K}_i$ on $\mathcal{X}_f$ is omitted in the sequel for the sake of brevity. Assuming an initial condition $x(t_0) \in \mathcal{K}_N$, the control objective is to achieve $x(t_0+N) \in \mathcal{X}_f$. To this end, the following optimization problem with varying horizon length $N_i = N-(t-t_0)$ should be solved at sampling instants $t = t_0, \ t_0+1, \ \cdots, \ t_0+N-1$:

$$\min_{u_0,\cdots,u_{N_i-1}} J_0(x_0,\cdots,x_{N_i},u_0,\cdots,u_{N_i-1}) \tag{4.6a}$$

$$\text{subject to } x_{k+1} = Ax_k + Bu_k + Ew_k,\ k = 0,\cdots,N_i-1 \tag{4.6b}$$

$$x_k \in \mathcal{X},\ u_k \in \mathcal{U},\ k = 0,\cdots,N_i-1 \tag{4.6c}$$

$$x_1 \in \mathcal{K}_{N_i-1} \tag{4.6d}$$

$$x_{N_i} \in \mathcal{X}_f \tag{4.6e}$$

$$x_0 = x(t) \tag{4.6f}$$

$$w_k = \bar{w}(t+k)\ k = 0,\cdots,N_i-1. \tag{4.6g}$$

The problem (4.6) has a prediction horizon that decreases by one for every subsequent sampling instant. In other words, at $t = t_0$ the horizon has length N, then at $t = t_0 + 1$ the horizon length is $N - 1$, and so on.

Constraint (4.6d) requires the predicted state x_1 to belong to the next robust controllable set in the sequence. The purpose of (4.6d) is to make the state evolve as $x(t_0+1) \in \mathcal{K}_{N-1}$, $x(t_0+2) \in \mathcal{K}_{N-2}$, $\cdots$, $x(t_0+N) \in \mathcal{K}_0$. This concept is illustrated in Figure 4.1.

In (4.6) a nominal disturbance $\bar{w}(t)$ is used to generate disturbance predictions w_k. In Paper A, the disturbance is assumed to be measurable so that $w(t)$ is known at instant t, but future values remain a priori unknown, so the measured disturbance $w(t)$ is used to generate the nominal disturbance sequence as $\bar{w}(t),\cdots,\bar{w}(t+N_i-1) = w(t)$.

It could be the case that the actual disturbance $w(t)$ does not match the predicted value, i.e. $w(t) \neq w_0$. This can happen in particular when the disturbance $w(t)$ cannot be measured accurately. When $w(t) \neq w_0$, the predicted state x_1 will not match the true state $x(t+1)$. In that case, constraints might be violated. For instance $x(t+1) \notin \mathcal{K}_{N_i-1}$ might happen, which would compromise the feasibility of the optimization problem at $t+1$. To address this issue, (4.6d) can be replaced with the following constraint:

$$\begin{bmatrix} x_0 \\ u_0 \end{bmatrix} \in \Phi_{N_i} \tag{4.7}$$

$$\text{with } \Phi_{N_i} = \left\{ \begin{bmatrix} x \\ u \end{bmatrix} \in \mathcal{X} \times \mathcal{U} : Ax + Bu + Ew \in \mathcal{K}_{N_i-1}(\mathcal{X}_f),\ \forall w \in \mathcal{W} \right\}. \tag{4.8}$$

If (4.7) is satisfied, then $x_1 = Ax_0 + Bu_0 + Ew_0 \in \mathcal{K}_{N_i-1}$ holds for any predicted disturbance $w_0 \in \mathcal{W}$. In turn, the next state will satisfy $x(t+1) = Ax(t) +$

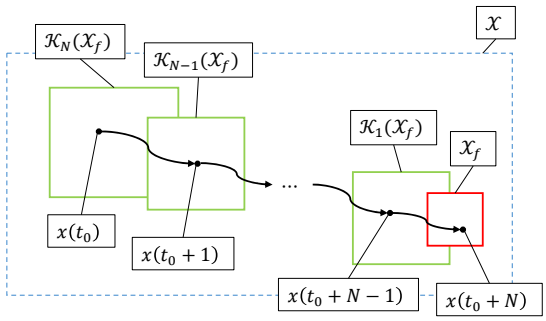

Figure 4.1: Schematic of a finite-time MPC scheme for controlling the state x into a goal set $\mathcal{X}_f$ in N steps. The state is controlled through a sequence of robust controllable sets $\mathcal{K}_N$, $\cdots$, $\mathcal{K}_1$.

$Bu(t) + Ew(t) \in \mathcal{K}_{N_i-1}$, as long as $w(t) \in \mathcal{W}$ and even if $w(t) \neq w_0$. To arrive at this claim, the fact that $x(t) = x_0$ and $u(t) = u_0^*$ were used.

Recall that an initial condition $x(t_0) \in \mathcal{K}_N$ is assumed. The feasibility of (4.6) at $t = t_0$ is not affected by introducing constraint (4.7). The previous claim follows from the fact that $\mathcal{K}_{N_i}$ is the projection of Φ_{N_i} onto the x-space (see Section 3.1 for details on projection). Solving problem (4.6) at instant t_0 and applying $u(t_0) = u_0^*$ achieves $x(t_0 + 1) \in \mathcal{K}_{N-1}$. This guarantees feasibility of the optimization problem at $t_0 + 1$.

The problem discussed in Paper A, stated also in Chapter 2, is to control the state of the system through a sequence of goal sets. The paper proposes a control strategy where a sequence of problems of the form (4.6) is solved. Results from a simulation example are discussed in Paper A to illustrate the robustness of the control strategy and the constraint-satisfaction properties.

CHAPTER 5

Summary of included papers

This chapter provides a summary of the included papers.

5.1 Paper A

Angel Molina Acosta, Paolo Falcone
Modelling and Control of Test-Scenarios for Validation of Autonomous Driving Functions
Published in *Proceedings of the American Control Conference*, pp. 2943-2948, May 2021.
 DOI: 10.23919/ACC50511.2021.9483214 .

This paper proposes a framework for modelling, analysis, and control suitable for leader-follower, multi-agent systems where the motion of the followers should be coordinated with the motion of an externally controlled leader. The motivation behind the framework is to conduct experimental validation of an autonomous car by exposing it to a traffic scenario at a test track, where mobile test-targets represent other road users.
The paper proposes a modelling approach based on a class of switched sys-

tems to represent a scenario consisting of a fixed sequence of stages. The feasibility of the scenario is investigated using backward reachability analysis to determine sets of feasible states. A constrained optimal control problem is formulated to control the state of the multi-agent system through a sequence of goal sets. Simulation results are provided for a scenario where two test-targets are coordinated around a vehicle-under-test that attempts to conduct an overtake.
I was responsible for developing the framework, conducting the numerical simulations, and writing the paper.

5.2 Paper B

Davide Liuzza, **Angel Molina Acosta**, Paolo Falcone, Luigi Glielmo
Distributed control invariant set: the linear affine case
In preparation for submission to *Automatica* .

This paper proposes a novel method for distributed computation of control invariant sets for large-scale systems. The paper considers a network of subsystems with linear affine dynamics. The coupling among the subsystems is due to dynamic dependencies as well as static constraints over their states and inputs. Linear constraints and bounded additive disturbances are considered. The advantages of the distributed method are more evident for the case of sparsely connected networks.
The paper describes an iterative procedure to compute the one-step backward reachable set in a distributed fashion, where the subsystems exchange with their neighbors information about local dynamics and constraints. The method provides local backward reachable sets which contain the minimum amount of information required to reconstruct the reachable set for the centralized system. The maximal robust control invariant set is obtained in a distributed fashion by recursive computation of distributed reachable sets.
A numerical example is provided to illustrate the advantages of the proposed method.
Regarding the content of this paper, I was responsible for designing the numerical example and implementing the algorithms to produce the numerical results.

CHAPTER 6

Concluding Remarks and Future Work

This thesis has provided contributions on two topics. The first contribution is a framework for modelling, analysis, and control of test-scenarios for validation of autonomous cars. The framework provides concrete answers to the research questions posed in Chapter 1. A key feature of the framework is the robustness it provides to an uncertain motion of the autonomous vehicle-under-test. Although it was conceived for validation of autonomous cars, the framework can be applied to other problems dealing with robust control of multi-agent systems.

The second contribution is a novel method for distributed computation of backward reachable sets for interconnected systems. The method is very general and can thus find applications in various fields. Moreover, distributed computation of reachable sets can make the proposed scenario framework more efficient. Further investigation is needed to identify for what class of test-scenarios a distributed computation of reachable sets would be beneficial.

The paragraphs below discuss limitations in the methods proposed here. Suggestions for future work are also provided.

The modelling approach described in Paper A assumes that the motion of the scenario participants can be described with linear dynamics. For scenarios

where the participants move along fixed paths, a lower-level path-following controller can be assumed to guide the motion of the participants. In that case, linear dynamics might suffice to describe the displacement along the paths and the framework proposed here can be used as a higher control layer. However, linear models might not suffice to describe more general cases where no fixed paths are given. Introducing nonlinear dynamics to the framework would require complex methods for backward reachability analysis that scale poorly with the dimension of the system [20]. It is preferable to restrict the framework to linear models and consider alternatives to account for possible nonlinear dynamics.

As discussed in Chapter 2 and Paper A, the uncertain motion of the vehicle-under-test can be described as a disturbance in the system. The feasibility analysis based on backward reachability is conducted without restrictions on how fast the disturbance can change between sampling instants. However, the disturbance is associated to variables such as the VUT's velocity which has in reality a limited rate of change. Hence, the feasibility analysis is conservative. Backward reachable sets for systems with bounds on the rate of change of the disturbance have been discussed in [21]. Their methods are less conservative but come with higher computational complexity. Further investigation is needed to evaluate the advantage of considering a bound on the rate of change of the disturbance in the reachability analysis.

Turning now to backward reachability, it has been discussed in Chapter 2 that the projection operation is the main source of complexity in the computation of reachable sets. Both the framework in Paper A and the methods in Paper B are affected by the complexity of polyhedral projection. A more efficient method for polyhedral projection is needed. A potential solution is to approximate the projection operation with simple polyhedra, as suggested in [15]. Projection of polyhedra in high dimensions remains a difficult problem that requires further investigation.

References

[1] European New Car Assessment Protocol. "Test Protocol - AEB Systems." (2017), [Online]. Available: https://cdn.euroncap.com/media/32278/euro-ncap-aeb-c2c-test-protocol-v201.pdf.

[2] A. Knauss, C. Berger, H. Eriksson, N. Lundin, and J. Schröder, "Proving Ground Support for Automation of Testing of Active Safety Systems and Automated Vehicles," in *Fourth Internetional Symposium on Future Active Safety Technology Toward zero traffic accidents (FASTzero) 2017*, 2017.

[3] R. Schram, "Euro NCAP'S First Step Towards Scenario-Based Assessment by Combining Autonomous Emergency Braking and Autonomous Emergency Steering," in *26th International Technical Conference and Exhibition on the Enhanced Safety of Vehicles*, Eindhoven, 2019.

[4] H. P. Schöner and W. Hurich, "Testing with coordinated automated vehicles," in *Handbook of Driver Assistance Systems: Basic Information, Components and Systems for Active Safety and Comfort*, 2015, pp. 261–276.

[5] Vinnova. "Chronos part 2." (2021), [Online]. Available: https://www.vinnova.se/en/p/chronos-part-2/.

[6] R. Goebel, R. G. Sanfelice, and A. R. Teel, "Hybrid Dynamical Systems: Robust stability and control for systems that combine continuous-time and discrete-time dynamics," *IEEE Control Systems Magazine*, vol. 29, no. 2, pp. 28–93, 2009.

[7] F. Borrelli, A. Bemporad, and M. Morari, *Predictive Control for Linear and Hybrid Systems*. UK: Cambridge University Press, 2015.

[8] F. Blanchini, "Set invariance in control," *Automatica*, vol. 35, no. 11, pp. 1747–1767, 1999.

[9] E. C. Kerrigan, "Robust Constraint Satisfaction: Invariant Sets and Predictive Control," Ph.D. dissertation, University of Cambridge, 2000.

[10] D. P. Bertsekas, "Infinite-Time Reachability of State-Space Regions by Using Feedback Control," *IEEE Transactions on Automatic Control*, vol. 17, no. 5, pp. 604–613, 1972.

[11] G. B. Dantzig and B. Curtis Eaves, "Fourier-Motzkin elimination and its dual," *Journal of Combinatorial Theory, Series A*, vol. 14, no. 3, pp. 288–297, 1973.

[12] C. N. Jones, E. C. Kerrigan, and J. M. Maciejowski, "Equality Set Projection : A new algorithm for the projection of polytopes in halfspace representation," Cambridge University, Tech. Rep., 2004.

[13] M. E. Dyer, "Complexity of Vertex Enumeration Methods," *Mathematics of Operations Research*, vol. 8, no. 3, pp. 381–402, 1983.

[14] C. N. Jones, E. C. Kerrigan, and J. M. Maciejowski, "On polyhedral projection and parametric programming," *Journal of Optimization Theory and Applications*, vol. 138, no. 2, pp. 207–220, Aug. 2008.

[15] T. Huynh, C. Lassez, and J. L. Lassez, "Practical issues on the projection of polyhedral sets," *Annals of Mathematics and Artificial Intelligence*, vol. 6, no. 4, pp. 295–315, 1992.

[16] M. Herceg, M. Kvasnica, C. N. Jones, and M. Morari, "Multi-parametric toolbox 3.0," in *2013 European Control Conference, ECC 2013*, IEEE Computer Society, 2013, pp. 502–510.

[17] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge: Cambridge University Press, 2004.

[18] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. M. Scokaert, "Constrained model predictive control: Stability and optimality," *Automatica*, vol. 36, no. 6, pp. 789–814, 2000.

[19] D. Limon, T. Alamo, and E. F. Camacho, "Robust MPC control based on a contractive sequence of sets," in *Proceedings of the IEEE Conference on Decision and Control*, vol. 4, 2003, pp. 3706–3711.

[20] S. Bansal, M. Chen, S. Herbert, and C. J. Tomlin, "Hamilton-Jacobi reachability: A brief overview and recent advances," in *2017 IEEE 56th Annual Conference on Decision and Control*, 2017, pp. 2242–2253.

[21] S. V. Raković, E. C. Kerrigan, D. Q. Mayne, and J. Lygeros, "Reachability analysis of discrete-time systems with disturbances," *IEEE Transactions on Automatic Control*, vol. 51, no. 4, pp. 546–561, Apr. 2006.