Leonhard Asselborn

Control of Stochastic Hybrid Systems based on Probabilistic Reachable Set Computation

kassel
university
press

Leonhard Asselborn

Control of Stochastic Hybrid Systems based on Probabilistic Reachable Set Computation

kassel
university
press

This work has been accepted by the Faculty of Electrical Engineering / Computer Science of the University of Kassel as a thesis for acquiring the academic degree of Doktor der Ingenieurwissenschaften (Dr.Ing.).

Supervisor: Prof. Dr.-Ing.Olaf Stursberg
Co-Supervisor: Prof. Dr. Erika Abraham
Defense day: 24th May 2018

Bibliographic information published by Deutsche Nationalbibliothek
The Deutsche Nationalbibliothek lists this publication in the Deutsche Nationalbibliografie; detailed bibliographic data is available in the Internet at http://dnb.dnb.de.

Zugl.: Kassel, Univ., Diss. 2018
ISBN 978-3-7376-0580-9 (print)
ISBN 978-3-7376-0581-0 (e-book)
DOI: http://dx.medra.org/10.19211/KUP9783737605810
URN: http://nbn-resolving.de/urn:nbn:de:0002-405819

Printed in Germany

Contents

Summary

This thesis proposes an algorithmic controller synthesis based on the computation of probabilistic reachable sets for stochastic hybrid systems. Hybrid systems consist in general of a composition of discrete and continuous valued dynamics, and are able to capture a wide range of physical phenomena. The stochasticity is considered in form of normally distributed initial continuous states and normally distributed disturbances, resulting in stochastic hybrid systems.

The reachable sets describe all states, which are reachable by a system for a given initialization of the system state, inputs, disturbances, and time horizon. For stochastic hybrid systems, these sets are probabilistic, since the system state and disturbance are random variables. This thesis introduces probabilistic reachable sets with a predefined confidence, which are used in an optimization based procedure for the determination of stabilizing control inputs. Besides the stabilizing property, the controlled dynamics also observes input constraints, as well as, so-called chance constraints for the continuous state.

The main contribution of this thesis is the formulation of an algorithmic control procedure for each considerd type of stochastic hybrid systems, where different discrete dynamics are considered. First, a control procedure for a deterministic system with bounded disturbances is introduced, and thereafter a probabilistic distribution of the system state and the disturbance is assumed. The formulation of probabilistic reachable sets with a predefined confidence is subsequently used in a control procedure for a stochastic hybrid system, in which the switch of the continuous dynamics is externally induced. Finally, the control procedure based on reachable set computation is extended to a type of stochastic hybrid systems with autonomously switching of the continuous dynamics.

Acknowledgments

First and foremost, enormous gratitude due to Prof. Dr.-Ing. Olaf Stursberg who has been my supervisor over the past years, and gave me the freedom, trust, and guidance to write this thesis. I also would like to thank the complete team of the Institute of Control and System Theory. I will miss the pleasant time with you guys. Moreover, I would like to thank Prof. Dr. Erika Abraham for her encouraging words, and for her willingness to examine my thesis. Finally, I would like to thank my family and Tatjana for their belief, love, and enduring support during the past years.

This thesis is dedicated to my son, Theodor.

1 Introduction

In system theory, the behavior of physical phenomena is captured by mathematical equations, which mostly describe the change of system variables in time. The continuous-valued system variable x represents the time-dependent value of the physical process and is used to describe the physical dependencies by mathematical equations. The continuous change of the system variable can be modeled by equations known as *ordinary differential equations* (ODE), and many real world processes can be described with ODE's. A very general and basic formulation of an ODE to describe the behavior of such a system is:

$$\dot{x}(t) = f(x(t), u(t)), \tag{1.1}$$

where $u(t)$ is the time-dependent continuous valued input, which is used to capture some externally induced power into the system, e.g. the change of input voltage in an electrical network. In (1.1) the current change of the continuous system state $x(t)$ is determined by a non-specific function $f(\cdot, \cdot)$, with the current state $x(t)$ itself and the continuous input $u(t)$ as parameters.

Besides the modeling of processes with ODE's and continuous state variables, there are many processes which cannot be captured by ODE's. These systems involve the execution of a transition which changes the value of a discrete-valued system mode z. Discrete-event systems can be used, for example to model the behavior of manufacturing systems, in which the dynamics are determined by discrete events, like the arrival of a work piece at a work station, which triggers the processing of the work piece, hence the discrete dynamics for the discrete mode z is not time-dependent, but event-triggered.

Many real world phenomena can be characterized by a combination of discrete- and continuous-valued processes, e.g. a bouncing ball or a power train of an automobile. The behavior of the bouncing ball in the free-fall phase can be modeled by a set of ODE's. However, the hit with the ground and the resulting change in the dynamics of the ball cannot be described by the same set of ODE's. The hit is a discrete event and triggers a change in the discrete system variable z, which leads to a switch in the continuous dynamics instantly. The switch results in a discontinuity of the continuous system variables, and the classical (non-)linear systems are not even capable of describing such discontinuity with ODE's, which motivates the extension to the system class known as *hybrid systems* (HS). In general, a HS models the interaction of a time driven process (solution of the ODE) and an event driven process (hit with the ground). The discrete events cause either a change of the continuous dynamics, i.e. after the discrete event the time driven process is determined by a different set of ODE's, or a discontinuity in the variables of the time driven process, or even both. This flexibility enables HS to capture many characteristics of real systems, such as a power train. With the evolution of time, variables like angular velocity and drive torque continuously vary in the power train. The gear changes in the transmission represent discrete events, which change the dynamic behavior of the power train and the available power. A mode of the HS is defined by each gear of the power train, and each mode is described by different continuous dynamics. However, while (1.1) describes an ODE of a continuous-time system, the considered systems in this thesis are discrete-time systems, resulting in a difference equation as follows:

$$x_{k+1} = f(x_k, u_k, z_k), \tag{1.2}$$

where the discrete-time variable $k \in \mathbb{N}_0$ is used in the subscript of the variables, and the value of the next state is computed by the evaluation of $f(\cdot, \cdot, \cdot)$. Note, that (1.2) is now a difference equation, and the right side of (1.2) requires three functionals, since the continuous dynamics are effected by the discrete dynamics of z_k. The consideration of discrete-time systems is reasonable, since the solution of continuous-time systems is in general computed with numerical methods. The base of these methods is a discretization of the time to approximate the solution of an ODE numerically.

In general, the mathematical modeling of physical processes in control theory serves for controller synthesis, and the objective is to modify the behavior of the process, such that a desired output is obtained. The controller synthesis for non-linear systems, as in (1.1), is challenging, due to the arbitrary specification of the nonlinear function. The consideration of an additional discrete mode z_k makes the controller synthesis even more challenging, and is thus a recent topic of interest in the field of control theory.

The two examples bouncing ball and power train of an automobile illustrate, that the discrete event may be externally induced (by the gear change) or internally triggered (by the hit of the ball with the ground). This motivates a separation of HS into switched systems (external) and switching systems (internal).

The exact modeling of a real world process is rarely possible and it is desirable to introduce uncertainties v_k to capture all relevant effects in the HS. The uncertainties can affect different properties of the HS, like the discrete event, parameters in the ODE's, disturbances, and so on, resulting in a non-deterministic behavior of the HS. Under the influence of uncertainties, the control of HS becomes even more challenging, and is addressed in this thesis.

The uncertainty is often assumed to be bounded ($v_k \in V$), in order to be able to provide a robust controller, which satisfies certain properties for the model. In other words, the investigation of non-deterministic HS enables to decide whether or not the HS satisfies a certain property under consideration of the uncertainty. In some applications, the considered disturbances are not bounded but stochastic distributed, which elevates HS to the system class of *stochastic hybrid system*(SHS).

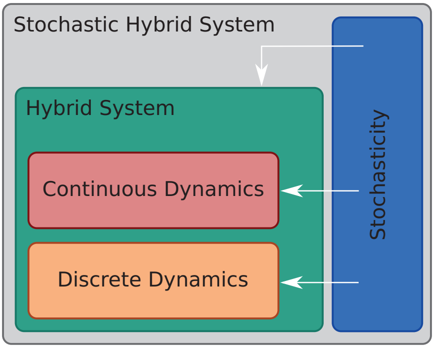


Figure 1.1: Components of stochastic hybrid systems: Stochastic hybrid system comprise a continuous and discrete system together with probabilistically modeled effects.

Stochastic hybrid systems are a very powerful modeling tool to capture an even wider class of physical phenomena than HS. The stochasticity can occur at different parts of the HS, e.g. the transition between two discrete-valued states might occur spontaneous at random points in time or the transition function itself is a probabilistic function such that the next discrete-valued state is given according to a probabilistic distribution. A graphical illustration of the composition of SHS is given in Fig. 1.1. Stochasticity (white arrows in Fig. 1.1) may also be present in the continuous dynamics of the HS, which typically results in a stochastic differential equation. An additive disturbance modeled as Gaussian noise can also add uncertainty in the continuous dynamics. The variety of stochastic effects in SHS results in countless variants of sub-classes, and the categorization follows from the considered random effects. For instance the sub-class piecewise deterministic Markov processes is governed by complete deterministic continuous dynamics and the transition of the discrete state is assumed to follow a Markov process. In [46], a rough overview of the sub-classes considered in literature is provided by Table 1.1. The price to pay for the wide applicability of SHS is, that the controller synthesis of SHS is, in general, much more difficult compared to the deterministic variant. Even for non-deterministic HS the controller design is easier than for SHS, due to the bounded uncertainties. Besides the topics of interest like stability and performance for non-hybrid systems, probabilistic reachability and safety verification is of interest for SHS.

In general, a reachable set of a dynamic system consists of all the states that can be reached by a trajectory of the dynamic system starting in a predefined set of initial states. The aim of safety verification is to show that starting from a set of initial conditions, the system will not enter an unsafe region, which is illustrated in Figure 1.2. It has to be verified, that the trajectories, starting from the initial set, and described by the reachable set do not enter the unsafe region, i.e. the reachable set must not intersect with the unsafe set. In a stochastic setting, the probabilistic safety verification provides a probabilistic estimation of the examined property, such that a value for the likelihood for safety can be provided.

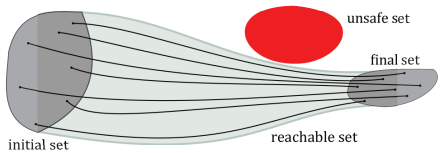


Figure 1.2: Safety Verification: The reachable sets of an SHS must not intersect with an unsafe region.

Reachability analysis is a key step of safety verification, but in addition, reachability analysis can be used for control and performance studies. It aims at showing that a target region is reached, when starting from a given set of initial conditions under the effect of all available control inputs. This thesis is concerned with the controller synthesis based on the computation of the reachable sets, in which the system dynamics are modified such that a predefined region is reached from an initial set of states. This as shown in Figure 1.3, where the desired terminal set is reached by all trajectories starting in the initial set. The above Figures 1.2 and

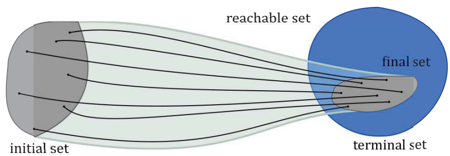


Figure 1.3: The reachable set computation can be used to find a feasible set of inputs to transfer all trajectories from the initial set into a terminal set.

1.3 do not show the stochastic setting of the considered problem, and therefore, only the qualitative statement “yes” or “no” is possible for the shown examples. Either the target region is reached with the available inputs, or not. The stochastic version of the example in Figure 1.3 is shown in 1.4, and it includes a probabilistic distribution over the continuous-state space, which is illustrated by different colors for different confidence levels δ. A desired high value for δ leads to an enlargement of the confidence reachable sets, and consequently the controller synthesis is made more difficult.

To be able to perform mathematical operations, needed for controller design, the exact reachable sets for dynamic systems are over-approximated by mathematical objects, such as polytopes, zonotopes, support functions, or ellipsoids. Each representation has its assets and drawbacks, which will be discussed later in the review of the literature.

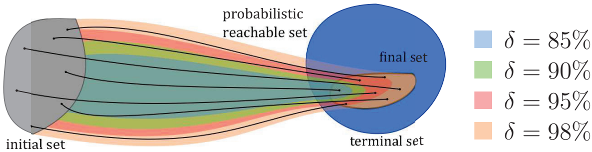


Figure 1.4: A probabilistic distribution leads to reachable sets with a certain confidence δ.

As pointed out, the system class of SHS is very versatile, and is able to describe a wide range of applications. However, the control of SHS is still challenging, due to the combination of continuous states and discrete modes, including probabilistic distributions. This thesis proposes a novel control approach based on the computation of probabilistic reachable sets. The fundamental objective of a control strategy is, in general, to achieve a desired and predefined behavior for the physical process. The uncertainty and probabilistic distribution of the system state in SHS motivates a set-valued interpretation of this control problem, which is introduced as a *set-to-target control problem*. This thesis formulates set-to-target control problems for different sub-classes of SHS, in which an initial set of states has to be transferred into a target set. The proposed algorithmic control procedure formulates an optimization problem in each discrete time step and the feasible solution provides a stabilizing control law for set-to-set transitions of the SHS. In addition to the stability criterion, the control law observes input constraints, and takes into account stochastic disturbances for a predefined confidence level.

Outline of the Dissertation

The remaining thesis continues in the next chapter with a review of the literature on reachability analysis for SHS and the considered approaches for control purposes. A classification of the thesis in the existing literature and the overall goals are presented in Chapter 2.

Chapter 3 introduces some general mathematical notations used in this thesis, as well as a general formulation of an SHS. Furthermore, multivariate distributions, set representations, and a short introduction into semi-definite programming (SDP) is given in this chapter.

Chapter 4 states a reachability problem for discrete-time affine systems (AS) with bounded disturbances, in which the ellipsoidal calculus is used to robustly over-approximate the reachable sets. The stabilizing control strategy is based on the solution of an SDP in every time step. It is shown, that the presented control algorithm can also be used for controller synthesis for nonlinear systems based on reachability computations. The nonlinear dynamics is approximated by a Taylor series and the resulting linearization error will be over-approximated and considered as an additional bounded disturbance to the affine dynamics. Furthermore, the provided algorithmic solution satisfies polytopic and ellipsoidal input constraints.

In Chapter 5, the reachability problem of an affine probabilistic system (APS) with additive disturbances is considered. The stochasticity in this setting is modeled as a normal distribution of the initial continuous state, and a normal distribution of the disturbances. The reachable sets of this stochastic model are here provided with a predefined confidence level δ, which results in probabilistic reachable sets. In addition to the input constraints, this chapter addresses state constraints. Due to the inherent stochasticity of the dynamics, a probabilistic interpretation of the constraints is required, which results in so called "state chance constraints". Chapter 5 contains a discussion of different approaches to formulate and satisfy the stated chance constraints.

Chapter 6 elevates the developed control algorithm from Chapter 5 to the system class of switched affine probabilistic systems (SAPS), in which the current active mode z_k is externally induced. The different choices for the discrete mode at each time step result in a decision tree, and tree search techniques are used. The controller synthesis is therefore twofold: first, the continuous control law is obtained by the solution of an SDP, and second, the feasible sequence of discrete modes z_k is computed by tree search. Besides the computation of a suitable continuous control input, an efficient search strategy of the decision tree is the challenging part of the hybrid control synthesis.

In Chapter 7 the methods developed in Chapter 5 and 6 are applied to the system class of piecewise affine probabilistic systems (PWAPS), in which the continuous state space is divided into finite polytopic regions with different continuous dynamics. The discrete state of this system class is determined by the region, which contains the current system state x_k, and the change of the discrete state is induced internally by the transition from one region to another. The challenging task in the problem setting with PWAPS is the intersection of the probabilistic reachable sets with multiple regions. The proposed control algorithm introduces a "push-branch-and-merge" procedure to (i) reduce the computational effort and (ii) to retain the ellipsoidal set representation of the reachable sets.

Chapter 8 concludes this thesis and proposes some future research directions in the topic of controller synthesis for SHS based on probabilistic reachability computations.

2 Literature Review

This chapter is dedicated to provide an overview of the relevant literature in the field of controller synthesis for SHS. As already pointed out, the investigated control problem in this thesis is a set-to-target control problem with the presence of uncertainties in the form of bounded and probabilistic disturbances. Such that the presented control approach in this thesis unites robust and stochastic control for hybrid systems.

In general, the field of robust control is concerned with the controller synthesis under the presence of disturbances, and in literature a vast number of publications exist on this topic for systems with continuous dynamics. The books in [9], [63], and [167] can be used as an introduction into the field of robust control.

Stochastic control theory is used to control stochastic processes, which can be described by stochastic differential equations. Economic systems or the financial market can, amongst others, be modeled by these equations, and it has, hence, gained much attention in the past decades (cf. [87], [149], [113]).

Furthermore, an introduction in the approaches on modeling and control of HS can be found in the book of [122] or in the overview article in [72].

The following Section 2.1 contains a detailed literature review on the existing approaches for controller synthesis based on reachability analysis for SHS. Since the representation of the reachable sets is a crucial aspect of these approaches, the different set representations, are reviewed in Sec. 2.2. A preview of the contributions of this thesis is given in the last section (Sec. 2.3) of this chapter.

2.1 Reachability Analysis of Stochastic Hybrid Systems

In many cases reachability analysis is motivated by verification purposes, in which it is checked whether the system states can reach an unsafe state set or not. But reachability analysis can also be used to achieve a certain safety property, i.e. it is suitable for controller synthesis, and this section reviews the available methods in literature. A mathematical foundation on reachability analysis for SHS, as well as an overview of the different types of SHS, can be found in [43].

The exact representation of the reachable set is difficult for complex system, and therefore, approximative methods are required. There are two main approaches integrating the approximation at different points. First, as already indicated, it is possible to use certain set representations (see Fig. 2.1) that can easily be represented and propagated through the system dynamics. In the second approach, the complex dynamics of SHS are approximated by a simpler abstraction, which yields a simpler model to solve the original reachability problem. In the following, some of the main contribution in the recent development of controller synthesis for SHS are presented.

Model Predictive Control

Model predictive control is an advanced method for controller synthesis, and it is based on a finite time-horizon optimization problem, which is solved at each discrete time step. The solution of each optimization is either an input vector (open loop control) or some feedback gain (closed loop control) for a finite horizon, but only the solution for the current time step is applied to the system. The repeated solution of the optimization problem at each time step is referred to as moving or receding horizon. The main advantage of model predictive control is the ability to predict future control behavior and take control actions accordingly. By applying only the current control action to the system, and neglecting the remaining entries of the solution vector, model predictive control can also deal with changes in the system dynamics. But the implementation of model predictive control requires an on-line solution of the optimization problem, and depending of the class of optimization problems, the demand on computational effort can be very high, such that model predictive control is limited to slow, or small processes.

For example, in [29] an approach is proposed, which formulates a control problem for a simple class of SHS as an mixed integer problem, and it is shown to be suitable for model predictive control. In [132], the authors propose an efficient approach based on stochastic model predictive control for a class of SHS. Similar approaches require a gridding of the state space to solve this problem for the considered class, but this approach solves explicitly the arising constraint control problem.

Dynamic programming

In mathematical optimization the term *dynamic programming* refers to the methodology of finding a solution of an optimization problem with a value function that observes Bellmann's "Principle of Optimality" ([28]). The basic idea is to simplify the optimization problem by breaking it down into a sequence of decisions, and the evaluation of each value function can be obtained by a recursive relationship (Bellman equation).

The connection of the reachability problem for a controlled discrete-time SHS, i.e. the continuous dynamics of the SHS are affected by a control input, and an finite-time horizon optimal control problem is first formulated by the authors in [16]. Therein, a methodology is developed to compute the maximum probability of remaining in a safe set by formulating an optimal control problem. The value function is defined as a multiplicative cost function, and the optimal value can be obtained by dynamic programming techniques (cf [31]). Besides the optimal control policy, the solution of the optimal control problem provides the safe set for a specified threshold probability. This safe set represents all initial conditions for which a control policy exists, such that the probability of remaining in the safe set is greater or equal the threshold probability. The extension to an infinite-time horizon together with a slightly new interpretation of the cost function is presented by the authors in [3, 7].

The introduction of a target set and obstacles leads to stochastic reach-avoid problem, which is considered in [153]. The work is based on the formulation of [3, 7], and extends it to the case of obstacles. The formulated optimal control problem maximizes the probability of hitting the target set while avoiding the obstacle (moving obstacle in [151, 152]), and is again solved by dynamic programming.

The reach-avoid problem in the framework of dynamic games is considered with two adversary players. The first player is the control input, which tries to keep the system state in a safe region, whereas the second player is the stochastic disturbance and tries to push the system state out of the safe region. While the formulation of dynamic games for stochastic systems is well-researched (see [146]), the formulation of the reach-avoid problem for discrete-time SHS is first formulated in [84, 57]. The authors extend the reach-avoid formulation for SHS in [153] to the stochastic game setting. The information pattern in the considered dynamic game is assumed to be non-symmetric, such that the player representing the disturbance has more information than the first player. This information pattern leads to a conservative behavior, since the first player make his decision without any information of the disturbance, and therefore, the worst case scenario is considered. But however, the formulation of the dynamic game leads to a value function which is maximized by dynamic programming algorithms. The solution provides a control law that maximizes the probability of entering the safe set, while avoiding the unsafe set.

A further extension suggested in [56, 117] addresses the reachability problem for SHS with only partial information of the current state. The solution of the optimal control problem is obtained by dynamic programming.

In general, the application of the dynamic programming methodology for an optimal control problem requires a gridding of the hybrid state and input space, and the numerical solution is convergent as the gridding parameter tends to zero. The computational burden for the evaluation of the value function associated with the gridding procedure becomes intractable in higher dimensions, such that the authors in [6] present an approximate dynamic programming approach. The idea is to approximate the iteration of the value function by analytic functions with finite parameters and known structure. Another approach to reduce the computational burden is suggested in [86], in which the use of approximative value functions enables the reformulation of the optimization program into a linear program with finite decision variables and constraints. However, this approach has some drawbacks concerning the probabilistic guarantees of the reach-avoid problem.

Markov-Chain Approximation

A popular method to deal with complex systems is to approximate them by a more simpler model. It has been shown, that Markov chains are a suitable system class to describe the behavior of stochastic processes in a discretized manner (cf. [113]), i.e. the continuous state and input space is partitioned into a finite space. The Markov chain originates from the definition of transition probabilities from one finite state to another. The application of the Markov approximation to continuous time stochastic differential equations is suggested in [83], in which the solution of a stochastic differential equation is approximated by a discrete-time Markov chain. The obtained Markov chain is used to estimate the probability that the system state will enter an unsafe set within a finite time horizon. The basic idea is extended to different variations of SHS and applied to air traffic management in [135, 134, 136].

The probabilistic reachability problem for continuous-time SHS is formulated in [93] as an optimization problem with a suitable value function represented by coupled Hamilton-Jacobi-Bellman equations. The authors developed a numerical approximation of the value function by Markov chains, and convergence is obtained for appropriate initial conditions as the discretization parameter tends to zero. However, it is not possible to provide any error bounds of the numerical method.

The problem of error bounds is addressed, amongst others by the authors in [4, 2] for autonomous SHS, and in [6] for controlled SHS. Therein, an abstraction of discrete-time SHS into a Markov set-chain is suggested, and likewise, this approach is based on a full discretization of the hybrid state and input space. The transition probabilities are computed by the integrals over finite subsets, and the computation of the integrals involve some approximative techniques. These techniques provide an error bound of the approximated integral, and thus of the approximated SHS. The abstraction is adopted in [5] for the purpose of model checking. The Markov set-chain is used to check a certain property, stated in terms of probabilistic temporal logic, and the model checking algorithm verifies if the property holds, or not. It is also shown, that the model checking algorithm can be used to compute a conservative approximation of the probabilistic safe set.

All the previously mentioned numerical approaches share the necessity of a full discretization of the continuous state and input space. This implies a massive computational demand and for higher dimension, the problem might get intractable due to the "curse of dimensionality". A naive approach, which is used in the aforementioned publications, is a uniform partitioning of the considered spaces. The authors in [148] propose an adaptive gridding algorithm, in which local information of the abstracted system is used to adapt the grid parameter. The effectiveness of the adaptive gridding algorithm is assured by some numerical benchmarks, and thus the method alleviates the curse of dimensionality.

Randomized Methods

Another possibility to deal with the inherent stochasticity of SHS are randomized methods, in which stochastic events are approximated by a finite number of samples or scenarios. The *scenario approach* was first introduced in [44] for solving uncertain convex optimization problems via randomization. Since then, the scenario approach is used in various application to approximate stochastic events. Model predictive control for discrete-time linear systems with parametric uncertainties in the system matrices is addressed in [30], in which a scenario-based optimization tree is build, where only the most relevant disturbance pattern are modeled. The author in [36] developed a framework for robust control of unmanned vehicles in an uncertain environment. Therein probabilistic requirements are satisfied by the scenario approach, too.

2.2 Set-Representation

As already mentioned, the exact computation of reachable sets even for linear systems with arbitrary (convex / non-convex) initial conditions and inputs is generally not possible, and approximative techniques are required. The degree of approximation is determined by the choice of the set representation, and the more complex the representation is, the more complex is the computation of the sets, involving a better approximation of the reachable set. The choice of the set representation is always a compromise between computational effort and accuracy. In literature two basic set representation are used as convex approximations: polytopes and ellipsoids. The class of polytopes can be divided in various sub-class, e.g. zonotopes, parrallelotopes, and rectangular polytopes.

A convenient introduction into the theory of convex sets in control can be found in [37], in which ellipsoids and polytopes are used to define invariant sets. An invariant set for a dynamic system describes a set, which cannot be leaved by the trajectories of the system, if the initial condition is inside this set. Therefore, an invariant set can be seen as an over-approximation of a reachable set. The propagation of reachable sets described as ellipsoids and polytopes is also contained in [37], which can serve as a start in the literature on reachability analysis. Although there is a relation between the theory of invariant sets and reachability analysis, this review focuses on the publications on reachability analysis, and a graphical illustration of the used set representations is shown in Fig. 2.1.

Polytopes

In order to perform reachability analysis for arbitrary dynamic systems several set-valued operations are required, such as affine transformation, intersection, union, and Minkowski sum/difference. An available implementation for Matlab to perform these operations on polytopes is the *Multi Parametric Toolbox* (MPT) [80].

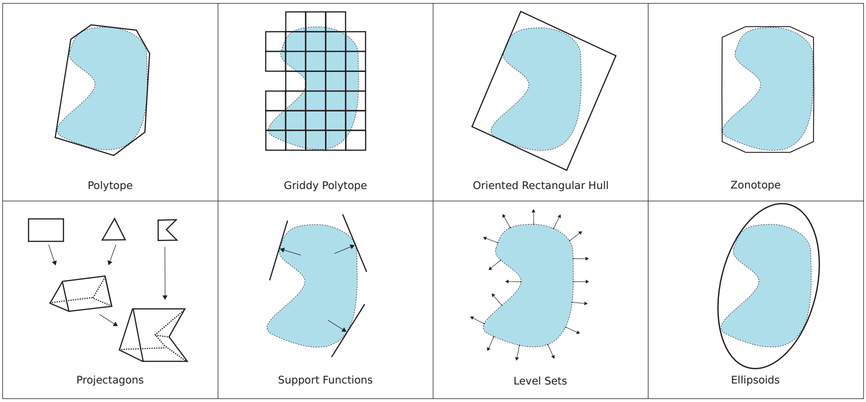


Figure 2.1: Set representation: Overview of set representations used for reachability computations.

In the reachability analysis of discrete-time systems, the reachable set for the next time step is computed by the Minkowski sum of the state set, the input set, and, if considered, the disturbance set. The computation of the Minkowski sum of two polytopes is based on the Double Description method [129], which computes a convex hull for a given set of vertices. The complexity of this algorithm scales linear with the number of vertices and exponential with the dimension of the state space. Therefore, the use of the toolbox is limited to low dimensional systems and even for low dimensional systems the number of vertices can grow very large with the number of time steps. Polytopic state and input sets are considered in [88], [89], [90] to perform a reachability analysis for discrete-time affine systems with bounded disturbances. The publications [140], [139] extend these result to the case of state- and input-dependent disturbances. A summary of the presented methods can be found in [142], which is based on the computation of backwards reachability sets for a finite number of time steps. The general idea of backwards reachability is to start in a desired set and compute a predecessor set of states for which an admissible input exists, such that the successor state is in the original set. If disturbances are considered as well in the computation, the developed backwards reachability approach in [142] provides a robust reachable set for every time step, together with the corresponding state feedback control law. However, the required computations on polytopes include the convex hull computation of the geometric difference, and therefore, the approach suffer from the curse of dimensionality. An attempt to reduce the computational complexity is to restrict the general polytopes to a specific shape. In [48], the authors perform reachability computations for rectangular hybrid systems, which are hybrid systems with rectangular bounds for all involved variables. The advantage of rectangular hybrid systems is, that the reachable sets can be exactly computed as a set of polytopes and no approximative methods are needed. The restriction to hyper-rectangles[1] leads to a constant number of vertices for the reachable sets, and thus, an efficient computation of the reachable sets as shown in [48].

Griddy Polytopes

The authors in [52] use *orthogonal* polytopes which consists of a finite union of hyper-rectangles, and the representation of each hyper-rectangle is easier compared to an arbitrary polytope. A sub-class of orthogonal polytopes are *griddy* polytopes which are generated by unit hyper-cubes and a canonical representation for any dimension is introduced in [40]. The proposed griddy polytopes are used in [20] for reachability computation of autonomous linear and autonomous piecewise linear systems, in which a method from [74] is used to find a trade-off between the accumulation of errors and the accuracy. The reachability computation with griddy polytopes for linear and hybrid systems is included in d/dt [51], but however, the computation includes the convex hull operation, which limits the dimension of the system and the time span of the calculated reachable set.

Oriented Rectangular Hulls and Parallelotopes

The disadvantage of griddy polytopes is the quality of approximation compared to arbitrary polytopes. The approximation of an arbitrary set by the axis-parallel faces of a griddy polytope can be seen as a zero-order approximation, which is always poorer than a first-order approximation provided by an arbitrary polytope. One approach to improve the approximation, while still restricting the shape of the polytope to rectangles, are known as oriented rectangular hulls, proposed in [150]. The idea is to compute an oriented rectangular polytope based on a given set of vertices. The orientation of this rectangle is determined by a technique which is known as *principle component analysis*. This technique uses the single value decomposition of a sample covariance matrix of the vertices to provide the orientation of the approximating rectangle. Due to the avoidance of the convex hull operation and the application of the singular value decomposition, this representation is much more efficient regarding the computational complexity in higher dimension. Reachability analysis and verification with oriented rectangular polytopes is included in CheckMate [49]. A natural extension to oriented rectangular polytopes are parallelotopes [91, 92], which can be seen as skewed oriented rectangles.

Zonotopes

Another sub-class of polytopes, which have received a certain amount of attention in the reachability community are zonotopes. A zonotope can either be defined

[1] A hyper-rectangle is a rectangle in arbitrary dimension.

by the projection of a hyper-cube, or the Minkowski sum of line segments, and is therefore always centrally symmetric [168]. The author in [69] propose the application of zonotopes for reachability computation on the strength of two main properties: 1. zonotopes are closed under linear transformation, i.e. the image of a zonotope by a linear map results again in a zonotope (this property holds in general for arbitrary polytopes), 2. zonotopes are closed under Minkowski sum operation, which does not hold for arbitrary polytopes and constitutes the main advantage of zonotopes. The Minkowski sum is an important operation in the reachability computation of discrete-time systems and the closeness property of zonotopes avoid further over-approximations in each time step. The propagation of approximation errors through the reachability computations is known as *wrapping effect*, which might lead to dramatic over-approximations of the reachable sets for large time horizons. The author in [97] introduces a method to construct zonotopes with arbitrary small effects of the wrapping. Another advantage of zonotopes compared to arbitrary polytopes is the efficient computation of the Minkowski sum, for which no convex hull algorithm is needed to obtain the resulting zonotope. The efficiency and accuracy of zonotopes for reachability analysis is first mentioned [96], and extended in [69] to systems with uncertain inputs. The work result in a wrapping free algorithm in [71]. The reachability analysis with zonotopes for safety assessment of autonomous cars is investigated in [12], in which a notion of probabilistic zonotopes is included. Besides the Minkowski sum, the Minkowski difference of the sets is of interest for backwards reachability analysis. The computation of the Minkowski difference of two zonotopes is presented in [13], and it is shown, that it outperforms the state-of-the-art algorithms for half-space representation of polytopes.

In general, all sub-classes of polytopes suffer from the problem of not being closed under intersection, e.g. the intersection of a griddy polytope/zonotope/parallelotope with a half-space is not a polytope/zonotope/parallelotope. Since the intersection is a necessary and important computation in reachability analysis of hybrid systems, zonotope bundles are introduced in [14] to overcome this disadvantage. The reachability computation with zonotopes for (hybrid), (non)-linear systems can be performed with the Matlab-toolbox CORA [11].

Projectagons

The representation of an arbitrary (possible non-convex) polytope in high dimensions by its projections onto two dimensional sub-spaces is considered by projectagons. The authors in [73] introduce this mathematical objects for reachability computations of circuit models in high dimensions. The full dimensional polytope is obtained by back-projecting the two dimensional polytopes and computing the intersection of those back-projections. Clearly, an advantage of projectagons is, that all geometric operations for reachability analysis take place in two dimensions, which is quite efficient. Although, the computational time for reachability analysis can be reduced by this technique, projectagons introduce new over-approximations, since some information about the initial full-dimensional polytope might be lost with the projection. For example, indentations on the surface cannot be captured by this approach. So far, the reachability techniques with projectagons seem to be quite useful for circuit verification and the methods are implemented in the MATLAB toolbox COHO [164].

Support Functions

Another representation of arbitrary convex sets are support functions, which are a standard tool in convex analysis [42]. The first consideration of support functions for reachability computations and controller synthesis for continuous-time linear systems with constrained initial states and inputs is in [161], and extended in [70, 116]. The advantage of support functions comprise the ability of being able to be computed efficiently for a fairly large class of sets, including polytope, zonotopes and ellipsoids. Furthermore, support functions behave nicely under most geometric operations like Minkowski sum and convex hull. Therefore, support functions are a very convenient and acclaimed way of representing reachable sets. The results from [116] are adapted in [115] for hybrid dynamics, in which the computation of the support function for the intersection of a convex set with hyper-plane is proposed. The intersection of the reachable set with a guard set triggers a change in the discrete variable in some variants of hybrid systems. The reachability computation for hybrid systems with continuous linear dynamics with support functions is mainly used in the toolbox SpaceEx [64, 65].

Level Sets

Level set methods were initially developed in [131] to describe the change of position and shape of dynamic boundaries, like breaking waves in the ocean, or flames in a fireplace. The idea is to define a level set function in a higher dimension than the actual set, and the zero level of this function returns the contour of the original set. The propagation in time of the boundary can be formulated as a level set equation, which is a partial differential equation. The solution to the differential equation is obtained by the mathematical theory of *viscosity solutions*. In [158] a reach-avoid problem is formulated for nonlinear hybrid systems and the solution for this problem is obtained by the techniques of level set methods. In detail, the optimal control input is computed by the viscosity solution of the Hamilton-Jacobi partial differential equation. The great challenge in the formulated partial differential equation for hybrid systems are the discontinuities in the variables due to a change in the discrete state variable. A great advantage of the proposed level set methods in [131] is that it can systematically handle these discontinuities, based on the viscosity solution. The basic level set approach is further improved in [126] with a new formulation of the partial differential equation with superior numerical properties and an improvement of the basic algorithm. Further improvements consider the fast and efficient computation of the solution of the partial differential equation [125], the application to continuous dynamic games [127, 156], and hybrid systems with differential algebraic equations in [128, 157, 50]. An implementation of the published level set methods can be found in the Matlab toolbox ToolboxLS [124]. Recently, the level set approach has been applied to a multi-vehicle collision avoidance problem in [47]. The graphical interpretation of the level set method is illustrated in Fig. 2.1 by the outwards pointing arrows, which indicate the solution of the partial differential equation.

Ellipsoids

Ellipsoids received much attention in the reachability computation, since the complexity of the representation does not grow in time, and it merely grows quadratic with the dimension of the state space. An excellent contribution on the ellipsoidal calculus for estimation and control can be found in [100]. Although ellipsoids are not closed under Minkowski sum and intersection, tight ellipsoidal approximations are proposed in [100], which can be computed efficiently and qualify themselves for reachability analysis of dynamic systems. Further research considered the tight inner and outer approximation of reachable set for continuous-time linear time varying systems in [102, 101, 103, 104], wherein the description of the sets can be obtained by the solution of an ordinary differential equation. Uncertain systems with unknown but bounded disturbances, and systems under state constraints has been investigated in [105] and [106, 109], respectively. The extension to hybrid systems, in which a change of the discrete system variable and a reset of the continuous variable may appear after crossing a hyper-plane, is suggested in [107, 108]. The results for continuous-time systems are adapted in [98] for the discrete-time case. The emphasis is made to the handling of singular state transition matrices, which leads to degenerate ellipsoids. The authors propose an over-approximating technique by bloating the original reach set to obtain a full-dimensional ellipsoid. The extension to the discrete-time case with bounded disturbances can be found in [112]. The results on the ellipsoidal calculus, basically developed in [100], are extended to the estimation of reachable sets for uncertain nonlinear systems in [60, 62, 61]. Therein, the ellipsoidal calculus is combined with the theory of integral funnel type evolution equations, which allows to describe the evolution of a set of a set-valued function.

A collection of all methods for the ellipsoidal calculus are implemented in the Matlab toolbox ET [110], in which the ellipsoidal approximation of the Minkowski sum and difference as well as the intersection of two ellipsoids is realized with methods known as propagation and fusion [143]. An earlier implementation of the methods developed in [100, 102] was included in the toolbox VeriShift, suggested in [39], however, the toolbox is no longer supported.

2.3 Contribution of this thesis

The review of the existing approaches for different sub-classes of SHS in Sec 2.1 shows, that the computational demand is an important aspect for the control of SHS. The reviewed methods are all based on a certain discretization of the state space or the probabilistic distribution by means of a sampling procedure, and in order to achieve a sufficiently good enough approximation, an excessive computational effort is needed. A remaining open question is the design of stabilizing controllers for SHS.

The aim of this thesis is to develop efficient control synthesis procedures for set-to-target control problems for different types of SHS. The control procedure is based on probabilistic reachability computations, and for this purpose, an efficient notation of probabilistic reachable sets with a specified confidence δ is introduced. These sets are formulated to contain the normally distributed state x_k with a probability of δ. The set based formulation is incorporated into an optimization based controller synthesis, and the obtained solution observes different constraints regarding the input and state variable.

The main contribution of this thesis is the introduction of an algorithmic control procedure for the set-to-target control for different specifications of SHS. Regarding the discrete dynamics of an SHS, three different types are considered: (i) the SHS has only one single discrete mode, such that no change of the discrete mode, and hence of the continuous dynamics occurs; (ii) the change of the discrete mode can be externally triggered, and hence be included as a degree of freedom in a control strategy; or (iii) the change of the discrete mode is triggered autonomously and cannot be affected directly by a control strategy but only indirectly by the continuous controls. Each of the described forms for the discrete dynamics has its own specific challenges, and this thesis introduces an algorithmic control procedure for each of them. The successful termination of each algorithm ensures the solution of the set-to-target control for the considered system class.

3 Theoretical Background

This chapter introduces some relevant basics on the mathematical concept and the notation used in this thesis. The transpose of a matrix or a vector is denoted by ^{T}. A set of eigenvalues $\lambda_i, i = \{1, \ldots, n\}$ for a given matrix $Q \in \mathbb{R}^{n \times n}$ will be denoted with $\Lambda(Q)$. $\lambda_{min}(Q)$ and $\lambda_{max}(Q)$ are the minimum and maximum eigenvalues of the matrix Q. The determinant of a matrix Q is given by $|Q|$. A matrix $Q \in \mathbb{R}^{n \times n}$ is called positive-definite, if it holds that $\lambda_i > 0,\ \forall \lambda_i \in \Lambda(Q)$, and positive-semidefinite, if $\lambda_i \geq 0,\ \forall \lambda_i \in \Lambda(Q)$. The same applies for negative-definite and negative-semidefinite. In this thesis the following short notation for the definiteness of a matrix will be used:

$$Q > 0, \tag{3.1}$$

where the inequality symbol in (3.1) specifies the definiteness of the matrix. A matrix $Q \in \mathbb{R}^{n \times n}$ is called symmetric, if it holds that: $Q = Q^T$. The Euclidean norm of a vector $x \in \mathbb{R}^n$ is given by

$$||x|| := \sqrt{x_1^2 + x_2^2 + \ldots + x_n^3}. \tag{3.2}$$

The interior and boundary of an arbitrary closed and compact set $W \subset \mathbb{R}^n$ is denoted by $int(W)$ and $\partial(W)$, respectively, and it holds that:

$$int(W) \cup \partial(W) = W. \tag{3.3}$$

The following subsection are structured such that a general formulation of SHS is introduce in Sec. 3.1. Sec. 3.2 presents the necessary representation for ellipsoidal, polytopic, and interval sets. Section 3.3 recalls some basics about multivariate random variables, since stochastic variables are used to model probabilistic uncertainties in the considered dynamic systems. Section 3.4 recaps basics about convex optimization, which is a fundamental part in the later presented control synthesis procedure.

3.1 Stochastic Hybrid System

Definition 3.1. *A stochastic hybrid System (SHS) is given by the following equations ($k \in \mathbb{N}_0$):*

$$
\begin{aligned}
x_{k+1} &= f(x_k, u_k, z_k) + G v_k, && (3.4a)\\
x_0 &\sim \mathcal{N}(q_{x,0}, Q_{x,0}), && (3.4b)\\
v_k &\sim \mathcal{N}(q_v, Q_v), && (3.4c)\\
u_k &\in U = P_H(R_u, b_u), && (3.4d)\\
x_k &\in \mathbb{X}_k = P_H(R_{x,k}, b_{x,k}), && (3.4e)\\
z_k &\in Z = \{1, \ldots, n_z\}, && (3.4f)
\end{aligned}
$$

where the initial state x_0, and the disturbance v_k are Gaussian distributed with expected value $q_{x,0} \in \mathbb{R}^n$, $q_v \in \mathbb{R}^n$, and covariance matrices $Q_{x,0} \in \mathbb{R}^{n\times n}$, and $Q_v \in \mathbb{R}^{n\times n}$, respectively. The input $u_k \in \mathbb{R}^m$ is bounded to a convex polytope $U \in \mathcal{P}$ with $R_u \in \mathbb{R}^{n_u \times m}$, and $b_u \in \mathbb{R}^{n_u}$. The discrete mode z_k is taken from a finite set of discrete modes Z, and it determines the active dynamics specified be the update function $f(x_k, u_k, z_k)$. △

The current value of the discrete mode z_k at time step k can either be determined by a functional relation between the continuous state x_k and the discrete mode z_k (switching system), or z_k can be externally chosen (switched system). Def. 3.1 introduces a general formulation of discrete-time SHS and a feasible execution is as follows:

1. sample the current state x_k and the disturbance v_k according to (3.4b) and (3.4c)
2. choose an input $u_k \in U$
3. determine the current discrete mode z_k by (i) a state dependent function, or (ii) externally chosen
4. evaluate the difference equation to get the consecutive state x_{k+1} according to (3.4a)

In the following chapters, this definition of an SHS serves as a template for the considered system class, and it is shown how to derive each system from this definition.

3.2 Set Representation

One important aspect in the verification and reachability computation of dynamic systems is the choice of the sets used for tight approximation of the exact reachable set. For many classes of dynamic systems and especially for hybrid systems, the computation of the exact reachable set is either impossible or associated with excessive computational effort. Thus, it is crucial to choose a set representation which is able of approximating the reachable sets closely, and which allows efficient computation of some geometric operations, like affine transformation and Minkowski addition.

Ellipsoid

In this thesis, ellipsoids are used for the approximation of the reachable sets:

Definition 3.2. *Let $\mathcal{E}$ denote the set of all ellipsoidal sets in $\mathbb{R}^n$. An* ellipsoidal set $\varepsilon(q,Q) \in \mathcal{E}$ *is parametrized by a center point $q \in \mathbb{R}^n$ and a symmetric and positive-definite shape matrix $Q \in \mathbb{R}^{n\times n}$, $Q = Q^T > 0$ according to:*

$$\varepsilon(q,Q) = \left\{x \in \mathbb{R}^n | (x-q)^T Q^{-1}(x-q) \leq 1\right\} \tag{3.5}$$

with ^{T} indicating the transpose of the vector. △

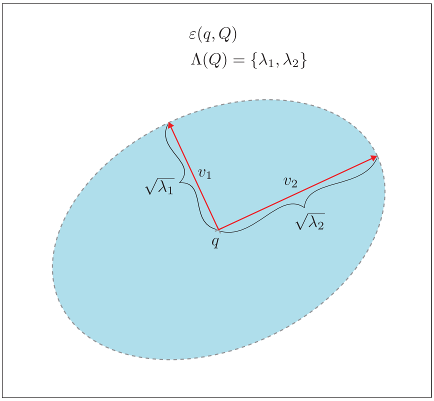


Figure 3.1: Definition of an ellipsoid: The direction of expansion of the ellipsoid is specified by the eigenvectors v_1 and v_2 and the length of each expansion is given by the square root of the eigenvalues λ_1 and λ_2.

The location of an ellipsoid is specified by its center point q, whereas the shape and orientation is defined by the shape matrix Q. The semi-major axis is the longest radius of an ellipsoid, and the semi-minor axis is the shortest radius of an ellipsoid. The length and orientation of the semi-axes of $\varepsilon(q,Q)$ is defined by the root of the eigenvalues $\lambda_i \in \Lambda(Q)$, and the eigenvectors v_i, respectively. Figure 3.1 illustrates the specification of an ellipsoid by the eigenvalues and eigenvectors of the shape matrix Q.

An affine transformation of an ellipsoid by a matrix $M \in \mathbb{R}^{n\times n}$ and a vector $b \in \mathbb{R}^n$ results again in an ellipsoidal set:

$$M \cdot \varepsilon(q, Q) + b = \varepsilon(M \cdot q + b, MQM^T) \tag{3.6}$$

Reachability analysis of dynamic systems comprise the computation of the reachable sets at any discrete time step and this includes the computation of the Minkowski sum of two sets.

Definition 3.3. *The Minkowski (or geometric) sum of two arbitrary, but bounded sets $W_1 \subset \mathbb{R}^n$ and $W_2 \subset \mathbb{R}^n$ is defined by:*

$$W_1 \oplus W_2 := \{w_1 + w_2 \mid w_1 \in W_1, w_2 \in W_2\} \tag{3.7}$$

△

The Minkowski sum of two ellipsoidal sets $\varepsilon(q_1, Q_1)$ and $\varepsilon(q_2, Q_2)$ is, in general, not an ellipsoidal set, but according to the following lemma, a tightly enclosing approximation can be obtained as an ellipsoidal set $\varepsilon(q_1 + q_2, Q(s))$:

Lemma 3.1. *(Lemma 2.2.1 (a) in [100])*
For $\varepsilon(q_1, Q_1) \in \mathcal{E}$ and $\varepsilon(q_2, Q_2) \in \mathcal{E}$, the ellipsoid $\varepsilon(q_1 + q_2, Q(s))$ with

$$Q(s) = (1 + s^{-1})Q_1 + (1 + s)Q_2, \ \ s > 0, \tag{3.8}$$

is an outer approximation of the Minkowski sum $\varepsilon(q_1, Q_1) \oplus \varepsilon(q_2, Q_2)$, i.e,

$$\varepsilon(q_1, Q_1) \oplus \varepsilon(q_2, Q_2) \subseteq \varepsilon(q_1 + q_2, Q(s)) \tag{3.9}$$

for any $s > 0$.

Proof. See [100]. □

The volume of the enclosing ellipsoid $\varepsilon(q_1 + q_2, Q(s))$ is determined by s, and there exists a unique value for s for which the outer approximation of the Minkowski addition is of minimum volume, and this value determines a tight enclosure of the Minkowski sum.

Polytope

Another type of set, used in this thesis, is a convex polytope. A convex polytope P can be defined by two representations: (I) the half-space representation (H-representation, left in Fig. 3.2) and (II) the vertex representation (V-representation, right in Fig. 3.2). In the half-space representation, the polytope is defined by the intersection of n_p half-spaces $\mathcal{H}$. A half-space is obtained by the division of the n-dimensional Euclidean space with a hyperplane into two convex sets. A half-space is defined by:

$$\mathcal{H} := \{x \in \mathbb{R}^n \mid r_1 \cdot x \leq b_1\}, \tag{3.10}$$

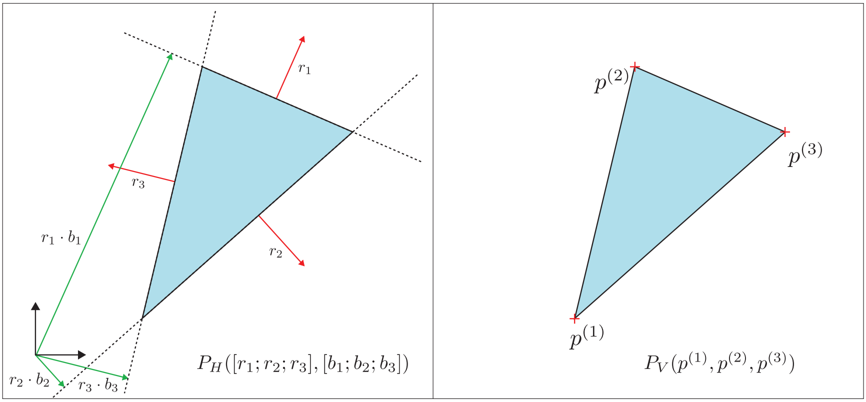


Figure 3.2: (left) The polytope $P_H([r_1; r_2; r_3], [b_1; b_2; b_4])$ is given by the intersection of $n_p = 3$ half-spaces. The green arrows indicate the distance of each hyperplane along the corresponding normal vector r_i to the origin. (right) Specification of a polytope in V-representation by a set of vertices $p^{(1)}, \ldots, p^{(n_r)}$.

where the vector $r_1 \in \mathbb{R}^{1\times n}$ (red arrows on the left of Fig. 3.2) is assumed to be a normalized vector ($||r_1|| = 1$) and $b_1 \in \mathbb{R}$ is the distance of the cutting plane to the origin along the vector r_1 (green arrows on the left of Fig. 3.2). A polytope is specified by the intersection of n_p half-spaces, such that 3.10 has to be fulfilled for each half-space:

Definition 3.4. *Let $\mathcal{P}$ denote the set of all polytopic sets in $\mathbb{R}^n$. A convex polytope $P_H(R, b) \in \mathcal{P}$ in H-representation with n_p half-spaces is defined by a matrix $R \in \mathbb{R}^{n_p \times n}$ and a vector $b \in \mathbb{R}^n$ according to:*

$$P_H(R, b) = \left\{ x \in \mathbb{R}^n \mid R \cdot x \leq b, \quad R \in \mathbb{R}^{n_p \times n}, b \in \mathbb{R}^{n_p} \right\} \tag{3.11}$$

The matrix R contains the normalized vectors r_i for each hyperplane and the vector b is the collection of the distance values b_i. △

The vertex representation specifies a polytope by the convex hull of a finite set of vertices $p^{(i)} \in \mathbb{R}^n$.

Definition 3.5. *Let $\mathcal{P}$ denote the set of all polytopic sets in $\mathbb{R}^n$. A convex polytope $P_V\left(p^{(1)}, \ldots, p^{(n_r)}\right) \in \mathcal{P}$ in V-representation with n_r vertices $p^{(i)}, i \in \{1, \ldots, n_r\}$ is defined by the linear combination of each vertex:*

$$P_V\left(p^{(1)}, \ldots, p^{(n_r)}\right) = \left\{ \sum_{i=1}^{n_r} \alpha_i p^{(i)} \mid p^{(i)} \in \mathbb{R}^n, \alpha_i \in \mathbb{R}, \alpha_i \geq 0, \sum_{i=1}^{n_r} \alpha_i = 1 \right\} \tag{3.12}$$

△

Definition 3.6. *For a bounded convex polytope $P_V\left(p^{(1)},\ldots,p^{(n_r)}\right)$, given in V-representation, the geometrical center is determined by the function:*

$$centroid\left(P_V\left(p^{(1)},\ldots,p^{(n_r)}\right)\right) = \frac{1}{n_r}\sum_{i=1,\ldots,n_r} p^{(i)} \tag{3.13}$$

△

Interval in Dimension n

Another important class of sets in this thesis is the class of n-dimensional intervals. In general, a closed interval is given by two boundary values w_{min} and w_{max} and it holds that any value between these boundary values is also contained in the interval:

$$[w_{min}, w_{max}] = \{w \mid w_{min} \le w \le w_{max}\} \tag{3.14}$$

For ease of notation, an interval for the variable w will be written as:

$$\lceil w \rfloor := [w_{min}, w_{max}] \tag{3.15}$$

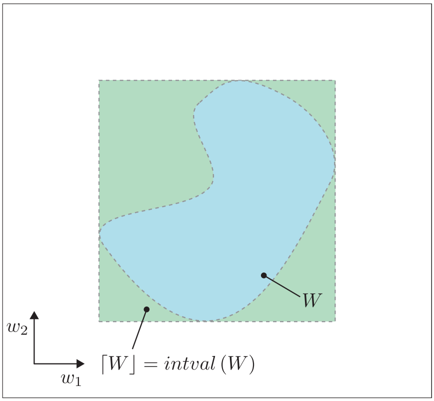


Figure 3.3: Definition of an interval in n dimension: The function $intval()$ returns an interval $\lceil W \rfloor$ of an arbitrary set W, and it holds that $W \subseteq \lceil W \rfloor$. The interval consists of the Cartesian product of the two intervals in each dimension.

The extension to an interval in n-dimensions can be done by the Cartesian product of n intervals:

$$\lfloor W \rceil = \lfloor w_1 \rceil \times \lfloor w_2 \rceil \times \ldots \times \lfloor w_n \rceil, W \in \mathbb{R}^n, w_{1,\ldots,n} \in \mathbb{R} \tag{3.16}$$

The resulting n-dimensional set can be seen as a rectangle whose facets are parallel to the coordinate axis (see Fig. 3.3). The set representation by intervals in n dimensions can be used as an over-approximation of any non-/convex set, which is obtained by the following function:

Definition 3.7. *Let $\mathcal{I}$ denote the set of all interval sets in $\mathbb{R}^n$. For an arbitrary bounded and connected set $W \in \mathbb{R}^n$, the following function returns an interval $\lfloor W \rceil \in \mathcal{I}$ as a vector of the smallest intervals in each dimension of $\mathbb{R}^n$, which completely contains W:*

$$\lfloor W \rceil = intval\,(W) = \begin{bmatrix} [\min_{w \in W} w_1, \max_{w \in W} w_1] \\ \vdots \\ [\min_{w \in W} w_n, \max_{w \in W} w_n] \end{bmatrix} \supseteq W \tag{3.17}$$

$\triangle$

An interval is a special form of a polytope ($\mathcal{I} \subseteq \mathcal{P}$) with two bounding spaces for each dimension, and the bounding hyperplanes are parallel two the coordinate axis.

3.3 Multivariate Random Distributions

The stochastic distributions used in this thesis are introduced in the following. One main source of uncertainty considering the modeling of physical systems is noise, since many biological and physical measurements have lots of sources of inaccuracy and noise. In most cases, this inaccuracy is modeled by a stochastic disturbances with a Gaussian/normal distribution. The Gaussian distribution is a reasonable choice for the modeling of the uncertainty, since the *Central Limit Theorem* states, that the sum of many random variables, each independent but arbitrary distributed, is again Gaussian (see [67]). Furthermore, the Gaussian distribution is very convenient, since it is fully specified by its first- and second-order moment, which is the mean and the variance. In contrast to the higher-order moments, the first two moments are easy to measure, and thus makes the Gaussian distribution a very useful and powerful tool.

Let ξ denote an n-dimensional random vector with *multivariate normal distribution*. This distribution is parametrized by the mean vector $q_\xi \in \mathbb{R}^n$ and the symmetric covariance matrix $Q_\xi = Q_\xi^T \in \mathbb{R}^{n \times n}$ according to:

$$\xi \sim \mathcal{N}(q_\xi,\ Q_\xi). \tag{3.18}$$

In the special case of a standard normal distribution, the mean is a zero vector $0 \in \mathbb{R}^n$, and the covariance matrix equals an identity matrix I_n of dimension n, i.e.: $\xi \sim \mathcal{N}(0,\ I_n)$. If the covariance matrix of a multivariate normal distribution is positive definite, $Q_\xi > 0$, the distribution is non-degenerate, and the *probability density function* (PDF) is given by:

$$f_N(\xi) = \frac{1}{\sqrt{(2\pi)^n |Q_\xi|}} \exp\left(-1/2(\xi - q_\xi)^T Q_\xi^{-1}(\xi - q_\xi)\right). \tag{3.19}$$

The *cumulative distribution function* (CDF) of a multivariate normal distribution is defined by the integral of the PDF over an arbitrary set W:

$$F_N(\xi, W) = \int_{w \in W} f_N(\xi) dw = \int_{w \in W} \mathcal{N}(q_\xi, Q_\xi). \tag{3.20}$$

The CDF provides the probability of ξ being inside the considered set W:

$$Pr(\xi \in W) := F_N(\xi, W). \tag{3.21}$$

In many applications the region of integration W is a polytope or an ellipsoid. There are numerical approximation for the CDF of a multivariate normal distribution over these sets (see [68]). However, in this thesis both types of regions will be of interest, and the different approaches to evaluate the multidimensional integral in (3.20) are used in this thesis.

For two random vectors in $\mathbb{R}^n$ with normal distribution, i.e. $\xi_1 \sim \mathcal{N}(q_{\xi_1},\ Q_{\xi_1})$, $\xi_2 \sim \mathcal{N}(q_{\xi_2},\ Q_{\xi_2})$, the sum is also normally distributed with:

$$\xi_1 + \xi_2 \sim \mathcal{N}(q_{\xi_1} + q_{\xi_2},\ Q_{\xi_1} + Q_{\xi_2}). \tag{3.22}$$

The χ^2-distribution with n degrees of freedom results when n independent standard normal random variables $\xi_1 \dots \xi_n$ with $\xi_i \in \mathbb{R}$ are squared and summed. Thus, for a vector $\xi = (\xi_1, \dots, \xi_n)^T$, it applies that:

$$\xi^T \xi = \xi_1^2 + \dots + \xi_n^2 = \sum_{i=1}^{n} \xi_i^2 =: \sigma \sim \chi^2(n), \tag{3.23}$$

σ is a χ^2 distributed random variable with the following PDF and CDF:

$$f_{\chi^2}(\sigma, n) = \frac{e^{-\sigma/2}\sigma^{n/2-1}}{2^{n/2}\Gamma\left(\frac{n}{2}\right)}, \tag{3.24}$$

$$F_{\chi^2}(c, n) = \int_0^c f_{\chi^2}(\sigma, n) d\sigma = \int_0^c \frac{e^{-\sigma/2}\sigma^{n/2-1}}{2^{n/2}\Gamma\left(\frac{n}{2}\right)} d\sigma = \frac{\gamma\left(\frac{1}{2}n, \frac{1}{2}c\right)}{\Gamma\left(\frac{n}{2}\right)}, \tag{3.25}$$

where $\gamma(n, c)$ is an incomplete gamma function and $\Gamma(n)$ is a gamma function (cf.[8]). $F_{\chi^2}(c, n)$ returns the probability of $\sigma = \xi^T \xi \in [0,\ c]$:

$$Pr\left(\sigma \in [0,\ c]\right) := F_{\chi^2}(c, n). \tag{3.26}$$

One key feature of the presented methods in this thesis is the determination of a confidence set for a multivariate normal distribution. This confidence set contains all realizations of the random variable ξ with a typically high probability δ. In the following, the χ^2 distribution will be used to derive such confidence regions for multivariate normal distributions.

Lemma 3.2. *Let a vector* $\xi = (\xi_1, \ldots, \xi_n)^T$ *of* n *independent random variables with standard normal distribution, parametrized by the mean* q_ξ *and the covariance matrix* Q_ξ*, be given:*

$$\xi \sim \mathcal{N}(q_\xi, Q_\xi). \tag{3.27}$$

Furthermore, let $F_{\chi^2}(c, n)$ *be the cumulative distribution function of the* χ^2*-distribution. Then, if* $Q_\xi^\delta = Q_\xi \cdot c_\xi$ *and* $c_\xi = \left(F_{\chi^2}\right)^{-1}(\delta, n)$*, the probability of* ξ *being inside the ellipsoid* $W^\delta := \varepsilon(q_\xi, Q_\xi^\delta)$ *is given by*

$$Pr(\xi \in W^\delta) = \delta. \tag{3.28}$$

△

Proof: The surfaces of equal density of the PDF of a multivariate normal distribution are ellipsoids. Any ellipsoid for a fixed density is determined by the mean vector q_ξ, the covariance matrix Q_ξ, and a scaling value σ (see [95]):

$$(\xi - q_\xi)^T Q_\xi^{-1} (\xi - q_\xi) = \sigma. \tag{3.29}$$

The change of variables according to $\beta = Q_\xi^{1/2}(\xi - q_\xi)$ transforms the multivariate normal distribution into a standard normal distribution with $\beta \sim \mathcal{N}(0, I_n)$, thus modifying (3.29) to $\beta^T\beta = \sigma$, and σ is according to (3.23) a χ^2-distributed random variable. The probability for σ being lower than an upper bound c_ξ is according to (3.26) evaluated by the CDF of the χ^2 distribution:

$$Pr(\sigma \le c_\xi) = F_{\chi^2}(c_\xi, n). \tag{3.30}$$

The desired probability is denoted with δ and the upper bound c_ξ follows from the inverse of the CDF:

$$c_\xi = \left(F_{\chi^2}\right)^{-1}(\delta, n). \tag{3.31}$$

Using these equations in combination with the backwards transformation of the variables, a confidence ellipsoid is obtained and it holds that:

$$\begin{aligned}
\delta &:= Pr(\sigma \le c_\xi) = F_{\chi^2}(c_\xi, n) \\
&= Pr(\beta^T\beta \le c_\xi) \\
&= Pr((\xi - q_\xi)^T Q_\xi^{-1} (\xi - q_\xi) \le c_\xi) \\
&= Pr((\xi - q_\xi)^T (Q_\xi c_\xi)^{-1} (\xi - q_\xi) \le 1) \\
&= Pr(\xi \in \varepsilon(q_\xi, Q_\xi c_\xi)) = Pr(\xi \in \varepsilon(q_\xi, Q_\xi^\delta)) = Pr(\xi \in W^\delta),
\end{aligned} \tag{3.32}$$

i.e. W^δ contains the percentage δ of realizations of the random variable ξ. □

The ellipsoid $W^\delta = \varepsilon(q_\xi, Q_\xi^\delta)$ is called *confidence ellipsoid* in the following, and the shape matrix of the ellipsoid is Q_ξ^δ. The superscript δ indicates that the shape matrix stems from a covariance matrix (Q_ξ) multiplied with the scaling factor c_ξ, in order to obtain the desired confidence δ. Lemma 3.2 is a numerical approximation of the initially formulated integral in (3.20) for the probability over an ellipsoidal set W^δ, since the evaluation of the CDF for a χ^2-distribution is done with look-up tables. Actually, Lemma 3.2 describes a special case of (3.20), in which the considered ellipsoidal set W^δ is centered at the mean value q_ξ of the normal distribution, whereas the formulation in (3.20) refers the more general case. The approximation of (3.20) over an arbitrary ellipsoidal set can be found in [68].

3.4 Semidefinite Programming

The methods presented in this thesis rely on the formulation and solution of optimization problems. Nowadays, convex optimization problems play a crucial role in computational mathematics, since many applications have been discovered in areas like control systems, estimation, signal processing, statistics, finance, and many others. In general an optimization problem has the form

$$\min_{\eta} \quad J(\eta) \tag{3.33a}$$

$$\text{subject to} \quad \psi_i(\eta) \leq 0, \quad i = 1, \ldots, n_c. \tag{3.33b}$$

The goal of an optimization problem is to minimize the objective function $J(\eta) : \mathbb{R}^{n_o} \to \mathbb{R}$ with the optimization vector $\eta \in \mathbb{R}^{n_o}$, while observing the inequality constraint functions $\psi_i : \mathbb{R}^{n_o} \to \mathbb{R}$ for $i = 1, \ldots, n_c$. The optimization problems as in (3.33) are classified into families of optimization problems by the characteristics of the objective function $J(\cdot)$, and the constraint function $\psi_i(\cdot)$. One very important and well understood class of problems are convex optimization problems, in which the objective function and the constraint function are convex, i.e. they satisfy

$$J(\alpha\eta_1 + \beta\eta_2) \leq \alpha J(\eta_1) + \beta J(\eta_2), \tag{3.34}$$

for all $\eta_1, \eta_2 \in \mathbb{R}^{n_o}$ and all $\alpha, \beta \in \mathbb{R}_{\geq 0}$ with $\alpha + \beta = 1$. The solution of an optimization problem (3.33) consists of finding an optimal vector η^*, which minimizes the objective function $J(\eta^*)$, and satisfies each constraint $\psi_i(\eta^*) \leq 0$. Since the late 1940's many algorithms have been developed for various sub-classes of convex optimization problems (see [42]), and the growth of the processing power in the past decades enables the development of even more and powerful algorithms. However, in this thesis a sub-class of convex optimization problems will be used for controller synthesis, namely *semidefinite programs* (SDP).

An SDP is characterized by a linear objective function $J(\eta)$, and the constraint functions $\psi_i(\eta)$ are defined by *linear matrix inequalities* (LMI), such that

$$\psi_i(\eta) = \eta_1\Delta_1 + \eta_2\Delta_2 + \ldots + \eta_{n_o}\Delta_{n_o} + \Delta_0 \geq 0, \tag{3.35}$$

where $\eta \in \mathbb{R}^{n_o}$ is the optimization variable, and the symmetric matrices $\Delta_i \in \mathbb{R}^{n_o \times n_o}$ are given. Note, the inequality symbol in (3.35) means, that the left side of the inequality has to be positive semi definite.

The formulation of LMI's arise in many results in the analysis of dynamic systems, and the optimization problems including LMI's can be solved numerically very efficiently. Two excellent books on the usage of LMI's in control theory including a historical overview of the development of LMI's and many standard problems can be found in [41] and [58]. In this thesis, the solution of the formulated SDP's is obtained with the help of the open-source software package YALMIP [121], which is a modeling language for convex and non-convex optimization problems. Within YALMIP the commercial solver MOSEK [19] is used to solve the optimization problem.

In many cases, the formulation of an SDP in control theory first leads to a non-linear matrix inequality, which cannot be efficiently solved by the usual algorithms. A very useful relation to overcome this drawback is the, so called, "*Schur's complement*". It is named after the mathematician *Issai Schur*, who published a lemma on matrix theory in [145, pp 215-216], and Emilie Haynsworth introduced the lasting term *Schur complement* in [78, 77]. Schur's complement can be used to characterize symmetric positive definite matrices and is formulated as follows:

Lemma 3.3 (Lemma 2.8 in [58])**.** *Let the partitioned matrix*

$$M = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix} \tag{3.36}$$

be symmetric. Then,

$$\begin{aligned} & M < 0 \Leftrightarrow \\ & M_{11} < 0, M_{22} - M_{21} M_{11}^{-1} M_{12} < 0 \Leftrightarrow \\ & M_{22} < 0, M_{11} - M_{12} M_{22}^{-1} M_{21} < 0 \end{aligned} \tag{3.37}$$

or

$$\begin{aligned} & M > 0 \Leftrightarrow \\ & M_{11} > 0, M_{22} - M_{21} M_{11}^{-1} M_{12} > 0 \Leftrightarrow \\ & M_{22} > 0, M_{11} - M_{12} M_{22}^{-1} M_{21} > 0. \end{aligned} \tag{3.38}$$

△

Proof. See Lemma 2.8 in [58]. □

In Lemma 3.3 the Schur complement is formulated for strict inequalities, and it is shown in [41], that under further conditions the Schur complement is also valid for non-strict matrix inequalities, i.e. $M \geq 0 \Leftrightarrow M_{11} \geq 0, M_{22} - M_{21} M_{11}^{-1} M_{12} \geq 0$, and so on. The Schur complement is very important for the controller synthesis, since it enables the transformation from a quadratic to a linear matrix inequality.

4 Robust Control of Affine Systems with Reachable Set Computation

Reachable set computation for deterministic and even linear dynamic systems can become arbitrary difficult, depending on the used type of sets and the desired accuracy of the computation. The problem gets even harder, if not only the reachable sets are of interest, but instead a controller synthesis is intended in combination with stochastic effects in the dynamics.

This chapter is intended to introduce a set-to-target control problem for deterministic systems as a preparation for the later sections with stochastic effects incorporated in the system dynamics. In many real world applications, the initial state is not exactly known, and can only be measured with a certain, but known, accuracy. This uncertainty affects not only the initial state, but also introduces an uncertainty in each time step, such that the evolution of the state can only be assumed to be in certain bounds, defined by the measurement accuracy. This chapter formulates a set-to-target control problem, which takes into account the mentioned uncertainties in the system state and consequently in the evolution of the state, and the suggested controller synthesis, based on reachable set computation, is able to solve this control problem.

The basic iterative control procedure with a repeated solution of an SDP is introduced here, and the individual LMI's are used later for the probabilistic case with slight adaptions. A naive initial assumption might be, that the computation of probabilistic reachable sets impedes the controller synthesis in contrast to deterministic reachable sets, but, as will be shown later, the controller synthesis is actually alleviated due to different reasons.

This chapter is primarily concerned with the controller synthesis for uncertain systems, in which the uncertainties arise from an uncertain initialization of the system state and disturbed system dynamics, in particular, discrete-time linear dynamics with bounded additive disturbances. The initial state is assumed to be contained in a bounded set, and the control problem is formulated as a reachability problem, such that the uncertain system is transferred from an initial state set into a terminal target region.

The introduced algorithmic solution procedure casts the reachability problem into a solution of an SDP in each time step to obtain a suitable control sequence. The solution of the SDP provides a control input, which guarantees the satisfaction of the input constraint, as well as a stability condition. In addition, this chapter includes the formulation of LMI constraints for two different kinds of set representations for the input constraints, namely polytopic and ellipsoidal sets.

Furthermore, it is shown, that the presented control algorithm for uncertain linear systems can be extended for the controller synthesis for uncertain discrete-time nonlinear systems. Nonlinear dynamics are in general a big challenge in control system theory, and a common approach is linearization. A first-order Taylor series approximation is used to obtain the linear system dynamics, which are again suitable for reachability analysis. In order to provide a reachability computation, which are robust against linearization errors, an over-approximation of the linearization error is considered as an additional disturbance. A core component of this procedure is to compute one-step reachable sets for the substitute models conservatively, i.e. such that any behavior of the original model under effect of a chosen control law and for any possible disturbance value and linearization error is included. Due to the local linearization of the nonlinear system, the substitute system is a switching system, since its dynamics are defined by the current linearization point. This motivates a more sophisticated stability criterion, as for switching systems Hurwitz stability is not sufficient anymore (see [118]). An adequate stability criterion is an relaxation of the well known Lyapunov stability and introduced in this chapter.

The main ideas of this chapter were already reported in [21].

Literature review

The consideration of systems with uncertainties were taken into account amongst others in [99] and [142] for the case of linear dynamics, while representing the reachable set by ellipsoids, or polytopes respectively. Reachable set computation for nonlinear dynamics with uncertainties for the purpose of safety verification was covered in [15] with specifying reachable sets by zonotopes. The latter technique has in common with [144] and [165] that over-approximations of the reachable set are determined based on local linearization of the dynamics around the current (estimated) state. The linearizations are evaluated in conservative manner over regions of the state and input space by using interval arithmetics – this principle is adopted here within the context of the control problem stated above. The description of trajectories of nonlinear systems with uncertain initial conditions and parameters has been considered in [27]. Therein, a two-step procedure is proposed, which provides an ellipsoidal approximation of the one step reachable set for the uncertain system.

The system class of affine systems and the reachability analysis is introduced in Sec. 4.1, and Section 4.2 specifies the chosen control law. The derivation of the LMI's for the SDP is contained in Section 4.3. Therein a distinguished consideration of polytopic and ellipsoidal input constraints is included. The presented control algorithm for affine systems in Section 4.3 is extended in Section 4.4 to nonlinear dynamics. Section 4.5 presents a numerical example of the control algorithm, and Section 4.6 completes this chapter with a discussion of the presented methods.

4.1 Reachability Problem of Discrete-Time Affine Systems

Definition 4.1. *An affine system (AS) is described by the following equations:*

$$x_{k+1} = Ax_k + Bu_k + Gv_k, \tag{4.1a}$$
$$x_0 \in X_0, \tag{4.1b}$$
$$v_k \in V, \tag{4.1c}$$
$$u_k \in U, \tag{4.1d}$$

where $x_k \in \mathbb{R}^n$ is the continuous system state, $u_k \in \mathbb{R}^m$ is the continuous system input, and $v_k \in \mathbb{R}^n$ is the disturbance. The convex and bounded sets X_0 and V are the initial state set and the disturbance set, respectively. The input u_k is bounded to a convex polytope $U = P_H(R_u, b_u) \in \mathcal{P}$ with $R_u \in \mathbb{R}^{n_u \times m}$ and $b_u \in \mathbb{R}^{n_u}$. △

An AS consists of linear difference equations (4.1a), defined by the matrices (A, B, G), with an additional drift term v_k, which is used to model disturbances in the dynamics. The disturbances v_k in AS are assumed to be non-deterministic but bounded to an ellipsoidal set $V = \varepsilon(q_v, Q_v)$.

AS can be interpreted as an uncertain, linear, and non-hybrid version of SHS defined in Def. 3.1, where the random distribution of the initial state and disturbance is replaced by a consideration of bounded sets for each variable. An obvious attempt is to assume the variables to be uniformally distributed over the bounded sets, but this is not mandatory. The lack of information concerning the distribution leads to an uncertain, or non-deterministic system. The update function (3.4a) for the continuous state is determined by a linear difference equation, specified by the matrices A, B, and G. The discrete mode is $z_k = 1, \ \forall\ k$, hence it is omitted for ease of notation. In AS, the system state is not bounded to a state set, such that $\mathbb{X}_k = \mathbb{R}^n$.

The following assumption is a necessary condition for the solvability of the general set-to-target control problem, and ensures the existence of an equilibrium point.

Assumption 4.1. *Let for system* (4.1) *exist an input $\bar{u} \in U$, for which* (4.1a) *has an equilibrium point $\bar{x}$, if the disturbance set V is centered in $\bar{v} = q_v$, i.e. $\bar{x} = q_T = (I - A)^{-1} \cdot (B\bar{u} + Gq_v)$.*

A reachable set $X_k \subseteq \mathbb{R}^n$ for AS at a certain discrete time step k describes all states x_k, that are reachable from a bounded set at the previous time step $k-1$ with all admissible inputs $u_k \in U$ and disturbances $v_k \in V$. In other words, given a bounded and compact set $X_k \in \mathbb{R}^n$, the one-step reachable set for AS at the next time step $k+1$ is the set of states reachable from any $x_k \in X_k$ under the effect of any control input in U and any disturbance in V:

$$X_{k+1} = \{x \in \mathbb{R}^n \mid \exists x_k \in X_k, u_k \in U, v_k \in V \ : \ x_{k+1} = Ax_k + Bu_k + Gv_k\} \tag{4.2}$$

For ease of notation, the set-valued mapping corresponding to the one-step execution of the system dynamics in (4.1a) starting from X_k will be written as:

$$X_{k+1} = AX_k \oplus BU \oplus GV, \tag{4.3}$$

where $\oplus$ is the operator for the Minkowski sum of two sets (cf. Def. 3.3). Eq. (4.3) is the base equation for the following control procedure, which is based on reachability computation. This chapter is concerned with the design of a suitable input to determine a desired reachable set X_{k+1}. The objective is to synthesize a control law $\kappa : \mathbb{R}^n \times \mathbb{N}_0 \to \mathbb{R}^m$ with

$$u_k = \kappa(x_k, k), \tag{4.4}$$

which transfers the system into the target region on a bounded time domain $\tau = \{0, 1, \ldots, N\}, N \in \mathbb{N}_0$. The set-to-target control problem under consideration in this thesis for discrete-time affine systems can be written as follows:

Problem 4.1. *Let an AS according to Def. 4.1 with a terminal region $\mathbb{T} = \varepsilon(q_T, Q_T) \subset \mathbb{R}^n$, which is centered in an equilibrium point ($\bar{x} = q_T$), be given. Determine a control law $\kappa(x_k, k)$ according to* (4.4) *for $x_k \in X_k$ on a time domain $k \in \tau$ such that:*

- *a finite $N \in \mathbb{N}$ exists, for which it applies:*

$$X_{k+1} = AX_k \oplus BU_k \oplus GV, k \in \{0, 1, \ldots, N-1\}, X_N \subseteq \mathbb{T}, \tag{4.5}$$

  i.e. any initial state $x_0 \in X_0$ is transferred into the terminal set $\mathbb{T}$ for $k = N$.

- *the input constraint $u_k \subset U$ is satisfied $\forall k \in \{0, 1, \ldots, N-1\}$*

So far, the general reachable set X_k is only assumed to be compact and bounded, but no specification of the shape has been made yet. As already indicated in the introduction, the considered sets for the reachability computation will be ellipsoids. Even if the true reachable set X_k is not an ellipsoid, efficient methods to compute an over-approximating ellipsoid $\hat{X}_k$, known as *Löwner-John-ellipsoid* (see [79]), exists:

$$X_k \subseteq \hat{X}_k = \varepsilon(q_{x,k}, Q_{x,k}), \tag{4.6}$$

with the center point $q_{x,k} \in \mathbb{R}^n$ and the symmetric and positive-definite shape matrix $Q_{x,k} \in \mathbb{R}^{n \times n}$. A Löwner-John-ellipsoid is a minimum volume ellipsoid containing a given convex set. An established algorithm to compute the minimum volume ellipsoid is known as Khachiyan's algorithm, see [155]. The subsequent sections elaborate on the usage of these reachable sets for controller synthesis.

4.2 Control Law Specification

The objective is to develop an algorithmic method to solve Problem 4.1, i.e. to stabilize the system (4.1) from an initial set X_0 with the given input constraints U and the bounded disturbances V. To detail the control law in (4.4), a local state feedback control law of affine structure is assumed:

$$u_k = \kappa(x_k, k) = -K_k x_k + d_k. \tag{4.7}$$

The set-valued mapping of (4.7) with the ellipsoidal representation of the reachable set results in:

$$\bar{U}_k = \Big\{ u_k \in U \mid \forall x_k \in \hat{X}_k \ : \ u_k = -K_k x_k + d_k \Big\} \to \ \bar{U}_k \in \mathcal{E}. \tag{4.8}$$

Since (4.8) is an affine transformation of the reachable ellipsoid $\hat{X}_k$, the control input set $\bar{U}_k$ is also an ellipsoidal set.

Proposition 4.1. *Given an ellipsoidal over-approximation $\hat{X}_k = \varepsilon(q_{x,k}, Q_{x,k}) \in \mathcal{E}$ of the current reachable set X_k ($X_k \subseteq \hat{X}_k$), and the set-valued mapping of the control law in* (4.8)*, the true reachable set X_{k+1} at the next time step, given by* (4.3)*, is contained in an ellipsoid $\hat{X}_{k+1} = \varepsilon(q_{x,k+1}, Q_{x,k+1})$ computed by:*

$$\hat{X}_{k+1} \supseteq A_{cl,k}\hat{X}_k \oplus GV + Bd_k, \ \textit{with}\ A_{cl,k} = (A - BK_k). \tag{4.9}$$

Proof. By the use of Lemma 3.1, it is possible to find a tight ellipsoidal approximation $\hat{X}_{k+1}$, which contains the result of a Minkowski sum. The application of the set-valued control law in (4.8) results in:

$$\hat{X}_{k+1} \supseteq A\hat{X}_k \oplus B\bar{U}_k \oplus GV, \tag{4.10a}$$
$$\supseteq \Big[A\hat{X}_k + B(-K_k\hat{X}_k + d_k)\Big] \oplus GV, \tag{4.10b}$$
$$\supseteq \underbrace{(A - BK_k)}_{A_{cl,k}} \hat{X}_k \oplus GV + Bd_k. \tag{4.10c}$$

Such that:

$$X_{k+1} \subseteq A_{cl,k}\hat{X}_k \oplus GV + Bd_k \subseteq \hat{X}_{k+1}. \tag{4.11}$$

□

Remark 4.1. *Note, that the Minkowski addition in* (4.10b) *is replaced by an element-wise addition of the reach set $\hat{X}_k$, and the control input ellipsoid $\bar{U}_k$. This is feasible, since the Minkowski sum of the state and input set is unnecessary due to the choice of the state feedback control law, which defines a specific input value $u_k \in \bar{U}_k$ for each state $x_k \in \hat{X}_k$.*

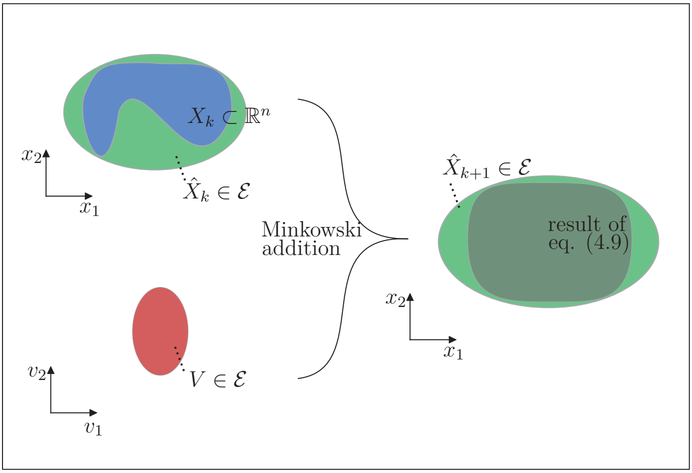


Figure 4.1: Illustration of the over-approximation of a reachable set by the Minkowski addition of two ellipsoids $\hat{X}_k$ and V.

Figure 4.1 illustrates the conclusion of Proposition 4.1. Therein, the non-convex true reachable set X_k at time step k is over-approximated by an ellipsoid $\hat{X}_k$. The disturbance set V is assumed to be of ellipsoidal shape, and furthermore, it is assumed that a feasible control law $u_k = -K_k x_k + d_k$ is available. The application of the controller leads to the closed-loop dynamics, as in (4.9), and the result of the Minkowski addition is shown as the dark green set. This set is neither a polytope, nor an ellipsoid, and therefore, this set has to be over-approximated by an ellipsoid $\hat{X}_{k+1}$. This procedure is repeated in every time step of the control procedure, and in order to handle the amount of over-approximations, the control law in (4.7) has to be chosen, such that the size of the next reachable set does not increase over time, and a convergence of size is achieved.

Besides the convergence of the size, a convergence of the location is desired, that is, the center point $q_{x,k}$ of the reachable ellipsoid should head into the origin, in order to stabilize the system. The components of the chosen controller in (4.7) can be interpreted as follows: the feedback gain K_k has to be chosen such that a contraction of the ellipsoid $\hat{X}_k$ in step k is achieved. The aim of the affine controller component d_k is the convergence of the center point to the origin. To make this obvious, (4.9) can be split as follows: first, the combination of the affine transformation of an ellipsoid in (3.6) and Lemma 3.1 allows to affect the size of the resulting ellipsoid with the state feedback matrix K_k. Second, the dynamics of the center point $q_{x,k}$

can be influenced by both controller parameters (K_k, d_k):

$$q_{x,k+1} = A_{cl,k} q_{x,k} + B d_k + G q_v, \tag{4.12a}$$
$$Q_{x,k+1} = (1 + s^{-1}) A_{cl,k} Q_{x,k} A_{cl,k}^T + (1 + s) G Q_v G^T, \tag{4.12b}$$

with

$$s > 0. \tag{4.13}$$

The controller synthesis will provide stabilizing controller parameters (K_k, d_k) in each time step k, and Section 4.3 derives the control procedure to solve the stated control problem.

4.3 Algorithmic Solution Based on Semi-Definite Programming

The formulated control problem in Problem 4.1 is recast into an iterative optimization problem by formulating linear matrix inequalities to observe the input constraint, the stability property, and to over-approximate the next reachable sets by ellipsoids. This optimization problem needs to be solved in every time step k, in order to transfer the initial reachable set X_0 into the terminal region $\mathbb{T}$, while taking into account all possible disturbances $v_k \in V$.

To satisfy the stability property for the center point $q_{x,k}$, a time-invariant quadratic Lyapunov function

$$V(q_{x,k}) = q_{x,k}^T M q_{x,k} \tag{4.14}$$

is employed[1]. Note, that the actual formulation of the Lyapunov function should use the difference vector $(q_{x,k} - q_T)$ for evaluation. But Assumption 4.2 always allows to find a suitable coordinate transformation, which recasts the problem into a transition-to-origin problem.

The Lyapunov stability is enforced by the following formulation:

$$V(q_{x,k+1}) - \rho V(q_{x,k}) \leq 0, \tag{4.15a}$$
$$q_{x,k+1}^T M q_{x,k+1} - \rho q_{x,k}^T M q_{x,k} \leq 0, \tag{4.15b}$$

with $q_{x,k+1}$ according to (4.12), $\rho \in (0, 1]$, and M is positive-definite symmetric matrix ($M = M^T > 0$). The parameter ρ can be used to define a certain threshold for the decrease of the Lyapunov function depending on the current value of $V(q_{x,k})$ at time step k. (4.15a) is a convex quadratic inequality and will be used as a

[1] The theory on Lyapunov stability and Lyapunov functions for discrete-time linear systems is well known, and therefore not repeated at this point

constraint in the optimization problem to ensure the convergence of the center point $q_{x,k}$.

The over-approximation of the reachable ellipsoid $\hat{X}_{k+1}$, based on the information of the current reachable ellipsoid $\hat{X}_k$, is obtained by linear matrix inequalities and formulated in the following lemma:

Lemma 4.1. *Given an AS, as in* (4.1)*, with an ellipsoidal reachable set* $\hat{X}_k = \varepsilon(q_{x,k}, Q_{x,k})$ *and an ellipsoidal disturbance set* $V = \varepsilon(q_v, Q_v)$*. The resulting reachable set at the next time step is over-approximated by the ellipsoid* $\varepsilon(q_{x,k+1}, S_{k+1})$*, if center point* $q_{x,k+1}$ *follows from* (4.12)*, and the following LMI holds for the shape matrix* S_{k+1}*:*

$$\begin{bmatrix} S_{k+1} & A_{cl,k}Q_{x,k} & GQ_v \\ Q_{x,k}A_{cl,k}^T & (1-\nu)Q_{x,k} & 0 \\ Q_vG^T & 0 & \nu Q_v \end{bmatrix} \geq 0, \tag{4.16}$$

with

$$\nu \in [0,1]. \tag{4.17}$$

Proof. The reachable ellipsoid $\hat{X}_{k+1}$ must contain the Minkowski sum of the current reachable ellipsoid $\hat{X}_k$ and the disturbance ellipsoid V, as shown in (4.9). According to Lemma 3.1, the shape matrix of the resulting ellipsoid in (4.12b) is a weighted combination of the added shape matrices. The shape matrix is over-approximated by a new matrix S_{k+1}:

$$S_{k+1} \geq Q_{x,k+1}, \tag{4.18a}$$

$$S_{k+1} \geq (1+s^{-1})A_{cl,k}Q_{x,k}A_{cl,k}^T + (1+s)GQ_vG^T, \tag{4.18b}$$

$$S_{k+1} - (1+s^{-1})A_{cl,k}Q_{x,k}A_{cl,k}^T - (1+s)GQ_vG^T \geq 0. \tag{4.18c}$$

By applying the Schur complement, (4.18c) is transferred into

$$\begin{bmatrix} S_{k+1} & A_{cl,k}Q_{x,k} & GQ_v \\ Q_{x,k}A_{cl,k}^T & \frac{1}{(1+s^{-1})}Q_{x,k} & 0 \\ Q_vG^T & 0 & \frac{1}{(1+s)}Q_v \end{bmatrix} \geq 0. \tag{4.19}$$

Due to the nonlinear scalar multiplication with the factor s, the above inequality is not linear, hence a further transformation is necessary. The scalar value $s > 0$ is replaced by a new variable $\nu \in [0,1]$ with the following relation:

$$\nu = \frac{1}{1+s} \leftrightarrow s = \frac{1-\nu}{\nu}, \tag{4.20}$$

which results in

$$\left.\frac{1}{1+s}\right|_{s=\frac{1-\nu}{\nu}} = \nu, \quad \left.\frac{1}{1+s^{-1}}\right|_{s=\frac{1-\nu}{\nu}} = 1-\nu. \tag{4.21}$$

With this transformation, (4.19) becomes a linear matrix inequality as in (4.16). $\square$

The LMI (4.16) can be used to consider the size and shape of the next reachable ellipsoid in the value function of an optimization problem.

LMI-Formulation for Polytopic Input Constraints

The input constraint has to be satisfied in each time step of the controller synthesis. To this end, two different formulations of the input constraint set U are possible. First, the input set can be considered as a polytope $U \in \mathcal{P}$, which is quite intuitive assumption, since most of the input signals are bounded by intervals, in practice. An interval in n-dimension is (see Sec. 3.2) a subclass of polytopes. The second possibility to represent input constraints are ellipsoids $U \in \mathcal{E}$. The discussed advantages in Sec. 2.2 of ellipsoids compared to polytopes in reachability computation might suggest the use of ellipsoids for the input constraints as well. But, as will be presented in the following, the consideration of ellipsoidal input constraints is not as convenient as considering polytopes.

The considered polytopic input set will be denoted as:

$$U = P_H(R_u, b_u) \in \mathcal{P}, \tag{4.22}$$

with $R_u \in \mathbb{R}^{m \times n_u}, b \in \mathbb{R}^{n_u}$.

Proposition 4.2. *Given a polytopic input set U, as in (4.22), the input constraint $u_k = -K_k x_k + d_k \in \bar{U}_k \subseteq U$ holds for K_k, d_k, and $x_k \in \hat{X}_k = \varepsilon(q_{x,k}, Q_{x,k})$, if the following LMI holds:*

$$\begin{bmatrix} (b_{u,i} - r_{u,i}(d_k - K_k q_{x,k}))I_n & -r_{u,i} K_k Q_{x,k}^{1/2} \\ (-r_{u,i} K_k Q_{x,k}^{1/2})^T & b_{u,i} - r_{u,i}(d_k - K_k q_{x,k}) \end{bmatrix} \geq 0, \quad \forall i = \{1, \ldots, n_u\} \tag{4.23}$$

Proof. The half-plane representation of U is given by:

$$R_u u_k \leq b_u, \tag{4.24}$$

and by substituting the input u_k in the constraint (4.24) with the control law (4.7), the input constraint has to be satisfied for each $x_k \in \hat{X}_k$:

$$R_u(-K_k x_k + d_k) \leq b_u, \ \forall x_k \in \hat{X}_k \tag{4.25}$$

By a row-wise maximization of the linear inequality in (4.25), the universal quantifier can be eliminated and the following condition is obtained:

$$W = \{w \in \mathbb{R}^m \mid w = -K_k x_k + d_k, \ x_k \in \hat{X}_k\}, \tag{4.26a}$$

$$\max_{w \in W} r_{u,i} w \leq b_{u,i}, \ \forall i = \{1, \ldots, n_u\}, \tag{4.26b}$$

where $r_{u,i}$ is the i-th row of the matrix R_u, and $b_{u,i}$ is the i-th entry of the vector b_u. The ellipsoid $\hat{X}_k$ can be mapped into a unit ball by a change of variables according to $\theta = Q_{x,k}^{-1/2}(x_k - q_{x,k})$ $(||\theta||_2 \leq 1)$. This yields:

$$W = \{w \in \mathbb{R}^m \mid w = -K_k Q_{x,k}^{1/2}\theta - K_k q_{x,k} + d_k,\ ||\theta||_2 \leq 1\}. \tag{4.27}$$

The maximization problem in (4.26b) subject to (4.27) can be recast as follows:

$$\max_{w \in W} r_{u,i} w \leq b_{u,i}, \tag{4.28a}$$

$$\max_{w \in W} r_{u,i}(-K_k Q_{x,k}^{1/2}\theta) - r_{u,i}K_k q_{x,k} + r_{u,i}d_k \leq b_{u,i}, \tag{4.28b}$$

$$\max_{w \in W} || - r_{u,i}K_k Q_{x,k}^{1/2}||_2 \leq b_{u,i} - r_{u,i}(d_k - K_k q_{x,k}). \tag{4.28c}$$

Finally, the Euclidean norm in (4.28c) can be expressed as an LMI, according to [130], resulting in (4.23). □

Note, that Proposition 4.2 generates a set of n_u LMI constraints for the optimization problem.

LMI-Formulation for Ellipsoidal Input Constraints

Besides the polytopic input constraint an ellipsoidal input constraint is presented in this thesis. Whereas the polytopic input constraint is an ellipse-in-polytope problem, the ellipsoidal input constraint is an ellipse-in-ellipse problem, which is much more difficult to embed into an SDP as LMI. The considered ellipsoidal input set will be denoted as:

$$U = \varepsilon(p, P) \in \mathcal{E}, \tag{4.29}$$

with $p \in \mathbb{R}^m$, $P = P^T \in \mathbb{R}^{m \times m}$, $P \geq 0$. To satisfy the input constraint

$$\bar{U}_k \subseteq U \tag{4.30}$$

in each time step k, the LMI formulation for an *ellipse-in-ellipse* problem is recalled:

Lemma 4.2. *([41]) Given two ellipsoids $E_0 = \varepsilon(p_0, P_0^2)$ and $E_1 = \varepsilon(p_1, P_1^2)$, the ellipsoid E_0 is contained in E_1, if and only if it holds for every $p \in E_0$ that:*

$$(p - p_1)^T P_1^{-2}(p - p_1) \leq 1. \tag{4.31}$$

Using the S-procedure (see [41]), this is equivalent to the existence of a nonnegative $s \in \mathbb{R}^{\geq 0}$, satisfying $\forall p \in E_0$:

$$(p - p_1)^T P_1^{-2}(p - p_1) - s(p - p_0)^T P_0^{-2}(p - p_0) \leq 1 - s, \tag{4.32}$$

or respectively:

$$\begin{bmatrix} -P_1^2 & p_1 - p_0 & P_0 \\ (p_1 - p_0)^T & s-1 & 0 \\ P_0 & 0 & -sI \end{bmatrix} \leq 0. \tag{4.33}$$

Therefore, the ellipsoid of largest volume contained in E_1 is obtained by solving the convex program:

$$\min_{P_0, p_0, s} \; logdet(P_0^{-1}), \tag{4.34}$$

$$subject\ to\ (4.33), P_0 > 0,\ s \geq 0. \tag{4.35}$$

Proof. See Sec. 3.7.3 of [41] □

Proposition 4.3. *Given an ellipsoidal input set $U \in \mathcal{E}$, as in (4.29), the input constraint $u_k = -K_k x_k + d_k \in \bar{U}_k \subseteq U$ holds for K_k, d_k, and $x_k \in \hat{X}_k = \varepsilon(q_{x,k}, Q_{x,k})$, if the following set of LMI holds:*

$$\begin{bmatrix} \bar{P}_k & K_k Q_{x,k} \\ Q_{x,k} K_k^T & Q_{x,k} \end{bmatrix} \geq 0, \tag{4.36a}$$

$$\begin{bmatrix} -P & p + K_k q_k - d_k & \tilde{P}_k \\ (p + K_k q_{x,k} - d_k)^T & s-1 & 0 \\ \tilde{P}_k & 0 & -sI \end{bmatrix} \leq 0, s \geq 0, \tag{4.36b}$$

$$\bar{P}_k \leq \alpha_P^2 P - \sum_{jh}^{m} \left(\left. \frac{\partial g(\tilde{P}_k)}{\partial \tilde{p}_{k,jh}} \right|_{\tilde{P}_k = P_L} e_j^T (\tilde{P}_k - \alpha_P P^{1/2}) e_h \right), \tag{4.36c}$$

$$0 \leq \tilde{P}_k \leq P^{1/2},, \tag{4.36d}$$

with $\bar{P}_k = \bar{P}_k^T \in \mathbb{R}^{m \times m}$, $\tilde{P}_k = \tilde{P}_k^T \in \mathbb{R}^{m \times m}$, $\bar{P}_k, \tilde{P}_k \geq 0$, $\alpha_P \in [0,1]$, and the non-convex function $g(\tilde{P}_k) : \mathbb{R}^m \to \mathbb{R}^m$. The identity $\tilde{p}_{k,jh} = e_j^T \tilde{P}_k e_h$ with the unit vector $e_j \in \mathbb{R}^{m \times 1}$ is used.

Proof. Lemma 4.2 enables to specify an LMI formulation for the ellipse-in-ellipse problem in (4.30). The shape matrix of the feasible input ellipsoid $\bar{U}_k \in \mathcal{E}$ is given by $K_k Q_{x,k} K_k^T$. In order to obtain an LMI, the shape matrix is over-approximated by the matrix variable $\bar{P}_k$:

$$\bar{P}_k \geq K_k Q_{x,k} K_k^T. \tag{4.37}$$

With the Schur complement, (4.37) is transformed into:

$$\begin{bmatrix} \bar{P}_k & K_k Q_{x,k} \\ Q_{x,k} K_k^T & Q_{x,k} \end{bmatrix} \geq 0 \tag{4.38}$$

Given Lemma 4.2, the following formulation ensures that the ellipsoid $\varepsilon(-K_k q_{x,k} + d_k, \bar{P}) \supseteq \bar{U}_k$ is contained in U:

$$\begin{bmatrix} -P & p + K_k q_{x,k} - d_k & \bar{P}_k^{1/2} \\ (p + K_k q_{x,k} - d_k)^T & s-1 & 0 \\ \bar{P}_k^{1/2} & 0 & -sI \end{bmatrix} \leq 0,\ s \geq 0. \tag{4.39}$$

Obviously, (4.39) is nonlinear in the matrix variable $\bar{P}_k$. To obtain a linear inequality, the matrix $\tilde{P}_k \geq \bar{P}_k^{1/2}$ is introduced, and (4.39) follows to:

$$\begin{bmatrix} -P & p + K_k q_k - d_k & \tilde{P}_k \\ (p + K_k q_{x,k} - d_k)^T & s - 1 & 0 \\ \tilde{P}_k & 0 & -sI \end{bmatrix} \leq 0. \tag{4.40}$$

The relation:

$$\tilde{P}_k \geq \bar{P}_k^{1/2} \tag{4.41}$$

is nonconvex, and therefore not directly usable within an SDP. To cope with the nonconvexity, a constraint function $g(\tilde{P}_k) \leq 0$ is defined first:

$$g(\tilde{P}_k) = \bar{P}_k - \tilde{P}_k^2 \leq 0. \tag{4.42}$$

A first-order Taylor approximation around a linearization point $P_L \in \mathbb{R}^{m \times m}$ is used to obtain a convex constraint $\tilde{g}(\tilde{P}_k)$:

$$g(\tilde{P}_k) \approx \tilde{g}(\tilde{P}_k) = g(P_L) + dg(\tilde{P}_k) \leq 0. \tag{4.43}$$

The total derivative of $dg(\tilde{P}_k)$ is obtained by a matrix differentiation, and the conservatively approximating constraint becomes:

$$\tilde{g}(\tilde{P}_k) = g(P_L) + \sum_{j,h}^{m} \left(\left. \frac{\partial g(\tilde{P}_k)}{\partial \tilde{p}_{k,jh}} \right|_{\tilde{P}_k = P_L} d\tilde{p}_{k,jh} \right), \tag{4.44a}$$

$$\begin{aligned} &= g(P_L) + \left. \frac{\partial g(\tilde{P}_k)}{\partial \tilde{p}_{k,11}} \right|_{\tilde{P}_k = P_L} d\tilde{p}_{k,11} + \dots \\ &\quad \dots + \left. \frac{\partial g(\tilde{P}_k)}{\partial \tilde{p}_{k,mm}} \right|_{\tilde{P}_k = P_L} d\tilde{p}_{k,mm}, \end{aligned} \tag{4.44b}$$

$$\begin{aligned} &= \bar{P}_k - P_L^2 + \left. \frac{\partial g(\tilde{P}_k)}{\partial \tilde{p}_{k,11}} \right|_{\tilde{P}_k = P_L} d\tilde{p}_{k,11} + \dots \\ &\quad \dots + \left. \frac{\partial g(\tilde{P}_k)}{\partial \tilde{p}_{k,mm}} \right|_{\tilde{P}_k = P_L} d\tilde{p}_{k,mm} \leq 0, \end{aligned} \tag{4.44c}$$

with:

$$d\tilde{p}_{k,jh} = \tilde{p}_{k,jh} - p_{L,jh}.$$

The range of the over-approximating matrix $\tilde{P}_k$, and therefore, of the linearization point P_L is limited by the shape matrix of the input ellipsoid U:

$$0 \leq \tilde{P}_k \leq P^{1/2} \rightarrow P_L = \alpha_P P^{1/2},\ \alpha_P \in [0, 1]\,. \tag{4.45}$$

With this expression, the resulting constraints for $\bar{P}_k$ and $\tilde{P}_k$ are:

$$0 \leq \tilde{P}_k \leq P^{1/2}, \tag{4.46}$$

$$\bar{P}_k \leq \alpha_P^2 P - \left.\frac{\partial g(\tilde{P}_k)}{\partial \tilde{p}_{k,11}}\right|_{\tilde{P}_k = P_L} (\tilde{p}_{k,11} - \alpha_P p_{11}^{1/2}) - \ldots$$
$$\ldots - \left.\frac{\partial g(\tilde{P}_k)}{\partial \tilde{p}_{k,mm}}\right|_{\tilde{P}_k = P_L} (\tilde{p}_{k,mm} - \alpha_P p_{mm}^{1/2}). \tag{4.47}$$

In order to obtain a matrix inequality, which is linear in $\bar{P}_k$ and $\tilde{P}_k$, (4.47) can be rewritten by the use of the identity $\tilde{p}_{k,jh} = e_j^T \tilde{P}_k e_h$ with the unit vector $e_j \in \mathbb{R}^{m\times 1}$, and (4.36c) is obtained, which completes the proof. □

The set of LMI's in (4.36) includes a first order Taylor linearization of the non-convex constraint in (4.41) at a linearization point P_L, which is specified by the parameter α_P in (4.45). Since the approximation error, and thus the conservatism, tends to increase with an increasing distance between the linearization point P_L and the evaluation point $\tilde{P}_k$, it is advisable to use several linearization points $\alpha_{P,i} \in [0,1]$, $i = \{1,\ldots,n_{\alpha P}\}$. A graphical illustration of this linearization is given in Fig. 4.2 for a scalar example.

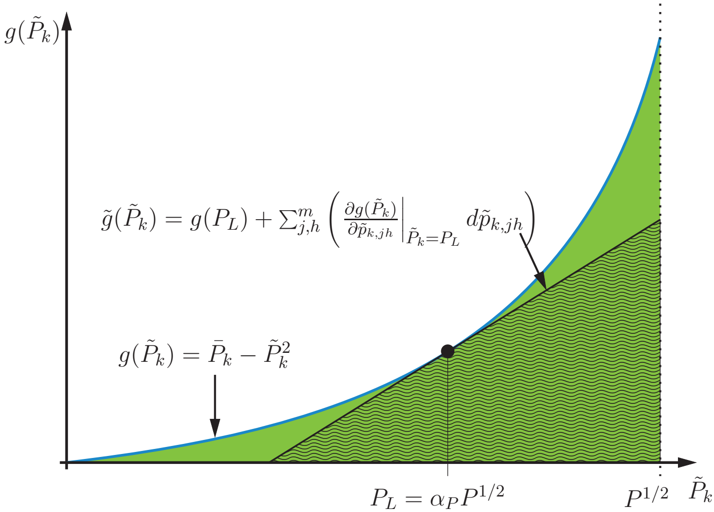


Figure 4.2: This graphic illustrates the application of the first order Taylor approximation to the non convex function $g(\tilde{P}_k)$. The approximating linear constraint $\tilde{g}(\tilde{P}_k) \leq 0$ is reformulated in (4.47) to obtain an LMI.

For ease of notation, and without loss of generality, the input constraint is assumed to be of polytopic shape ($U \in \mathcal{P}$) in the remaining parts of this thesis, keeping in mind, that the following formulations for the controller synthesis also hold for the case of ellipsoidal input constraints ($U \in \mathcal{E}$).

Control Algorithm based on Reachability Computation

The collection of the previously introduced LMI constraints results in the following SDP, and the solution of this SDP provides stabilizing controller parameters (K_k, d_k):

$$\min_{S_k, K_k, d_k, \nu} J(\hat{X}_{k+1}) = \text{trace}\left(\begin{bmatrix} \mu_0 S_k & 0 & 0 \\ 0 & \mu_1 ||q_{x,k+1}|| & 0 \\ 0 & 0 & \mu_2 ||u_k|| \end{bmatrix}\right), \tag{4.48a}$$

subject to:

$$q_{x,k+1}^T M q_{x,k+1} - \rho q_{x,k}^T M q_{x,k} \leq 0, \tag{4.48b}$$

$$q_{x,k+1} = A_{cl,k} q_{x,k} + B d_k + G q_v, \tag{4.48c}$$

$$\begin{bmatrix} S_{k+1} & A_{cl,k} Q_{x,k} & G Q_v \\ Q_{x,k} A_{cl,k}^T & (1-\nu) Q_{x,k} & 0 \\ Q_v G^T & 0 & \nu Q_v \end{bmatrix} \geq 0, \tag{4.48d}$$

$$\text{trace}(S_k) \leq \text{trace}(Q_{x,k}), \tag{4.48e}$$

$$\begin{bmatrix} (b_{u,i} - r_{u,i}(d_k - K_k q_{x,k})) I_n & -r_{u,i} K_k Q_{x,k}^{1/2} \\ (-r_{u,i} K_k Q_{x,k}^{1/2})^T & b_{u,i} - r_{u,i}(d_k - K_k q_{x,k}) \end{bmatrix} \geq 0, \tag{4.48f}$$

$$\forall i = \{1, \ldots, n_u\}.$$

The value function of the optimization problem (4.48) is chosen with a weighted combination of the trace of the over-approximating shape matrix S_k, the norm of the next center point $q_{x,k+1}$, and the control effort u_k to evaluate the reachable ellipsoid of the next time step. In general, different value functions $J(\cdot)$ can be used to evaluate the size and shape of an ellipsoids. The two most important cases are $J(\varepsilon(q_{x,k}, Q_{x,k})) = \det(Q_{x,k})$ and $J(\varepsilon(q_{x,k}, Q_{x,k})) = trace(Q_{x,k})$. The first one, the determinant of the shape matrix, scales with the volume of an ellipsoid, whereas the latter one equals the sum of squared semi-axis [100].

The use of the volume of an ellipsoid for the value function of an optimization problem might lead to a solution, where only one semi-axes or equivalently one eigenvalue is zero. A semi-axis of zero length leads to a volume of zero, and hence to a minimum of the value function. But the remaining semi-axis can get arbitrary large, since the determinant of a matrix is the product of each eigenvalue, and if a single eigenvalue is zero the result will be zero, no matter how large the remaining eigenvalues are. Furthermore, the choice of zero for the length of a semi-axes, and the corresponding eigenvalue, leads to a degenerate ellipsoid, i.e. the ellipsoid is defined on $\mathbb{R}^n$, but due to the singular shape matrix the ellipsoid has only a dilation in $\mathbb{R}^{n-1}$ (see [100]).

The second optimality criterion $trace(Q_{x,k})$ is more suitable to assess the shape of an ellipsoid. The trace of a square matrix is defined by the sum of the diagonal elements, and the diagonal elements of a shape matrix define the squared size of the semi-axis (see Sec. 3.2 and [100]). Thus, the trace of a shape matrix is the sum of the squared semi-axis. By minimizing the trace, the size of each semi-axis has to decrease, and a value of zero is only possible, if the eigenvalues of each semi-axes is zero (reduction to a singleton). The unpleasant effects of degenerate ellipsoid, and a combination of very large and small eigenvalues can be omitted by the use of the trace as a value function. Therefore, the value function in (4.48a) evaluates the trace of the over-approximated matrix (S_k). The evaluation of the norm of the next center point leads to an attractiveness of the center point to the origin. The consideration of the norm in the value function of the optimization problem is, in addition to the concept of Lyapunov, an additional tool to compel a stabilizing behavior of the controller. The third entry of the value function (4.48a), the norm of the control input u_k, allows to tune the impact of the controller on the system state. If the current application is sensitive against large input values, the weight μ_2 can be used to generate a solution of the optimization problem with small values for the input value, if existing.

As already introduced at the beginning of this section, the first two constraints (4.48b)-(4.48c) are responsible for the stabilization of the center point $q_{x,k}$ by applying the Lyapunov stability criterion. The LMI in (4.48d), together with the constraint in (4.48e), leads to a contraction of the reachable ellipsoid $\hat{X}_{k+1}$, and depending on the current input constraint the LMIs in (4.48f) ensure the satisfaction of the input constraints. In combination of all the LMI constraints, the solution of the optimization problem provides a stabilizing controller for discrete-time uncertain affine systems.

The complete procedure to compute a stabilizing controller based on the over-approximating reachability analysis is described in Algorithm 4.1. The initial step solves the stated SDP in (4.48), and if no feasible solution can be computed, the algorithm, and hence the controller synthesis, fails. In this case, a new parametrization of the ECA-AS could help to render the SDP solvable, e.g. the variable ρ for the threshold of the Lyapunov function could be adjusted, or the considered disturbance set V and the initial uncertain state set X_0 could be adjusted.

Assuming a feasible solution is available, the consecutive reachable ellipsoid $\hat{X}_{k+1}$ is computed according to (4.9) in step two. Algorithm 4.1 iteratively solves the optimization problem and terminates successfully, if the over-approximating reachable set $\hat{X}_k$ is contained in the target set $\mathbb{T}$. If the optimization problem provides a feasible solution in every iteration of the algorithm, but no reduction of the distance between the origin and the center point $q_{x,k}$ occurs, an additional exit criterion is used:

$$\pi_{k+1} = ||q_{x,k+1} - q_{x,k}|| \leq \pi_{min}. \tag{4.49}$$

This criterion is computed in the third step of the algorithm and, uses the decreasing rate π_{k+1} to evaluate the step size between $q_{x,k}$ and $q_{x,k+1}$, and if this decreasing rate falls under a lower bound π_{min} the algorithm terminates, as well.

Algorithm 4.1.
Ellipsoidal Control Algorithm for Affine Systems (ECA-AS)

Given: AS as in (4.1)*, initial set* $X_0 = \varepsilon(q_{x,0}, Q_{x,0})$, $v_k \in V = \varepsilon(q_v, Q_v)$*, and* $U \in \mathcal{P}$*, as well as* $\mathbb{T}$*,* π_{min}*,* $\rho \in (0, 1]$*.*
Define: $k := 0$, $\pi_0 = \pi_{min}$, $\hat{X}_0 \supseteq X_0$
while $\hat{X}_k \nsubseteq \mathbb{T}$ *and* $\pi_k \geq \pi_{min}$ ***do***

1. *solve the optimization problem* (4.48)
   if *no feasible solution is found* ***do***
   stop algorithm (synthesis failed)
   end if
2. *compute the reachable set* $\hat{X}_{k+1} = \varepsilon(q_{x,k+1}, Q_{x,k+1})$ *according to* (4.9).
3. $\pi_{k+1} := ||q_{x,k+1} - q_{x,k}||$
4. $k := k + 1$

end while

Lemma 4.3. *If Algorithm 4.1 terminates with* $\hat{X}_N \subseteq \mathbb{T}$*, Problem 4.1 is successfully solved and a control law* (4.7) *exists, which steers any initial state* $x_0 \in X_0$ *into the target set* $\mathbb{T}$ *in* N *steps for all possible disturbances* $v_k \in V$*. Furthermore, the input constraint* $u_k \in U$ *holds for all* $0 < k < N$*, and the center point* $q_{x,k}$ *of the reach set* $\hat{X}_k$ *asymptotically converges to the origin.*

Proof. According to Proposition 4.1, it holds that $X_k \subseteq \hat{X}_k$, i.e. the true reach set at each time step is over-approximated by the ellipsoid $\hat{X}_k$. The reach set $\hat{X}_{k+1}$ at time $k + 1$ is computed in Algorithm 4.1 for all $x_k \in \hat{X}_k \supseteq X_k$ and all disturbances $v_k \in V$. Thus, it follows, that $X_{k+1} \subseteq \hat{X}_{k+1}$, and by induction $x_k \in X_k \subseteq \hat{X}_k$ for all $k > 0$. Successful termination of Algorithm 4.1 implies that $\hat{X}_N \in \mathbb{T}$ and consequently $x_N \in X_N \subseteq \mathbb{T}$ holds for all initial states $x_0 \in X_0$ and all disturbances $v_k \in V$. It follows, that the input constraint $u_k \in U$ holds at each time step, if $u_k = -K_k x_k + d_k \in U$ holds for all $x_k \in \hat{X}_k$, which is established in Proposition 4.2/4.3. Finally, the convergence of the center point $q_{x,k}$ is achieved by the satisfaction of the Lyapunov condition (4.15a), which follows from the successful solution of the SDP in each time step k. □

4.4 Nonlinear Systems

The behavior of many real world processes cannot be characterized by strict linear difference equations, as in AS. The majority of real world processes is described by nonlinear difference equations, and this section extends the previously introduced reachability based controller synthesis for nonlinear systems.

Definition 4.2. *A discrete-time nonlinear affine disturbed system (NADS) is described by the following equations:*

$$x_{k+1} = f(x_k, u_k) + Gv_k, \tag{4.50a}$$
$$x_0 \in X_0, \tag{4.50b}$$
$$v_k \in V, \tag{4.50c}$$
$$u_k \in U, \tag{4.50d}$$

where $x_k \in \mathbb{R}^n$ is the continuous system state, $u_k \in \mathbb{R}^m$ is the continuous system input, and $v_k \in \mathbb{R}^n$ is the disturbance. The convex and bounded sets X_0 and V are the initial state set and the disturbance set, respectively. The input u_k is bounded to a convex polytope $U = P_H(R_u, b_u) \in \mathcal{P}$ with $R_u \in \mathbb{R}^{n_u \times m}$ and $b_u \in \mathbb{R}^{n_u}$. △

The transfer of the current state x_k to the next state x_{k+1} under the impact of the current input value u_k is determined by the nonlinear function $f(x_k, u_k) : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$.

Assumption 4.2. *Let for system* (4.50) *exist an input $\bar{u} \in U$, for which* (4.50a) *has an equilibrium point $\bar{x}$, if the disturbance set V is centered in $\bar{v} = q_v$, i.e. $\bar{x} = q_T = (I - A)^{-1} \cdot (B\bar{u} + Gq_v)$.*

Equivalently to the linear case, Assumption 4.2 ensures that the nonlinear system has an equilibrium point in q_T.

In general, this type of system class is hard to control, if no assumptions on the structure of $f(x_k, u_k)$ is made, especially if the control approach is based on reachability computation. Even if the reachable set at time step k is convex, the set valued evaluation of (4.50a) can produce a non-convex set for the successive reachable set. This non-convex set could be again over-approximated by an ellipsoid, but the conservatism of this approach can grow arbitrary large. The goal is to introduce an algorithmic approach with a minimum amount of over-approximation with an acceptable computational effort.

Given a bounded and compact set $X_k \in \mathbb{R}^n$, the one-step reachable set for NADS at the next time step $k+1$ is the set of states reachable from any $x_k \in X_k$ under the effect of any control input $u_k \in U$, and any disturbance in V:

$$X_{k+1} = \{x \in \mathbb{R}^n \mid \exists x_k \in X_k, u_k \in U, v_k \in V \ : \ x_{k+1} = f(x_k, u_k) + Gv_k.\} \tag{4.51}$$

For ease of notation, the set-valued mapping corresponding to the one-step execution of the system dynamics in (4.50a) starting from X_k will be written as:

$$X_{k+1} = F(X_k, U) \oplus GV, \tag{4.52}$$

The objective is to synthesize a control law $\kappa : \mathbb{R}^n \times \mathbb{N}_0 \to \mathbb{R}^m$ with

$$u_k = \kappa(x_k, k), \tag{4.53}$$

which transfers the system into the target region on a bounded time domain $\tau = \{0, 1, \ldots, N\}, N \in \mathbb{N}_0$.

By analogy to AS, the control problem for NADS is formulated as follows:

Problem 4.2. *Let an NADS according to Def. 4.50 with a terminal region $\mathbb{T} = \varepsilon(q_T, Q_T) \subset \mathbb{R}^n$, which is centered in an equilibrium point ($\bar{x} = q_T$), be given. Determine a control law $\kappa(x_k, k)$ according to (4.53) for $x_k \in X_k$ on a time domain $k \in \tau$ such that:*

1. *a finite $N \in \mathbb{N}$ exists, for which it applies:*

$$X_{k+1} = F(X_k, U_k) \oplus GV, k \in \{0, 1, \ldots, N-1\}, X_N \subseteq \mathbb{T}, \tag{4.54}$$

   i.e. any initial state $x_0 \in X_0$ is transferred into the terminal set $\mathbb{T}$ for $k = N$.

2. *$\bar{U}_k \subset U$ is a bounded and compact set $\forall k \in \{0, 1, \ldots, N-1\}$*

This control problem is quite more challenging compared to the linear case and Problem 4.1, since the non-linear set-valued equation in (4.54) requires some additional computational effort. The proposed approach to solve Problem 4.2 is introduced in the following sections.

4.4.1 Conservative Linearization Procedure

In order to be able to develop a control algorithm, which builds on local linearizations of the nonlinear difference equation (4.50a) while accounting for set-based computation, this subsection addresses a conservative linearization procedure. For concise notation, a vector $\zeta_k = [x_k, u_k]^T \in (X_k \times U)$, $X_k \in \mathcal{E}, U \in \mathcal{P}$ is introduced, and a linearization point is denoted by $\bar{\zeta}_k = [\bar{x}_k, \bar{u}_k]^T$. The function $f(x_k, u_k) = f(\zeta_k)$ is approximated by a first-order Taylor series with a Lagrange remainder $L(\zeta_k, h_L) \in \mathbb{R}^n$:

$$f(\zeta_k) \approx f(\bar{\zeta}_k) + \left.\frac{\partial f(\zeta_k)}{\partial \zeta_k}\right|_{\zeta_k = \bar{\zeta}_k} (\zeta_k - \bar{\zeta}_k) + L(\bar{\zeta}_k, h_L), \tag{4.55}$$

where h_L is a point in the space $X_k \times U$ defined by $h_L \in \left\{\bar{\zeta}_k + \alpha_h(\zeta_k - \bar{\zeta}_k) \mid \alpha_h \in [0,1]\right\}$.

According to the mean value theorem [8], a point h_L exists for ζ_k being chosen in the neighborhood of $\bar{\zeta}_k$ such that the approximation by the Taylor series becomes exact:

$$\begin{aligned} &\exists h_L \in \left\{\bar{\zeta}_k + \alpha_h(\zeta_k - \bar{\zeta}_k) \mid \alpha_h \in [0,1]\right\} \leftrightarrow \\ &f(\zeta_k) = f(\bar{\zeta}_k) + \left.\frac{\partial f(\zeta_k)}{\partial \zeta_k}\right|_{\zeta_k = \bar{\zeta}_k} (\zeta_k - \bar{\zeta}_k) + L(\bar{\zeta}_k, h_L)^{.} \end{aligned} \tag{4.56}$$

The Lagrange remainder $L(\bar{\zeta}_k, h_L)$ exactly accounts for all terms of order higher than one in $(\zeta_k - \bar{\zeta}_k)$, see e.g. [17]. The i-th component $L_i(\bar{\zeta}_k, h_L)$ of the Lagrange remainder is a second-order Taylor term formulating the linearization error by:

$$L_i(\bar{\zeta}_k, h_{L,i}) = \frac{1}{2}(\zeta_k - \bar{\zeta}_k)^T \frac{\partial^2 f_i(\zeta_k)}{\partial^2 \zeta_k}\bigg|_{\zeta_k = h_{L,i}} (\zeta_k - \bar{\zeta}_k). \tag{4.57}$$

Using the Lagrange remainder with appropriately chosen h_L, the nonlinear difference equation (4.50a) can be transformed into:

$$\begin{aligned} x_{k+1} = f(\bar{x}_k, \bar{u}_k) + &\underbrace{\frac{\partial}{\partial x_k} f(x_k, u_k)\bigg|_{\substack{x_k=\bar{x},\\ u_k=\bar{u}}}}_{A_k} (x_k - \bar{x}_k) + \dots \\ &\dots \underbrace{\frac{\partial}{\partial u_k} f(x_k, u_k)\bigg|_{\substack{x_k=\bar{x},\\ u_k=\bar{u}}}}_{B_k} (u_k - \bar{u}) + \dots \\ &\dots L(\bar{\zeta}_k, h_L) + G v_k, \end{aligned} \tag{4.58}$$

where the partial derivatives of the function $f(\zeta_k)$ are formulated separately in original coordinates x_k and u_k.

Furthermore, the first-order partial derivatives determine the matrices A_k and B_k, which are equipped with a subscript k, indicating the time-varying nature of the approximation. Since the Taylor series approximation is a local linearization of the nonlinear dynamics, and the linearization point $\bar{\zeta}_k$ changes its position in every time step k, the resulting approximating dynamics are time-varying, but at least linear. The Lagrange remainder is an additional term in the linearization, describing the linearization error of the approximation. The further consideration of the linearization error in the computation and controller synthesis renders the controller robust against the approximation errors. In order to keep the linearization errors as small as possible, a natural choice for the linearization point $\bar{\zeta}_k$ are the center points of the set $X_k \times U$:

$$\bar{\zeta}_k = \begin{bmatrix} \bar{x}_k \\ \bar{u}_k \end{bmatrix} = \begin{bmatrix} q_{x,k} \\ p \end{bmatrix}, \tag{4.59}$$

where $q_{x,k}$ is the center point of the reachable ellipsoid $\hat{X}_k$, and $p \in \mathbb{R}^m$ is the geometric center of the input set U, i.e. with (3.13) and $p := \mathit{centroid}(U)$.

This section extends the previously presented control algorithm 4.1 for AS, in order to generate a stabilizing robust controller for the nonlinear system based on reachability computations. To this end, it is necessary to consider the linearization error, represented by the Lagrange remainder in (4.57), in the reachability computations. In order to establish (4.58), such that it corresponds to the model (4.50a) for sets of input arguments, the Lagrange remainder has to be specified for set based arguments, too. To obtain $L(\bar{\zeta}_k, h_L)$ for a neighborhood of $\bar{\zeta}_k$, in fact for any $h_L \in \hat{X}_k \times U$, it can be over-approximated by means of interval arithmetics, as proposed in [12]. In order to make interval arithmetics applicable, the reachable set $\hat{X}_k \in \mathcal{E}$ and the input space U are over-approximated by hyperboxes by applying the function introduced in Def. 3.7:

$$\lfloor \hat{X}_k \rceil := intval\left(\hat{X}_k\right), \quad \lfloor U \rceil := intval\left(U\right), \quad \lfloor \Psi_k \rceil = \begin{bmatrix} \lfloor \hat{X}_k \rceil \\ \lfloor U \rceil \end{bmatrix}. \tag{4.60}$$

The set-valued evaluation of the Lagrange remainder $L(\bar{\zeta}_k, h_L)\big|_{h_L = \lfloor \Psi_k \rceil}$ for the interval $\lfloor \Psi_k \rceil$ leads to a hyperbox-valued over-approximation $L_{box}(\bar{\zeta}_k)$ of the linearization error $L(\bar{\zeta}_k, h_L)$ for the domain $\hat{X}_k \times U$:

$$L_{box}(\bar{\zeta}_k) \supseteq \{L(\bar{\zeta}_k, h_L) \mid h_L = \bar{\zeta}_k + \alpha_h(\zeta_k - \bar{\zeta}_k), \alpha_h \in [0,1], \zeta_k \in \lfloor X_k \rceil \times \lfloor U \rceil\}. \tag{4.61}$$

With this over-approximation of the linearization error, the set-valued computation of the reachable follows to:

$$X_{k+1} \subset A_k(X_k - q_{x,k}) \oplus B_k(U - p) \oplus GV \oplus L_{box}(\bar{\zeta}_k) + f(q_{x,k}, p). \tag{4.62}$$

By considering the disturbance set V and the over-approximation $L_{box}(\bar{\zeta}_k)$ of the linearization error on the right of (4.62), the resulting set is robust against all possible linearization errors and disturbances.

In order to obtain an analogon to the set-based mapping (4.10) for the linearized dynamics, such that the ellipsoidal calculus can be applied, the error box $L_{box}(\bar{\zeta}_k)$ has to be tightly enclosed by a Löwner-John ellipsoid, denoted by $L_E(\bar{\zeta}_k) = \varepsilon(q_{L,k}, Q_{L,k}) \supseteq L_{box}(\bar{\zeta}_k)$:

$$\hat{X}_{k+1} \supseteq A_k(X_k - q_{x,k}) \oplus B_k(U - p) \oplus GV \oplus L_E(\bar{\zeta}_k) + f(q_{x,k}, p). \tag{4.63}$$

The true reach set X_{k+1} is over-approximated in a two-step procedure. First, the nonlinear difference equation (4.50a) is conservatively linearized with a first-order Taylor series (4.58), while considering the linearization error with the Lagrange remainder $L_E(\bar{\zeta}_k)$. The obtained set-valued linear equation to compute the subsequent reachable set in (4.63) includes the Minkowski addition of several ellipsoids. Second, as already mentioned, the Minkowski addition of ellipsoids does not result in an ellipsoid, but the resulting, compact and convex set can be over-approximated by an ellipsoid $\hat{X}_{k+1} \in \mathcal{E}$. This procedure results in similar expression for the computation of over-approximating reachable sets for nonlinear systems, as for affine systems. But, in addition to the consideration of the over-approximated linearization error $L_E(\bar{\zeta}_k)$, the obtained linearized dynamics are time-varying. Even though the conservative linearization of the nonlinear dynamics simplifies the dynamics, the consideration of the time-varying dynamics and the linearization error impedes the controller synthesis compared to pure linear affine systems, as in Sec. 4.3. The following section presents a formulation of an optimization problem, which is able to generate an robustly stablizing controller, and ultimately results in an extension of Algorithm 4.1 for nonlinear systems.

4.4.2 Extension of Control Algorithm for Affine Systems

This section is about the controller synthesis for nonlinear affine system, as defined in Def. 4.2. The main idea is to extend the approach, presented in Sec. 4.3 for AS. Due to the conservative local linearization of the nonlinear dynamics, and the resulting time-varying linear dynamics, the extension consists on one hand on an enhanced formulation of the Lyapunov stability criterion, and consideration as an constraint in the optimization problem. And on the other hand, the over-approximation linearization error has to be considered for a robust reachability computation of the nonlinear system.

The structure of the control law for NADS is the same as for AS (see (4.7)), and the set-valued, closed-loop dynamic of the linearized system in (4.63) becomes:

$$\hat{X}_{k+1} \supseteq A_k(\hat{X}_k - q_{x,k}) \oplus B_k(\bar{U}_k - p) \oplus GV \oplus L_E(\bar{\zeta}_k) + f(q_{x,k}, p), \tag{4.64a}$$

$$\supseteq A_k(\hat{X}_k - q_{x,k}) \oplus B_k(-K_k\hat{X}_k + d_k - p) \oplus GV \oplus L_E(\bar{\zeta}_k) + \dots \; f(q_{x,k}, p), \tag{4.64b}$$

$$\supseteq \underbrace{(A_k - B_kK_k)}_{A_{cl,k}}\hat{X}_k \oplus GV \oplus L_E(\bar{\zeta}_k) + f(q_{x,k}, p) - A_kq_{x,k} + \dots \; B_k(d_k - p). \tag{4.64c}$$

Regarding the reachable ellipsoid $\hat{X}_k$, the control task is again twofold: First, the center point $q_{x,k}$ has to be stabilized by the control law, and second, the size of the reachable ellipsoid should decrease, or at least be constant over time. To this end, the set-valued formulation for the closed-loop behavior can be split, as follows:

$$q_{x,k+1} = (A_k - B_kK_k)q_{x,k} + q_{L,k} + f(q_{x,k}, p) - A_kq_{x,k} + B_k(d_k - p) + Gq_v, \tag{4.65a}$$

$$= B_k(-K_kq_{x,k} + d_k - p) + q_{L,k} + f(qx, k, p) + Gq_v, \tag{4.65b}$$

for the center point. The computation of the next reachable ellipsoid in (4.64) requires a twice application of the Minkowski addition, and leads in the following to a bi-linear matrix inequality. In order to retain the linearity of the matrix inequalities, the linearization error $L_E(\bar{\zeta}_k)$ is first neglected in the formulation of the evolution of the shape matrix $Q_{x,k}$:

$$Q_{x,k+1} = (1 - \nu)A_{cl,k}Q_{x,k}A_{cl,k}^T + \nu GQ_vG^T, \;\; \nu \in [0, 1]. \tag{4.66}$$

As for AS the above equation will be reformulated as an LMI, and in order to over-approximate a reachable ellipsoid for NADS.

Due to the time-varying dynamics of NADS, finding stabilizing control parameter (K_k, d_k) for (A_k, B_k) in each time step k is not sufficient to enforce convergence of the center point $q_{x,k}$ to the origin. In order to achieve convergence anyway, the concept of flexible Lyapunov functions is employed. Thus, a time-invariant Lyapunov function $V(q_{x,k}k) = q_{x,k}^T M q_{x,k}$ with positive definite matrix M is used for the initial dynamics (A_0, B_0).

However, it may be impossible to find a quadratic Lyapunov function, which monotonically decreases (i.e. $V(q_{x,k+1}) \leq \rho V(q_{x,k}),\ \rho \in [0,1)$) for the nonlinear dynamics for each time step $k > 0$. To relax this condition, the concept of flexible Lyapunov functions (cf. [114]) introduces slack variables $\alpha_k \in \mathbb{R}_{\geq 0}$:

$$q_{x,k+1}^T M q_{x,k+1} - \rho q_{x,k}^T M q_{x,k} \leq \alpha_k, \tag{4.67}$$

$$q_{x,k+1} = B_k(-K_k q_{x,k} + d_k - p) + q_{L,k} + f(q_{x,k}, p) + G q_v. \tag{4.68}$$

As a result, the Lyapunov condition is flexible in the sense that the Lyapunov function may be locally non-monotone (increasing), in contrast to a monotone decrease in the standard Lyapunov condition, as used in (4.15) for AS. Nonetheless, asymptotic convergence is guaranteed if $\alpha_k \to 0$ for $k \to \infty$, which is obtained by

$$\alpha_k \leq \max_{i \in \{1,\dots,k\}} \omega^i \alpha_{k-i}, \tag{4.69}$$

with $\omega \in [0,1)$, see [114].

This concept is used here to couple the problem at time step k to the problems at $k-i$ to enforce convergence of the center point $q_{x,k}$ of $\hat{X}_k$ over the iterations of the control procedure. (4.67)-(4.69) are the first set of constraints, considered in the optimization problem to compute a feasible controller for NADS. The formulation of an approximation for the shape matrix of the reachable set follows directly from the formulation in (4.16) for AS. But for NADS, the LMI contains time-varying system matrices (A_k, B_k):

$$\begin{bmatrix} S_{k+1} & A_{cl,k} Q_{x,k} & G Q_v \\ Q_{x,k} A_{cl,k}^T & (1-\nu) Q_{x,k} & 0 \\ Q_v G^T & 0 & \nu Q_v \end{bmatrix} \geq 0, \quad \nu \in [0,1] \tag{4.70}$$

The input constraint formulation is identical to AS, and follows from Prop. 4.2. The combination of the mentioned constraints results in an SDP, which is very similar to (4.48), but ensures a convergence of the center point even for time-varying system dynamics:

$$\min_{S_k,K_k,d_k,\nu,\alpha_k,} J(\hat{X}_{k+1}) = \text{trace}\left(\begin{bmatrix}\mu_0 S_k & 0 & 0\\ 0 & \mu_1||q_{x,k+1}|| & 0\\ 0 & 0 & \mu_2||u_k||\end{bmatrix}\right), \tag{4.71a}$$

subject to:

$$q_{x,k+1}^T M q_{x,k+1} - \rho q_{x,k}^T M q_{x,k} \leq \alpha_k, \tag{4.71b}$$

$$q_{x,k+1} = B_k(-K_k q_{x,k} + d_k - p) + q_{L,k} + f(qx,k,p) + Gq_v, \tag{4.71c}$$

$$\alpha_k \leq \max_{i\in\{1,\dots,k\}} \omega^i \alpha_{k-i}, \tag{4.71d}$$

$$\begin{bmatrix} S_{k+1} & A_{cl,k}Q_{x,k} & GQ_v\\ Q_{x,k}A_{cl,k}^T & (1-\nu)Q_{x,k} & 0\\ Q_v G^T & 0 & \nu Q_v\end{bmatrix} \geq 0, \tag{4.71e}$$

$$\text{trace}(S_k) \leq \text{trace}(Q_{x,k}), \tag{4.71f}$$

$$\begin{bmatrix} (b_{u,i} - r_{u,i}(d_k - K_k q_{x,k}))I_n & -r_{u,i}K_k Q_{x,k}^{1/2}\\ (-r_{u,i}K_k Q_{x,k}^{1/2})^T & b_{u,i} - r_{u,i}(d_k - K_k q_{x,k})\end{bmatrix} \geq 0, \tag{4.71g}$$

$$\forall i = \{1,\dots,n_u\}.$$

The convex optimization problem must be solved in every time step k, and the controller synthesis terminates successfully, if the reachable set $\hat{X}_k$ is contained in the target set $\mathbb{T}$.

Note, that in the above formulation only the impact of the disturbance set V on the reachable ellipsoid $\hat{X}_{k+1}$ is considered in the LMI (4.71e). The approximating ellipsoid for the linearization error $L_E(\bar{\zeta}_k)$ is applied afterwards to generate the robustly over-approximating reachable set $\hat{X}_{k+1}$, in order to reduce the conservatism of the reachable set.

Reduced Linearization Error

The follow-up consideration of the linearization error $L_E(\bar{\zeta}_k)$ in the reachability computation is an advantage, in the sense of conservatism. Once a solution of the optimization problem (4.71) is available, it is possible to reduce the conservatism of the linearization error by reducing the considered input set. Prior to the solution of the SDP, the complete input set U has to be considered in the evaluation of the Lagrange remainder (see (4.60)(4.61)), and a-posteriori the control input set $\bar{U}_k \subseteq U$ is available. The fact, that $\bar{U}_k$ is most likely smaller than U results consequently in a smaller set for the linearization error $L_E(\bar{\zeta}_k)$, and thus in a reduction of the conservatism. But the use of a reduced input set has to be done very cautiously. Since the system matrices (A_k, B_k) are directly connected with the linearization point, it is crucial to use an input set with the same center point as U. Therefore, a hyperbox $\lfloor \check{U}_k \rceil \supseteq \bar{U}_k$ is introduced, which contains the control input set $\bar{U}_k$ and

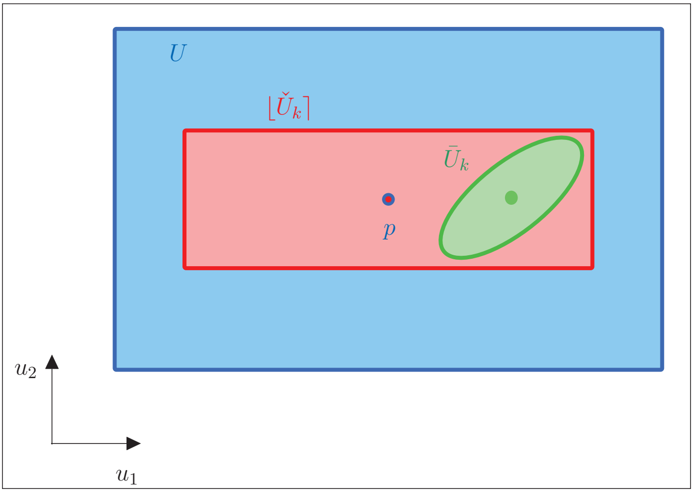


Figure 4.3: A two dimensional example of the construction of the reduced input hyperbox $\lfloor \check{U}_K \rceil$. It holds that $\bar{U}_k \subseteq \lfloor \check{U}_k \rceil \subseteq U_p$, and $p = centroid(U) = centroid(\lfloor \check{U}_k \rceil)$.

is centered at p:

$$p = centroid(U) = centroid(\lfloor \check{U}_k \rceil). \tag{4.72}$$

The reduced linearization error $\check{L}_E(\bar{\zeta}_k)$ is then:

$$\check{L}_E(\bar{\zeta}_k) \subseteq \{L(\bar{\zeta}_k, h) \mid h = \bar{\zeta}_k + \alpha_h(\zeta_k - \bar{\zeta}_k), \alpha_h \in [0,1], \zeta_k \in \lfloor X_k \rceil \times \lfloor \check{U}_k \rceil\}. \tag{4.73}$$

The construction of the reduced input hyperbox $\lfloor \check{U}_k \rceil$ is illustrated in Fig. 4.3. Obviously, the size of $\lfloor \check{U}_k \rceil$ is determined by the current location of the feasible input ellipsoid $\bar{U}_k$. If $\bar{U}_k$ is close to the boundary of U, such that their boundaries lie on top of each other, the reduced input set $\lfloor \check{U}_k \rceil$ has the same size and shape as U, and no reduction of the linearization error is obtained. But for all remaining cases, the linearization error can be reduced by the use of $\lfloor \check{U}_k \rceil$.

Under any circumstances, it has to be ensured that the reduced input set $\lfloor \check{U}_k \rceil$ has the same center point as U, otherwise the linearized dynamics in (4.63) gets invalid. A new linearization at center point of $\bar{U}_k$ is inadmissible, since the linearized system (A_k, B_k) has already been used in the SDP to compute the control law, and a new linearization at a different linearization point would lead to new system matrices (A_k, B_K). The convergence properties and input constraint would not hold for the application of the control law to the new system matrices. To retain the consistence of control law and system matrices, the center point of $\lfloor \check{U}_k \rceil$ has to be at p. Nevertheless, as long as the linearization point p is consistent with the system matrices (A_k, B_k), the use of $\check{L}_E(\bar{\zeta}_k)$ in the reachable set computation guarantees the consideration of all possible linearization errors, such that

$$\check{X}_{k+1} \supseteq (A_k - B_k K_k)\check{X}_k \oplus GV \oplus \check{L}_E(\bar{\zeta}_k) + f(q_{x,k}, p) - A_k q_{x,k} + B_k(d_k - p), \tag{4.74}$$

is a robust over-approximation of the true reachable set.

Algorithmic Control Procedure for NADS

The complete iterative procedure to compute a stabilizing controller is formulated in Algorithm 4.2. The initial step computes the hyperbox $\lfloor \check{X}_k \rceil$ of the current reduced reachable set $\check{X}_k$, and the non-linear system is linearized in the second step to get the system matrices A_k and B_k. The third step solves the optimization problem (4.71). If no feasible solution for the SDP exists, the controller synthesis fails with the current parameters of Algorithm 4.2. Regarding the concept of flexible Lyapunov functions, the parameters ρ, α_0, and ω can be adjusted, or the size of the initial state set and the disturbance set can be adjusted, with the objective to render the SDP feasible.

Assuming a feasible solution is available, the reduced linearization error $\check{L}_E(\bar{\zeta}_k)$ is computed in step four, and is used in step five to compute the next reachable set $\check{X}_{k+1}$. The sixth step of the algorithm computes π_{k+1} to evaluate distance traveled by $q_{x,k+1}$.

Algorithm 4.2.
Ellipsoidal Control Algorithm for Nonlinear Affine Systems (ECA-NADS)

Given: NADS as in Def. 4.2, initial set $X_0 = \varepsilon(q_{x,0}, Q_{x,0})$, $v_k \in V = \varepsilon(q_v, Q_v)$, *and* $U \in \mathcal{P}$, *as well as* $\mathbb{T}$, π_{min}, $\rho \in (0, 1]$, *and* $\alpha_0 \in [0,\ 1]$, $\omega \in [0, 1)$.
Define: $k := 0$, $\pi_0 = \pi_{min}$, $\check{X}_0 = \hat{X}_0 \supseteq X_0$
while $\check{X}_k \nsubseteq \mathbb{T}$ *and* $\pi_k \geq \pi_{min}$ ***do***

1. *Compute hyperbox* $\lfloor \check{X}_k \rceil$ *according to* (4.60)
2. *Apply linearization procedure according to Sec. 4.4.1 to get* A_k, B_k
3. *solve the optimization problem* (4.71)
   if *no feasible solution is found* ***do***
      stop algorithm (synthesis failed)
   end if
4. *Compute the reduced linearization error* $\check{L}_E(\bar{\zeta}_k)$ *based on* $\lfloor \check{X}_k \rceil$ *and* $\lfloor \check{U}_k \rceil$.

5. *Compute the reachable set* $\check{X}_{k+1} = \varepsilon(q_{x,k+1}, Q_{x,k+1})$ *according to* (4.74).

6. $\pi_{k+1} := ||q_{x,k+1} - q_{x,k}||$

7. $k := k + 1$

end while

Lemma 4.4. *If Algorithm 4.2 terminates with* $\check{X}_N \subseteq \mathbb{T}$*, Problem 4.2 is successfully solved and a control law* (4.7) *exists, which steers any initial state* $x_0 \in X_0$ *into the target set* $\mathbb{T}$ *in* N *steps for all possible disturbances* $v_k \in V$*. Furthermore, the input constraint* $u_k \in U$ *holds for all* $0 < k < N$*, and the center point* $q_{x,k}$ *of the reach set* $\check{X}_k$ *converges to the origin.*

Proof. According to Proposition 4.1 it holds that $X_k \subseteq \check{X}_k$, i.e. the true reach set at each time step is over-approximated by the ellipsoid $\check{X}_k$. The reach set $\check{X}_{k+1}$ at time $k+1$ is computed in Algorithm 4.2 for all $x_k \in \check{X}_k \supseteq X_k$ and all disturbances $v_k \in V$, including the linearization error $\check{L}_E(\bar{\zeta}_k)$. Thus, it follows that $X_{k+1} \subseteq \check{X}_{k+1}$ is a robust over-approximation of the true reachable set X_{k+1}, and by induction $x_k \in X_k \subseteq \check{X}_k$ for all $k > 0$. Successful termination of Algorithm 4.2 implies that $\check{X}_N \in \mathbb{T}$ and consequently $x_N \in X_N \subseteq \mathbb{T}$ holds for all initial states $x_0 \in X_0$ and all disturbances $v_k \in V$. It follows, that the input constraint $u_k \in U$ holds at each time step, according to Proposition 4.2. Finally, the flexible Lyapunov condition is employed by (4.67) and (4.68), which ensures convergence of the center point $q_{x,k}$ of $\check{X}_k$ (cf. Lemma III.4 in [114]) over the considered horizon N is achieved. Furthermore asymptotic convergence of $q_{x,k}$ is achieved for $\alpha_0 = 0$. □

4.5 Numerical Example

In order to show the capability of the proposed algorithm, it is applied to a numerical example of a three-tank-system. The dynamics of the fluid height inside each tank can be described by nonlinear differential equations. In the following example a time discretization leads to nonlinear difference equations for each tank, and hence the proposed control algorithm 4.2 from Section 4.4.2 is used for the control task.

Modelling

The considered tank system consists of three cylindrical tanks with different base areas. The fluid inside the tanks can flow from one tank to another through connecting pipes at the bottom of the tanks. The inlet is located at the top of the tanks, whereas the outlet is located at the bottom. Within this example two different kinds of outlets are considered. First, there is an outlet with a controllable valve, and second, there is an outlet without any affecting device (uncontrollable).

The volume flow for the inlet is constant, generated by pumps and controlled by valves.

The control task for this example is to fill the empty tanks to a predefined reference height. An illustration of the three-tank-system with the considered configuration of inlet and outlet is given in Fig. 4.4. The connecting pipes have the

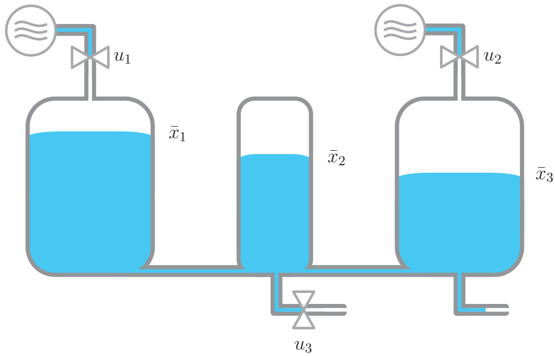


Figure 4.4: This graphic illustrates the three-tank-system, which is used to show the capability of the proposed algorithm. The control inputs consists of two inflow valves (tank 1 and 3) and one outflow valve at the bottom of tank 2. The uncontrollable outflow of tank 3 can be seen as a known disturbance. At the beginning of the algorithm the tanks are completely empty and the goal is to fill the tanks to the reference height $\bar{x}$.

same cross-section area as the outlets, which is denoted by a. The cross section of each tank is denoted by $E_i, i = \{1, 2, 3\}$. The input is modeled as the percentage of a fully opened valve, which effects the inflow into the first and third tank, and the outflow of the second tank; $u_{i,k} \in [0, 1],\ i = 1, 2, 3$. The system state $x_k \in \mathbb{R}^3$ describes the fluid height in each tank. The desired fluid height in each tank is given by the vector $\bar{x}$, which is also an equilibrium point of the system. A mathematical model of this system can be developed by the use of the mass balance equation for a single tank, in which the change of the fluid volume/height is determined by the sum of the inflow and outflow rate. The discrete-time nonlinear dynamics are given in (4.76), wherein a simplified version of Toricelli's law (cf. [59]) is used. Furthermore, the nonlinear dynamics are adapted, in order to formulate a suitable control problem. To this end the auxiliary state

$$\tilde{x}_{i,k} := x_{i,k} - \bar{x}_i, i = \{1, 2, 3\}, \tag{4.75}$$

is introduced. The auxiliary system dynamics, defined with $\tilde{x}_k$ in (4.76), is used to solve the control problem and stabilize the system from an initial system state $\tilde{x}_0$

into the origin, which corresponds to the reference height $\bar{x}$ in the original coordinates x_k. The complete formulation of the system is:

$$\tilde{x}_{k+1} = \begin{bmatrix} \tilde{x}_{1,k} + \left(\frac{1}{A_1}(v_{max}u_{1,k} - \mathrm{a}0.5g\tanh(x_{1,k} - x_{2,k}))\right) + v_{1,k} \\ \tilde{x}_{2,k} + \frac{a}{A_2}0.5g\left(\tanh(x_{1,k} - x_{2,k}) - \tanh(x_{2,k} - x_{3,k}) - \tanh(x_{2,k})u_{2,k}\right) + v_{2,k} \\ \tilde{x}_{3,k} + \left(\frac{a}{A_3}0.5g\left(\tanh(x_{2,k} - x_{3,k}) - \tanh(x_{3,k})\right) + \frac{1}{A_3}v_{max}u_{3,k}\right) + v_{3,k} \end{bmatrix},$$
$$E_1 = E_3 = 4,\ E_2 = 2,\ a = 0.04,\ v_{max} = 1.1, \tag{4.76}$$

with an initial set of states:

$$X_0 = \varepsilon\left(-\bar{x}, 1e^{-4}\begin{bmatrix} 2 & 0 & 0 \\ 0 & 8 & 0 \\ 0 & 0 & 4 \end{bmatrix}\right),\ \bar{x} = \begin{bmatrix} 0.7113 \\ 0.6091 \\ 0.5067 \end{bmatrix}. \tag{4.77}$$

The admissible input set is given by a box constraint:

$$u_k \in U = P_H(R_u, b_u), \tag{4.78}$$

with

$$R_u = \begin{bmatrix} 1 & 0 & 0 \\ -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & -1 \end{bmatrix},\ b_u = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \end{bmatrix}. \tag{4.79}$$

The disturbance represents a possible uncertainty in the sensor units for the height measurement, and the disturbance is assumed to be taken from the ellipsoid:

$$v_k \in V = \varepsilon\left([0,0,0]^T, 1e^{-5}\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}\right). \tag{4.80}$$

The target set is defined as ellipsoid centered in the origin:

$$\mathbb{T} = \varepsilon\left([0,0,0]^T, 1e^{-3}\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}\right). \tag{4.81}$$

Discussion of the Results

The initial Lyapunov function (4.14) is parametrized with an identity matrix and the remaining parameters of Algorithm 4.2 are chosen as follows: $\alpha_0 = 1e{-}4,\ \omega = 0.98,\ \rho = 0.98,\ \pi_{min} = ||\bar{x}||$. The reachable set $\check{X}_k$ is steered in $N = 14$ steps by the *Ellipsoidal Control Algorithm* from the initial set X_0 into the target region $\mathbb{T}$ with a total computational time of $19.58s$. The average solution time for one single LMI problem is $0.311s$. This result enables to provide a feasible control law for any initial state $x_0 \in X_0$. Fig. 4.5 illustrates a simulation for 100 randomly initialized states in X_0 (X_0 is covered by "*"-markers, hence it is not visible). The graphic shows the propagation of 100 samples under the influence of the computed control law. The reachable sets are completely covered by the '*'-markers, which represents the states $x_k^i, i = 1...100$. For $k > 3$, it even seems to converge to one single point and one cannot recognize the 100 different state with this resolution.

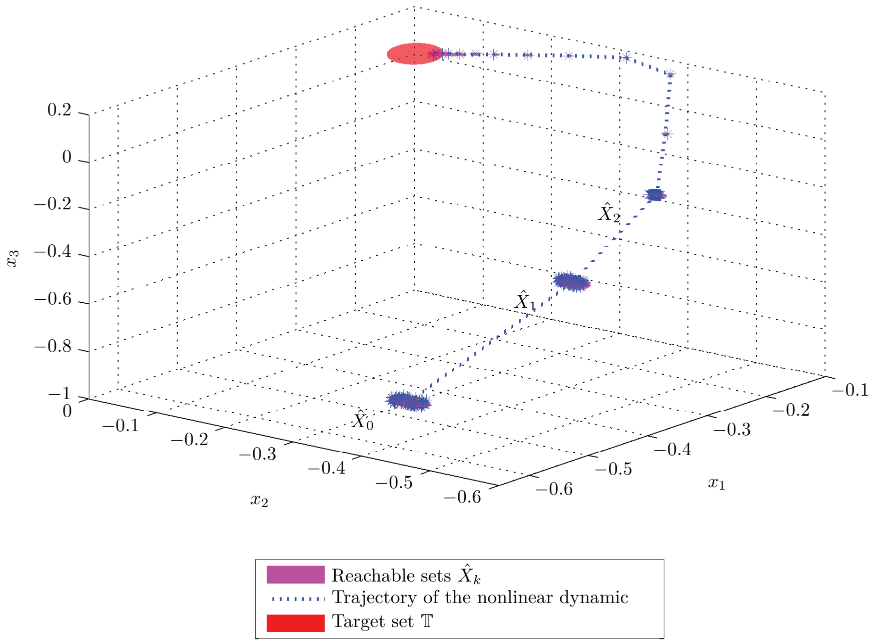


Figure 4.5: Numerical example: Algorithm 4.1 transfers the initial set X_0 into the terminal set $\mathbb{T}$. In addition, the evolution of 100 randomly chosen samples is shown.

The simulated trajectories are solutions of the nonlinear dynamic, given in (4.76). Obviously, the distribution of the 100 states for $k > 0$ is around the center point of the current reachability set. This emphasizes the effect of the feedback gain K_k, in that it has a stabilizing effect on the state vector $\tilde{x}_k$. The feed-forward part d_k ensures, that the center point converges to the origin, and hence to the target region

$\mathbb{T}$. The reachability sets $\check{X}_k$ are obviously conservative over-approximations of the true reachability set X_k incorporating all disturbances and linearization errors.

The optimization problem (4.71) was solved with Matlab 7.12.0 with YALMIP 3.0 and SeDuMi 1.3. The reachability computations were performed with the ellipsoidal toolbox ET, [111].

4.6 Discussion

This chapter introduces the basics of reachability computation for uncertain discrete-time systems, and the possibilities to use reachable set computations for controller synthesis. The reachability computation is based on the assumption, that the initial set is a bounded set, for which an ellipsoidal over-approximation exists. By the use of the well known ellipsoidal calculus, it is possible to compute the subsequent reachable set, given the sets for the states, inputs and disturbances. The control problem is formulated as set-to-target problem, in which the initial state set should be transferred into a target set within finite time, and under consideration of the input constraint and all disturbances. The solution approach is based on an iterative solution of an SDP, to steer the system into the target region. The SDP contains LMI constraints to achieve the convergence of the center point, and shape matrix, as well as a formulation to satisfy the input constraints. In this section, LMI formulations for two different set representation for the input set are shown: (I) polytopic input constraint, and (II) ellipsoidal input constraints. The former results in an LMI for each half-plane in the input constraint. The latter is a bit more elaborate, since it includes the linearization of a non-convex matrix inequality. This results in multiple linearization points, and on account of this, the complete SDP has to be solved for each linearization point. Due to the computational advantage of polytopic input constraints, the remaining sections of this thesis will only consider polytopic input constraints, well knowing that ellipsoidal input constraints are possible, too.

The solution of the optimization problem provides control parameters for an affine control structure. The algorithmic procedure, in which the optimization problem is solved in every iteration, ensures the solution of the overall control problem for linear affine systems. Furthermore, it is shown that a minor extension makes the algorithm suitable for nonlinear discrete-time systems as well. The extended control algorithm is able to compute a stabilizing controller for nonlinear affine systems. Therein, the controller synthesis is again based on a formulation of an SDP, with an a-priori local linearization of the nonlinear difference equations by a first-order Taylor series. The conservative linearization at each time step leads to time-varying affine dynamics, since the system matrices are connected to the current linearization point. This requires an adaption of the stability criterion for the center point of the reachable set, since a standard stability formulation is not sufficient for time-varying dynamics. The extended algorithm includes a relaxation of the standard Lyapunov stability criterion, in which a temporary increase of the Lyapunov function is feasible, as long as an overall convergence is achieved.

An extension of the proposed control procedure might address the convergence of the shape matrix of the reachable ellipsoid in a similar way as the flexible Lyapunov function. Instead of requiring a strict reduction of the trace in (4.71f), a slack variable could be introduced to allow some flexibility in the convergence of the shape matrix, too. This extension would enhance the probability for a successful termination of ECA-AS and ECA-NADS.

The conservative linearization introduces a linearization error, and in order to take into account the linearization error, a set valued over-approximation of the Lagrange remainder is included in the reachability computations. Although it is not considered in the solution of the SDP, and thus in the controller synthesis, it is minded within the computation of the next reachable set. By computing the linearization error $L_E(\bar{\zeta}_k)$ after the control law (K_k, d_k) is available, the control input $\bar{U}_k$ set can be used instead of the whole feasible input set U. The advantage of this procedure is the reduced conservatism due to the smaller linearization error.

A direct consideration of the Lagrange remainder in the optimization problem would result in a bilinear matrix inequality, which could not by solved by standard solvers for SDPs. But if the disturbance set V and the linearization error ellipsoid $L_E(\bar{\zeta}_k)$ are combined to a new disturbance set $\tilde{V}$ by Minkowski addition, the new disturbance set $\tilde{V}$ could be considered in the LMI formulation for the next reachable set. But this would lead to additional conservatism, since $\tilde{V}$ can only be over-approximated by an ellipsoid. Eventually, the separate consideration of the linearization error, with the possibility of reduction has been chosen in this thesis.

Furthermore, the general reachability computations in this section include the Minkowski addition of two ellipsoidal sets, and as already pointed out, the resulting set is in general not an ellipsoids. The repeated over-approximation of the result of the Minkowski addition leads to a cumulation of over-approximations, known as wrapping effect. This is a general bottleneck of this approach, and to resolve this situation a quantitative evaluation of the induced over-approximation could be a first step to improve this approach. Moreover the control approach includes several over-approximations within each iteration of the algorithm, and in combination of the over-approximation from the Minkowski addition, the control algorithm is very sensitive against any changes in the parametrization. To overcome this drawbacks, the remaining sections of this thesis will focus on system with probabilistic elements. First, the introduction of stochasticity might seem to impede the control problem, but as will be shown in the following sections, the need of over-approximation will decrease.

5 Probabilistic Reachable Set Computation for Controller Synthesis

The control synthesis procedure in the previous chapter is based on reachable set computations, which take into account all possible disturbances and, if present, linearization errors. The provided robust control law solves the stated set-to-target control problem, i.e. a successful termination of Algorithm 4.1 transfers a given set of initial states into a target set, for all specifications of the bounded disturbances. The set-based consideration of the disturbances and linearization errors is a conservative procedure, and hence robust. The use of bounded disturbances is necessary, if no further information on the disturbance is available and a worst-case consideration is required. But in most cases, the disturbances stem from measurement noise, where the state information is distributed around the true value. Using a set of noisy measurements, the actual distribution can be estimated and the now known probabilistic distribution of the uncertainty can be used for controller synthesis. The additional information on the distribution of the uncertainty allows to extend the reachable set computations to probabilistic reachable sets. While the reachable sets of the previous chapter are guaranteed to contain all states x_k for any disturbance $v_k \in V$, the probabilistic reachable sets, introduced later in this chapter, contain the system state with a user-defined confidence δ. This confidence level is a further degree of freedom in the control synthesis procedure, and allows to adapt the chosen confidence in the control synthesis procedure.

In many cases, it is possible and quite convenient to use a multivariate normal distribution to specify the initial state x_0. For example, the acquisition of information about a system state is in practice done by sensors with a certain measurement uncertainty. This uncertainty can be in a given interval, or can be described by a probabilistic distribution. The most common distribution for measurement uncertainties is the multivariate normal distribution: the initial state $x_k \in \mathbb{R}^n$ is known to be in the neighborhood of an expected value $q_{x,0} \in \mathbb{R}^n$, and the size and shape of the "neighborhood" can be described by a covariance matrix $Q_{x,0} \in \mathbb{R}^{n \times n}$. At first glance, the introduction of randomness might seem to impede the control problem, but it will be shown, that the additional information on the distribution still permits reachable set computations.

The aim of this chapter is to present a method to solve a control problem with probabilistic reachable set computations for a system class that involves probabilistic uncertainties in the initial state and the disturbance. The target is to control the system from an initial set into a target set with a certain confidence. Robustness is here understood as the requirement to reach the target with a specified probability, which is possible under the assumption of an a-priori known probabilistic distribution of the random variables. The idea underlying the proposed approach is to compute reachable sets under the effect of time-variant control laws in order to realize the transition into the target. The presented method of this chapter is partially published in [26].

Besides the aim of entering a target set, the control problem of this chapter includes the avoidance of entering an unsafe set, formulated as state constraints. This state constraints arise with the specification on the states to be in a particularly desired range of the state space. For example, the transition of a vehicle from standstill to a certain target velocity can be performed in many ways, but, by formulating state constraints, the engine speed can be held in an engine sparing range. Assuming, that the state information is collected by sensors with data probabilistically distributed around an expected value, the satisfaction of the state constraints cannot be robustly ensured for all realizations of the random state variables. A probabilistic relaxation of the state constraints results in so called *chance constraints*, which take into account the probabilistic distribution of the state, and demand a satisfaction of state constraint with high likelihood. A detailed discussion of existing methods to include chance constraints, also known as soft constraints, within controller synthesis is a particular aspect of this chapter.

Literature review

In literature, three different variants for approximating chance constraints within control design have been reported: (i) the conservative approximation via linear inequalities for Gaussian distributions ([137, 33, 35, 162]), (ii) the evaluation of chance constraints via scenario-based methods ([44, 34, 22, 133]), and (iii) set-based methods ([160, 159, 123]). The first approach was initially formulated in [137] for stochastic programming problems, and extended for controller synthesis in [33]. It is typically computational less demanding, but is limited to specific distributions. The second approach can handle non-Gaussian distributions, but is computationally intensive, if large numbers of samples need to be considered to obtain the desired confidence. This approach can be divided in techniques for approximating the chance constraint [34, 22], and for bounding the chance constraint [44, 133]. The third approach results in a more conservative control behavior compared to the other two approaches. This chapter includes a formulation and comparison of the mentioned approaches to handle chance constraints within the set-to-target control problem based on probabilistic reachable sets.

The chapter is organized such that the considered probabilistic model class and probabilistic reachable sets are introduced in Section 5.1. It defines a suitable type of stability for the system class and formally states the control problem. Different methods to handle chance constraints for the considered control problem are described in Section 5.2. The algorithmic solution procedure to synthesize the feedback control laws is presented in Section 5.3. A numerical example and a discussion on the different approaches for the consideration of chance constraints is provided in Section 5.4. Section 5.5 completes this chapter.

5.1 Affine Probabilistic Systems

A stochastic version of AS, as in Def. 4.1, results in the system class of *affine probabilistic systems* (APS), where the initial state x_0, as well as the disturbance v_k are assumed to be random variables with known distributions. The dynamics of APS are also defined by linear difference equations for the continuous state. In general, stochasticity in the system dynamics impede the robust control synthesis, since robustness cannot be guaranteed for all realizations of the random variables. In the stochastic case, the deterministic robustness is replaced by "probabilistic robustness", since only a certain percentage of realizations of the random variable has to satisfy the specifications. Furthermore, the information about the probabilistic distribution of x_0 and v_k can be extremely helpful for controller synthesis, whereas in the deterministic case, no information about the distribution of the disturbance v_k is available and a worst-case treatment with $v_k \in V$ is necessary for a robust control design.

5.1.1 System Definition and Probabilistic Reachable Computations

APS are a variant of SHS (see Def. 3.1), since the general nonlinear function in (3.4a) is in APS replaced by linear dynamics, specified by the matrices A and B. Furthermore, in APS no discrete dynamic is present, such that $z_k = 1 \ \forall\, k$, hence it is for ease of notation omitted. APS are a stochastic version of AS, since the initial state x_0 is not taken from a closed set, but instead, x_0 and v_k are random variables with a multivariate normal distribution. The complete system definition of an affine probabilistic system is given in Def. 5.1.

Definition 5.1. *An affine probabilistic system (APS) is given by the following equations ($k \in \mathbb{N}_0$):*

$$x_{k+1} = Ax_k + Bu_k + Gv_k, \tag{5.1a}$$

$$x_0 \sim \mathcal{N}(q_{x,0}, Q_{x,0}), \tag{5.1b}$$

$$v_k \sim \mathcal{N}(q_v, Q_v), \tag{5.1c}$$

$$u_k \in U = P_H(R_u, b_u), \tag{5.1d}$$

where the initial state x_0 and the disturbance v_k are Gaussian distributed with expected value $q_{x,0} \in \mathbb{R}^n$, $q_v \in \mathbb{R}^n$, and covariance matrices $Q_{x,0} \in \mathbb{R}^{n\times n}$ and $Q_v \in \mathbb{R}^{n\times n}$, respectively. The input $u_k \in \mathbb{R}^m$ is bounded to a convex polytope $U \in \mathcal{P}$ with $R_u \in \mathbb{R}^{n_u\times m}$ and $b_u \in \mathbb{R}^{n_u}$. △

A feasible execution of an APS is given as follows:

Definition 5.2. *Sample an initial continuous state x_0, the sequence of states x_k, $k \in \mathbb{N}_0$ is called "admissible", if for every k, x_{k+1} is determined by the following order of operations:*

1. *given the continuous state x_k*
2. *sample the disturbance $v_k \sim \mathcal{N}(q_v, Q_v)$*
3. *choose a suitable continuous input $u_k \in U$*
4. *compute x_{k+1} according to 5.1a*

△

APS are valuable in modeling physical processes with noisy observations of the initial state x_0, due to measurement noise. Such noise is present in each measurement of the system state, modeled by the additive stochastic disturbance v_k.

As an important aspect of the further derivations, the evolution of the state is subject to constraints. The probabilistic state distribution according to (5.1b) suggests a probabilistic notion of constraints, commonly referred to as chance constraint in literature (see [138]). (Note that the Gaussian distributions in (5.1b) would render strict state constraints to be violated in any case.) The chance constraints are formulated to probabilistically bound the state x_k to a polytopic state set $\mathbb{X}_k \in \mathcal{P}$, i.e. x_k has to be inside of $\mathbb{X}_k \subset R^n$ with a probability not smaller than $\delta_x \in]0, 1[$ in every time step k:

$$Pr(x_k \in \mathbb{X}_k) \geq \delta_x,\ \mathbb{X}_k = P_H(R_{x,k}, b_{x,k}),\ R_{x,k} \in \mathbb{R}^{n_x\times n}, b_{x,k} \in \mathbb{R}^{n_x}. \tag{5.2}$$

The subscript k for the feasible state set $\mathbb{X}_k$ indicates the possibility of time-varying state constraints, and n_x is the number of half-planes defining the feasible state set $\mathbb{X}_k$.

In this chapter, the set-to-target control problem is formulated in a probabilistic manner, wherein the initial reachable set has to be transferred into a target region $\mathbb{T}$ with a specified confidence probability. The center point of the target region is denoted by q_T, and for the solvability of the set-to-target control problem, it is assumed that an equilibrium $(\bar{x} = q_T, \bar{u})$ exists.

Assumption 5.1. *Let for system* (5.1) *exist at least one input $\bar{u} \in U$, for which* (5.1a) *has an equilibrium point $\bar{x}$, if the disturbance assumes its expected value $\bar{v} = q_v$, i.e. $\bar{x} = q_T = (I - A)^{-1} \cdot (B\bar{u} + Gq_v)$.*

In general, the reachable set of a dynamic system contains the subset of states that are reachable for the given dynamic starting from an initial state set, and an admissible input set over a specified (possibly infinite) span of time (see Sec. 4.1). For probabilistic systems (as of type APS), it is reasonable to include the stochasticity into the definition of reachable sets. To this end, confidence sets are introduced with the interpretation, that a percentage δ of the states for a system with given initialization are, at a chosen point of time, contained within this set. For the APS according to (5.1), a *reachable set with confidence* δ at time k is denoted by X_k^δ (here, the superscript δ is used to distinguish between the deterministic and probabilistic reachable sets). For the multivariate normal distributions used for x_0 and v_k, a certain level of probability corresponds to an ellipsoidal set in the space of the vectors x_0, and v_k, respectively (cf Sec. 3.3). It is thus straightforward to use ellipsoidal sets to represent the probabilistic reachable sets of APS, as well. Thus, $X_k^\delta \in \mathcal{E}$ applies, and with use of Lemma 3.2, X_k^δ can be computed for given mean $q_{x,k}$, covariance matrix $Q_{x,k}$, and confidence level δ.

For the initialization of the random variable x_0, an ellipsoid is specified as:

$$X_0^\delta := \varepsilon(q_{x,0}, Q_{x,0} \cdot c_x), \tag{5.3}$$

where the parameter c_x scales the covariance matrix $Q_{x,0}$ of the distribution according to (3.31), in order to obtain the shape matrix $Q_{x,0}^\delta := Q_{x,0} \cdot c_x$. The parameter is chosen, such that $Pr(x_0 \in X_0^\delta) = \delta$ holds for X_0^δ with (3.28).

The evolution of the ellipsoidal reachable set with confidence δ over time, briefly called *confidence ellipsoid* X_k^δ from here on, is parametrized by the mean vector $q_{x,k}$ and the covariance matrix $Q_{x,k}$. These quantities are computed by the following update functions for $k \in \mathbb{N}$:

$$q_{x,k} := Aq_{x,k-1} + Bu_{k-1} + Gq_v, \tag{5.4a}$$
$$Q_{x,k} := AQ_{x,k-1}A^T + GQ_vG^T. \tag{5.4b}$$

These equations make use of the fact, that the sum of Gaussian variables again has a Gaussian distribution, following (3.22). It is noticeable from (5.4), that $q_{x,k}$ can be directly influenced by the continuous control u_{k-1}, while this is not possible for the covariance matrix $Q_{x,k}$. However, as will be shown later, an impact on the shape of the confidence ellipsoid X_k^δ is possible by choosing a state feedback control law for u_{k-1}. With $q_{x,k}$ and $Q_{x,k}$, the confidence ellipsoid X_k^δ is obtained with $Q_{x,k}^\delta := Q_{x,k} \cdot c_x$ from:

$$X_k^\delta = \varepsilon(q_{x,k}, Q_{x,k}^\delta). \tag{5.5}$$

Equivalently to (5.3), the factor c_x scales the covariance matrix of the multivariate normal distribution to let X_k^δ contain the percentage δ of possible realizations of the random variable x_k at any time step k.

5.1.2 Definition of the Set-To-Target Control Problem for APS

In stochastic control theory, various definitions of stability can be found. A comprehensive survey of different stability definitions can be found in [94]. The most important being mean-square stability (which is a notable case of moment stability) and almost sure stability. Moment stability of order k requires the asymptotic convergence to zero of the k-th moment of the state norm. Conversely, almost sure stability is equivalent to convergence to zero of almost all realizations of the state.

In order to prepare the problem definition involving a stability requirement for the APS, a suitable type of stability for probabilistic systems has to be identified, first.

Definition 5.3. *Given a bounded time domain $\tau = \{0, 1, \dots, N\}$, $N \in \mathbb{N}_0$, and continuous inputs $\{u_k\}$ for $k \in \tau$, the APS* (5.1) *is called* attractive with confidence δ *on the domain τ, if for any initial condition $x_0 \in X_0^\delta$ and any $v_k \in \varepsilon(q_v, Q_v c_x)$, finite parameters $\bar{q} \in \mathbb{R}^n$ and $\bar{Q} \in \mathbb{R}^{n \times n}$ exist such that:*

$$||q_{x,N}|| \leq ||\bar{q}||, \quad ||Q_{x,N}|| \leq ||\bar{Q}||. \tag{5.6}$$

The system is said stable with confidence δ *on τ if in addition:*

$$||q_{x,k+1}|| < ||q_{x,k}||, \quad ||Q_{x,k+1}|| \leq ||Q_{x,k}||, . \tag{5.7}$$

holds for any $0 \leq k \leq N-1$. △

This definition of probabilistic stability for APS is justified by the requirement that the expected value $q_{x,k}$ of the state vector x_k converges to the origin, and that the size of the confidence ellipsoids (expressed by the shape matrix $Q^\delta_{x,k}$) decreases. The introduced stability definition in Definition 5.3 is a slightly weaker condition in comparison to mean square stability, since no convergence to a steady state distribution is required, where the expected value $q_{x,k}$ and the covariance matrix $Q_{x,k}$ is fixed to constant values. In this definition, only the parameters are bounded from above. The purpose is to allow rotation of the covariance matrix $Q_{x,k}$

The objective of control synthesis is to determine the sequence $\{u_k\}$ in the definition above, such that the requirements of stability, and decrease of the reachable set size are met. Hence, the task is to synthesize a control law $\kappa : \mathbb{R}^n \times \mathbb{N}_0 \to \mathbb{R}^m$ with:

$$u_k = \kappa(x_k, k). \tag{5.8}$$

The set-to-target control problem for APS can be written as follows:

Problem 5.1. *Let a APS according to* (5.1)*, a set of chance constraints* (5.2)*, and a terminal region $\mathbb{T} = \varepsilon(q_T, Q_T) \subset \mathbb{R}^n$ (i.e. centered in an equilibrium point) be given. Determine a control law $\kappa(x_k, k)$ according to* (5.8) *for $x_k \in X_k^\delta$ on a time domain $k \in \tau = \{0, 1, \dots, N\}$ such that:*

- *the controlled system is stable with confidence* δ*,*
- *the chance constraints* (5.2) *are satisfied for any* $k \in \tau$*,*
- *and the following terminal constraint is satisfied for a finite* N*:*

$$X_N^\delta \subseteq \mathbb{T}, \tag{5.9}$$

i.e. any initial state $x_0 \in X_0^\delta$ *is transferred into* $\mathbb{T}$ *with probability* δ *within* N *steps.* △

In combination of Assumption 5.1, it is always possible to find a suitable coordinate transformation, which recasts the Problem 5.1, such that the terminal set $\mathbb{T}$ is centered in the origin.

In the following sections, an algorithmic method to solve problem 5.1 is proposed. For the control law (5.8), a local time-variant, continuous, affine state feedback controller of the following form is selected:

$$u_k = -K_k x_k + d_k \in U, \ \forall\ x_k \in X_k^\delta. \tag{5.10}$$

Since x_k is taken from a confidence ellipsoid $X_k^\delta \in \mathcal{E}$, the resulting required input set is also an ellipsoid in the input space:

$$\begin{aligned} \bar{U}_k &:= \{u_k \in U \mid \forall\ x_k \in X_k^\delta : u_k = -K_k x_k + d_k\}, && (5.11a)\\ &= \varepsilon(-K_k q_k + d_k, K_k Q_{x,k}^\delta K_k^T) \in \mathcal{E}. && (5.11b) \end{aligned}$$

A feasible solution of Problem 5.1 is a set of control tuples $(K_k, d_k)\ \forall\ k \in \{0, 1, \ldots, N-1\}$, satisfying the conditions of the problem statement with:

$$\bar{U}_k \subseteq U. \tag{5.12}$$

The control law (5.10) leads to the following closed-loop dynamics for (5.1a):

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k + Gv_k, && (5.13a)\\ &= \underbrace{(A - BK_k)}_{:=A_{cl,k}} x_k + Bd_k + Gv_k. && (5.13b) \end{aligned}$$

With (5.13) and starting from $x_0 \sim \mathcal{N}(q_{x,0}, Q_{x,0})$, the propagation of the confidence ellipsoids of the controlled system results from the following update of center points and shape matrices:

$$\begin{aligned} q_{x,k+1} &= A_{cl,k} q_{x,k} + Bd_k + Gq_v, && (5.14a)\\ Q_{x,k+1} &= A_{cl,k} Q_{x,k} A_{cl,k}^T + GQ_v G^T, && (5.14b) \end{aligned}$$

for $k \in \{0, 1, \ldots, N-1\}$.

The following section elaborates on the different methods to consider the state chance constraints in the controller synthesis procedure.

5.2 Review of Methods for the Approximation of Chance Constraints

In order to evaluate the chance constraint (5.2), a multi-dimensional integral of the Gaussian probability density function over the polytopic state set $\mathbb{X}_k$ has to be solved:

$$Pr(x_k \in \mathbb{X}_k) = \int_{\mathbb{X}_k} f_N(x_k)dx_k \geq \delta_x. \tag{5.15}$$

Since the analytic solution is only possible in special cases, numerically approximating techniques have to be applied, and different alternatives were proposed in literature. The following subsections are for the most part published in [26], and formulate some of these alternatives in the context of the given control problem.

5.2.1 Set-Based Evaluation of Chance Constraints

A procedure aiming to get completely rid of the stochasticity of the constraint is to consider ellipsoidal confidence sets X_k^δ for normally distributed random variables (see [160, 159, 123]), and is referred to as *Set-Based Evaluation of Chance Constraints* (SBE-CC) in this thesis. In SBE-CC, the optimization problem incorporates no probabilistic distribution, because the distribution of x_k is replaced by a convex set X_k^δ, and it has to be verified, that X_k^δ is inside the convex state set $\mathbb{X}_k$, as shown in Figure 5.1 (left). Note, $r_{x,k+1,i}$ is the i-th row of the matrix $R_{x,k}$, which defines the state polytope $\mathbb{X}_k$. If it holds, that $X_k^\delta \subseteq \mathbb{X}_k$ for any k, it follows that:

$$Pr(x_k \in \mathbb{X}_k) \geq \delta. \tag{5.16}$$

The initial chance constraint (5.2) is satisfied as long as the probability δ_x of the chance constraint is less or equal to the probability δ of the confidence set:

$$\delta_x \leq \delta. \tag{5.17}$$

5.2.2 Locally Linear Approximation of Multivariate CDF's

One approach, which will be referred to as *Locally Linear Approximation of Chance Constraints* (LLA-CC) in the following, has been developed in [137] and extended in [33] in the context of discrete-time linear systems, and a finite-horizon optimal control problem. It uses a splitting method, in which the joint multivariate distribution is divided into n_x uni-variate distributions using Boole's inequality. The joint chance constraint is then over-approximated by n_x uni-variate chance constraints, which can be evaluated by the cumulative distribution function. The LLA-CC approach evaluates each half-plane separately ($\epsilon_{vio,i}$), possibly leading to $X_k^\delta \nsubseteq \mathbb{X}_k$, yet satisfying $Pr(x_k \in \mathbb{X}_k) \geq \delta$ (see Fig. 5.1). A local linearization of the CDF allows to include this approach in the SDP-based controller synthesis.

The following lemma summarizes the basic idea for the LLA-CC approach, formulated in [33].

Lemma 5.1. *The chance constraint* $Pr(x_k \in \mathbb{X}_k) \geq \delta_x$ *is satisfied, if it holds for* $i \in \{1, 2, \ldots, n_x\}$ *that:*

$$Pr(r_{x,k,i}x_k > b_{x,k,i}) = \epsilon_{vio,i}, \quad \sum_{i=1}^{n_x} \epsilon_{vio,i} < 1 - \delta_x. \tag{5.18}$$

Proof: The chance constraint implies that $Pr(x_k \notin \mathbb{X}_k) < 1 - \delta_x$. The complementary set to $\mathbb{X}_k$ can be described as the union of halfspaces:

$$x_k \notin \mathbb{X}_k \Longleftrightarrow x \notin \bigcap_{i=1}^{n_x} \{x_k \mid r_{x,k,i}x_k \leq b_{x,k,i}\} \Longleftrightarrow x_k \in \bigcup_{i=1}^{n_x} \{x_k \mid r_{x,k,i}x_k > b_{x,k,i}\}, \tag{5.19}$$

where $r_{x,k,i}$ denotes the i-th row of the matrix $R_{x,k}$. It is well known, that the following inequality holds for a countable set of stochastic events $\Xi_1, \ldots, \Xi_{n_x}$:

$$Pr\left(\bigcup_{i=1}^{n_x} \Xi_i\right) \leq \sum_{i=1}^{n_x} Pr(\Xi_i). \tag{5.20}$$

By interpreting the evaluation of $r_{x,k,i}x_k > b_{x,k,i}$ as an event Ξ_i, the inequality:

$$Pr(x_k \notin \mathbb{X}_k) \leq \sum_{i=1}^{n_x} Pr(r_{x,k,i}x_k > b_{x,k,i}) \tag{5.21}$$

implies (5.18). □

By applying Lemma 5.1, no further evaluation of a multi-dimensional integral is needed, but instead, the use of a cumulative distribution function is possible. Let $y_{k,i} := r_{x,k,i}x_k \sim \mathcal{N}(q_{y,k,i}, Q_{y,k,i})$ define an uni-variate auxiliary variable with mean and covariance matrix according to:

$$q_{y,k,i} = r_{x,k,i}q_{x,k} \in \mathbb{R}, \;\; Q_{y,k,i} = r_{x,k,i}Q_{x,k}r_{x,k,i}^T \in \mathbb{R}_{\geq 0}. \tag{5.22}$$

The probability $Pr(y_{k,i} > b_{x,k,i})$ in (5.18) can be obtained for $i \in \{1, \ldots n_x\}$ through the evaluation of the cumulative distribution function of a standard normal distribution. Using (3.25) and (3.26), as well as the CDF of normal distributions:

$$F_N(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-\frac{w^2}{2}} dw,$$

the probability $Pr(y_{k,i} > b_{x,k,i})$ satisfies:

$$Pr(y_{k,i} > b_{x,k,i}) = 1 - F_N\left(\frac{b_{x,k,i} - q_{y,k,i}}{\sqrt{Q_{y,k,i}}}\right). \tag{5.23}$$

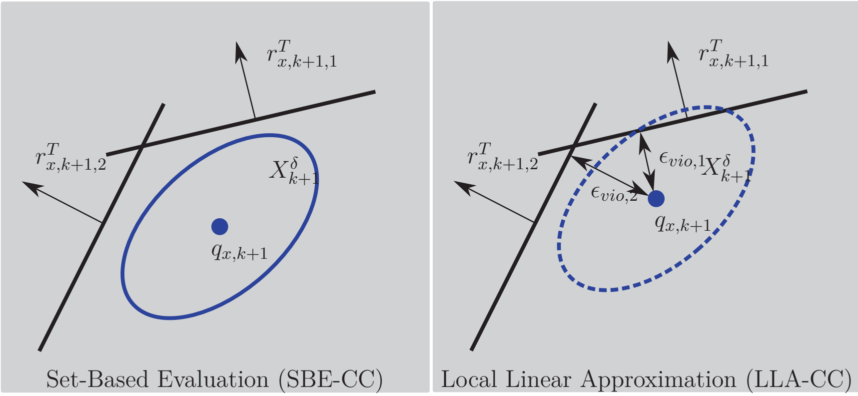


Figure 5.1: Illustration of the basic idea for SBE-CC (left) and LLA-CC (right).

The function $F_N(x)$ cannot be evaluated in closed form, but precise look-up tables or closely approximating functions can be used. According to [10], the function:

$$F_N(x) \approx F_{N,appr}(x) := 2^{-22^{1-41^{x/10}}}, \tag{5.24}$$

transfers the right part of (5.18) into nonlinear constraints:

$$\sum_{i=1}^{n_x} \epsilon_{vio,i} = \sum_{i=1}^{n_x} \left(1 - F_{N,appr}\left(\frac{b_{x,k,i} - q_{y,k,i}}{\sqrt{Q_{y,k,i}}}\right)\right) < 1 - \delta_x. \tag{5.25}$$

The idea of LLA-CC is illustrated in Figure 5.1 (right), in which $\epsilon_{vio,i}$ denotes the probability of violating the $i-$th half-plane. In [26], a local linearization of the nonlinear constraint (5.25) is used in order to obtain a semi-definite program for controller synthesis. Due to variable transformation in (5.22), the CDF depends on $q_{y,k}$, $Q_{y,k}$, and $b_{x,k}$. With linearization points $\bar{q}_{y,k}$ and $\bar{Q}_{y,k}$, chosen as the expected valuem and the covariance matrix in step k (see (5.22)), the first-order Taylor series approximation of F_N is:

$$\begin{aligned} F_{N,appr}(q_{y,k}, Q_{y,k}, b_{x,k}) &\approx F_{N,appr}(\bar{q}_{y,k}, \bar{Q}_{y,k}, b_{x,k}) + \dots \\ &\dots + \nabla_{q_{y,k}} F_{N,appr}(q_{y,k} - \bar{q}_{y,k}) + \nabla_{Q_{y,k}} F_{N,appr}(Q_{y,k} - \bar{Q}_{y,k})m \\ &=: \bar{F}_{N,appr}(q_{y,k}, Q_{y,k}, b_{x,k}), \end{aligned} \tag{5.26}$$

where the ∇-operator is used to indicate the differentiation according to the variable in the subscript.

With (5.26), the set of constraints used for the SDP becomes:

$$\epsilon_{vio,i} = 1 - \bar{F}_{N,appr}\left(q_{y,k,i}, Q_{y,k,i}, b_{x,k,i}\right), \tag{5.27a}$$

$$\sum_{i=1}^{n_x} \epsilon_{vio,i} < 1 - \delta_x, \ \forall i = \{1, 2, \dots n_x\}. \tag{5.27b}$$

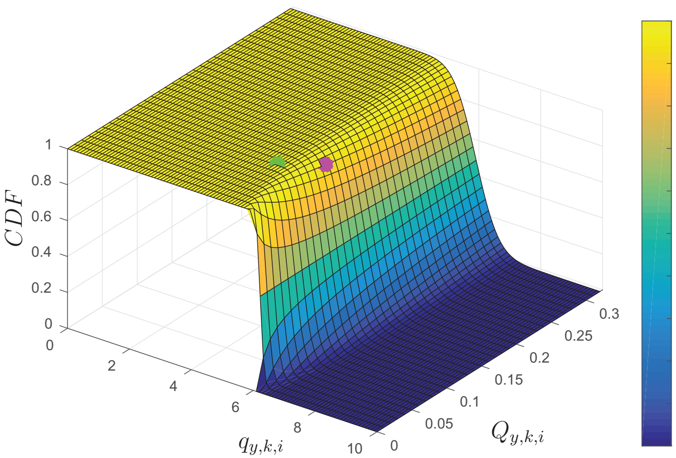


Figure 5.2: Exact values of the CDF for the current values $q_{y,k,1} \in \mathbb{R}$ and $Q_{y,k,1} \in \mathbb{R}$ (green) and the consecutive values (magenta).

Approximation Error of the Nonlinear PDF

Unfortunately, the linear approximation is not guaranteed to be an over-approximation of the uni-variate CDF, such that in some cases the true value of the CDF $F_{N,appr}$ is lower than the approximation by $\bar{F}_{N,appr}$. This issue needs to be addressed in the formulation of an algorithmic solution procedure, and is discussed in detail below.

The aim is to formulate an SDP, the solution of which constitutes a control law at time step k, and the application of this control law satisfies inter alia the chance constraint for $k+1$. But the approximation of the chance constraints by the LLA-CC approach includes a non-conservative linearization of the CDF, in which the linearization points are chosen as:

$$\bar{q}_{y,k} = R_{x,k} q_{x,k}, \;\; \bar{Q}_{y,k} = R_{x,k} Q_{x,k} R_{x,k}^T. \tag{5.28}$$

That is, if LLA-CC is chosen within the optimization problem to synthesize a controller at time step k, the satisfaction of the chance constraint for $(q_{x,k+1}, Q_{x,k+1})$ is evaluated based on a linearization at $(q_{x,k}, Q_{x,k})$. In the LLA-CC approach, instead of the original distribution an auxiliary distribution with the parameters in (5.22) is used to evaluate the contribution of each half-plane to the chance consraint. An illustration of an exemplary shape of an uni-variate CDF for $y_{k,1} \sim \mathcal{N}(q_{y,k,1}, Q_{y,k,1})$ for different values of $q_{y,k,1}$ and $Q_{y,k,1}$ is shown in Fig. 5.2, where $b_{x,k,1}$ is fixed to 6 (for $i = 1$), and the linearization points are $\bar{q}_{y,k,1} = 4.93$ and $\bar{Q}_{y,k,1} = 0.082$, shown as the green dot. Note, that in the uni-variate case, the covariance matrix reduces to a scalar value, such that: $Q_{y,k,1} \in \mathbb{R}_{\geq 0}$.

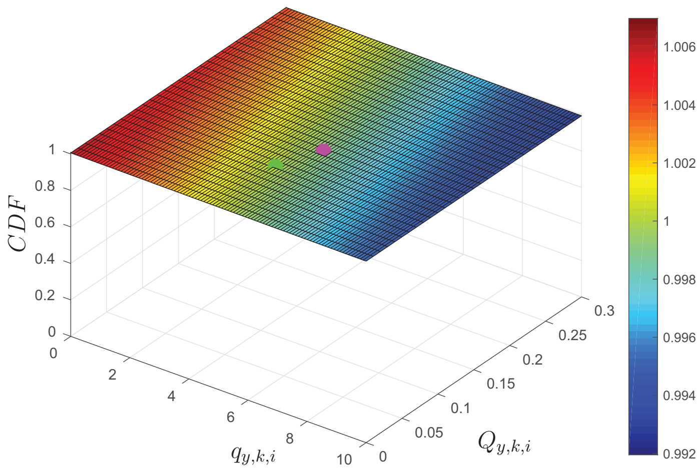


Figure 5.3: Approximated values of the CDF based on a linearization at $\bar{q}_{y,k,1}$ and $\bar{Q}_{y,k,1}$.

In this example it is further assumed, that a feasible controller is available, which transfers the current distribution of the state x_k to a consecutive distribution $x_{k+1} = \mathcal{N}(q_{x,k+1}, Q_{x,k+1})$. The considered chance constraint for the example in Fig 5.2 is specified by $\delta_x = 0.95$, and an evaluation of the CDF at the auxiliary variables $q_{y,k+1,1}$ and $Q_{y,k+1,1}$ is shown as the magenta colored dot. The value of the CDF for the magenta colored dot is $F_N(q_{y,k+1,1}, Q_{y,k+1,1}, b_{x,k,1}) = 0.9223$, which is less than the desired value of 0.95 for the chance constraint. This illustrates, that the chance constraint might be violated, even if a feasible controller exist, and the reason of this problem is the mismatch between the linearization points, and the evaluation points for the CDF.

The formulation of linear constraints does not enable the SDP solver to detect the curved shape of the function, as shown in Fig. 5.2, instead, the solver uses the plane, shown in Fig. 5.3. This figure illustrates the linearization of the true CDF function at , where the green dot is again the linearization point at $\bar{q}_{y,k,1} = 4.93$ and $\bar{Q}_{y,k,1} = 0.082$, and the magenta colored dot specifies the value of the CDF for the generated values for $q_{y,k+1,1}$ and $Q_{y,k+1,1}$; the point used by the solver for the SDP. The value of the CDF in the linearization is 0.9990 for $k+1$, and thus satisfies the chance constraint in the SDP. The choice of the current distribution as linearization points in time step k to approximate the CDF is inevitable, due to the lack of information. But the mismatch between the evaluated parameters $(q_{x,k+1}, Q_{x,k+1})$ for the chance constraint and the used parameters for the linearization $(q_{x,k}, Q_{x,k})$ might lead to a result in which the chance constraint is not satisfied for the true distribution, given by $(q_{x,k+1}, Q_{x,k+1})$.

This example illustrates a weak point of the non-conservative linearization procedure in LLA-CC. In order to use this method in the control synthesis procedure, a verification of the chance constraint is required, once the SDP is solved.

A comparison of the SBE-CC approach with the LLA-CC approach can be found in [33], and it is shown, that the consideration of elliptical bounded sets for the chance constraints results in a conservative solution, i. e. the SBE-CC approach might fail to find a feasible solution of the controller synthesis, even though one exists. In other words, it is possible to satisfy the chance constraint (5.2), even without the requirement of $X_k^\delta \subseteq \mathbb{X}_k$ (for $\delta_x = \delta$) and this leads to a conservatism when using the SBE-CC approach.

Furthermore, the consideration of a constraint of the type *ellipsoid-in-polytope* would lead to nonlinear inequalities, and thus, does not seem suitable for the intended solution of controller synthesis by SDP (Sec. 5.3.1). It is thus not further considered in this thesis.

5.2.3 Scenario-Based handling of Chance Constraints

A further alternative to approximate the chance constraint without solving the multi-dimensional integral is a *scenario-based* (SB-CC) approach [44, 133]: it uses a finite number of samples, so-called *scenarios* of the random variable $x_{k+1} \sim \mathcal{N}(q_{x,k+1}, Q_{x,k+1})$. The scenarios $x_{k+1}^{(i)}$, $i = \{1, \dots, N_{p,SB}\}$, are generated by drawing $N_{p,SB}$ samples $v_k^{(i)}$ of the disturbance $v_k \sim \mathcal{N}(q_v, Q_v)$, and by applying the computed control input $\kappa(x_k, k)$ to obtain:

$$x_{k+1}^{(i)} = A_{cl,k} q_{x,k} + B d_k + G v_k^{(i)}, \quad i = \{1, \dots, N_{p,SB}\}. \tag{5.29}$$

The control synthesis in [44, 133] is tackled by using scenarios in the optimization, hence not the original chance constraint is used, but the condition that any scenario has to satisfy:

$$R_{x,k+1} x_{k+1}^{(i)} \leq b_{x,k+1},\ i = \{1, \dots, N_{p,SB}\}. \tag{5.30}$$

Thus, for any k, a number of $N_{p,SB} \cdot n_x$ deterministic constraints has to be considered. Of course, the solution of the optimization does only refer to the selected set of samples, leading to the crucial question of how many samples are required to obtain a certain level of confidence. In [44], a formulation for a minimum sample size to achieve a certain confidence is suggested, and further refined in [45]. The following condition, derived in [45], specifies the required number $N_{p,SB}$ in order to get a desired confidence level β:

$$N_{p,SB} \geq \frac{2}{(1-\delta_x)} \left(n_o - 1 + \ln\left(\frac{1}{1-\beta} \right) \right). \tag{5.31}$$

Here, δ_x is the desired probability used in the formulation of the chance constraint, and β is the confidence parameter. n_o is the number of degrees of freedom in the optimization problem. If $N_{p,SB}$ is chosen according to (5.31), the chance constraint (5.15) is satisfied with probability not smaller than β. This second confidence parameter is needed to evaluate the reliability of a solution with a certain choice of samples. Figure 5.4 (left) shows a feasible solution, in which the linear constraint (5.30) is satisfied for any scenario $x_{k+1}^{(i)}$.

Eventually, the *scenario-based approximation* of chance constraints by using *mixed-integer* formulations (SBMI-CC) is conceivable [34, 22]: an intuitive solution is that not all scenarios have to satisfy (5.30), but only the percentage corresponding to the given probability δ_x. Let $N_{p,SBMI}$ realizations $x_{k+1}^{(i)}$, $i = \{1, \dots, N_{p,SBMI}\}$, be generated as in (5.29). The share of at least $\delta_x \cdot N_{p,SBMI}$ realizations have to satisfy (5.30). By introducing a binary vector $h_b \in \mathbb{R}^{N_{p,SBMI}}$, $h_{b,i} = \{0, 1\}$, $i = \{1, \dots, N_{p,SBMI}\}$, the above formulation can be stated as:

$$R_{x,k+1} \cdot x_{k+1}^{(i)} \leq b_{x,k+1} + h_i \cdot M, \quad i = \{1, \dots, N_{p,SBMI}\}, \tag{5.32a}$$

$$\sum_{i}^{N_{p,SBMI}} h_{b,i} \leq (1 - \delta_x) \cdot N_{p,SBMI}. \tag{5.32b}$$

In (5.32a), a *big-M*-formulation is used, enabling to fulfill the inequality for $x_{k+1}^{(i)} \notin \mathbb{X}_{k+1}$, if the corresponding entry of h_b is forced to 1 (with a large $M \in \mathbb{R}_{>0}$, see [18]). The inequality (5.32b) ensures that at most $(1 - \delta_x) \cdot N_{p,SBMI}$ samples are outside of the admissible state set $\mathbb{X}_{k+1}$. This is illustrated in Figure 5.4, where $x_{k+1}^{(i)} \notin \mathbb{X}_{k+1}$ is allowed for a certain percentage of the scenarios.

An important question is also here, how many scenarios $N_{p,SBMI}$ are required to obtain a sufficiently good approximation of the distribution, and hence of the chance constraint. To answer this question, a function for the probability of violating the chance constraint is defined:

$$H(x_k) = Pr(x_k \notin \mathbb{X}_k), \tag{5.33}$$

and it is required that $H(x_k) \leq 1 - \delta_x$ holds. The expression in (5.32b) can be seen as an empirical test of violating the chance constraint, and is denoted by:

$$H_{N_{p,SBMI}}(x_k^{(i)}) = \frac{\sum_i^{N_{p,SBMI}} h_{b,i}}{N_{p,SBMI}}. \tag{5.34}$$

The quality of approximating (5.33) by (5.34) can be quantified by the difference between both quantities depending on $N_{p,SBMI}$. The upper bound of the difference will be denoted by a parameter ϑ, and is used as a measure for the quality of the approximation. The following Lemma specifies a minimum number of scenarios $N_{p,SBMI}$ needed to obtain a given level of confidence β with a specified precision ϑ.

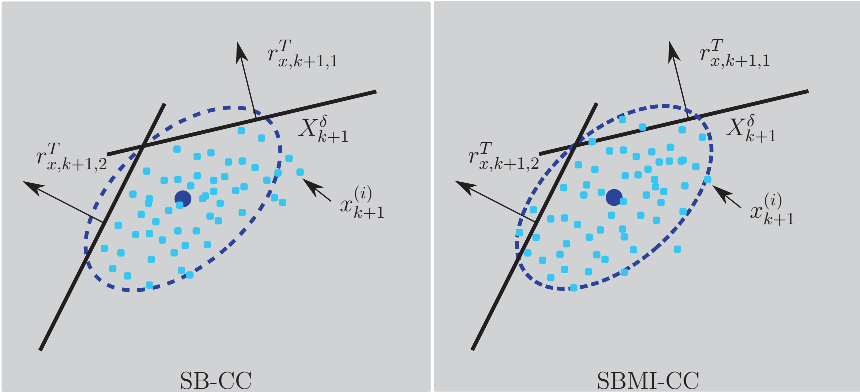


Figure 5.4: Basic idea of the two scenario-based approaches to consider chance constraints: On the left, the scenarios $x^{(i)}_{k+1}$ are used to formulate $N_{p,SB}$ linear constraints of an optimization problem, and the optimal solution is feasible for all considered scenarios. On the right, the scenarios are used to formulate $N_{p,SBMI}$ binary constraints of an optimization problem, and the optimal solution is feasible for $\delta_x N_{p,SBMI}$ scenarios, i. e. $(1-\delta_x)N_{p,SBMI}$ scenarios may lie outside of $\mathbb{X}_k$.

Lemma 5.2. *Let* (5.33) *specify the probability of violating the chance constraint* (5.2)*, and* (5.34) *formulates the empirical probability. The confidence of*

$$|H_{N_{p,SBMI}}(x^{(i)}_k) - H(x_k)| \leq \vartheta \tag{5.35}$$

is greater than β, if it holds that:

$$N_{p,SBMI} \geq \frac{1}{2\vartheta^2} \ln\left(\frac{2}{1-\beta}\right). \tag{5.36}$$

Proof The confidence β can be expressed as:

$$Pr\left(|H_{N_{p,SBMI}}(x^{(i)}_k) - H(x_k)| \leq \vartheta\right) = \beta. \tag{5.37}$$

Following Hoeffdingers inequality [81], it holds that

$$Pr\left(|H_{N_{p,SBMI}}(x^{(i)}_k) - H(x_k)| \leq \vartheta\right) \geq 1 - 2\exp(-2\vartheta^2 N_{p,SBMI}). \tag{5.38}$$

By inserting (5.37) in (5.38) and by algebraic reformulation, (5.38) leads to the condition for $N_{p,SBMI}$ in (5.36). □

Lemma 5.2 specifies the required number of scenarios to obtain a sampled approximation of the distribution with precision ϑ and confidence β. It is obvious, that the effort to increase the precision ϑ is much higher, than to increase the confidence δ.

The different means sketched above to handle chance constraints are now used within a procedure to solve the set-to-target control problem 5.1.

5.3 Controller Synthesis for Affine Probabilistic Systems

This section proposes an algorithm to solve problem 5.1 by semi-definite programming, while considering the approximations of chance constraints as addressed before. The SDP provides the continuous control parameters (K_k, d_k) for any time step k.

The following subsections derive the SDP to be evaluated and proposes an algorithmic solution procedure for controller synthesis, in which the solution of the optimization problem is embedded.

5.3.1 Semi-Definite Programming for Continuous Input

The objective of obtaining the parameters (K_k, d_k) for the continuous control law $\kappa(x_k, k)$ is addressed by solving an SDP. The optimization problem contains linear matrix inequalities, which constrain the mean vector $q_{x,k}$, the covariance matrix $Q_{x,k}$, and ensure that the input constraint and the chance constraint (according to LLA-CC, SB-CC, SBMI-CC) are satisfied. In the following, the relevant sets of LMIs for formulating the SDPs are derived.

The covariance matrix $Q_{x,k+1}$ is over-approximated by a matrix variable S_{k+1}, which serves as a degree of freedom in the optimization:

$$S_{k+1} \geq Q_{x,k+1} = A_{cl,k} Q_{x,k} A_{cl,k}^T + G Q_v G^T. \tag{5.39}$$

By applying the Schur complement (see in [41]), the inequality can be written as LMI:

$$\begin{bmatrix} S_{k+1} & A_{cl,k} Q_{x,k} & G Q_v \\ Q_{x,k} A_{cl,k}^T & Q_{x,k} & 0 \\ Q_v G^T & 0 & Q_v \end{bmatrix} \geq 0. \tag{5.40}$$

This linear matrix inequality ensures $S_{k+1} \geq Q_{x,k+1}$ for a suitable choice of the feedback gain matrix K_k according to (5.13b). The additional constraint:

$$\operatorname{trace}(S_{k+1}) \leq \operatorname{trace}(Q_{x,k}) \tag{5.41}$$

implements a contraction of the covariance matrix $Q_{x,k+1}$. The LMI in (5.40) is similar to the approximating LMI for AS in (4.48d), but here one advantage of the probabilistic formulation appears. Since the new covariance matrix for the next time step is directly defined by the sum of the current covariance matrices $Q_{x,k}$ and Q_v, no scaling variable ν is required, as in (4.48d). The resulting SDP has one degree of freedom lesser.

Since the closed-loop dynamics for the center point $q_{x,k}$ in (5.13a) is a time-varying affine system, the concept of flexible Lyapunov function is adapted here to enforce a convergence of $q_{x,k}$. As shown in Chapter 4, the following equations are required to implement the flexible Lyapunov function:

$$q_{x,k+1}^T M q_{x,k+1} - \rho q_{x,k}^T M q_{x,k} \leq \alpha_k, \tag{5.42}$$

$$q_{x,k+1} = A_{cl,k} q_{x,k} + B d_k + G q_v, \tag{5.43}$$

with a matrix M symmetric, positive definite ($M = M^T \geq 0$), and

$$\alpha_k \leq \max_{i \in \{1,\ldots,k\}} \omega^i \alpha_{k-i}, \tag{5.44}$$

with $\omega \in [0,1)$, see [114].

As previously introduced in Proposition 4.2, the following LMI constraints enforce that the input constraints (5.10) are satisfied in each time step.

Proposition 5.1. *The input constraint $u_k = -K_k x_k + d_k \in \bar{U}$ holds for K_k, d_k, and all $x_k \in X_k^\delta = \varepsilon(q_{x,k}, Q_{x,k}^\delta)$ if:*

$$\begin{bmatrix} (b_{u,i} - r_{u,i}(d_k - K_k q_{x,k}))I_n & -r_{u,i} K_k (Q_{x,k}^\delta)^{1/2} \\ (-r_{u,i} K_k (Q_{x,k}^\delta)^{1/2})^T & b_{u,i} - r_{u,i}(d_k - K_k q_{x,k}) \end{bmatrix} \geq 0, \quad \forall i = \{1,\ldots,n_u\} \tag{5.45}$$

As already elaborated in Section 5.2, different means to consider the chance constraints within an SDP exists. The three different approaches (LLA-CC, SB-CC, SBMI-CC), considered in this thesis, are embedded and evaluated regarding their computational complexity, and effect on the control performance. The necessary LMI constraints are now collected to state the SDP problem for each alternative of considering the chance constraints.

Here, the trace of the shape matrix $J(\varepsilon(q,Q)) = \text{trace}(Q)$ is chosen, and supplemented by the weighted norms of $q_{x,k+1}$ and input u_k. The minimization of the trace of the shape matrix Q results in the reduction of the semi-axes of the confidence reachable set simultaneously. The choice of the cost function, and especially the weights $\mu_1, \mu_2 > 0$, can promote a fast convergence of the center points and low control effort. The SDP problem to be solved for each time step k is:

$$\min_{S_{k+1},K_k,d_k,\alpha_k} \quad J_k = \operatorname{trace}\left(\begin{bmatrix} S_{k+1} & 0 & 0 \\ 0 & \mu_1\|q_{x,k+1}\| & 0 \\ 0 & 0 & \mu_2\|u_k\| \end{bmatrix}\right), \tag{5.46}$$

subject to:

(5.40), (5.41) for shape matrix convergence,

(5.42), (5.43), (5.44) for center point convergence,

(5.45) for input constraint,

(5.27) for the chance constraint according to LLA-CC, or ,

(5.30) for the chance constraint according to SB-CC, or ,

(5.32) for the chance constraint according to SBMI-CC .

Note that in contrast to (5.45), the constraints (5.40) use the covariance matrix $Q_{x,k}$ instead of the scaled covariance matrix $Q^{\delta}_{x,k} = Q_{x,k}c_x$. The use of $Q^{\delta}_{x,k}$ in (5.45) is essential, since it has to be ensured, that the control law and input constraint is satisfied for all $x_k \in X^{\delta}_k = \varepsilon(q_{x,k}, Q^{\delta}_{x,k})$. In contrast, the constraints (5.40) and (5.41) ensure the contraction of X^{δ}_k, which is mainly determined by $Q_{x,k}$. The scaling parameter c_x can be neglected.

Due to the randomized method in SB-CC/SBMI-CC for the satisfaction of the chance constraints, the result of finding a feasible solution thus depends on the current set of scenarios $x^{(i)}_{k+1}$ for these methods. A high value for the desired confidence level β clearly reduces the dependency on the drawn scenarios $x^{(i)}_k$, but also results in a large number of needed scenarios. Each scenario formulates a constraint in the SDP, which increases the computational complexity and makes the solution of the SDP impractical.

5.3.2 Algorithmic Solution Procedure

The determination of the controller in each time step to solve the stated control problem in Problem 5.1 is formulated in Algorithm 5.1. The feasible solutions of the embedded SDP in each time step provide a continuous control law $\kappa(x_k, k)$, satisfying the input and chance constraints, and contributing to drive the system towards the terminal set $\mathbb{T}$. The algorithm starts with a computation of the confidence reachable set X^{δ}_k (step 1) in each iteration by applying Lemma 3.2 to the distribution $x_k \sim \mathcal{N}(q_{x,k}, Q_{x,k})$. In the second step, the chance constraint is considered in terms of the chosen variant. The third step solves the stated SDP. The fourth step uses the control tuple (K_k, d_k) to compute the distribution of the following state x_{k+1} according to (5.14). If LLA-CC was chosen for the approximative chance constraint consideration, the satisfaction of the original chance constraint is verified in step five of the algorithm.

A feasible solution of the optimization problem might fail to satisfy the original chance constraint once the solution is computed (see Sec. 5.2.2). A way out of this dilemma is to solve the SDP again, but with the new information about the next distribution $(q_{x,k+1}, Q_{x,k+1})$ from the solution of the current SDP. This new information can be used as the linearization points in the Taylor series approximation of the CDF in (5.26). It is very likely, that the new solution of the SDP is similar to the initial one, such that the application of the controller leads to the same or a similar distribution $(q_{x,k+1}, Q_{x,k+1})$. The knowledge of the consecutive distribution leads to an improved approximation, due to the new location of the linearization point. Unfortunately, a satisfaction of chance constraint in the second solution of the SDP cannot be guaranteed, and if the re-verification of the second solution fails, the algorithm stops without success. An adequate countermeasures is to reduce either the probability δ_x of the chance constraint, or to reduce the confidence level δ for the reachable sets. Another countermeasure is to adapt the weights in the cost function of the SDP.

The computation terminates successfully, if a confidence ellipsoid $X_k^\delta \in \mathcal{E}$ is contained in the target set $\mathbb{T}$, meaning that Problem 5.1 is solved. To ensure termination of the algorithm, also if $\mathbb{T}$ is not reached in a reasonable number of iterations, the additional criterion:

$$\pi_{k+1} = ||q_{x,k+1} - q_{x,k}|| \leq \pi_{min}, \tag{5.47}$$

is used with a parameter $\pi_{min} \in \mathbb{R}$. Step six and seven compute the parameter π_{k+1} and increase k, respectively.

Let k_π denote the iteration in which the criterion holds first (if existing). If this applies (indicating termination without success), different countermeasures can be applied: If q_{x,k_π} and Q_{x,k_π} indicate a small distance to satisfying $X_k^\delta \subseteq \mathbb{T}$, a reduction of π_{min} can be reasonable. Another possibility is to reduce the confidence level δ, which results in smaller ellipsoids and thus increases the probability to terminate successfully.

Lemma 5.3. *Problem 5.1 is successfully solved, if Algorithm 5.1 terminates with* $X_N^\delta \subseteq \mathbb{T}$, $N \leq k_\pi$. *The solution provides a control law* (5.10) *which steers any initial state* $x_0 \in X_0^\delta$ *with probability* δ *into the target set* $\mathbb{T}$ *in* N *steps. The controlled dynamics with disturbances* $v_k \sim \mathcal{N}(q_v, Q_v)$ *is attractive with confidence* δ *according to Definition 5.3, and stable with confidence* δ *if* $trace(Q_{x,k+1}) < trace(Q_{x,k})$ $\forall k \in \{0, \ldots, N-1\}$ *holds. Furthermore, the input constraint* $u_k \in U$ *holds for all* $0 < k < N$. *The chance constraint* (5.2) *is satisfied with :*

- $Pr(x_k \in \mathbb{X}_k) \geq \delta_x$ *for all* $0 < k < N$ *(LLA-CC), or*
- $Pr(x_k \in \mathbb{X}_k) \geq \delta_x$ *with confidence* β *for all* $0 < k < N$ *(SB-CC), or*
- $Pr(x_k \in \mathbb{X}_k) \geq \delta_x$ *with confidence* β *and precision* ϑ *for all* $0 < k < N$ *(SBMI-CC).*

Algorithm 5.1. ***Probabilistic Ellipsoidal Control Algorithm for Affine Probabilistic Systems (PECA-APS)***

Given: (5.1) *with* $x_0 \sim \mathcal{N}(q_{x,0}, Q_{x,0})$, $v_k \sim \mathcal{N}(q_v, Q_{v,k})$, $\mathbb{X}_0 \in \mathcal{P}$, *and* $U \in \mathcal{P}$; $\mathbb{T}$, δ, π_{min}, $\rho \in (0, 1]$, *and* $\alpha_0 \in [0,\ 1]$, $\omega \in [0, 1)$; (5.2) *with* δ_x, β *(for SB-CC/SBMI-CC), or* ϑ *(for SBMI-CC)*

Define: $k := 0$, $\pi_0 := \pi_{min}$

while $X_k^\delta \nsubseteq \mathbb{T}$ *and* $\pi_k \geq \pi_{min}$ ***do***

1. *compute the confidence ellipsoid* X_k^δ *by* (5.5)
2. ***if*** *SB-CC or SBMI-CC* ***do***
   - *draw* $N_{p,SB/SBMI}$ *samples* $v_k^{(i)}$ *of* $v_k \sim \mathcal{N}(q_v, Q_v)$, *with* $N_{p,SB/SBMI}$ *according to* (5.31) *for SB-CC, or* (5.36) *for SBMI-CC*
   - *generate the scenario* $x_{k+1}^{(i)}$ *according to* (5.29)
   - *formulate the chance constraint according to* (5.30) *for SB-CC, or* (5.32) *for SBMI-CC*

   else
   - *formulate the chance constraint according to* (5.25) *for LLA-CC*

   end
3. *solve the optimization problem* (5.46)
4. *compute the distribution of* $x_{k+1} \sim \mathcal{N}(q_{x,k+1}, Q_{x,k+1})$ *with the control tuple* (K_k, d_k) *according to* (5.14)
5. ***if*** *LLA-CC* ***do***
   - *check chance constraint* (5.15)*:*

     if $\sum_{j=1}^{n_x} 1 - F_N(q_{y,k+1,j}, Q_{y,k+1,j}, b_{x,k,j}) > 1 - \delta_x$ ***do***
     - *Solve optimization problem* (5.46) *again and consider* $q_{x,k+1}, Q_{x,k+1}$ *for linearization in* (5.26).
     - ***if*** $\sum_{j=1}^{n_x} 1 - F_N(q_{y,k+1,j}, Q_{y,k+1,j}, b_{x,k,j}) > 1 - \delta_x$ ***do*** *EXIT* ***end***

     end

   end
6. *compute* π_{k+1} *according to* (5.47)
7. $k := k + 1$

end while

Proof

According to Lemma 3.2, it holds that $Pr(x_k \in X_k^\delta) = \delta$ with $X_k^\delta = \varepsilon(q_{x,k}, Q_{x,k}c_x)$, and following (5.14), the state distribution for $k+1$ is $x_{k+1} \sim \mathcal{N}(q_{x,k+1}, Q_{x,k+1})$. By computing $X_{k+1}^\delta = \varepsilon(q_{x,k+1}, Q_{x,k+1}c_x)$ with the scaling factor c_x according to (3.31), the Algorithm 5.1 ensures that $Pr(x_{k+1} \in X_{k+1}^\delta) = \delta$. Hence, it is guaranteed that $Pr(x_k \in X_k^\delta) = Pr(x_{k+1} \in X_{k+1}^\delta) = \delta$ for all $k \in \{0, 1, \ldots, N-1\}$, such that $Pr(x_N \in X_N^\delta) = \delta$ follows from induction.

Furthermore, a successful termination implies the existence of a feasible solution of the optimization problem (5.46) in any time step k. This further implies that the input constraint $u_k \in \bar{U}_k \subseteq U$ holds according to Lemma 5.1.

A feasible solution of the SDP in each time step provides a fulfillment of the chance constraint (5.2) with:

- for LLA-CC: $Pr(x_k \in \mathbb{X}_k) \geq \delta_x$ for all $0 < k < N$ according to (5.25), or
- for SB-CC: $Pr(x_k \in \mathbb{X}_k) \geq \delta_x$ with confidence β for all $0 < k < N$ with $N_{p,SB}$ according to (5.31), or
- for SBMI-CC: $Pr(x_k \in \mathbb{X}_k) \geq \delta_x$ with confidence β and precision ϑ for all $0 < k < N$ with $N_{p,SBMI}$ according to (5.36) and Lemma 5.2.

Attractiveness of the mean vector $q_{x,k}$ towards 0 in the sense of Def. 5.3 follows from the fact that $X_N^\delta = \varepsilon(q_{x,N}, Q_{x,N}^\delta) \subseteq \mathbb{T}$ implies that $||\bar{q}|| \geq ||q_{x,N}||$ exists as well as $||Q_T|| \geq ||Q_{x,N}^\delta||$ with $Q_T = \bar{Q}$. For $\alpha_k = 0 \ \ \forall k \in \{0, \ldots, N-1\}$, stability with confidence δ according to (5.7) follows, since: (i) (5.42) implies with $\rho \in [0, 1)$ that $||q_{x,k+1}|| < ||q_{x,k}||$; (ii) $||Q_{x,k+1}|| \leq ||Q_{x,k}||$ is enforced by (5.41). The upper bounds for the norm of the expected value $q_{x,N}$ and the covariance matrix $Q_{x,N}$ are given by parametrization of the ellipsoidal terminal set $\mathbb{T}$.

□

5.4 Illustration of the Control Algorithm

To illustrate the principle of the proposed algorithm, it is applied to a simple example with three continuous states and two inputs. The different solutions, obtained by the application of the algorithm for the different variants to consider chance constraints are used to evaluate and discuss each approach.

5.4.1 System Model

The system stems from a time discretization of a continuous-time linear system of an academic example, and the initial distribution of the system state and the disturbance are given by:

$$x_0 \sim \mathcal{N}(q_{x,0}, Q_{x,0}) \text{ with } q_{x,0} = \begin{bmatrix} 2 \\ 9 \\ 2 \end{bmatrix}, \quad Q_{x,0} = \begin{bmatrix} 0.2 & 0.5 & 0 \\ 0.5 & 0.1 & 0.5 \\ 0 & 0.5 & 0.2 \end{bmatrix}, \tag{5.48}$$

$$v_k \sim \mathcal{N}(0, \Sigma) \text{ with } \Sigma = \begin{bmatrix} 0.2 & 0.5 & 0 \\ 0.5 & 0.1 & 0.5 \\ 0 & 0.5 & 0.2 \end{bmatrix}. \tag{5.49}$$

The system dynamic is specified by the following difference equation with an unstable eigenvalue of the state matrix:

$$\begin{aligned} x_{k+1} = & \begin{bmatrix} 0.89 & 0.05 & -0.03 \\ 0 & 1.07 & 1.00 \\ 0 & 0.05 & 0.92 \end{bmatrix} x_k \\ & + \begin{bmatrix} 0.54 & 0.001 \\ -0.25 & 0.23 \\ 0.53 & 0.51 \end{bmatrix} u_k + \begin{bmatrix} 0.1 & 0 & 0 \\ 0 & 0.2 & 0 \\ 0 & 0 & 0.3 \end{bmatrix} v_k. \end{aligned} \tag{5.50}$$

The input u_k is constrained to:

$$u_k \in U = P_H(R_u, b_u), \tag{5.51}$$

with

$$R_u = \begin{bmatrix} 1 & 0 \\ -1 & 0 \\ 0 & 1 \\ 0 & -1 \end{bmatrix}, \; b_u = \begin{bmatrix} 7 \\ 7 \\ 7 \\ 7 \end{bmatrix}. \tag{5.52}$$

The feasible state space is given by:

$$\mathbb{X}_k = P_H(R_{x,k}, b_{x,k}), \tag{5.53}$$

with

$$R_{x,k} = \begin{bmatrix} 1 & -1 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 & 1 \end{bmatrix}^T, \tag{5.54}$$

$$b_{x,k} = \begin{bmatrix} 6 & 0.5 & 1 & 10.8 & 0.5 & 3 & 4 & 5 \end{bmatrix}^T. \tag{5.55}$$

The considered chance constraint for this region is:

$$Pr\left(x_k \in \mathbb{X}_k\right) \geq \delta_x = 0.95, \tag{5.56}$$

The target set defined as:

$$\mathbb{T} = \varepsilon\left(0, \begin{bmatrix} 0.96 & 0.64 & 0.24 \\ 0.64 & 0.8 & 0.64 \\ 0.24 & 0.64 & 0.96 \end{bmatrix}\right). \tag{5.57}$$

The remaining parameters of Algorithm 5.1 are chosen as follows:

- $\delta = 0.95$
- $\gamma_{min} = 0.01$
- for SB-CC and 3: $\beta = 0.9$
- for SBMI-CC: $\vartheta = 0.05$

The cost function evaluates the approximated covariance matrix by its trace, and the norm of the expected value: $J = \text{trace}\left(\begin{bmatrix} S_{k+1} & 0 \\ 0 & 0.8\|q_{k+1}\| \end{bmatrix}\right)$, $\mu_2 = 0$.

The computation is performed on a standard 4.2-Ghz-Quad-Core CPU with 16 GB of RAM. The SDP problems are solved by using the commercial solver MOSEK, which is implemented in Yalmip. The MISDP is solved by a standard branch-and-bound algorithm (Yalmip built-in), and the SDP in each node of the MISDP is solved by MOSEK.

5.4.2 Discussion of the different results

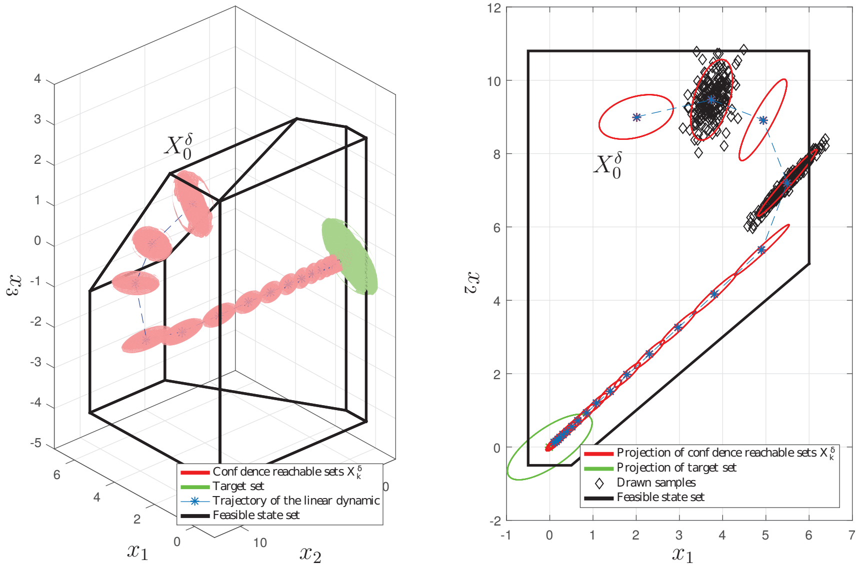
In order to compare the three mentioned variants, some characteristic values are listed in Table 5.1.

Method	Comp time		time steps	cum. cost
	total	avg.	k	$\sum_{k=0}^{N} J_k$
LLA-CC	8.66 s	0.4 s	20	66.40
SB-CC	31 s	1.48 s	21	70.75
SBMI-CC	35 s	1.75 s	20	66.00

Table 5.1: Comparison of the results with different variants to consider the chance constraint (5.15).

The initial confidence set X_0^δ is transferred in $N = 21$ steps for SB-CC, and $N = 20$ steps for LLA-CC and SBMI-CC into the terminal set $\mathbb{T}$. The cumulative costs $\sum_{k=0}^{N} J_k$, used as a criterion for comparison of the control behavior, turns out to be the lowest for the third variant. For this example, the three variants obviously differ in the required computational time. Table 5.1 shows the total time (total) and the average time (avg.). The total time specifies the time to execute the algorithm, and the average time indicates the time required to solve an optimization problem in one iteration of the algorithm.

The LLA-CC method is the fastest with a total computational time of 8.66 s. This is due to the average computational time per iteration, i.e. each SDP is solved faster compared to the variants SB-CC and SBMI-CC. The other two methods have a higher average computational time, because the SDP (MISDP) includes significantly more constraints, since each scenario is considered with a distinct set of constraints. It is noticeable, that the standard branch-and-bound algorithm of



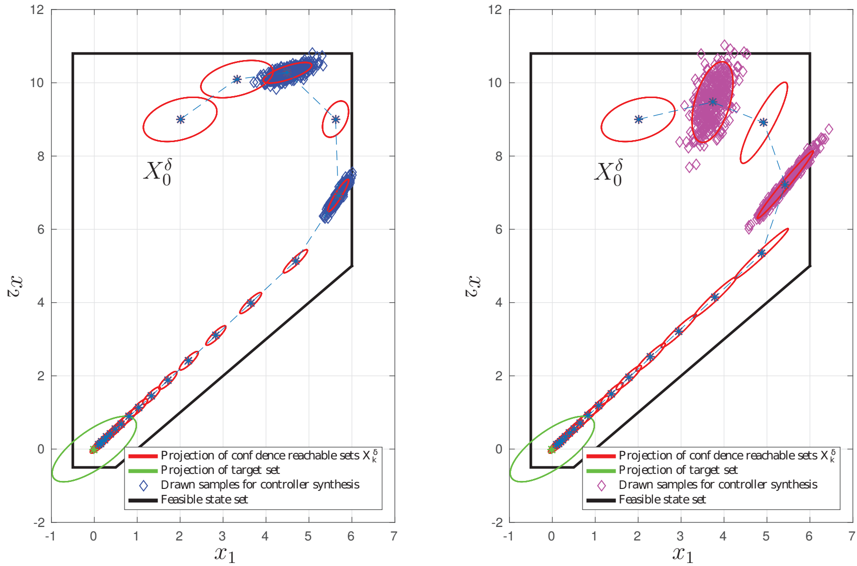
(a) Evolution with LLA-CC for the chance constraint consideration.

(b) Projection of the evolution with LLA-CC with drawn samples to illustrate, that a feasible .

Figure 5.5: Simulation of the evolution of system (5.1) with the computed controller by a feasible solution of Algorithm 5.1.

Yalmip is relatively slow, i.e. reduced computational times for SBMI-CC can be expected, when using a commercial solver for the branch-and-bound problem in combination with MOSEK for the solution of the MISDP.

The resulting evolution of the confidence reachable sets X_k^δ for LLA-CC is shown in Fig. 5.5a in three dimensions. The admissible polytope $\mathbb{X}_k$ is indicated as a black framed box, and the terminal set $\mathbb{T}$ (green) is partially outside of $\mathbb{X}_k$. For the purpose of analyzing the behavior, a projection of the confidence sets in the x_1-x_2-space is shown in Fig. 5.5b: the 2-dimensional ellipsoids are steered into the terminal region $\mathbb{T}$, and a successful termination of the algorithm guarantees that all constraints, including the input and chance constraints, are satisfied. Additionally, Fig. 5.5b contains 200 drawn scenarios from the current distribution for arbitrarily chosen times $k = 2$ and 4. It can be seen, that some samples are not in $\mathbb{X}_k$, and also the reachable set X_4^δ is not completely contained in $\mathbb{X}_k$, illustrating the stochastic character of the generated controller: Even if the probabilistic reachable set is considered with confidence $\delta = 0.95$, it is possible to satisfy a chance constraint with $\delta_x = 0.95$, and the reachable set being partially outside of $\mathbb{X}_k$. The solution



(a) Projection of the resulting evolution of the confidence reachable sets X_k^δ for SB-CC with drawn samples.

(b) Projection of the resulting evolution of the confidence reachable sets X_k^δ for SBMI-CC with drawn samples.

Figure 5.6: Simulation of the results with SB-CC (left) and SBMI-CC (right) with drawn samples used in the optimization problem to illustrate the differences of the both variants.

generated by the algorithm with SB-CC and SBMI-CC for the chance constraint consideration is shown in Fig. 5.6. The results in Fig. 5.6 include drawn samples as well, and it is apparent that all samples in Fig. 5.6a are inside the feasible state set, whereas the samples in 5.6b are partially outside. As required by a feasible solution of the optimization problem (5.46) with SB-CC, all samples have to satisfy eq. (5.30), which is not required for SBMI-CC in the corresponding formulation of the optimization problem. The necessity for the samples to be inside the feasible state set in SB-CC leads to a more conservative solution of the control problem. Comparing the evolution of the reachable set for the first four time steps, it is obvious that the solution with SB-CC needs five time steps to fall under the $x_2 = 6$ line, whereas the solution with SBMI-CC is at $k = 4$ already below this line. The remaining behavior is a straight movement to the target set and very similar in both simulations, since no boundaries of the feasible set are relevant for the solution.

Note, that the drawn samples in Fig. 5.5b are generated only for illustration, but the samples for SB-CC and SBMI-CC in Fig. 5.6 were drawn when solving the optimization problem. As already mentioned in Sec. 5.2.3 another set of samples could result in other solution, especially for SB-CC. It is not unlikely to fall outside of the feasible set with a new set of samples, and one way to reduce the randomness in the solution at this point, is to increase the number of samples.

5.4.3 General Evaluation of the Chance Constraint Approximation

So far, the three considered variants for the evaluation of chance constraints were compared based on the results for the numerical example, now a more general comparison is appended.

The parametrization of Algorithm 5.1 depends on the current dynamics and the control problem. The computational complexity is determined by the chosen variant used for the consideration of the chance constraints. Comparing the increase of memory consumption and computational effort for an SDP without any consideration of chance constraints and the consideration by one of the mentioned approaches, the LLA-CC method has the lowest increase of computational effort and memory consumption. In LLA-CC only $n_x + 1$ additional constraints in the optimization problem (5.46) have to be considered. The increase of memory consumption for SB-CC and SBMI-CC depends on the chosen level of confidence β, and the precision ϑ for SBMI-CC, since $N_{p,SB} \cdot n_x$ or $N_{p,SBMI} \cdot n_x + 1$ additional constraints have to be considered. In general, a high confidence and precision will lead to a significant increase in the number of constraints, and hence of memory consumption compared to LLA-CC. Additionally, the integer optimization in SBMI-CC complicates the optimization problem, and leads to an additional increase of the computational complexity and the memory consumption, compared to LLA-CC.

A general advise for the utilization of a specific variant for the approximation of the chance constraints is hard to provide. It depends very much on the current application, and on the desired features of the solution. However, if computational complexity and memory consumption is not critical, SB-CC and SBMI-CC are a good choice, since a high number of particles will provide a most-likely solution of the probabilistic behavior. The use of SB-CC or SBMI-CC is mandatory, if the probabilistic distribution is not Gaussian, since the controller synthesis is based on scenarios and not on the underlying distribution, as in LLA-CC. LLA-CC is a good choice, if computational time is important. The integration of only one constraint for each half-plane result in a relative compact, yet powerful formulation of the optimization problem. Furthermore, LLA-CC combines the two features of a slim formulation of the chance constraints and a non-conservative solution of the optimization problem. Even though, the compact formulation in LLA-CC can be solved efficiently, a verification of the solution is mandatory. A remaining drawback of LLA-CC is the possibility to over-estimate the true probability of being inside the feasible state set, due to the linearization. A reduction of this outcome can be achieved by a successive improvement of the linearization by adapting the linearization points.

Once the important features of the solution for the set-to-target control problem are identified, a suitable approximation of the chance constraints can be chosen as desired.

It is important to point out, that a different interpretation of the sampled scenarios in SB-CC and SBMI-CC applies: for SB-CC, the sampling procedure approximates the chance constraint in a convex program, and the optimal solution satisfies the chance constraint with a high confidence β. In contrast, the scenarios $x_k^{(i)}$ in SBMI-CC approximate the distribution $x_k \sim \mathcal{N}(q_{x,k}, Q_{x,k})$ with a defined precision ϑ, and the solution satisfies the chance constraint also with a confidence β. These different interpretations lead to two different sizes $N_{p,SB}$ and $N_{p,SBMI}$ of sample sets. The set sizes of both approaches tend to ∞ for $\beta \to 1$, i.e. a confidence with probability of 1 can only be achieved with infinitely many samples. Furthermore, the size of the sample set in SBMI-CC tends to infinity for $\vartheta = 0$. In LLA-CC, no confidence level or precision parameter is needed, since the multivariate distribution is divided into n_x uni-variate distributions, and hence, the CDF of a standard normal distribution is applicable. The approximations in LLA-CC stem from the use of Boole's inequality and the linearization of the CDF. The computational effectiveness and the small number of required constraints make LLA-CC a powerful approach to consider chance constraints for multivariate normally random variables. However, while SB-CC and SBMI-CC are computationally more demanding, these two approaches can handle any distribution and are not limited to normal distributed random variables as LLA-CC.

5.5 Discussion of the Controller Synthesis for Affine Probabilistic Systems

This chapter elevates the introduced reachable set computation for deterministic systems in Chapter 4 to a new system class, namely affine probabilistic systems. While the considered system class in Chap. 4 models merely deterministic effects, the system class of APS introduces randomness in terms of random distributions into the control task. The randomness affects the continuous dynamics, since the initial state and the affine disturbances are assumed to be Gaussian variables. Furthermore, the distribution of the system state leads to a probabilistic interpretation of the reachable set computation, and it is shown that the computed reachable sets X_k^δ contain the continuous system state x_k with a predefined probability δ.

Another new aspect of this chapter is the consideration of state constraints formulated as chance constraints due to the stochasticity of the system dynamics. These constraints restrict the system state to a state polytope, which can be time-varying. Different approximative approaches to integrate the chance constraints within the optimization based controller synthesis are presented, and discussed in this chapter. Algorithm 5.1 includes the different formulations for the chance constraints and completes the proposed method to synthesize a sequence of control parameters (K_k, d_k) for APS. The main asset of the scenario-based approaches is the flexibility concerning the probabilistic distribution: These approaches use solely the scenarios and, once determined, the underlying distribution does not matter anymore. While LLA-CC uses local approximations of the CDF for a Gaussian distribution, the independence of the type of distribution makes the scenario-based approaches in SB-CC and SBMI-CC very attractive. However, the most commonly used form of randomness in system dynamics are still the use of Gaussian variables. The price to pay for the flexibility is the computational effort to obtain the solution of the optimization problem. Each sampled scenario leads to a linear inequality constraint in the SDP, which increases the number of constraints in both approaches. Additionally in SBMI-CC, the number of discrete variables in the MISDP scales linearly with the number of scenarios. This increase of computational effort can exemplarily be seen in the numerical example, and should not be neglected. Concerning the computational complexity, LLA-CC is an efficient approach. The basic idea to separately consider the contribution of the multivariate normal distribution for each half-plane of the feasible state set, resulting in an auxiliary uni-variate distribution for each half plane. The application of Boole's inequality over-approximates the chance constraint. In the initial formulation from Blackmore in [33], the resulting optimization problem is nonlinear due to the CDF. In contrast, this thesis suggests a local linearization of the CDF to include the chance constraints as a set of linear inequalities in the SDP (cf. [26]). The increase of the linear inequalities in the optimization problem scales linearly with the number of half-planes n_x of the polytope $\mathbb{X}_k$. This number will be in general much lower than the number of scenarios needed for an appropriate approximation of the distribution. The efficient formulation is a favorable property of LLA-CC, but due to the local linearization, an a posteriori verification using the resulting control law is necessary to guarantee a satisfaction of the chance constraints.

The solution of the SDP is embedded into a control algorithm, and a successful termination provides a control law, which renders the system stable. The system is stable in the sense of Def. 5.3 for a specified and usually high percentage of realizations of the random state x_k. Furthermore, while the successful termination of the algorithm in Chap. 4 provides a controller being robust against all bounded disturbances, the statement of robustness is not valid for the solution generated by

Algorithm 5.1. In this chapter robustness is to be understood in a probabilistic sense, meaning that the control objective is achieved with a certain likelihood if probabilistic disturbances are present. At first, the introduction of randomness seem to impede the controller synthesis, but in fact it facilitates the synthesis in several ways. The transition from deterministic to probabilistic reachable sets is paired with the introduction of a new parameter δ, in order to specify the desired confidence of the set. This confidence level scales the size of the reachable ellipsoid X_k^δ and can be used as a tuning parameter in the control algorithm. If the algorithm terminates without success, δ can be reduced to increase the chance of terminating successfully.

Another advantage of the probabilistic consideration in APS over the deterministic consideration in AS can be seen in the LMI for the over-approximation of the next shape matrix S_{k+1}(see. (4.16) and (5.40)). Due to the Minkowski addition in the deterministic case, the shape matrix for $k+1$ is a linear combination of the shape matrices $Q_{x,k}$ and Q_v, and the quality of the approximation is determined by the optimization variable ν in (4.16). This additional optimization variable is eliminated in the probabilistic case, since the covariance matrix of the next time step is solely determined by the sum of the covariance matrices of the current time step. A reduction of the necessary optimization variables facilitates the solution of the SDP. Furthermore, as previously mentioned in Chap. 4, the Minkowski sum of two ellipsoidal is in general not an ellipsoid, and the presented formulation are used to over-approximation the true set by a new ellipsoid of minimum volume. This cumulation of over-approximations in each iteration of the algorithm is avoided in the computations of probabilistic reachable set for APS.

The algorithmic solution procedure includes a solution of the SDP in any iteration of the algorithm, and the required time to solve even one SDP (see Table 5.1 in Sec. 5.4) makes the procedure only suitable for an offline synthesis of a controller. In addition to the control parameters, the algorithm provides the whole sequence of probabilistic reachable sets, generated during the execution of the algorithm. The sequence of reachable sets is valuable for an a-posteriori detection of critical situations, in which e.g. the probabilistic reachable set is close to the boundary of the feasible state set $\mathbb{X}_k$. If desired, this information can be used to adapt some parameters of the algorithm, e.g. the weights μ_1 and μ_2 in (5.46) to influence the solution of the SDP, and hence the generated control parameters. Although no successful termination of Algorithm 5.1 can be guaranteed, it is a very powerful tool to verify if a controller synthesis with a chosen set of parameters is possible. A result with an unsuccessful termination of the proposed algorithm does not necessarily mean, that no feasible sequence of control inputs exist, since an adaption of the parameters might lead to a different and successful result.

6 Hybrid Controller Synthesis for Switched Affine Probabilistic Systems

In the previous chapter, the introduction of stochastic effects in the state distribution and disturbance, resulting in a consideration of probabilistic reachable sets for controller synthesis, was motivated for APS. The proposed control synthesis procedure computes a time-varying controller, which transfers any initial state x_0 from the initial confidence set X_0^δ into the terminal set $\mathbb{T}$ with confidence δ. The system state of the considered system class is solely continuous valued ($x_k \in \mathbb{R}^n$), but as already pointed out in the Chapter 1, many real world applications also comprise discrete events. Discrete events may have an impact on the continuous dynamics, e.g. the behavior of a power train is mainly determined by the current gear of the transmission, to recall one of the examples mentioned in the introduction of this thesis. The consideration of different discrete modes elevates APS to another system class of *stochastic hybrid systems*. In general, the changes of the discrete mode z_k may be spontaneous/autonomous (switching), or externally induced (switched). In this chapter the switching of the discrete mode z_k is assumed to be externally induced, as is true for the power train example. Concerning the controller synthesis, the introduction of an externally controllable mode, which determines the current continuous dynamics, requires a strategy to choose a feasible sequence of modes.

This chapter addresses the task of controlling discrete-time *switched affine probabilistic systems* (SAPS), which are uncertain with respect to the continuous state initialization and normally distributed disturbances. In addition to providing continuous feedback control laws, the task comprises to select the discrete mode as control input. An important aspect of this chapter is the determination of a feasible sequence of discrete modes. In order to select an appropriate discrete mode in any time step, a type of tree search is adopted, which steers the system evolution towards a given target set.

Literature Review

For piecewise-affine hybrid systems, [75, 76] presented a method for controller synthesis based on problem decomposition: first, the continuous reachability problem is solved on sets of simplices, such that any simplex is left through an exit facet. Secondly, a discrete control problem is solved to obtain a feasible discrete mode path. For the same system class with bounded disturbances, an optimal control problem is addressed by dynamic programming in [90]; this approach is extended to state- and input-dependent disturbances in [141]. The computation of invariant sets is considered in [54, 53] for constrained switched linear systems under dwell-time restriction. Therein, an admissible switching sequence of the discrete mode is derived from invariant sets for switched linear dynamics with no continuous inputs. The quadratic stabilization by a state-dependent switching sequence for switched linear systems with uncertain subsystems of polytopic type is addressed in [166]. Synthesis conditions using bilinear matrix inequalities (BMI) for exponential stabilization are derived in [119, 120], where the latter publication derives not only a state-dependent switching law, but also a state-feedback law for the continuous input. Both approaches consider bounded disturbances. The work in [22] studies a reach-avoid-problem for a class of nonlinear hybrid systems with probabilistic initialization using a type of particle-filter. In general, the computation of reachable sets for stochastic hybrid systems has gained much attention in the past years, either for the purpose of verification (see e.g. work by [134, 1, 43, 55]), or for control design, as by [82, 38, 46, 85, 7].

The main ideas of this chapter were already reported in [24], and extended in [26].

The chapter is organized, such that the considered probabilistic model class is introduced in Section 6.1. It defines a suitable type of stability for the system class, and formally states the control problem. The combination of a tree search procedure and an embedded solution of an SDP is described in Section 6.2, wherein the algorithmic solution procedure to synthesize the hybrid control laws is presented, too. A numerical example and a discussion on tree search procedure is contained in Section 6.3. Section 6.4 completes this chapter.

6.1 Switched Affine Probabilistic Systems

SAPS are able to capture a broad spectrum of engineering processes, where switches can change the dynamics of the plant, e.g. a chemical process in a bio-reactor depends on whether (I) a stirrer is active/deactivated, or (II) a heater is on/off, etc.. The stochastic elements in SAPS stem again from measurement uncertainties. In SAPS, the continuous dynamics is determined by a finite set $Z = \{1, \dots, n_z\}$ of difference equations, and the current active dynamics is specified by the discrete mode $z_k \in Z$. Concerning the controller synthesis, the discrete mode affects the optimization based controller synthesis in many ways, since it introduces an additional degree of freedom. In fact, it is not possible to include the search for a suitable sequence of discrete mode in the SDP formulation as used in the controller synthesis for AS and APS. It is rather required to perform a separate discrete optimization in addition. To this end, a control synthesis procedure for SAPS is proposed, which consists of a tree search for the discrete modes with an embedded solution of an

SDP for the continuous control input. In the following, the formal definition of an SAPS is provided.

6.1.1 System Definition and Probabilistic Reachable Computations

SAPS are a version of the more general SHS (see Def. 3.1), and the continuous dynamics in SAPS are determined by a set of n_z affine difference equations. The feasible input set is a bounded polytope U, and a feasible state set is not explicitly considered, i.e. $\mathbb{X}_k = \mathbb{R}^n$.

A system definition of a switched affine probabilistic system is given in Def. 6.1.

Definition 6.1. *A switched affine probabilistic system (SAPS) is modeled by the following equations and relations ($k \in \mathbb{N}_0$):*

$$x_{k+1} = A_{z_k} x_k + B_{z_k} u_k + G_{z_k} v_k, \tag{6.1a}$$
$$x_0 \sim \mathcal{N}(q_{x,0}, Q_{x,0}), \tag{6.1b}$$
$$v_k \sim \mathcal{N}(q_v, Q_v), \tag{6.1c}$$
$$u_k \in U, \tag{6.1d}$$
$$z_k \in Z := \{1, 2, \ldots, n_z\}, \tag{6.1e}$$

where the initial state x_0 and the disturbance v_k are normally distributed with expected values $q_{x,0} \in \mathbb{R}^n$ and $q_v \in \mathbb{R}^n$, as well as the covariance matrices $Q_{x,0} \in \mathbb{R}^{n \times n}$ and $Q_v \in \mathbb{R}^{n \times n}$, respectively. The input $u_k \in \mathbb{R}^m$ is bounded to a convex polytope $U = P_H(R_u, b_u) \in \mathcal{P}$ with $R_u \in \mathbb{R}^{n_u \times m}$ and $b_u \in \mathbb{R}^{n_u}$. The discrete mode z_k is taken from a finite set Z with n_z possible modes. △

An SAPS is a composition of different continuous dynamics, given by the matrix tuple $(A_{z_k}, B_{z_k}, G_{z_k})$. The subscript indicates the dependency of the discrete mode z_k, which determines the active mode of SAPS at time step k. The tuple $(A_{z_k}, B_{z_k}, G_{z_k})$ is called a *subsystem of an SAPS*. These systems are in general also known as "switched systems", since the switching is externally induced.

An admissible execution of SAPS is as follows:

Definition 6.2. *For a sample of the initial continuous state x_0, the sequence of states $\{x_0, x_1, \ldots, x_k, \ldots\}$ for $k \in \mathbb{N}_0$ is called "admissible", if x_{k+1} is determined for x_k by the following order of operations:*

1. *sample the disturbance $v_k \sim \mathcal{N}(q_v, Q_v)$*
2. *choose a suitable hybrid input $(u_k, z_k) \in U \times Z$*
3. *compute x_{k+1} for the matrix tuple $(A_{z_k}, B_{z_k}, G_{z_k})$ according to 6.1a*

△

As in the previous chapters, the control task for SAPS will comprise a set-to-target control problem, wherein the initial reachable set has to be transferred into a target region $\mathbb{T}$. The center point of this region is denoted by q_T, and for the solvability of the set-to-target control problem, it is assumed that an equilibrium $(\bar{x} = q_T, \bar{u})$ exists for at least one discrete mode $\bar{z}$.

Assumption 6.1. *Let for system* (6.1) *exist at least one hybrid input* $(\bar{u}, \bar{z})^T \in U \times Z$, *for which* (6.1a) *has an equilibrium point* $\bar{x}$, *if the disturbance assumes its expected value* $\bar{v} = q_v$, *i.e.* $\bar{x} = q_T = (I - A_{\bar{z}})^{-1} \cdot (B_{\bar{z}}\bar{u} + G_{\bar{z}}q_v)$.

Equivalently as for APS, the formulation of the reachable set computation is determined by the evolution of the distribution for x_k, specified by the expected value $q_{x,k}$ and covariance matrix $Q_{x,k}$.

The initial state set X_0^δ is given by (5.3) and $Pr(x_0 \in X_0^\delta) = \delta$ holds as well for SAPS.

The evolution of the confidence ellipsoid X_k^δ for SAPS, is very similar to APS in (5.4), but the system matrices in SAPS depend on the current active mode z_k and the update function for $q_{x,k}$ and $Q_{x,k}$ are:

$$q_{x,k} := A_{z_k} q_{x,k-1} + B_{z_k} u_{k-1} + G_{z_k} q_v, \tag{6.2a}$$

$$Q_{x,k} := A_{z_k} Q_{x,k-1} A_{z_k}^T + G_{z_k} Q_v G_{z_k}^T. \tag{6.2b}$$

Here once again, the fact is exploited, that the sum of normal variables has a normal distribution, too, following (3.22). For SPAS, two degrees of freedom exist for the selection of an input signal: (i) subsystem can be chosen by z_k, and (ii) a continuous input u_k is selected to modify the shape and position of the confidence ellipsoid X_k^δ, which is obtained with $q_{x,k}$ and $Q_{x,k}^\delta := Q_{x,k} \cdot c_x$ from:

$$X_k^\delta = \varepsilon(q_{x,k}, Q_{x,k}^\delta). \tag{6.3}$$

6.1.2 Definition of the Set-To-Target Control Problem for SAPS

As well as for APS, the formulation of the set-to-target control problem requires a suitable stability definition, which is for SAPS as follows:

Definition 6.3. *Given a bounded time domain* $\tau = \{0, 1, \dots, N\}$, $N \in \mathbb{N}_0$, *and continuous inputs* $\{u_k\}$ *for* $k \in \tau$, *the SAPS* (6.1) *is called* attractive with confidence δ *on the domain* τ, *if for any initial condition* $x_0 \in X_0^\delta$ *and any* $v_k \in \varepsilon(q_v, Q_v c_x)$, *finite parameters* $\bar{q} \in \mathbb{R}^n$ *and* $\bar{Q} \in \mathbb{R}^{n \times n}$ *exist such that:*

$$||q_{x,N}|| \leq ||\bar{q}||, \quad ||Q_{x,N}|| \leq ||\bar{Q}||. \tag{6.4}$$

The system is said stable with confidence δ *on a bounded time domain* $[0, N]$ *if in addition*

$$||q_{x,k+1}|| < ||q_{x,k}||, \quad ||Q_{x,k+1}|| \leq ||Q_{x,k}||. \tag{6.5}$$

holds for any $0 \leq k \leq N - 1$. △

This choice of stability definition is well motivated, since it implies a convergence of the expected value $q_{x,k}$ to the origin and a decrease of the confidence ellipsoid X_k^δ in size (or at least convergence to a constant size). The convergence in size follows from the convergence of the covariance matrix $Q_{x,k}$.

The control synthesis procedure aims at the computation of a stabilizing hybrid control law $\kappa : \mathbb{R}^n \times \mathbb{N}_0 \to \mathbb{R}^m \times Z$ according to:

$$\begin{pmatrix} u_k \\ z_k \end{pmatrix} = \kappa(x_k, k) := \begin{cases} \phi_c(x_k, k) \\ \phi_d(k) \end{cases}, \tag{6.6}$$

consisting of the continuous part $\phi_c : \mathbb{R}^n \times \mathbb{N}_0 \to \mathbb{R}^m$ and the switching logic $\phi_d : \mathbb{N}_0 \to Z$.

The set-to-target control problem for SAPS can now be stated as follows:

Problem 6.1. *Let a SAPS with disturbances and input constraints according to* (6.1)*, a terminal region $\mathbb{T} = \varepsilon(q_T, Q_T) \subset \mathbb{R}^n$ centered in an equilibrium point, and an initial confidence set of states X_0^δ be given. Find a time-varying hybrid control law:*

$$\kappa(x_k, k) = (\phi_c(x_k, k), \phi_d(k)), \quad x_k \in X_k^\delta, \quad \phi_d(k) \in Z, \tag{6.7}$$

which renders the closed-loop system stable with confidence δ, and for which a finite $N \in \mathbb{N}$ exists with:

$$X_N^\delta \subseteq \mathbb{T}, \tag{6.8}$$

i.e., any initial state $x_0 \in X_0^\delta$ is transferred into the terminal set $\mathbb{T}$ with probability δ within N steps.

□

Assumption 6.1 always allows to find a suitable coordinate transformation, which recasts the Problem 6.1, such that the terminal set $\mathbb{T}$ is centered in the origin.

The objective is to develop an algorithmic method to solve problem 6.1. As in the previous chapters, a time-variant, continuous, affine state feedback controller of the following form is selected:

$$u_k = \phi_c(x_k, k) = -K_k x_k + d_k \in U, \ \forall \ x_k \in X_k^\delta. \tag{6.9}$$

As for APS in (5.11), the set-valued application of the control law on X_k^δ results in an ellipsoid for the feasible input set:

$$\bar{U}_k = \varepsilon(-K_k q_{x,k} + d_k, K_k Q_{x,k}^\delta K_k^T) \in \mathcal{E}. \tag{6.10a}$$

The switching control $\phi_d(k)$ chooses a suitable discrete mode z_k at every time k, in order to obtain a stabilizing sequence of discrete modes. A feasible algorithmic solution of Problem 6.1 is a set of control tuples $(K_k, d_k, \phi_d(k))\ \forall\ k \in \{0, 1, \ldots, N-1\}$ satisfying the conditions of the problem statement for:

$$\bar{U}_k \subseteq U. \tag{6.11}$$

The control law (6.9) leads to the closed-loop dynamics for $q_{x,k}$ and $Q_{x,k}$ with $A_{cl,\phi_d(k)} = (A_{\phi_d(k)} - B_{\phi_d(k)} K_k)$:

$$q_{x,k+1} = A_{cl,\phi_d(k)} q_{x,k} + B_{\phi_d(k)} d_k + G_{\phi_d(k)} q_v, \tag{6.12a}$$
$$Q_{x,k+1} = A_{cl,\phi_d(k)} Q_{x,k} A^T_{cl,\phi_d(k)} + G_{\phi_d(k)} Q_v G^T_{\phi_d(k)}, \tag{6.12b}$$

for $k = \{0, 1, 2, \ldots, N\}$. It is noticeable that the control input $\phi_c(x_k, k)$ affects equation (6.12a) implicitly through the expected value. However, the state feedback gain K_k in (6.12) can influence both, $q_{x,k}$ and the covariance matrix $Q_{x,k}$. Therefore, K_k has to be chosen, such that the expected value for $q_{x,k}$ converges to the origin, and that the size of the confidence ellipsoid X^δ_k decreases over time. With (6.12b), X^δ_{k+1} is obtained from:

$$X^\delta_{k+1} = \varepsilon(q_{x,k+1}, \underbrace{Q_{x,k+1} c_x}_{=:Q^\delta_{x,k+1}}). \tag{6.13}$$

The following section elaborates on the discrete optimization approach and derives the LMI's for the SDP to obtain the sequence of stabilizing hybrid controllers.

6.2 Discrete Optimization with embedded SDP for Controller Synthesis

This section proposes an algorithm to solve problem 6.1 by a combination of a tree search with embedded semi-definite programming. The tree search procedure provides a feasible sequence of discrete modes $\phi_d(k),\ k \in \{0, \ldots, N-1\}$, while the solution of the SDP provides the continuous control parameters (K_k, d_k) for any selected $\phi_d(k)$. Thus, the hybrid control tuple $(K_k, d_k, \phi_d(k))$ is obtained for any $k \in \{0, \ldots, N-1\}$.

The following subsections describe the tree search procedure, first, and thereafter the SDP to be evaluated in any explored node of the tree is derived.

6.2.1 Tree Search for the Discrete Input

The determination of a feasible control sequence $(\phi_d(0), \ldots, \phi_d(N-1))$ of the discrete inputs leads to a discrete optimization problem of selecting $\phi_d(k) \in Z$. Within discrete optimization the field of tree search is quite mature, see [154]. In general, a tree structure Γ consists of a set of nodes:

$$\Gamma := \{\gamma_0, \gamma_1, \dots, \gamma_{n_T}\}, \tag{6.14}$$

where each γ_i is a node in the tree and the number of nodes in a tree Γ is given by n_T with $i = \{1, \dots, n_T\}$. An exemplary tree structure is shown in Figure 6.1. The nodes $\gamma_i \in \Gamma$ can be classified into root nodes, parent nodes, child nodes, and leaf nodes, e.g. the root node in Figure 6.1 has three child nodes, and the parent node for each γ_1, γ_2, and γ_3 is γ_0.

Each node in a tree can be linked to a set of parameters. The linked parameter set for each node γ_i contains the corresponding (i) parent node, (ii) the child nodes, (iii) the current reachable set X_k^δ, (iv) the resulting cost function J_k of the SDP, (v) the chosen mode z_k leading to the current reachable set, and (vi) the current time step k. In order to access these parameters, a set of functions is defined:

$$\begin{aligned}
Pre &: 2^\Gamma \to 2^\Gamma && \text{predecessor node,} && (6.15a)\\
Succ &: 2^\Gamma \to 2^\Gamma && \text{set of successor nodes,} && (6.15b)\\
ReachSet &: 2^\Gamma \to 2^{\mathbb{R}^n} && \text{current reachable set,} && (6.15c)\\
CostVal &: \Gamma \to \mathbb{R} && \text{cost value of the SDP,} && (6.15d)\\
Mode &: \Gamma \to Z && \text{chosen mode } z_{k-1}, && (6.15e)\\
TimeStep &: \Gamma \to \mathbb{N}_0 && \text{time step } k. && (6.15f)
\end{aligned}$$

With these functions each parameter can be accessed, e.g. in Figure 6.1 $Pre(\gamma_1) =$

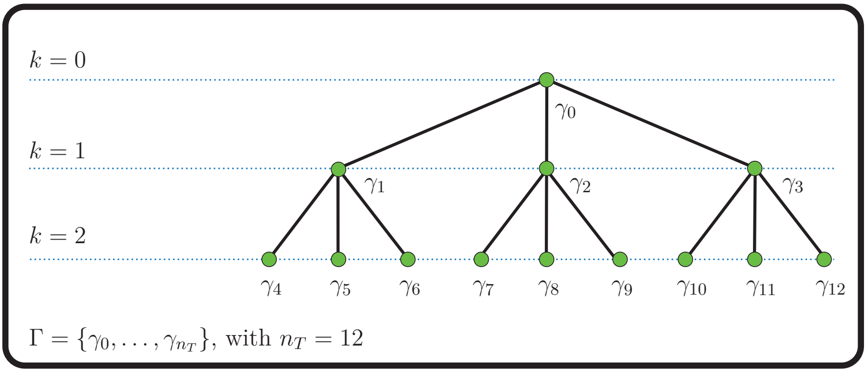


Figure 6.1: General tree structure

γ_0 and $Succ(\gamma_0) = \{\gamma_1, \gamma_2, \gamma_3\}$. For the considered application of the tree Γ, the number of nodes is mainly determined by the number of discrete modes n_z in SAPS, since each node γ_i in the tree can have n_z child nodes (except of the leaf nodes), and time steps k. Therefore, it holds that:

$$n_T \leq \sum_{i=0}^{k} n_z^i - 1. \tag{6.16}$$

Figure 6.1, furthermore, illustrates the relation between the time step k and the different layers of the tree, i.e. the node $\gamma_4, \ldots, \gamma_{12}$ are in the same layer of the tree, and for each node in this layer the current time step k for the evolution of the continuous dynamics (6.1a) is $k = 2$, but each node in this layer is reached by a different choice of discrete modes z_k.

In the considered control problem, the tree structure is used to represent the possible choices for the discrete mode z_k for every $k \in \{0, 1, 2, \ldots, N-1\}$. Each node γ_i in the tree can only be reached by one sequence of discrete modes, which is called a branch of the tree. At the beginning of the control synthesis procedure the tree is unexplored, and the challenge is to find at least one branch in the tree, which specifies a feasible sequence of discrete modes to solve Problem 6.1.

The tree is rooted in γ_0 with the initialization of the control problem, and it is iteratively explored and extended by a tree search procedure.

In order to apply the tree search procedure, three sets of tree nodes are introduced:

$$\mathcal{O}_C \subseteq \Gamma \ : \ \text{set of current nodes, used for exploration,} \tag{6.17a}$$
$$\mathcal{O}_W \subseteq \Gamma \ : \ \text{set of waiting nodes, not yet used for exploration,} \tag{6.17b}$$
$$\mathcal{O}_P \subseteq \Gamma \ : \ \text{set of past nodes, already used for exploration,} \tag{6.17c}$$

and it holds that:

$$\Gamma = \mathcal{O}_W \cup \mathcal{O}_P \cup \mathcal{O}_C. \tag{6.18}$$

The exploration of a node γ_i means to check, which of the options $z_k \in Z$ leads to a feasible solution of the embedded SDP (6.20) – if feasibility is obtained for z_k, a child node γ_{i+1} is included in the waiting node set $\mathcal{O}_W$ for further exploration in a subsequent iteration of the search procedure. The candidates for exploration in each iteration of the search procedure are contained in $\mathcal{O}_C$.

The feasible solutions of the embedded SDP in each node provide a continuous control law $\phi_c(x_k, k)$, satisfying the input constraints, and contributing to drive the system towards the terminal set $\mathbb{T}$. But if no feasible solution is obtained for an explored node, this node γ_{i+1} is not further considered for the exploration of the tree, and is included in the set of past nodes $\mathcal{O}_P$. This is illustrated in Figure 6.2, where the nodes are colored green if a feasible solution exists, and red if no feasible solution for the SDP is found. The blue branch in Figure 6.2 represents a feasible sequence of discrete modes for the first three time steps. In the current iteration of the tree search procedure ($k = 3$) the set of waiting nodes $\mathcal{O}_W$ only includes the nodes γ_{10} and γ_{11}, since these two nodes have not been used for the

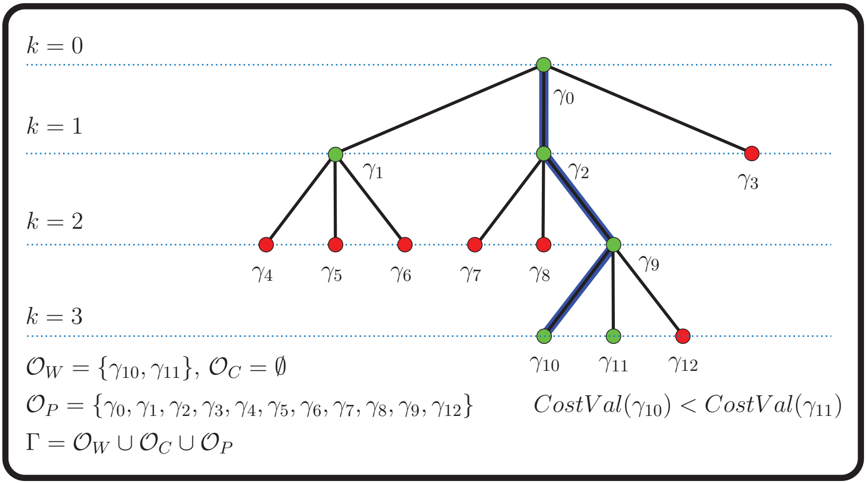


Figure 6.2: Explored tree structure

tree exploration, so far. The remaining nodes of the tree have either been already used for exploration, or no feasible solution of the SDP is available. These nodes are included in the node set of past nodes $\mathcal{O}_P$, since these nodes are not further considered for the tree search procedure.

While several variants exists to establish the order of tree exploration, two basic approaches to search for an optimal path are common. In *depth-first search* (DFS), a branch of the tree is explored as far as a feasible extension of the path is impossible. If the current branch provides no feasible extension, backtracking is applied, and an alternative branch is explored. Typically, DFS is a strategy to obtain a feasible solution fast, but without exploring the complete tree, and therefore, no guarantee for global optimality. Depending on the actual problem, the number of explored nodes often is small relative to all possible nodes in the tree.

In the second approach, namely *breadth-first search* (BFS), the tree is primarily explored in breadth. Given a set of feasible nodes $\mathcal{O}_W$, BFS explores the n_z child nodes for each $\gamma_i \in \mathcal{O}_W$, such that any node leading to a feasible extension of a path is explored. This approach typically leads to high computational time, but if a solution is computed, it is guaranteed to be one with minimal possible number of nodes.

In the example in Figure 6.2 the number of discrete modes is $n_z = 3$, and BFS was applied. The first iteration results in two feasible solutions for $k = 0$ and γ_1 and γ_2 are included in $\mathcal{O}_W$ for further exploration of the tree. For $k = 1$, the SDP has to be solved for overall six nodes, but only for one node a feasible solution of the SDP is obtained, such that $\mathcal{O}_W = \{\gamma_9\}$ in the second iteration of the search procedure. In the last iteration of the example, two feasible solutions are available. The resulting optimal sequence of discrete modes for this example is then $\{\phi_0 = Mode(\gamma_2), \phi_1 = Mode(\gamma_9), \phi_2 = Mode(\gamma_{10})\}$, which is marked as a blue branch in Figure 6.2.

In [23], DFS was used in a similar setting, but only a small share of possible nodes was explored with the aim of only finding a feasible solution. If control performance (or optimality) of the solution is the focus, a better coverage of the search space is desirable. In this case, a combination of BFS and DFS is a reasonable alternative, while the search heuristics, of course, will depend on the specific parametrization of the SAPS and the control problem. To this end, a *Tree Search Heuristic* (TSH) is introduced:

$$TSH \; : \; \mathcal{O}_W \to \mathcal{O}_C \times \mathbb{N}_0. \tag{6.19}$$

The function takes the current set of waiting nodes $\mathcal{O}_W$ as an argument, and returns a set of nodes $\mathcal{O}_C$ for further exploration and the layer of the tree. This does not necessarily mean an exploration in depth with $k+1$, but can also include back stepping with $k-1$, and explore a new branch of the tree.

For different problem instances, a suitable specification of $TSH(\mathcal{O}_W)$ is one in which a few steps of BFS (to cover $\mathbb{X}_k$) and several steps of DFS (to advance each branch in depth and X_k^δ in direction of $\mathbb{T}$) alternate. The relation of the steps for BFS and DFS establish a trade-off between computational tractability and optimality of the solution.

6.2.2 Synthesis of the Continuous Control Law via SDP

A feasible solution of the embedded SDP in the tree search heuristic stabilizes the system, such that the probabilistic reachable ellipsoid X_k^δ is steered towards the terminal set $\mathbb{T}$. The formulation of the SDP for SAPS can be derived from a combination of the constraints used in the SDP for NADS and APS. The LMI's for the convergence of the shape matrix $Q_{x,k}$ in (5.40) and (5.41) can be taken from APS, as well as the LMI's for the input constraint (5.45). Since, the continuous dynamics of SAPS is switched, and is hence time-variant, the formulation for the center point convergence can also be adopted from APS, in which a concept of flexible Lyapunov functions is used in (5.42) and (5.44). The complete SDP to be solved in any explored node of the search tree is then:

$$\min_{S_{k+1},K_k,d_k,\alpha_k} \quad J_k = \text{trace}\left(\begin{bmatrix} S_{k+1} & 0 & 0 \\ 0 & \mu_1\|q_{x,k+1,\gamma_i}\| & 0 \\ 0 & 0 & \mu_2\|u_k\| \end{bmatrix}\right), \tag{6.20a}$$

subject to:

$$q^T_{x,k+1,\gamma_i} M q_{x,k+1,\gamma_i} - \rho q^T_{x,k,Pre(\gamma_i)} M q_{x,k,Pre(\gamma_i)} \leq \alpha_k, \tag{6.20b}$$

$$q_{x,k+1,\gamma_i} = A_{cl,\phi_d(k)} q_{x,k,Pre(\gamma_i)} + B_{\phi_d(k)} d_k + G_{\phi_d(k)} q_v, \tag{6.20c}$$

$$\alpha_k \leq \max_{i \in \{1,\dots,k\}} \omega^i \alpha_{k-i}, \tag{6.20d}$$

$$\begin{bmatrix} S_{k+1} & A_{cl,\phi_d(k)} Q_{x,k,Pre(\gamma_i)} & G_{\phi_d(k)} Q_v \\ Q_{x,k,Pre(\gamma_i)} A^T_{cl,\phi_d(k)} & Q_{x,k,Pre(\gamma_i)} & 0 \\ Q_v G^T_{\phi_d(k)} & 0 & Q_v \end{bmatrix} \geq 0, \tag{6.20e}$$

$$\text{trace}(S_k) \leq \text{trace}(Q_{x,k,Pre(\gamma_i)}), \tag{6.20f}$$

$$\begin{bmatrix} (b_{u,i} - r_{u,i}(d_k - K_k q_{x,k,Pre(\gamma_i)}))I_n & -r_{u,i} K_k (Q^\delta_{x,k,Pre(\gamma_i)})^{1/2} \\ (-r_{u,i} K_k (Q^\delta_{x,k,Pre(\gamma_i)})^{1/2})^T & b_{u,i} - r_{u,i}(d_k - K_k q_{x,k,Pre(\gamma_i)}) \end{bmatrix} \geq 0, \tag{6.20g}$$

$$\forall i = \{1, \dots, n_u\}.$$

The concept of flexible Lyapunov functions is adapted here for SAPS in (6.20b)-(6.20d), where $q_{x,k+1,\gamma_i}$ is computed with the matrices of the switched closed-loop dynamics $(A_{cl,\phi_d(k)}, B_{\phi_d(k)}, G_{\phi_d(k)})$. The matrix M in (6.20b) is symmetric, positive definite, and employed for all possible subsystems $(A_{cl,\phi_0}, B_{\phi_0}, G_{\phi_0})$, $\forall\ \phi_0 \in Z$ [1]. In order to evaluate (6.20b) with two consecutive center points in the tree Γ, an additional subscript is used to indicate the dependency of the current and predecessor node.

By (6.20d), the solution for the current Lyapunov equation is coupled with the past solutions to enforce convergence for the center point $q_{x,k}$ (see Chapter 4 for a detailed explanation). The convergence of the shape matrix $Q^\delta_{x,k}$, or equivalently of the covariance matrix $Q_{x,k}$, is enforced by (6.20e) and (6.20f). Both constraints correspond to the LMI for APS ((5.40),(5.41)), since it is exactly the same requirement for the size and shape of the reachable ellipsoid X^δ_k. Only the matrix for the closed loop dynamics in (5.40) is adapted to $A_{cl,\phi_d(k)}$ in (6.20e). The last LMI enforces the satisfaction of the input constraint, and is basically an ellipse-in-polytope problem, which is stated in Proposition 4.2.

The tree search procedure with the embedded solution of the SDP is formulated algorithmically in Algorithm 6.1. The first step of the algorithm is the "tree exploration step", in which the search heuristic determines a set of nodes $\mathcal{O}_C$ for

[1] This is in general referred to as a common Lyapunov function. Another possibility is to employ a Lyapunov function for each subsystem with n_z different matrices M_{z_k}.

exploration. The SDP is solved for each node $\gamma_i \in \mathcal{O}_C$, and if a feasible solution exists, the node is included in the node set $\mathcal{O}_W$, otherwise in $\mathcal{O}_P$. The second removes the predecessor nodes of all nodes in $\mathcal{O}_C$ from the current set of waiting nodes $\mathcal{O}_W$, and the removed nodes are considered in the set of past nodes $\mathcal{O}_P$. The third step verifies, if a criterion for termination is satisfied.

The algorithm terminates successfully, if a confidence ellipsoid $X_k^\delta \in \mathcal{E}$ is contained in the target set $\mathbb{T}$ for any node $\gamma_i \in \Gamma$, meaning that Problem 6.1 is solved. The algorithm is formulated to terminate with the first feasible solution of Problem 6.1, but no global optimality is guaranteed. To ensure termination, also if $\mathbb{T}$ is not reached in a reasonable number of iterations, the additional criterion:

$$\pi_{k+1} = \min_{\gamma_i \in \Gamma} ||q_{x,k+1,\gamma_i} - q_{x,k,Pre(\gamma_i)}|| \leq \pi_{min}, \tag{6.21}$$

is used with a parameter $\pi_{min} \in \mathbb{R}$. π_k evaluates the difference in the $2-$norm of two consecutive expected values $q_{x,k+1}$ and $q_{x,k}$. Since the explored decision tree allows different branches, it has to be ensured, that only two consecutive nodes in a branch are considered for the parameter. To this end, the norm of expected value of node γ_i for $k+1$ is compared to the expected value of the parent node $Pre(\gamma_i)$ at time k. If π_k is lower than a user-defined threshold, no significant change of the expected value can be detected, and hence the algorithm stops without success.

Algorithm 6.1. ***Probabilistic Control Algorithm for Switched Affine Probabilistic Systems (SAPS)***

Given: (6.1) *with* $x_0 \sim \mathcal{N}(q_{x,0}, Q_{x,0})$, $v_k \sim \mathcal{N}(q_v, Q_v)$, *and* $U \in \mathcal{P}$;
$\mathbb{T}$, *TSH*$(\cdot)$, δ, π_{min}, $M \in R^{n\times n}$, $M = M^T \geq 0$, $\rho \in (0,1]$, $\alpha_0 \in [0,\,1]$, *and* $\omega \in [0,1)$
Define: $\pi_0 := \pi_{min}$, $\mathcal{O}_W := \{\gamma_0\}$, $\mathcal{O}_P = \emptyset$, $X_0^\delta = \varepsilon(q_{x,0}, Q_{x,0}^\delta)$
while $\mathcal{O}_W \neq \emptyset$ *and* $\pi_k \geq \pi_{min}$ ***do***

1. *apply* $[\mathcal{O}_C, k] :=$*TSH*$(\mathcal{O}_W)$
   for *any* $\gamma_i \in \mathcal{O}_C$ ***do***
   - *solve the optimization problem* (6.20) *for* γ_i
     if *feasible solution available* ***do***
     - *compute the distribution of* $x_{k+1} \sim \mathcal{N}(q_{x,k+1}, Q_{x,k+1})$ *with the control tuple* $(K_k, d_k, Mode(\gamma_i))$ *according to* (6.12)
     - *compute the confidence ellipsoid* X_{k+1}^δ *by* (6.3)
     - *compute* π_{k+1} *according to* (6.21)
     - *compute* $\gamma_i = \Big(Pre(\gamma_i), \emptyset, X_{k+1}^\delta, J_k, Mode(\gamma_i), k+1\Big)$
     - $\mathcal{O}_W := \mathcal{O}_W \cup \{\gamma_i\}$

     else
     - $\mathcal{O}_P := \mathcal{O}_P \cup \{\gamma_i\}$

     end

   end
2. $\mathcal{O}_W := \mathcal{O}_W \setminus (Pre(\mathcal{O}_C))$, $\mathcal{O}_P := \mathcal{O}_P \cup (Pre(\mathcal{O}_C))$
3. ***if*** $X_k^\delta \subseteq \mathbb{T}$ *for any* $\gamma_i \in \mathcal{Q}_W$ ***or*** (6.21) *holds* ***do*** *terminate* ***end***

end while

Let k_π denote the iteration, in which the criterion holds first (if existing). If this applies (indicating termination without success), different countermeasures can be applied: If q_{x,k_π} and Q_{x,k_π} indicate a small distance to satisfying $X_k^\delta \subseteq \mathbb{T}$, a reduction of π_{min} can be reasonable. Another possibility is to reduce the confidence level δ, which results in smaller ellipsoids, and thus increases the probability to terminate successfully. The algorithm terminates also, if no feasible solution is found in each node of the explored tree, such that $\Gamma = \mathcal{O}_P$, and $\mathcal{O}_W = \mathcal{O}_C = \emptyset$.

Lemma 6.1. *Problem 6.1 is successfully solved, if Algorithm 6.1 terminates with* $X_N^\delta \subseteq \mathbb{T}$, $N \leq k_\pi$. *The solution provides a hybrid control law* (6.9)*, which steers any initial state* $x_0 \in X_0^\delta$ *with probability* δ *into the target set* $\mathbb{T}$ *in* N *steps. The controlled dynamics with disturbances* $v_k \sim \mathcal{N}(q_v, Q_v)$ *is attractive with confidence* δ

according to Definition 6.3, and stable with confidence δ *if* $\alpha_k = 0 \;\; \forall k \in \{0, \dots, N-1\}$ *holds. Furthermore, the input constraint* $u_k \in U$ *holds for all* $0 < k < N$.

Proof From Lemma 3.2, it holds that $Pr(x_k \in X_k^\delta) = \delta$, with $X_k^\delta = \varepsilon(q_k, Q_k c_x)$, and following (6.12), the state distribution for $k+1$ is $x_{k+1} \sim \mathcal{N}(q_{x,k+1}, Q_{x,k+1})$. By computing $X_{k+1}^\delta = \varepsilon(q_{k+1}, Q_{k+1} c_x)$ with the scaling factor c_x according to (3.31), the Algorithm 6.1 ensures that $Pr(x_{k+1} \in X_{k+1}^\delta) = \delta$. Hence, it is guaranteed that $Pr(x_k \in X_k^\delta) = Pr(x_{k+1} \in X_{k+1}^\delta) = \delta$ for all $k \in \{0, 1, \dots, N-1\}$, such that $Pr(x_N \in X_N^\delta) = \delta$ follows from induction, along the feasible branch of the tree Γ.

A successful termination of Algorithm 6.1 provides a feasible sequence of discrete modes $\phi_d(k)$, and implies $X_N^\delta \subseteq \mathbb{T}$, i.e. $x_N \in X_N^\delta \subseteq \mathbb{T}$ holds for all initial states $x_0 \in X_0^\delta$ and all disturbances v_k.

Furthermore, a successful termination implies the existence of a feasible solution of the optimization problem (6.20) in any time step k. This further implies that the input constraint $u_k \in \bar{U}_k \subseteq U$ holds according to Proposition 5.1.

Attractiveness of the mean vector $q_{x,k}$ towards 0 in the sense of Def. 6.3 follows from the fact, that $X_N^\delta = \varepsilon(q_{x,N}, Q_{x,N}^\delta) \subseteq \mathbb{T}$ implies that $||\bar{q}|| \geq ||q_{x,N}||$ exists, as well as $||Q_T|| \geq ||Q_{x,N}^\delta||$ with $Q_T = \bar{Q}$. For $\alpha_k = 0 \;\; \forall k \in \{0, \dots, N-1\}$, stability with confidence δ according to (6.5) follows, since: (i) (4.67) implies with $\rho \in [0,1)$ that $||q_{x,k+1,\gamma_i}|| < ||q_{x,k,Pre(\gamma_i)}||$ is obtained from $V(q_{x,k+1,\gamma_i}) < V(q_{x,k,Pre(\gamma_i)})$ and (4.14); (ii) $||Q_{x,k+1,\gamma_i}|| \leq ||Q_{x,k,Pre(\gamma_i)}||$ is enforced by the constraint $\text{trace}(S_{k+1}) \leq \text{trace}(Q_{x,k})$ of the SDP.

The upper bounds for the norm of the expected value $q_{x,N}$ and the covariance matrix $Q_{x,N}$ are given by parametrization of the ellipsoidal terminal set $\mathbb{T}$. □

6.3 Exemplary Application of Tree-Search-Heuristics

To illustrate the principle of the proposed algorithm with consideration of the tree search procedure, the synthesis is applied to a arbitrary chosen example. First, the specification of an SAPS is given, and thereafter, the simulation results are used to exemplify the effect of different parametrizations of the tree search heuristic.

6.3.1 Generic System Model of an SAPS

The control procedure is applied to an example with four discrete modes $Z = \{1, 2, 3, 4\}$, three continuous states $x_k \in \mathbb{R}^3$, and two continuous inputs $u_k \in \mathbb{R}^2$. The system matrices $(A_{z_k}, B_{z_k}, G_{z_k})$ stem from a discretization of a continuous-time system for each $z_k \in Z$. The initial distribution of the system state and the disturbances are given by:

$$x_0 \sim \mathcal{N}(q_{x,0}, Q_{x,0}) \text{ with } q_{x,0} = \begin{bmatrix} 15 \\ 10 \\ -10 \end{bmatrix}, \quad Q_{x,0} = I_3, \tag{6.22}$$

$$v_k \sim \mathcal{N}(q_v, Q_v) \text{ with } q_v = \begin{bmatrix} 0 \\ 0.3 \\ 0 \end{bmatrix}, \quad Q_v = 1e-1 \begin{bmatrix} 1 & 0.5 & 0 \\ 0.5 & 1.5 & 0.5 \\ 0 & 0.5 & 1 \end{bmatrix}. \tag{6.23}$$

The continuous dynamics is specified by the following system matrices:

$$\begin{aligned} A_1 &= \begin{bmatrix} 0.78 & 0 & 0.71 \\ 0.34 & 0.61 & 0.15 \\ 0 & 0 & 1.13 \end{bmatrix}, \quad A_2 = \begin{bmatrix} 1.28 & 0.12 & 0 \\ 0 & 0.72 & 0 \\ 0.05 & 0.20 & 0.88 \end{bmatrix}, \\ A_3 &= \begin{bmatrix} 0.93 & 0.13 & 0.71 \\ 0 & 1.08 & 0.49 \\ 0 & 0 & 0.88 \end{bmatrix}, \quad A_4 = \begin{bmatrix} 0.56 & 0 & 0 \\ 0.37 & 0.93 & 0 \\ 0.66 & 0.26 & 1.13 \end{bmatrix}, \end{aligned} \tag{6.24}$$

$$B_1 = \begin{bmatrix} 0 & 0.09 \\ 0.20 & 0.01 \\ 0 & 0.27 \end{bmatrix}, B_2 = \begin{bmatrix} 0.28 & 0.02 \\ 0 & 0.21 \\ 0.01 & 0.03 \end{bmatrix}, B_3 = \begin{bmatrix} 0.25 & 0.09 \\ 0 & 0.06 \\ 0 & 0.24 \end{bmatrix}, B_4 = \begin{bmatrix} 0.19 & 0 \\ 0.29 & 0.25 \\ 0.12 & 0.3 \end{bmatrix}, \tag{6.25}$$

$$G_1 = G_2 = G_3 = G_4 = I_3. \tag{6.26}$$

Note, that all four subsystems are chosen to have unstable state matrices to have a challenging example. A feasible matrix for the initial Lyapunov function is given by:

$$M = \begin{bmatrix} 1.0005 & 0.0111 & 0.0148 \\ 0.0111 & 1.2589 & 0.3440 \\ 0.0148 & 0.3440 & 1.4571 \end{bmatrix}. \tag{6.27}$$

The input u_k is constrained to a hyper-box:

$$u_k \in U = P_H(R_u, b_u), \tag{6.28}$$

with

$$R_u = \begin{bmatrix} 1 & 0 \\ -1 & 0 \\ 0 & 1 \\ 0 & -1 \end{bmatrix}, \; b_u = \begin{bmatrix} 5 \\ 5 \\ 5 \\ 5 \end{bmatrix}. \tag{6.29}$$

The target set is specified as:

$$\mathbb{T} = \varepsilon \left(\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 3.6 & 2.4 & 0 \\ 2.4 & 4.5 & 2.4 \\ 0 & 2.4 & 4 \end{bmatrix} \right). \tag{6.30}$$

The remaining parameters of the Algorithm 6.1 are chosen to: $\delta = 0.95$, $\pi_{min} = 10^{-4}$, $\alpha_0 = 10^{-4}$, $\omega = 0.8$ and $\rho = 0.98$. The cost function evaluates the approximated covariance matrix by its trace and the norm of the expected value: $J = \text{trace}\left(\begin{bmatrix} S_{k+1} & 0 \\ 0 & 0.8\|q_{k+1}\| \end{bmatrix}\right)$, $\mu_2 = 0$.

The computation is performed on a standard 4.2-Ghz-Quad-Core CPU with 16 GB of RAM. The SDP problems are solved by using the commercial solver MOSEK, which is implemented in Yalmip. The MISDP is solved by a standard branch-and-bound algorithm (Yalmip built-in), and the SDP in each node of the MISDP is solved by MOSEK.

6.3.2 Discussion of the Impact of the Tree Search Heuristic

The control problem for the stated system dynamics is solved for different variants of the function $TSH(\mathcal{O}_W)$, and the different results are discussed now.

In fact, the chosen heuristic function for the tree search is based on a combination of BFS and DFS, where each strategy is alternating applied for a predefined number of iterations. Table 6.1 summerizes the simulation result for some exemplary specifications: The second and third column of the table specifies the used number of

Simulation	steps BFS	steps DFS	Comp time total	Comp time avg.	tree nodes $\lvert\Gamma\rvert$	time steps k	cum. cost $\sum_{k=0}^{N} J_k$
1	0	2	46.2 s	0.59 s	81	16	184.61
2	2	2	57.3 s	0.6 s	97	15	171.48
3	2	0	7.7 h	0.62 s	44493	13	148.33
4	3	1	101.2 s	0.57 s	177	15	168.24

Table 6.1: Comparison of the simulation results for different specifications of $TSH(\Gamma_k)$.

steps for DFS and BFS, e.g. the first simulation is an example for a heuristic, where only DFS is used as a strategy. The fourth and fifth column describe the required computational time to solve the each SDP in any node of the tree, and the number of tree nodes is shown in column six. The seventh column indicates the required number of time steps to transfer the initial reachable set X_k^δ into the terminal set $\mathbb{T}$ with the resulting hybrid control law. The last column shows the cumulated costs for $k = \{0, \ldots, N\}$. The variants for the specification of the heuristic are chosen, such that a pure DFS (Simulation 1) and a pure BFS (Simulation 3) can be compared, as well as a balanced specification (Simulation 2). The last simulation shows a specification with a slightly shifted focus on BFS.

A striking entry of Table 6.1 is the number of tree nodes in Simulation 3, which shows the required computational effort to find a feasible solution with a pure BFS strategy. The number of nodes clearly leads to a very high computational time, but the spend time and effort results in the lowest costs (148.33) and a minimum number of time steps (13). The maximum number of nodes for a whole tree is given by:

$$|\Gamma| \leq \sum_{k=0}^{N} n_z^k. \tag{6.31}$$

That is, the maximum number of the tree nodes for Simulation 3 is 89478484, and although the number of explored nodes during the execution of the algorithm is enormous, only 0.05% of the tree is explored.

Furthermore, the table illustrates, that the required computational time for a successful termination of the algorithm scales with the explored nodes, since the SDP has to be solved in every node of the tree. Another impact on the computational time and the number of nodes stems from the specification of steps for BFS and DFS. The results in Table 6.1 show, that an increase of steps for BFS leads to an increase of tree nodes, which is comprehensable, since the tree is explored in breadth, and hence all available nodes with a feasible solution for the SDP are further explored. In contrast, the simulations with a focus on DFS generate a tree with less tree nodes, since this strategy uses only one node of $\mathcal{O}_W$ for a further exploration of the tree. The remaining nodes in $\mathcal{O}_W$ can have a feasible solution, but are not further considered, and are therefore leaf nodes. The last two columns show, that it is worthwhile to consider a reasonable number of steps for BFS, since the value for the cumulative cost decreases with the steps for BFS. But in many cases, a termination of the algorithm is needed in less then 7.7 h, as for the pure BFS solution in the shown example, and therefore, a combination with DFS can provide a solution in less time. A simulation with a balanced result concerning the computational time and cumulative cost is provided by the Simulation 2. Even if the cumulative cost is 15% above the best solution in Simulation 3, it takes only a fraction of the computational time. Simulation 4 illustrates, that even though an increase of steps for BFS leads to an improved result for the cumulative cost, it is still 13% above the best value, and hence only 2% better than Simulation 2. The improvement of 2% compared to Simulation 2 is achieved at the cost of an approximately doubled computational time. In summary, Table 6.1 shows, that the combination of different steps for BFS and DFS for the tree search heuristic is a powerful tool to find a suitable solution with either a balanced result concerning the achieved cost and computational time, or a result with a focus on optimality or time.

The resulting evolution of the probabilistic reachable sets for Simulation 2 is shown in Fig. 6.3, where the color of each reachable ellipsoid indicates the chosen discrete mode z_k, with red for $z_k = 1$, green for $z_k = 2$, blue for $z_k = 3$, and magenta for $z_k = 4$. The terminal set $\mathbb{T}$ is shown as a black colored ellipsoid. The simulation shows the reduction of reachable ellipsoid in each time step, and Algorithm 6.1

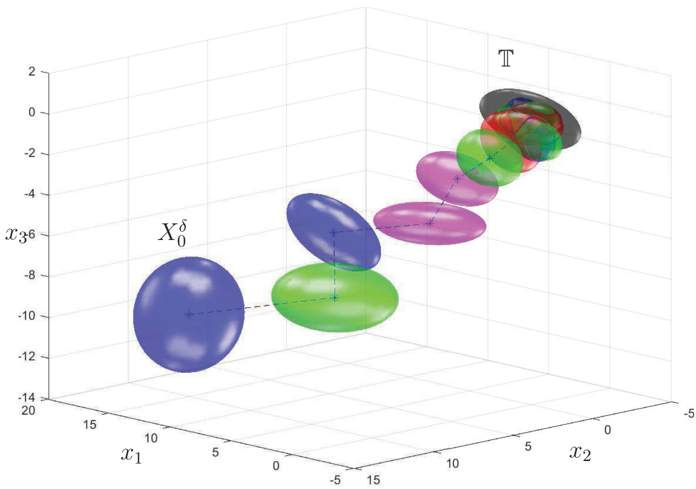


Figure 6.3: Illustration of Simulation 2.

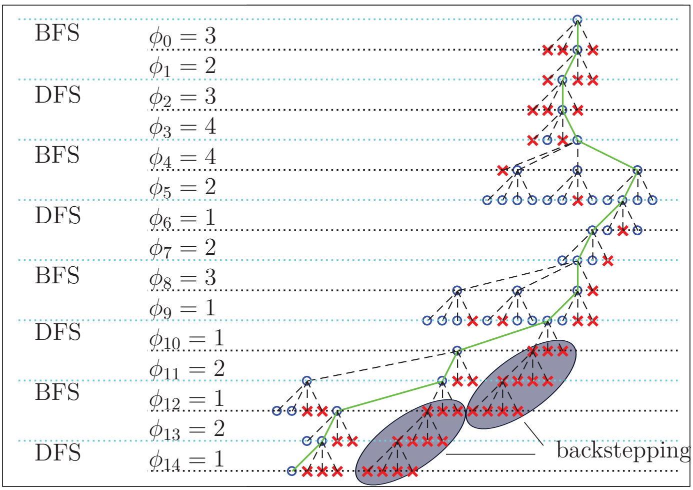


Figure 6.4: Resulting tree for Simulation 2

terminates succesfully with $X_N^\delta \subseteq \mathbb{T}$. In addition to the evolution of the reachable ellipsoids Fig. 6.4 shows the search tree explored during the execution of Algorithm 6.1 in Simulation 2. The figure shows on the left side the used search strategy and the chosen discrete mode $\phi_d(k)$. The explored tree is shown on the right of the figure, where the blue dots and red crosses indicate a feasible and infeasible solution of the SDP. The grey colored ellipsoids highlight the branches of the tree in which backstepping is required. Initially the two branches were chosen by the tree search heuristic, but later on it turned out to be a dead end, since each of the branch led to the case with an infeasible solution of the SDP for each discrete mode. In this case, the search heuristic applies backstepping and the corresponding nodes are marked as nodes with infeasible solutions. Eventually, the search heuristic found a feasible sequence of discrete modes, marked as a green line in the figure.

6.4 Discussion

This chapter elevates the controller synthesis for APS based on reachable set computation introduced in Chapter 5 to a new system class, namely switched affine probabilistic systems. While the considered system class in Chapter 5 already includes probabilistic effects, the system class of SAPS introduces switched dynamics in the sense of a choosable discrete mode z_k. The discrete mode can be externally specified at each time step k, and it selects the current active subsystem $(A_{z_k}, B_{z_k}, G_{z_k})$ for the continuous state x_k. An SAPS also incorporates the probabilistic effects as in APS, such that the principles for the probabilistic reachable set computation hold for SAPS as well. The proposed controller synthesis in this chapter provides not only a feasible control law for the continuous valued controller input u_k, but also a feasible sequence of discrete modes z_k.

The combination of the externally chosen discrete mode and continuous input motivates the introduction of a hybrid input (u_k, z_k), which determines the evolution of the system state x_k at time step k. The discrete mode is an additional degree of freedom in the general control problem, and the search for a feasible sequence of discrete modes can be described as a combinatorial problem. Which subsystem should be active at time step k to transfer the initial probabilistic reachable set X_k^δ into the terminal set $\mathbb{T}$?

This chapter presents an algorithmic procedure with a combination of a tree search for the discrete mode and an embedded solution of SDP's for the continuous input. While the formulation of the SDP's is very similar to the ones in Chapter 4 and 5, the discrete optimization is new in the current chapter. The field of discrete optimization is well-researched (see [163]), and this chapter proposes a general heuristic function to solve the combinatorial problem. The problem consists of finding a feasible branch in a decision tree, and each branch in the tree represents a sequence of discrete modes. Initially the tree is unexplored and only the root node is given. Based on a specification of the search heuristic, a set of nodes is considered for further exploration of the tree, and the SDP is solved in every explored node. The objective of the search heuristic is twofold: first, a solution must be found within a minimum of explored nodes, and hence within an minimum of time. Second, the solution must provide an optimal solution concerning the cumulative cost of the SDP. As discussed in Chapter 6.3.2, a combination of the well-known BFS and DFS strategies is able to provide a solution, which satisfies both objectives. The combination consists of an alternating search strategy between DFS and BFS for several number of time steps k. If computational time is of main interest, the search heuristic can be formulated, such that the DFS strategy is more dominant, and if optimality of the solution is of interest BFS should be more dominant. The discussion in Chapter 6.3.2 shows that the required computational time and the optimality do not scale linearly, i.e. the increase of explored nodes does not lead to the same increase of optimality. In fact, the numerical examples shows that a balanced specification of time steps for BFS and DFS provides a very balanced result concerning the computational time and optimality. But the provided solutions of the tree search heuristics are not guaranteed to be globally optimal. A globally optimal solution can only be provided with a fully explored tree, but in most cases the number of nodes becomes to large and the computation is impractical.

The optimization problem for the continuous input u_k is very similar to the SDP's in Chapter 4 and 5, but nevertheless, a few minor adaptions due to the discrete mode are necessary. As for NADS the concept of flexible Lyapunov function is applied for the convergence of the expected value $q_{x,k}$. But whereas for NADS the matrix M should satisfy the Lyapunov condition only for the initial dynamics (A_0, B_0), the Lyapunov condition in SAPS has to be satisfied for each subsystem $(A_{z_k}, B_{z_k}, G_{z_k})$, $z_k \in Z$. This ensures a convergence of the expected value $q_{x,k}$ by the use of the Lyapunov function, no matter which subsystem is currently active. The remaining LMI's of the SDP are directly adopted from the SDP for APS to ensure a convergence of the ellipsoid size and a satisfaction of the input constraint.

The following chapter is also concerned with hybrid dynamics, but the change of the discrete mode z_k is not externally induced, as in SAPS. Instead, the discrete mode changes autonomously based on the current value of the continuous state x_k. The continuous state space is divided into several sub-spaces and the value of the discrete mode is determined by the sub-space containing the current state x_k.

7 Control of Piece-Wise Affine Probabilistic Systems

The previous chapter considered controller synthesis for hybrid dynamics via reachable set computations. The current chapter addresses the task of controlling discrete-time piecewise affine probabilistic systems (PWAPS), which model a partition of the state space and specific affine dynamics valid in each region. While the change of the discrete mode for SAPS is externally induced, the change in PWAPS is triggered autonomously, based on the partition of the state space. The consideration of the autonomous change of the discrete mode is the main challenge of this chapter.

Probabilistic uncertainties with respect to the initial state and additive disturbances are considered in terms of normal distribution, as before. In general, piecewise affine (PWA) systems are convenient mathematical models for many practical applications, since discontinuities arising from saturation constraints, hysteresis, or friction can be encoded. Furthermore, PWA systems enable to encode linearizations of originally nonlinear dynamics for a finite number of state space regions [147]. The focus of this contribution is to provide stabilizing time-variant state feedback control laws for set-to-set transitions of the system state, while ensuring a given probability level.

PWAPS are a special class of stochastic hybrid systems, and the question of whether a target set is reached from an initial set is undecidable in the general case.

The main ideas of this chapter were already reported in [25].

The chapter first introduces the class of systems and control problem (Sec. 7.1), and Sec. 7.2 specifies the control law, and covers the main challenges of reachability computations for PWAPS. The optimization-based solution procedure is proposed in Sec. 7.2.2. Numerical results for an example are provided in Sec. 7.3, before Sec. 7.4 concludes the paper.

7.1 Piece-Wise Affine Probabilistic Systems

Piece-wise affine probabilistic systems (PWAPS) are very similar to SAPS, but in contrast to SAPS the switching is not induced by a discrete input variable, but instead given by the partitioning of the state space. The switching is a part of the hybrid dynamics, and a huge variety of physical processes in which autonomous switching occurs can be modeled by PWAPS. A motivation for PWAPS stems from the possibility to encode linearizations of originally nonlinear dynamics for a finite number of state regions, as suggested in [147]. A linearization of the nonlinear dynamics in each state region yields a PWAPS, and the proposed procedure in this contribution provides a control law to reach a target set, while taking into account stochastic effects, like uncertain initialization and disturbances.

7.1.1 System Definition and Probabilistic Reachable Computations

PWAPS are a version of the more general SHS (see Def. 3.1), and the feasible state set $\mathbb{X}_k$ for the continuous state x_k is partitioned into n_z polytopic regions. The continuous dynamics in each region is specified by a different affine difference equation, and the discrete mode z_k is determined by the current region containing the continuous state x_k. Hence, the discrete mode is not externally induced, and is not available as an additional input. The feasible input set for the continuous input u_k is a bounded polytope U.

The formal system definition of a PWAPS is as follows:

Definition 7.1. *A piece-wise affine probabilistic system (PWAPS) is described by the following equations:*

$$x_{k+1} = A_{z_k} x_k + B_{z_k} u_k + G_{z_k} v_k, \tag{7.1a}$$
$$x_0 \sim \mathcal{N}(q_{x,0}, Q_{x,0}), \tag{7.1b}$$
$$v_k \sim \mathcal{N}(q_v, Q_v), \tag{7.1c}$$
$$u_k \in U, \tag{7.1d}$$
$$z_k \in Z = \{1, 2, \dots, n_z\}, \tag{7.1e}$$
$$\mathbb{X} = \bigcup_{i=1}^{n_z} \Theta^{(i)}, \tag{7.1f}$$
$$\bar{\Theta} = \{\Theta^{(1)}, \dots, \Theta^{(n_z)}\}, \tag{7.1g}$$
$$getAdjPart \ : \ \mathbb{R}^n \to 2^Z, \tag{7.1h}$$

where the initial continuous state x_0 and the disturbance v_k are normal distributed with expected value $q_{x,0} \in \mathbb{R}^n$, $q_v \in \mathbb{R}^n$ and covariance matrix $Q_{x,0} \in \mathbb{R}^{n \times n}$, and $Q_v \in \mathbb{R}^{n \times n}$, respectively. PWAPS posses a hybrid state, which consists of the continuous state x_k, and the discrete mode z_k. The continuous state space is partitioned into a set $\bar{\Theta}$ of polytopic subsets $\Theta^{(i)}$, $i = \{1, \dots, n_z\}$, and the current discrete mode z_k is determined by the partition element $\Theta^{(i)}$, which contains x_k at time step k. The continuous input in constraint to a convex polytope $U = P_H(R_u, b_u) \in \mathcal{P}$ with $R_u \in \mathbb{R}^{n_u \times m}$ and $b_u \in \mathbb{R}^{n_u}$. The function $getAdjPart(x_k)$ defines the current active discrete mode z_k. △

The polytopic subsets $\Theta^{(i)} \in \mathcal{P}$ are defined by:

$$\Theta^{(i)} = P_H(R_x^{(i)}, b_x^{(i)}), \tag{7.2}$$

with $R_x^{(i)} \in \mathbb{R}^{n_{x,i} \times n}$ and $b_x^{(i)} \in \mathbb{R}^{n_{x,i}}$, where $n_{x,i}$ is the number of half-planes defining the polytope $\Theta^{(i)}$. For a partition, it holds that:

$$\mathbb{X} = \bigcup_{i=1}^{n_z} \Theta^{(i)}, \tag{7.3}$$

with:

$$int\left(\Theta^{(i)}\right) \bigcap int\left(\Theta^{(j)}\right) = \emptyset, \text{ for any pair } (i,j) \in Z \times Z, i \neq j. \tag{7.4}$$

For $\Theta^{(i)} \in \bar{\Theta}$, the union of all partitions yields the feasible state space $\mathbb{X}$, and the interior of the partitions do not overlap. To any partition element $\Theta^{(i)}$, a different dynamics for the continuous state is assigned by the evaluation of the function *getAdjPart*. The resulting discrete mode z_k encodes the current partition element $\Theta^{(z_k)}$, and thus determines the active continuous dynamics $(A_{z_k}, B_{z_k}, G_{z_k})$ valid in time step k.

If $x_k \in int(\Theta^{(i)})$, the result of (7.1h) is the current partition $z_k = i$, while for $x_k \in \partial(\Theta^{(i)})$, the result is the subset of Z, which determines all adjacent partitions, i.e. those, which share the boundary $\partial(\Theta^{(i)})$ in x_k. A feasible evolution of PWAPS, which solves the multiple assignment of z_k, if $x_k \in \partial(\Theta^{(i)})$ is as follows:

Definition 7.2. *Given an initial continuous state $x_0 \in \Theta^{(i)}$, the initial discrete state $z_0 = i$ is determined by the partition, which contains x_0. A sequence of pairs (x_k, z_k), $k \in \{0, 1, \ldots\}$ is called "'admissible"', if for every $k \in \mathbb{N}_0$, x_{k+1} and z_{k+1} are determined by the following order of computations:*

1. *sample the disturbance $v_k \sim \mathcal{N}(q_v, Q_v)$*
2. *choose a suitable input $u_k \in U$*
3. *compute x_{k+1} with the tuple $(A_{z_k}, B_{z_k}, G_{z_k})$ according to (7.1a)*
4. *compute z_{k+1}:*
   if $x_{k+1} \in int(\Theta^{(i)})$, $i \in Z$ ***do*** $z_{k+1} = i$
   elseif $x_{k+1} \in \partial(\Theta^{(i)})$, $i \in Z$ ***do***
     $Z_{adj,k+1} := getAdjPart(x_{k+1})$
     if $z_k \in Z_{adj,k+1}$ ***do*** $z_{k+1} := z_k$
     else $z_{k+1} = \min_{z \in Z_{adj,k+1}} z$
     end
   end

△

Step 4 in Definition 7.2 is a semantic convention, if x_k is on any boundary. The set $Z_{adj,k+1}$ contains the information about all adjacent partitions for the next time step, and if the current state z_k is contained in $Z_{adj,k+1}$, it will be chosen also for the next time step. If z_k is not contained in $Z_{adj,k+1}$, the semantic convention defines the minimum value in the set as the consecutive discrete mode z_{k+1}, e.g. if $Z_{adj,k+1} = \{2, 3\}$, then $z_{k+1} = 2$.

As in the previous chapters, the control task for PWAPS will comprise a set-to-target control problem, wherein the initial reachable set has to be transferred into a target region $\mathbb{T}$ with a given confidence. The center point of this region is denoted by q_T, and for the solvability of the set-to-target control problem, it is assumed that an equilibrium $(\bar{x} = q_T, \bar{u})$ exists for at least one discrete mode $\bar{z}$ (compare with Assumption 7.1).

Assumption 7.1. *Let for system* (7.1) *exist at least one input $\bar{u} \in U$, for which* (7.1a) *has an equilibrium point $\bar{x}$ in at least one partition element $\Theta^{\bar{z}}$, if the disturbance assumes its expected value $\bar{v} = q_v$, i.e. $\bar{x} = q_T = (I - A_{\bar{z}})^{-1} \cdot (B_{\bar{z}}\bar{u} + G_{\bar{z}}q_v)$.*

If the equilibrium point is on a border between two or more partition elements, Assumption 7.1 states, that it is an equilibrium point for the affine dynamics in each adjacent partition element.

Equivalently as for SAPS and APS, the formulation of the reachable set computation is determined by the evolution of the distribution for x_k, specified by the expected value $q_{x,k}$ and covariance matrix $Q_{x,k}$.

For the initialization of the random variable x_0, an ellipsoid is specified as:

$$X_0^\delta := \varepsilon(q_{x,0}, Q_{x,0} \cdot c_x). \tag{7.5}$$

The evolution of the confidence ellipsoid X_k^δ for PWAPS is exactly the same as for APS, shown in (6.2). The reachable set with confidence δ for any $k \in \{0, \dots, N\}$ is given by:

$$X_k^\delta = \varepsilon(q_{x,k}, Q_{x,k}^\delta). \tag{7.6}$$

7.1.2 Definition of the Set-To-Target Control Problem PWAPS

As for APS and SAPS, a stability definition for PWAPS is stated. Formally, *attractivity and stability with confidence* δ for PWAPS is defined as follows:

Definition 7.3. *Given a bounded time domain $\tau = \{0, 1, \dots, N\}$, $N \in \mathbb{N}_0$, and continuous inputs $\{u_k\}$ for $k \in \tau$, the PWAPS* (7.1) *is called* attractive with confidence δ *on the domain τ, if for any initial condition $x_0 \in X_0^\delta$, and any $v_k \in \varepsilon(q_v, Q_v c_x)$, finite parameters $\bar{q} \in \mathbb{R}^n$ and $\bar{Q} \in \mathbb{R}^{n \times n}$ exist, such that:*

$$||q_{x,N}|| \leq ||\bar{q}||, \quad ||Q_{x,N}|| \leq ||\bar{Q}||. \tag{7.7}$$

The system is said stable with confidence δ *on a bounded time domain* $[0, N]$*, if in addition*

$$||q_{x,k+1}|| < ||q_{x,k}||, \quad ||Q_{x,k+1}|| \leq ||Q_{x,k}||, . \tag{7.8}$$

holds for any $0 \leq k \leq N - 1$. △

As in the previous chapter, the choice of this stability definition is well motivated, since it implies a convergence of the expected value $q_{x,k}$ to the origin and a decrease of the confidence ellipsoid X_k^δ in size (or at least convergence to a constant size). The convergence in size follows from the convergence of the covariance matrix $Q_{x,k}$.

The control synthesis procedure aims at the computation of a stabilizing control law $\kappa : \mathbb{R}^n \times \mathbb{N}_0 \to \mathbb{R}^m$ according to:

$$u_k = \kappa(x_k, k). \tag{7.9}$$

The set-to-target control problem for PWAPS can now be stated as follows:

Problem 7.1. *Let a PWAPS with disturbances and input constraints according to (7.1), a terminal region $\mathbb{T} = \varepsilon(q_T, Q_T) \subset \mathbb{R}^n$ centered in an equilibrium point in the origin, and an initial confidence set of states X_0^δ be given. Find a time-varying control law:*

$$\kappa(x_k, k), \quad x_k \in X_k^\delta, \tag{7.10}$$

which renders the closed-loop system stable with confidence δ, and for which a finite $N \in \mathbb{N}$ exists with:

$$X_N^\delta \subseteq \mathbb{T}, \tag{7.11}$$

i.e., any initial state $x_0 \in X_0^\delta$ is transferred into the terminal set $\mathbb{T}$ with probability δ within N steps.

□

In combination of Assumption 7.1 it is always possible to find a suitable coordinate transformation, which recasts the Problem 7.1, such that the terminal set $\mathbb{T}$ is centered in the origin.

The following section elaborates on techniques for controller synthesis for PWAPS by the use of reachable sets.

7.2 Controller Synthesis for PWAPS

This section proposes an algorithm to solve Problem 7.1.

One main challenge in the controller synthesis for PWAPS is the fact, that the confidence reachable set may be located in more than one region $\Theta^{(i)}$ at the same time step. This leads to numerous problems for the further computation of the reachable sets and controller synthesis, e.g. the consideration of only a certain slice of the reachable set would require a reachability computation for a non-ellipsoidal set. Since the reachable set computation in this thesis is based on an ellipsoidal set formulation, an alternative approach is proposed later in 7.2.1 to retain the ellipsoidal sets.

The following formulations of the control law and closed-loop dynamics for PWAPS are already known from the previous chapters, but are repeated for completeness.

To specify the structure of the control law (7.10), the already known local time-variant, affine state feedback controller is selected:

$$u_k = \kappa(x_k, k) = -K_k x_k + d_k \in U, \ \forall \ x_k \in X_k^\delta. \tag{7.12}$$

Thus, a solution of Problem 7.1 is established by a set of control tuples $(K_k, d_k) \ \forall \ k \in \{0, 1, \ldots, N-1\}$ satisfying the conditions of the problem statement while considering:

$$\bar{U}_k := \{u_k \mid \forall \ x_k \in X_k^\delta : u_k = -K_k x_k + d_k\} \subseteq U. \tag{7.13}$$

The control law (7.12) leads to the following closed-loop dynamics for the parameters of the state distribution $q_{x,k}$ and $Q_{x,k}$, which is similar to (6.12), but z_k is not a degree of freedom:

$$q_{x,k+1} = A_{cl,k,z_k} q_{x,k} + B_{z_k} d_k + G_{z_k} q_v, \tag{7.14a}$$

$$Q_{x,k+1} = A_{cl,k,z_k} Q_{x,k} A_{cl,k,z_k}^T + G_{z_k} Q_v G_{z_k}^T, \tag{7.14b}$$

with $A_{cl,k,z_k} := A_{z_k} - B_{z_k} K_k$. The confidence ellipsoid X_{k+1}^δ is obtained from:

$$X_{k+1}^\delta = \varepsilon(q_{x,k+1}, \underbrace{Q_{x,k+1} c_x}_{=:Q_{x,k+1}^\delta}). \tag{7.15}$$

In [24], the following semi-definite program has been introduced, which is solved for any $k \in \{0, \ldots, N-1\}$ to provide the controller tuples (K_k, d_k), and thus X_{k+1}^δ.

The basic semi-definite program for the synthesis of a controller tuple (K_k, d_k) for one time step is very similar to the previously introduced SDP's for AS, APS, and SAPS:

$$\min_{S_{k+1}, K_k, d_k} \ \text{trace}\left(\begin{bmatrix} S_{k+1} & 0 & 0 \\ 0 & \mu_1 \|q_{x,k+1}\| & 0 \\ 0 & 0 & \mu_2 \|u_k\| \end{bmatrix}\right), \tag{7.16a}$$

subject to:

$$q_{k+1}^T M q_{k+1} - \rho q_{x,k}^T M q_{x,k} \leq \alpha_k, \tag{7.16b}$$

$$q_{k+1} = A_{cl,k,z_k} q_{x,k} + B_{z_k} d_k + G_{z_k} q_v, \tag{7.16c}$$

$$\alpha_k \leq \max_{l \in \{1, \ldots, k\}} \omega^l \alpha_{k-l}, \tag{7.16d}$$

$$\begin{bmatrix} S_{k+1} & A_{cl,k,z_k} Q_{x,k} & G_{z_k} Q_v \\ Q_{x,k} A_{cl,k,z_k}^T & Q_{x,k} & 0 \\ Q_v G_{z_k}^T & 0 & Q_v \end{bmatrix} \geq 0, \tag{7.16e}$$

$$\text{trace}(S_{k+1}) \leq \text{trace}(Q_{x,k}), \tag{7.16f}$$

$$\begin{bmatrix} (b_{u,i} - r_{u,i}(d_k - K_k q_{x,k})) I_n & -r_{u,i} K_k (Q_{x,k}^\delta)^{1/2} \\ (-r_{u,i} K_k (Q_{x,k}^\delta)^{1/2})^T & b_{u,i} - r_{u,i}(d_k - K_k q_{x,k}) \end{bmatrix} \geq 0, \tag{7.16g}$$

$$\forall i = \{1, \ldots, n_c\}.$$

The concept of flexible Lyapunov functions is again used here for PWAPS in (7.16b)-(7.16d), where $q_{x,k+1}$ is computed with the matrices of the switched closed-loop dynamics $(A_{cl,z_k}, B_{z_k}, G_{z_k})$. The matrix M in (7.16b) is symmetric, positive definite ($M = M^T \geq 0$), and employed for all possible subsystems $(A_{cl,z_0}, B_{z_0}, G_{z_0})$, $\forall\ z_0 \in Z$[1]. By (7.16d), the solution for the current Lyapunov equation is coupled with the past solutions to enforce convergence for the center point $q_{x,k}$ (see Chapter 4 for a detailed explanation). The convergence of the shape matrix $Q^{\delta}_{x,k}$, or equivalently of the covariance matrix $Q_{x,k}$, is enforced by (6.20e) and (6.20f). Both constraints correspond to the LMI for APS ((5.40),(5.41)), since it is exactly the same requirement for the size and shape of the reachable ellipsoid X^{δ}_k. Only the matrix for the closed loop dynamics in (5.40) is adapted to A_{cl,z_k} in (7.16e). The last LMI enforces the satisfaction of the input constraint, and is basically an ellipse-in-polytope problem, which is stated in Proposition 4.2.

The above formulation for the over-approximation of the reachable ellipsoid by an SDP formulation does not consider any intersection of the reachable set X^{δ}_k with more than one regions $\Theta^{(i)}$. But this case is the challenging one, and the consequences and approaches to avoid this case are discussed in the following section.

7.2.1 Push, Branch, and Merge Procedure

The following subsections describe a procedure to handle the difficult case of

$$X^{\delta}_k \cap \Theta^{(i)} \neq X^{\delta}_k, \tag{7.17}$$

where the reachable set X^{δ}_k is partially in one or more regions $\Theta^{(i)}$. If this case is unavoidable, a procedure called *branching* is introduced, and an alternative to branching is a procedure called *pushing*. In the following, these two approaches are motivated and described in detail.

Branching

The closed-loop evolution of the reachable ellipsoid, as specified by (7.14), does not consider the partition of the continuous state space for PWAPS, i.e. it does not take into account whether X^{δ}_{k+1} is completely contained in region $\Theta^{(i)}$, or not. If containment applies ($X^{\delta}_{k+1} \cap \Theta^{(i)} = X^{\delta}_{k+1}$ for any $i \in Z$), no further attention is required. But if not, i.e. if X^{δ}_{k+1} intersects with two or more partition elements ($X^{\delta}_{k+1} \cap \Theta^{(i)} \neq X^{\delta}_{k+1}$), *branching* is needed.

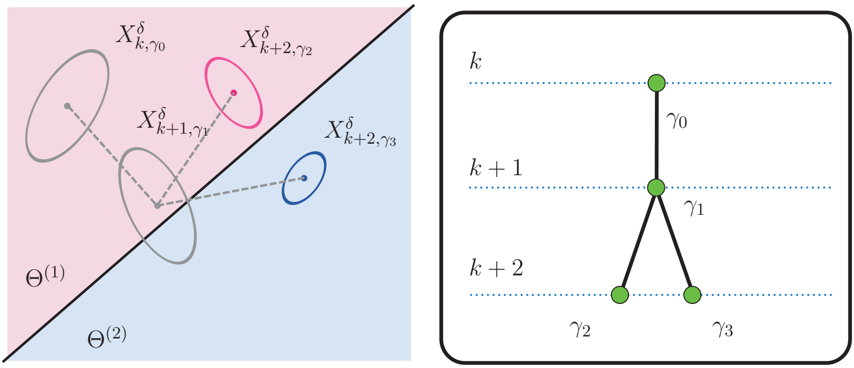
The formulation of the SDP in (7.16) assumes the same mode z_k for all $x_k \in X^{\delta}_k$ for the controller synthesis, and in the described case multiple modes z_k are valid for the states in the reachable ellipsoid X^{δ}_k. It is divided by one or more hyper-planes of the polytopes $P_H(R^{(i)}_x, b^{(i)}_x)$, and then a proper controller synthesis requires a

[1]Alternatively, a separate Lyapunov function defined by M_{z_k} could be employed for the convergence of the center point $q_{x,k}$.

partial consideration of the reachable ellipsoid. But the division of an ellipsoid by a hyper-plane leads to two non-ellipsoidal sets, which tremendously impedes the controller synthesis, and above all, it makes the SDP formulation in (7.16) useless. The foundation of the whole control synthesis procedure in this thesis is the optimization problem build on an ellipsoidal set formulation of the reachable sets.

It is crucial to retain the ellipsoidal shape of the sets, and to this end, the *branching* procedure is proposed, in which the SDP is solved several times for each intersecting region, and therefore, for different modes z_k. That is, instead of a partial consideration of the reachable ellipsoid for each intersecting region in the controller synthesis, the complete reachable ellipsoid X_k^δ is considered in the solution of the SDP for every intersecting region z_k.

The case, when *branching* is required, is explained with an example, shown in Figure 7.1.



(a) (b)

Figure 7.1: Left: intersection of X^δ_{k+1} with more than one region $\Theta^{(i)}$; Right: resulting tree structure when branching occurs.

It illustrates on the left (Fig. 7.1a) the branching procedure, where the set X^δ_{k+1} intersects with both regions $\Theta^{(1)}$ and $\Theta^{(2)}$. At time step $k+1$ the SDP is solved twice: first with $z_{k+1} = 1$, and second with $z_{k+1} = 2$. This results in two different control tuples (K_{k+1}, d_{k+1}), which lead to two different reachable sets for time step $k+2$. Depending in which region the current state x_{k+1} is contained, the corresponding control tuple is chosen to evaluate the closed-loop dynamics.

In order to keep track of the different confidence reachable sets at any time step k, a tree structure Γ is employed. A general tree structure is previously defined in 6.2.1, but the interpretation of the tree nodes is completely different. The tree structure in Chapter 6 is used to apply a tree search procedure, and any node γ_i is specified by a sequence of discrete modes to reach this node from the root node γ_0. Furthermore, at any time step k, the control algorithm 6.1 provides only one single node of the tree to obtain the control tuple (K_k, d_k). In contrast to that, the tree structure in this chapter is used to keep track of all possible reachable sets valid at any time step k. In Chapter 6 only one reachable ellipsoid can be valid at time step k, whereas in the current chapter, due to the branching procedure, multiple reachable ellipsoids can be valid (as shown in Figure 7.1 for $k+2$).

The spanned tree for this example is shown on the right (Fig. 7.1b), where the tree Γ is rooted in γ_0 for time step k. The reachable ellipsoid in time step $k+1$ intersects with more than one region, and therefore, the node γ_1 has multiple child nodes, such that: $Succ(\gamma_1) = \{\gamma_2, \gamma_3\}$. Such that for $k+2$ two reachable sets have to be considered for further evaluation of the system dynamic, since the system state can be contained in both sets. The different nodes with the same time index k are referred to as a layer of the tree, and the tree structure in PWAPS is characterized by the fact, that all nodes in a layer must be further explored for a proper evaluation of PWAPS dynamics.

Another difference to the tree structure used in Chapter 6 is the set of parameters linked to a node γ_i. In PWAPS a node $\gamma_i \in \Gamma$ is linked to a parameter tuple $\gamma_i := (Pre(\gamma_i), Succ(\gamma_i), ReachSet(\gamma_i), CtrlPrm(\gamma_i), NgetIntReg(\gamma_i), NSetProb(\gamma_i), \ldots$ $Mode(\gamma_i), TimeStep(\gamma_i))$. The functions $Pre(\gamma_i)$, $Succ(\gamma_{k,i})$, $ReachSet(\gamma_{k,i})$, $Mode(\gamma_{k,i})$, and $TimeStep(\gamma_i)$ are defined as in (6.15), and the remaining functions are defined as follows:

$$NgetIntReg(\gamma_i) \;:\; 2^{\Gamma} \to 2^{Z}, \tag{7.18}$$

$$NSetProb(\gamma_i) \;:\; \Gamma \to [0,1]\,, \tag{7.19}$$

$$SetProb(X^{\delta}_{k+1}, \Theta^{(i)}) \;:\; \mathcal{E} \times \mathcal{P} \to [0,1]\,, \tag{7.20}$$

$$CtrlPrm(\gamma_i) \;:\; \Gamma \to \mathbb{R}^{m\times n} \times \mathbb{R}^{m} \tag{7.21}$$

where $NgetIntReg(\gamma_i)$ returns a set of discrete modes $Z_{int,k}$ of intersecting regions, and $NSetProb(\gamma_i)$ and $SetProb(X_{k+1}, \Theta^{(i)})$ return the respective share of $X^{\delta}_{k+1} \cap \Theta^{i}$, but with different arguments. The function $CtrlPrm(\gamma_i)$ returns the current valid control tuple (K_k, d_k) for the evaluation of the control law.

For ease of notation, the probabilistic reachable set in a node γ_i is in the following denoted by X^{δ}_{k,γ_i}, and it holds that:

$$X^{\delta}_{k,\gamma_i} := ReachSet(\gamma_i). \tag{7.22}$$

The following function is employed to determine all discrete states, for which intersections of $\Theta^{(i)} \in \bar{\Theta}$ and X^{δ}_{k+1} exist:

$$Z_{int,k+1} := getIntReg(X^{\delta}_{k+1}, \bar{\Theta}) \subset Z, \tag{7.23}$$

with

$$getIntReg \;:\; 2^{\mathbb{R}^n} \times \bar{\Theta} \to 2^{Z}. \tag{7.24}$$

If the result is a singleton, i.e. $|Z_{int,k+1}| = 1$, no further action is needed. Otherwise, the reachable set overlaps with more than one region, and branching is needed.

The edges of the tree are established by the successor and predecessor relation between two subsequent confidence sets (e.g. $X^{\delta}_{k+1,\gamma_1} \to X^{\delta}_{k+2,\gamma_2}$ in Fig. 7.1). If branching occurs, as in the case of the figure, $X^{\delta}_{k+2,\gamma_2}$ and $X^{\delta}_{k+2,\gamma_3}$ stem from $X^{\delta}_{k+1,\gamma_1}$, the distribution $x_{k+1} \sim \mathcal{N}(q_{x,k+1}, Q_{x,k+1})$ is converted into two distributions $x_{k+2,\gamma_2} \sim \mathcal{N}(q_{x,k+2,\gamma_2}, Q_{x,k+2,\gamma_2})$ and $x_{k+2,\gamma_3} \sim \mathcal{N}(q_{x,k+2,\gamma_3}, Q_{x,k+2,\gamma_3})$ by two different control laws.

A separation of the confidence reachable set $X^{\delta}_{k+1,\gamma_1}$ in Fig 7.1a results in a different interpretation of the confidence parameter δ for the consecutive reachable sets. Since the probability of being in one of the separated parts of the ellipsoid is in general different, a further parameter for a correct probability interpretation is needed. This parameter is denoted by ϵ_{RS,γ_i}, and encodes the probability of being in one of the separated parts. It is given by the share of the reachable ellipsoid X^{δ}_{k,γ_i} in a region $\Theta^{(i)}$, and it holds

$$\epsilon_{RS,\gamma_i} := NSetProb(\gamma_i). \tag{7.25}$$

The share ϵ_{RS,γ_i} is determined by the probability of x_{k+1} being inside of $\Theta^{(i)} \cap X^{\delta}_{k+1}$. Computing this probability relies on solving a multi-dimensional integral of the normal probability density function over the set $\Theta^{(i)}$:

$$Pr\left(x_{k+1} \in \Theta^{(i)}\right) = \int_{\zeta \in \Theta^{(i)}} \mathcal{N}(q_{x,k+1}, Q_{x,k+1}) d\zeta. \tag{7.26}$$

The shares ϵ_{RS,γ_2} and ϵ_{RS,γ_3} are computed according to (7.27). The fact, that $X^{\delta}_{k+2,\gamma_2}$ is more likely to occur, since it is more likely for x_k to be inside $\Theta^{(1)}$ than in $\Theta^{(2)}$, is considered by $\epsilon_{RS,\gamma_2} = 0.6 > \epsilon_{RS,\gamma_3} = 0.4$ in the example in Figure 7.1.

While it is hard to solve (7.26) analytically, this integral can be approximated by using a combination of $n_{x,i}$ uni-variate distributions, which can be evaluated by the cumulative distribution function. The approximation used in Sec. 5.2.2 for evaluating the chance constraints, is also suitable for the evaluation of (7.26). As elaborated in Sec. 5.2.2, the evaluation of the chance constraint with ϵ_{vio} in (5.25) is an over-approximation, and in order to avoid an exceeding of the cumulative probability of 1, the share has to be scaled as follows:

$$\epsilon_{RS,\gamma_i} = \frac{Pr(x_{k+1} \in \Theta^{(i)})}{\sum_{i \in Z_{int,k+1}} Pr(x_{k+1} \in \Theta^{(i)})}, \sum_{i \in Z_{int,k+1}} \epsilon_{RS,\gamma_i} = 1. \tag{7.27}$$

The described procedure specifies a possibility for probabilistic reachable set computation for PWAPS, and the branching introduces an additional probability variable, namely ϵ_{RS,γ_i}. This new value has to be considered in the probability interpretation of the reachable set X^{δ}_{k}, since due to the branching procedure a realization of the random variable x_k is not necessarily in a single confidence ellipsoid X_k^δ. Instead, it is possible to have several confidence ellipsoids X_{k,γ_i}^δ at time step k, and the evaluation of the probability of x_k being in any of the available X_{k,γ_i}^δ is done by:

$$Pr(x_k \in X_{k,\gamma_i}^\delta) = \epsilon_{RS,\gamma_i} \cdot \delta, \tag{7.28}$$

and with (7.27), it holds that:

$$\sum_{i \in Z_{int,k}} Pr(x_k \in X_{k,\gamma_i}^\delta) = \delta. \tag{7.29}$$

The following approach introduces a modification of the SDP in (7.16) to avoid *branching*, in order to keep the number of branches in the tree low, and hence the computational effort low, as well.

Pushing

The SDP in (7.16) does not take into account any branching of the reachable ellipsoids. It is rather a straightforward application of a reachable set computation with $\mathbb{X}_k = \mathbb{R}^n$. But as already mentioned, the state space is partitioned, and requires a cautious computation of the reachable sets. The above described procedure of branching generates a possibly large number of tree branches, and one approach to reduce the number of branches in the tree will be referred to as *pushing*. If an intersection with more than one region $\Theta^{(i)}$ is detected ($|Z_{int,k+1}| > 1$), before branching, *pushing* is applied, and the SDP (7.16) is solved again with an additional constraint to push the computed ellipsoid in one of the intersecting regions.

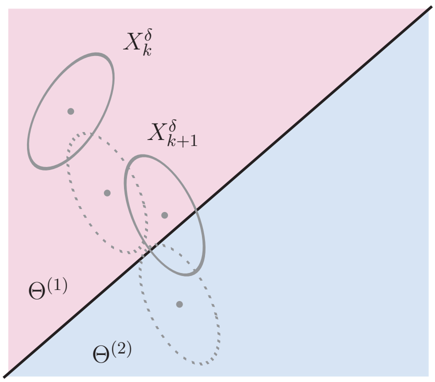


Figure 7.2: *Pushing* of X_{k+1}^δ into one region $\Theta^{(i)}$: if intersection is detected, pushing tries to push X_{k+1}^δ into $\Theta^{(1)}$ **OR** $\Theta^{(2)}$.

For this purpose, the initial SDP is modified in order to avoid intersection with the region boundary. Figure 7.2 shows this case: to obtain one of the dotted ellipsoids (instead of X^{δ}_{k+1}), an additional constraint is formulated, such that the distance between the center point $q_{x,k+1}$ and the boundary $\partial(\Theta^{(i)})$ is greater than the length of the semi-major axis of X^{δ}_{k}. This proceeding is reasonable, since the exact shape and orientation of the resulting ellipsoid for $k+1$ is not known in advance. But due to the constraint (7.16e) in the SDP, it is known that the resulting semi-major axis of the reachable ellipsoid decreases for $k+1$. This is used to formulate a constraint in the SDP, such that a feasible solution of the optimization problem ensures containment in one of the intersecting regions.

The length of the semi-major axis of an ellipsoid is given by the square root of the maximum eigenvalue of the shape matrix (see Sec. 3.2). If a hyper-plane of $\partial(\Theta^{(i)})$, to which the distance has to be adjusted, is parametrized by the normal vector $r^{(i)}_{x,j}$ and the distance to the origin $b^{(i)}_{x,j}$ (see Sec. 3.2 and (7.2)), the distance to the center point $q_{x,k+1}$ is given by:

$$D_H\left((r^{(i)}_{x,j}, b^{(i)}_{x,j}), q_{x,k+1}\right) := r^{(i)}_{x,j} q_{x,k+1} - b^{(i)}_{x,j} \geq \sqrt{\max\{\Lambda(Q^{\delta}_{x,k})\}}. \tag{7.30}$$

The extended SDP, i.e. (7.16) with (7.30), has to be solved for every discrete mode $z_k \in Z_{int,k+1} := getIntReg(X^{\delta}_{k+1}, \bar{\Theta})$, until a feasible solution is found. If multiple feasible solutions exists for $z_k \in Z_{int,k+1}$, the best solution according to the cost function (7.16a) is chosen. If no alternative solution for $z_k \in Z_{int,k+1}$ is available, the pushing attempt fails, and branching is unavoidable. The Algorithm 7.1 summarizes the needed computations for the push and the previously mentioned branch procedure.

The algorithm has to be applied, if for any node γ_i and the corresponding reachable set $ReachSet(\gamma_i)$ $|Z_{int,k+1}| > 1$ is detected. The first step solves the SDP for every discrete mode of the intersecting regions, with the additional distance constraint in (7.30). The feasible or infeasible solutions are used in the second step of the algorithm to decide whether pushing or branching is necessary. If one or more feasible solution exist, the best solution according to the cost function of the SDP is chosen to compute the distribution for x_{k+1} and the reachable ellipsoid X_{k+1}. The set of intersecting regions $Z_{int,k+1}$ is then updated, and used in the next iteration to verify, if branching/pushing is needed. The probability ϵ_{RS} for the consecutive node γ_j is the same as for γ_i, since no branching is necessary, and hence computed by $NSetProb(\gamma_i)$. Amongst others, these parameters are used to generate the new node γ_j, where j is computed by the number of nodes in the tree Γ plus one, and the predecessor is γ_i, and the successor is empty. Furthermore, the new node γ_j is set as one of the successor nodes of γ_i, and integrated in the tree Γ. The sets $\mathcal{O}_W$ and $\mathcal{O}_P$ have already been used in the previous chapter for the tree search procedure, and are useful for PWAPS as well. The set $\mathcal{O}_W$ contains all nodes of the deepest layer of the tree Γ, and $\mathcal{O}_P$ contains all remaining nodes, such that $\Gamma = \mathcal{O}_W \cup \mathcal{O}_P$ holds. Therefore, the node γ_i is removed from $\mathcal{O}_W$ and put into $\mathcal{O}_P$, and the new node

Algorithm 7.1. ***Push Algorithm***

given: (7.1a) *with* $x_k \sim \mathcal{N}(q_{x,k}, Q_{x,k})$, $v_k \sim \mathcal{N}(q_v, Q_v)$, $\bar{\Theta}$, *and* $U \in \mathcal{P}$; δ, $\rho \in (0, 1]$, $\alpha_k \in [0,\ 1]$, $\omega \in [0, 1)$, $Z_{int,k+1}$, γ_i, $\mathcal{O}_W$, $\mathcal{O}_P$

1. ***for*** $\bar{z} \in Z_{int,k+1}$ ***do***
   - *solve SDP* (7.16) *with the additional constraint* (7.30), *and* $z_k = \bar{z}$.

   end

2. ***if*** *a feasible solution exists* ***do "pushing"***
   - *choose the discrete mode* $z^\star$ *with the best solution according to* J_k
   - *compute distribution for* x_{k+1} *from* (7.14) *with* $z_k = z^\star$
   - *compute* X^δ_{k+1} *according to* (7.15)
   - $Z_{int,k+1} := getIntReg(X^\delta_{k+1,\gamma_{k+1,i}}, \bar{\Theta})$
   - $\epsilon_{RS} = NSetProb(\gamma_i)$
   - $\gamma_j := \big(\gamma_i, \emptyset, X^\delta_k, (K_k, d_k), Z_{int,k+1}, \epsilon_{RS}, z^\star, k+1\big)$, *with* $j = |\Gamma| + 1$
   - $Succ(\gamma_i) := Succ(\gamma_i) \cup \gamma_j$
   - $\Gamma := \Gamma \cup \gamma_j$; $\mathcal{O}_W = (\mathcal{O}_W \cup \gamma_j) \setminus \gamma_i$; $\mathcal{O}_P = \mathcal{O}_P \cup \gamma_i$

   end

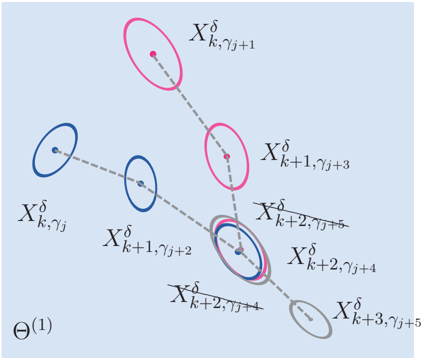
3. ***return*** Γ, $\mathcal{O}_W$, $\mathcal{O}_P$

γ_j is put into the set $\mathcal{O}_W$. These two sets will be also used in the overall control synthesis algorithm, which will be introduced later.

But, if the computations in the first step of Algorithm 7.1 leads to no feasible solution of the SDP, and *pushing* fails, *branching* is needed.

Merging

Once branching is required, the reachable set computation and controller synthesis has to be applied for each branch in the tree. But in some cases, it is worthwhile to inspect the reachable ellipsoids in the tree nodes in detail, in order to reduce the number of branches by a procedure called *merging*, i.e. two or more branches can be merged to a single branch under certain conditions. This is only reasonable, if the considered distributions in each branch at the same time step k are very similar, i.e. the expected values have to be in a close neighborhood to each other, and the covariance matrices should have similar entries. A graphical illustration of this attempt to reduce the number of branches in the tree is shown in Figure 7.3.



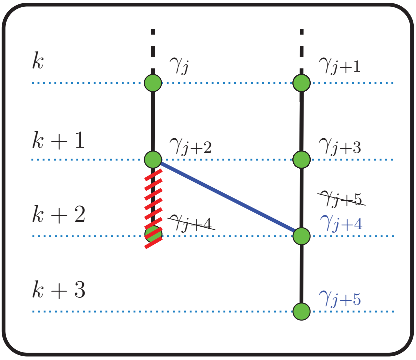


Figure 7.3: *Merging* of $X^\delta_{k+2,\gamma_{j+5}}$ and $X^\delta_{k+2,\gamma_{j+4}}$ to a new distribution. Left: the two distribution of $X^\delta_{k+2,\gamma_{j+5}}$ and $X^\delta_{k+2,\gamma_{j+4}}$ are replaced by a new distribution (grey colored). Right: the merging of the two nodes changes the edges in the tree, and the former node γ_{j+4} is replaced with the merged node.

Therein, the evolution of the reachable ellipsoid for two branches is shown (blue and pink), and at time step $k+2$ the confidence ellipsoids $X^\delta_{k+2,\gamma_{j+4}}$ and $X^\delta_{k+2,\gamma_{j+5}}$ are almost completely overlapping. The overlapping reachable ellipsoids indicate a very similar distribution for $x_{k+2,\gamma_{j+4}}$ and $x_{k+2,\gamma_{j+5}}$, and a very similar result of the controller synthesis can be expected for both branches. Therefore, it is obviously appropriate to merge these to branches, and consider only one confidence ellipsoid for further evaluation and controller synthesis. On the right side of Figure 7.3, the changes in the tree by the merging procedure are shown. Before merging is applied, the nodes γ_{j+4} and γ_{j+5} belong to different branches, and after the merging procedure the old node γ_{j+4} is replaced by a new node γ_{j+4} (indicated by the blue color) with $Pre(\gamma_{j+4}) = \{\gamma_{j+2}, \gamma_{j+3}\}$, and the node γ_{j+5} becomes obsolete. The new node γ_{j+4} contains the merged distribution, and is now considered for further evaluation of the system dynamics. The successor node of γ_{j+4} is the new node γ_{j+5} for time step $k+3$, and is generated by a controller synthesis for the merged distribution in γ_{j+4}.

The similarity of two distributions in the pre-merged nodes γ_{j+4} and γ_{j+5} can be evaluated by the *Bhattacharyya distance* (see [32]). For the case of two normal distributions, the Bhattacharyya distance is defined as follows:

Definition 7.4. *Given two multivariate normal distributions* $\xi_1 \sim \mathcal{N}(q_{\xi_1}, Q_{\xi_1})$ *and*

$\xi_2 \sim \mathcal{N}(q_{\xi_2}, Q_{\xi_2})$, *the Battacharyya distance* D_B *is given by:*

$$D_B\left(\mathcal{N}(q_{\xi_1}, Q_{\xi_1}), \mathcal{N}(q_{\xi_2}, Q_{\xi_2})\right) = \frac{1}{8}(q_{\xi_1} - q_{\xi_2})^T Q_\xi^{-1}(q_{\xi_1} - q_{\xi_2})... \\ + \frac{1}{2} ln\left(\frac{|Q_\xi|}{\sqrt{|Q_{\xi_1}||Q_{\xi_2}|}}\right), \tag{7.31}$$

with

$$Q_\xi = \frac{Q_{\xi_1} + Q_{\xi_2}}{2}. \tag{7.32}$$

△

The Bhattacharyya distance provides a simple to compute parameter, which can be used to decide whether a merging of the branches is admissible or not. Once merging is required, an open question is the computation of the resulting distribution for the system state x_k, which represents the two merged distributions.

In this thesis, a weighted mean distribution is suggested, in which the weights are determined by the occurrence probabilities ϵ_{RS} for the two merged nodes γ_i and γ_j. Furthermore, it has to be specified, in which node the information about the merged distribution is stored and used further for exploration. To this end, the node with the smallest subscript value is used: $\gamma_{\min(i,j)}$. Consequently γ_4 is used in the above example to contain the merged distribution.

The resulting distribution for $\gamma_{\min(i,j)}$ is computed as follows:

$$x_{k,\gamma_{\min(i,j)}} = \mathcal{N}(q_{k,\gamma_{\min(i,j)}}, Q_{k,\gamma_{\min(i,j)}}), \tag{7.33}$$

with

$$Q_{x,k,\gamma_{\min(i,j)}} = \epsilon_{RS,\gamma_i} \cdot Q_{x,k,\gamma_i} + \epsilon_{RS,\gamma_j} \cdot Q_{x,k,\gamma_j}, \tag{7.34a}$$

$$q_{x,k,\gamma_{\min(i,j)}} = \epsilon_{RS,\gamma_i} \cdot q_{x,k,\gamma_i} + \epsilon_{RS,\gamma_j} \cdot q_{x,k,\gamma_j}. \tag{7.34b}$$

The resulting occurrence probability $\epsilon_{RS,\gamma_{\min(i,j)}}$ for the reachable set $X^\delta_{k,\gamma_{\min(i,j)}}$ in the new generated node $\gamma_{\min(i,j)}$ is obtained by the sum of the occurrence probabilities of the two merged nodes γ_i and γ_j:

$$\epsilon_{RS,\gamma_{\min(i,j)}} = \epsilon_{RS,\gamma_i} + \epsilon_{RS,\gamma_j}. \tag{7.35}$$

The needed computations for the merging procedure are summarized in Algorithm 7.2.

Therein, the set $\mathcal{O}_W$ is used again to determine all nodes in the deepest layer of the tree, since only this nodes can be merged. The algorithm checks for each combination of two nodes in $\mathcal{O}_W$ whether a predefined threshold $\bar{D}_B$ for the Bhattacharrya distance is exceeded or not. If the Bhattacharyya distance is less than

Algorithm 7.2. ***Merge Algorithm***

given: Γ, $\mathcal{O}_W$, $\bar{D}_B \in \mathbb{R}$

1. ***for each*** $\gamma_i, \gamma_j \in \mathcal{O}_W$, *with* $i \neq j$ ***do***

- *compute the Bhattacharyya distance* D_B *of the two distributions for* $ReachSet(\gamma_i)$ *and* $ReachSet(\gamma_j)$ *according to* (7.31)
- ***if*** $D_B \leq \bar{D}_B$ ***do***
  - *compute merged distribution* x_{k+1} *according to* (7.34), *and the confidence ellipsoid* X^{δ}_{k+1}
  - ***if*** $X^{\delta}_{k+1} \supseteq ReachSet(\gamma_i) \cup Reachset(\gamma_j)$ ***do***
    * $\gamma_h = \Big(Pre(\gamma_{\min(i,j)}), \emptyset, X^{\delta}_{k+1}, getCtrlPrm(\gamma_{\min(i,j)}), \ldots$
      $NgetIntReg(\gamma_{\min(i,j)}), NSetProb(\gamma_i) + NSetProb(\gamma_j), \ldots$
      $Mode(\gamma_{\min(i,j)}), k+1\Big)$, *with* $h = \min(i,j)$
    * $Succ(Pre(\gamma_i)) := Succ(Pre(\gamma_i)) \cup \gamma_h$; $Succ(Pre(\gamma_j)) := Succ(Pre(\gamma_j)) \cup \gamma_h$
    * $\Gamma := (\Gamma \setminus \{\gamma_i, \gamma_j\}) \cup \gamma_h$; $\mathcal{O}_W = (\mathcal{O}_W \setminus \{\gamma_i, \gamma_j\}) \cup \gamma_h$

    end

  end

end

2. ***return*** Γ, $\mathcal{O}_W$

$\bar{D}_B$, the new distribution and the confidence ellipsoid are computed. In order to guarantee the confidence δ for the new confidence set, it has to verified that the set contains the two merged confidence sets. If containment applies, a new node with the parameters of the merged distribution is generated and added to the tree Γ. The inclusion of the new node γ_h requires a removal of the two merged nodes of the tree Γ and of the set $\mathcal{O}_W$.

The new node γ_h also affects the predecessor nodes of γ_i and γ_j, such that the successor node of $Pre(\gamma_i)$ and $Pre(\gamma_j)$ changes to γ_h. By this rearrangement of successor and predecessor it is guaranteed, that the sequence of nodes of the involved branches will lead to γ_h for both branches.

In the following, the push-branch and merge algorithm are embedded into a control algorithm to solve Problem 7.1.

7.2.2 Algorithmic Synthesis of the Continuous Control Law

The procedure to obtain the sequence of control laws $(K_{k,\gamma_i}, d_{k,\gamma_i})$ for all $\gamma_i \in \Gamma$ and for $k \in \{1, \ldots, N\}$ to solve the problem 7.1 is now formulated as an algorithm. The Algorithm 7.3 steers the initial distribution $x_0 \sim \mathcal{N}(q_{x,0}, Q_{x,0})$ into the terminal region $\mathbb{T}$, and the computation terminates successfully with $k = N$, if the confidence ellipsoids $X^{\delta}_{N,\gamma} \in \mathcal{E}$ of all nodes γ_i of layer N are contained in the target set $\mathbb{T}$.

The main loop of the algorithm is executed until the terminal region $\mathbb{T}$ is not yet reached, and the confidence sets sufficiently approach $\mathbb{T}$ in the each step. The latter criterion, which is included to avoid an unreasonably large number of iterations, can be modeled by:

$$\pi_{k+1} = \min_{\gamma_i \in \mathcal{O}_W} || \left(q_{x,k+1,\gamma_i} - q_{x,k,Pre(\gamma_i)} \right) || \geq \pi_{min} \tag{7.36}$$

with a parameter $\pi_{min} \in \mathbb{R}$.

In the first step of the algorithm, the first computation determines the set of intersecting regions, and the optimization problem (7.16) is solved for every intersecting region with $z_k = \bar{z}$. For each feasible solution of the SDP, a reachable set for the next time step X^{δ}_{k+1} is computed. The further execution of the algorithm depends on the number of intersecting regions for the next time step: if more than one region intersects with the reachable set, "pushing" is applied and Algorithm 7.1 is executed. Otherwise, the node γ_j and its parameters are generated and added to the set of nodes for the next time step. Once each of the nodes in $\mathcal{O}_W$ are processed, the new set of nodes $\mathcal{O}_W$ is examined, if a pairwise merging of branches is possible. This is done by executing Algorithm 7.2, only if more than one node is contained in $\mathcal{O}_W$. The third and fourth step of the algorithm computes the additional criterion for termination and increases the time index k. A successful termination of Algorithm 7.3 provides a set of control parameters for each node in the tree, and the application of the controller solves Problem 7.1. The used tree structure in the control algorithm, and the presence of multiple control tuples for a single time step requires a clarification of the feasible execution of a controlled PWAPS. To this end, an additional function for the tree structure is needed to access the active node γ_c for a given time step k and a discrete mode z_k:

$$activeNode(\Gamma, k, z_k) \; : \; \Gamma \times \mathbb{N}_0 \times Z \rightarrow \Gamma. \tag{7.37}$$

Definition 7.5. *Given a PWAPS according to* (7.1) *and a tree structure* Γ*, resulting from a successful termination of Algorithm 7.3. A sequence of pairs* (x_k, z_k)*,* $k \in \{0, 1, \ldots\}$ *is called "'admissible"', if for every* $k \in \mathbb{N}_0$*,* x_{k+1} *is determined by the following order of computations:*

1. *sample the disturbance* $v_k \sim \mathcal{N}(q_v, Q_v)$
2. *sample the state* $x_k \sim \mathcal{N}(q_{x,k}, Q_{x,k})$

Algorithm 7.3. ***Probabilistic Control Algorithm***

given: (7.1a) *with* $x_0 \sim \mathcal{N}(q_{x,0}, Q_{x,0})$, $v_k \sim \mathcal{N}(q_v, Q_v)$, $\bar{\Theta}$, *and* $U \in \mathcal{P}$; $\mathbb{T}$, δ, π_{min}, $\rho \in (0, 1]$, $\alpha_0 \in [0,\, 1]$, $\omega \in [0, 1)$, *and* $\bar{D}_B \in [0, 0.5]$
define: $k := 0$, $\pi_0 := \pi_{min}$, $\gamma_0 = (\emptyset, \emptyset, X_0^\delta, \emptyset, \emptyset, 1, \emptyset, 0)$, $\mathcal{O}_W := \{\gamma_0\}$, $\mathcal{O}_P = \emptyset$
while $\exists\, \gamma_i \in \mathcal{O}_W$ *with* $X_{k,\gamma_i}^\delta \nsubseteq \mathbb{T}$ ***and*** $\pi_k \geq \pi_{min}$ ***do***

1. ***for*** $\gamma_i \in \mathcal{O}_W$ ***do***
   - *compute* $Z_{int,k} := NgetIntReg(\gamma_i)$
   - ***for*** $\bar{z} \in Z_{int,k}$ ***do***
     - *solve the optimization problem* (7.16) *with* $z_k = \bar{z}$
     - *compute the distribution of* x_{k+1} *from* (7.14)
     - *compute* X_{k+1}^δ *according to* (7.15)
     - $Z_{int,k+1} := getIntReg(X_{k+1}^\delta, \bar{\Theta})$
     - ***if*** $|Z_{int,k+1}| > 1$ ***do "pushing"***
       * *execute Algorithm 7.1*
     - ***else***
       * $\gamma_j := (\gamma_i, \emptyset, X_{k+1}^\delta, (K_k, d_k), Z_{int,k+1}, NgetSetProb(\gamma_i), \bar{z}, k+1)$, *with* $j = |\Gamma| + 1$
       * $Succ(\gamma_i) := Succ(\gamma_i) \cup \gamma_j$
       * $\Gamma := \Gamma \cup \gamma_j$; $\mathcal{O}_W = (\mathcal{O}_W \cup \gamma_j) \setminus \gamma_i$; $\mathcal{O}_P = \mathcal{O}_P \cup \gamma_i$

   end, end, end
2. ***if*** $|\mathcal{O}_W| > 1$ ***do "merging"***
   - *execute Algorithm 7.2*

   end
3. *compute* π_{k+1} *according to* (7.36)
4. $k := k + 1$

end while
return (K_k, d_k) *for all* $\gamma_i \in \Gamma$ *and* $0 \leq k \leq N - 1$

3. *compute current discrete mode* z_k:
   if $x_k \in int(\Theta^{(i)})$, $i \in Z$ ***do*** $z_k = i$
   elseif $x_k \in \partial(\Theta^{(i)})$, $i \in Z$ ***do***
   $Z_{adj,k} := getAdjPart(x_k)$

$z_k = \min_{z \in Z_{adj,k}} z$
end

4. *compute current active node* $\gamma_c = activeNode(\Gamma, k, z_k)$
5. *extract feasible control tuples:* $(K_k, d_k) := CtrlPrm(\gamma_c)$
6. *compute* $(q_{x,k+1}, Q_{x,k+1})$ *with the tuple* $(A_{z_k}, B_{z_k}, G_{z_k})$ *and* (K_k, d_k) *according to* (7.14)
7. $x_{k+1} \sim \mathcal{N}(q_{x,k+1}, Q_{x,k+1})$

$\triangle$

Lemma 7.1. *Problem 7.1 is successfully solved, if Algorithm 7.3 terminates with* $X^{\delta}_{N,\gamma} \subseteq \mathbb{T}$, $\forall \gamma_i \in \mathcal{O}_W$. *The solution provides a sequence of control laws* (7.12), *which steer any initial state* $x_0 \in X^{\delta}_0$ *with probability* δ *into the target set* $\mathbb{T}$ *in* N *steps. Furthermore, the input constraint* $u_k \in U$ *holds for all* $0 < k < N$.

Proof: For any $k \in \{0, 1, \dots, N-1\}$ one of the following cases applies for the evolution of the reachable ellipsoids:

1. $|Z_{int,k+1}| = 1$**:** X^{δ}_{k,γ_i} is mapped by (7.14), (7.15) into $X^{\delta}_{k+1,Succ(\gamma_i)} \subseteq \Theta^{(i)}$ for one $\Theta^{(i)} \in \bar{\Theta}$;
2. $|Z_{int,k+1}| > 1$**:** X^{δ}_{k,γ_i} is mapped into $X^{\delta}_{k+1,Succ(\gamma_i)} \nsubseteq \Theta^{(i)}$ for any $\Theta^{(i)} \in \bar{\Theta}$. The execution of Algorithm 7.1 in each time step and the possibility to push the reachable set into one intersecting region leads to $|Z_{int,k+1}| = 1$;
3. $|Z_{int,k+1}| > 1$**:** The application of the push-and-branch Algorithm results in branching, hence $|\mathcal{O}_W| > 1$.

For the cases 1. and 2., the solution of (7.16) by definition preserves the confidence δ. From Lemma 3.2, it holds that $Pr(x_k \in X^{\delta}_k) = \delta$ with $X^{\delta}_k = \varepsilon(q_{x,k}, Q_{x,k} c_x)$, and following (7.14), the state distribution for $k+1$ is
$x_{k+1,Succ(\gamma_i)} \sim \mathcal{N}(q_{x,k+1,Succ(\gamma_i)}, Q_{x,k+1,Succ(\gamma_i)})$. By computing $X^{\delta}_{k+1,Succ(\gamma_i)} = \varepsilon(q_{x,k+1,Succ(\gamma_i)}, Q_{x,k+1,Succ(\gamma_i)} c_x)$ with the scaling factor c_x according to (3.31), the Algorithm 7.3 ensures for case 1. and 2. that $Pr(x_{k+1,Succ(\gamma_i)} \in X^{\delta}_{k+1,Succ(\gamma_i)}) = \delta$. Hence, it is guaranteed that $Pr(x_k \in X^{\delta}_k) = Pr(x_{k+1} \in X^{\delta}_{k+1}) = \delta$ for all $k \in \{0, 1, \dots, N-1\}$, such that $Pr(x_N \in X^{\delta}_N) = \delta$ follows from induction.

For case 3., where branching is needed, the assignment for the occurrence probability $\epsilon_{RS,Succ(\gamma_i)}$ for each following node in Algorithm 7.1 is obtained by multiplying the current occurrence probability ϵ_{RS,γ_i} with each share of intersection of $X^{\delta}_{k+1,suc(\gamma_i)}$ with $\Theta^{(j)}$, $j \in Z_{int,k+1}$, given by $SetProb\left(X^{\delta}_{k+1,Succ(\gamma_i)}, \Theta^{(j)}\right)$.

Once branching occurred in time step k, the confidence probability for $x_{k+1} \in X^{\delta}_{k+1,Succ(\gamma_i)}$ is determined according to (7.28). Furthermore it holds that

$$\sum_{\gamma_i \in \mathcal{O}_W} \epsilon_{RS\gamma_{k+1,i}} = 1, \tag{7.38}$$

and

$$\sum_{\gamma_i \in \mathcal{O}_W} Pr\left(x_{k+1} \in X^{\delta}_{k+1,\gamma_i}\right) = \delta. \tag{7.39}$$

With that, the induction holds as for the cases 1. and 2., since it holds that $Pr(x_k \in X^{\delta}_{k,\gamma_i}) = Pr(x_{k+1} \in X^{\delta}_{k+1,Succ(\gamma_i)}) = \delta$ for all $k \in \{0, 1, \ldots, N-1\}$, $\gamma_i \in \mathcal{O}_W$, such that $Pr(x_N \in \bigcup_{\gamma_i \in \mathcal{O}_W} X^{\delta}_{N,\gamma_{N,i}}) = \delta$ follows from induction.

In addition to the above mentioned cases, a merging of branches may also be reasonable during the execution of Algorithm 7.3. The resulting distribution of the system state is obtained by a weighted addition of the merged distributions, such that the resulting distribution is again a normal distribution, and a confidence ellipsoid can be computed. The execution of the sub-algorithm 7.2 ensures, that the new reachable set is a confidence ellipsoids for both merged branches with confidence δ. Concerning the occurrence probability, instead of splitting it as in the case of branching, the occurrence probability of the two merged branches is determined according to 7.35, such that the confidence probability for the merged reachable ellipsoid is retained, and the induction hold as for the cases 1., 2. and 3..

The satisfaction of the input constraints follows from the construction of (7.16).

In addition, attractivity with confidence δ and stability with confidence δ as defined in [24] follows for $\alpha_k = 0$, $\forall k \in \mathbb{N}_0$ and a successful termination of Algorithm 7.3. □

7.3 Numerical Example

To illustrate the principle of the control algorithm, it is applied to an arbitrary chosen PWAPS. Let the dynamics be modeled by affine dynamics, which differs in different regions of the state space.

7.3.1 Exemplary Model of an PWAPS

The corresponding PWAPS comprises two continuous states, two continuous inputs, and three discrete modes, hence the state space is partitioned into three regions $\Theta^{(i)} \subseteq \mathbb{R}^n$. The regions can be seen from the bold lines in Fig 7.4.

Let the initial distribution of the continuous state and the disturbances be given by:

$$x_0 \sim \mathcal{N}(q_{x,0}, Q_{x,0}) \text{ with } q_{x,0} = \begin{bmatrix} -10 \\ 50 \end{bmatrix}, \quad Q_{x,0} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix},$$

$$v_k \sim \mathcal{N}(0, Q_v) \text{ with } Q_v = \begin{bmatrix} 0.02 & 0.01 \\ 0.01 & 0.02 \end{bmatrix},$$

and the continuous dynamics by:

$$\begin{aligned}
A_1 &= \begin{bmatrix} 9.41 & 0.19 \\ -0.38 & 9.99 \end{bmatrix} 10^{-1}, \; B_1 = \begin{bmatrix} 1.98 & 0.02 \\ 3.96 & 2.00 \end{bmatrix} 10^{-1}, \\
A_2 &= \begin{bmatrix} 9.22 & 0.19 \\ -0.58 & 10.4 \end{bmatrix} 10^{-1}, \; B_2 = \begin{bmatrix} 1.96 & 0.02 \\ 4.02 & 2.04 \end{bmatrix} 10^{-1}, \\
A_3 &= \begin{bmatrix} 11.2 & -0.21 \\ 0.42 & 9.79 \end{bmatrix} 10^{-1}, \; B_3 = \begin{bmatrix} 2.12 & -0.04 \\ 0.04 & 3.96 \end{bmatrix} 10^{-1}, \\
G_1 &= G_2 = G_3 = \begin{bmatrix} 0.1 & 0.05 \\ 0.08 & 0.2 \end{bmatrix}.
\end{aligned}$$

Note that the second and third subsystems have unstable state matrices. The inputs u_k are constrained according to: $-4 \leq u_{1,k} \leq 4$, $-8 \leq u_{2,k} \leq 4$, and the target set is defined to: $\mathbb{T} = \varepsilon\left(0, \begin{bmatrix} 0.96 & 0.64 \\ 0.64 & 0.8 \end{bmatrix}\right)$. The algorithm 7.3 is parametrized by $\delta = 0.95$, $\pi_{min} = 0.01$, $\alpha_0 = 10^{-4}$, $\omega = 0.8$ and $\rho = 0.98$. The cost function is selected to $J = \text{trace}\left(\begin{bmatrix} S_{k+1} & 0 \\ 0 & \mu_1\|q_{k+1}\| \end{bmatrix}\right)$ with $\mu_1 = 0.8$ (and $\mu_2 = 0$). The threshold for the Bhattacharyya distance is selected to $\bar{D}_B = 0.1$.

7.3.2 Discussion of the Numerical Results

The proposed Algorithm 7.3 is able to compute a control law while considering the different dynamics in each region and the stochastic disturbances.

The successful termination of the synthesis algorithm is shown in Fig. 7.4, illustrating the confidence reachable sets X_k^δ. Three iterations after starting from X_0^δ, the ellipsoid cannot be pushed into one region, thus X_3^δ intersects with $\Theta^{(3)}$ and $\Theta^{(1)}$, i.e. branching is required with shares $\epsilon_{RS,\gamma_3} = 0.89$ and $\epsilon_{RS,\gamma_4} = 0.11$. For the subsequent iterations, the SDP has to be solved for the two branches, while the evolution of the confidence sets converge to each other. At time step $k = 16$, the Bhattacharyya distance is below the chosen threshold and the merging procedure is applied. The blue and red branch are merged, such that only the red branch is further evolved.

Following the Algorithm 7.2 the new occurrence probability is then $\epsilon_{RS,\gamma_{31}} = 1$. A few time steps later, the pushing of the reachable set can be seen, where the reachable set is a bit displaced compared to the successor and predecessor sets, such that the transition from $\Theta^{(1)}$ to $\Theta^{(2)}$ proceeds without branching. After $N = 32$ iterations, the confidence reachable sets contained in the terminal set $\mathbb{T}$, and the algorithm terminates successfully.

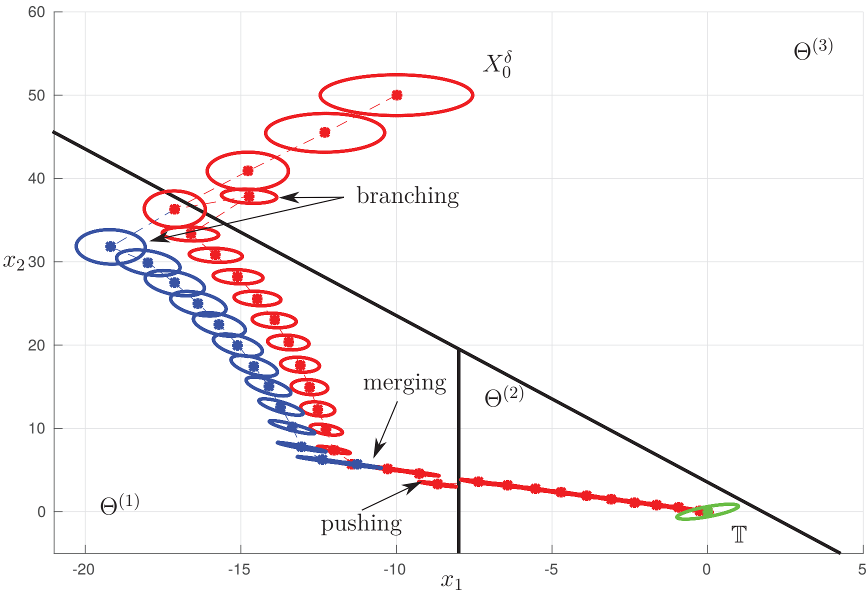


Figure 7.4: Control result for the example: The initial confidence set with mean vector $[-10,\ 50]^T$ is steered to the origin in 32 iterations. Branching occurs after 3 steps with $\epsilon_{RS,\gamma_1} = 0.89$ and $\epsilon_{RS,\gamma_2} = 0.11$. The two branches are merged at $k = 16$.

The overall time of computation is 36.98 sec on a standard PC (Intel Core $i7-6700$ CPU, 16GB RAM, and Matlab 2016a). The SDP problem is built with YALMIP and solved by MOSEK.

The shown result of the numerical example illustrates the advantages of applying the three procedures "pushing", "branching", and merging". The overall goal is to retain the ellipsoidal formulation of the reachable sets, in order to be able to compute the control law by solving an SDP. The branching procedure is a suitable tool to maintain the ellipsoidal shape of the reachable set, even though the separation of an ellipsoid by a half-plane results in two none-ellipsoidal pieces. Moreover, a tree structure Γ is used to keep track of all branches. If the confidence ellipsoids evolve to each other, and are for the most part overlapping, the merging procedure allows a reduction of the number of branches. The similarity of the confidence sets in both branches is evaluated by the Bhattacharyya distance D_B. The shown example in Figure 7.4 includes branching only once, but if the state space is divided into smaller regions, branching may be needed far more times, which leads to more branches and an increase of computational effort and time. Therefore, the possibility of merging two branches is very important to keep the computational effort reasonable.

In addition, by the use of the same Lyapunov condition (7.16b)-(7.16d) within the SDP problems for each branch, the evolution of the reachable sets in each branch to each other is supported, such that a branching procedure is likely followed by a merging procedure.

7.4 Discussion of the Method

Many physical phenomena can be mathematically modeled by nonlinear difference equations in discrete time, and the robust control of such nonlinear systems is still subject of ongoing research. This section introduces a method based on reachability computation to find a suitable control law to transfer a piece-wise affine probabilistic system into a target set. PWAPS can be obtained by a piece-wise linearization of a nonlinear system in different regions of the state space, such that different dynamics are valid in different parts of the state space. Hence, the presented control algorithm in 7.3 is suitable for a control problem involving nonlinear dynamics.

The general formulation of the control problem in Problem 7.1, and the optimization problem in (7.16) are very similar to the formulations in the previous chapters. The main challenge of the presented approach is the computation of the ellipsoidal reachable set computation in a subdivided state space. As explained before, the reachable set may intersect the boundary between two regions, where two different dynamics are valid for the contained states in the reachable ellipsoids. The proposed approach to retain the ellipsoidal set formulation is the Push Algorithm 7.1, which first tries to push the ellipsoid into an intersecting region, and, if this attempt fails, branching is unavoidable. The SDP is then solved for each intersecting region, and from thereon different branches for the evolution of the state exist. This method allows a further usage of the ellipsoidal set formulation. Hence, a continued use of the optimization problem is possible, since stability properties and the input constraint satisfaction are already included therein. But the drawback of this approach is twofold: first, the computational complexity for the controller synthesis increases with every branch, since the SDP has to be solved in each branch, and second, even though only a part of the intersecting reachable set is inside a region, the solution of the SDP is computed, as if the whole set is inside the intersecting region. This may result in a certain conservatism of the controller input, because for each new branch a greater set is used for the controller synthesis and the actual control inputs are applied only to the states inside the region. This trade-off in performance and computational effort has to be made for a consistent computation with ellipsoids for the reachable sets. A trivial approach to avoid the conservatism and derive a controller only for the actual states contained in a region, could be to over-approximate the partial ellipsoids by new ellipsoids. Concerning the set computation, this approach might be reasonable, but within the context of probabilistic reachable sets, the probabilistic interpretation would get lost. The presented branch approach allows a further probabilistic interpretation of the reachable sets in the different branches, since to each branch an occurrence probability is assigned for the considered branch/set. In this way, the original distribution for the state is maintained and used for the further evolution.

In addition to the push-and-branch procedure, this chapter suggests a merging procedure in order to reduce the mentioned computational complexity, stemming from multiple applications of the branch procedure. The merging procedure is based on the assumption, that two very similar distributions of different branches with close expected values and similar entries in the covariance matrices will result in similar SDP's, and therefore, will most likely result in the same control parameters. This motivates an approach, in which the different but similar distributions of two branches are merged, and the optimization problem is solved for the resulting distribution only once. This is a possibility to reduce the computational effort, if branching was required before during the execution of Algorithm 7.3.

A core component of the merging procedure is the evaluation of the similarity of two distributions. In literature, different parameters can be found for a distance measure between probabilistic distributions, namely the Bhattacharyya distance, Kulback-Leibler divergence, and the Hellinger distance (all related to each other). This thesis follows the proposition in [66], and uses the Bhattacharyya distance, because of its simplicity for normal distributions. The control synthesis procedure is based on a computation of probabilistic reachable sets with confidence δ, and these probability statement for the merged distribution has to be computed cautiously. The merge algorithm checks, if the probabilistic reachable set of the resulting distribution contains the two reachable ellipsoids of the two merged distributions. This is a very important condition to uphold the probability interpretation of ”set with confidence δ“. In fact, if containment applies, the probability is over-approximated, since the new reachable ellipsoid might include parts, which are in none of the merged confidence sets. The merge procedure is, in this thesis, only intended for distributions in different branches at the same time step k. But it is conceivable that the similarity of different distributions also applies for different time steps and merging could be reasonable. This case is notably challenging, since it has to be defined, which time index is valid for the new, merged distribution. Furthermore, this may lead to a difference greater then one in the time index k, which has to be clarified first, in order to merge distributions of different time steps.

The outlined case of merging distributions with different time indices is not pursued in this thesis and may be subject of future research directions.

8 Conclusion

This thesis has proposed control synthesis procedures based on reachability computations for different types of SHS. The considered control problem is a set-to-target control problem, and solved by an algorithmic approach for each considered sub-class of SHS. Although an individual control algorithm is presented for each sub-class, the common core of all algorithms is the optimization-based computation of a stabilizing control law for the continuous input u_k. The computation of the one-step reachable set, i.e. the set of states reachable at the next time step starting from the current set of states, is formulated as an optimization problem. The LMI's in the optimization problem over-approximate either the resulting reachable set for deterministic AS, or over-approximate the resulting covariance matrix for the next time step for the probabilistic systems (APS, SAPS, PWAPS). Furthermore, ellipsoidal input constraints or polytopic input constraints can be included in the optimization based controller synthesis.

In general, there is a fundamental difference between the considered reachable sets for AS, and the reachable sets with confidence δ for APS, SAPS, and PWAPS. The reachable sets for AS are determined by the Minkowski sum of two or more sets, and consequently the considered disturbances has to be considered by a bounded set(ellipsoid). The successful termination of the control algorithm provides a control law, which satisfies the input constraint, and is robust against the considered disturbances. The control approach for linear affine systems is extended to nonlinear affine systems, as well, but the additional linearization error impedes the convergence of the reachable ellipsoids in size. Although, the algorithm includes an attempt to reduce the size of the linearization error, the ellipsoidal representation of the linearization error is a remaining bottleneck of the control approach for NADS.

The usage of reachable sets with confidence δ for APS, SAPS, and PWAPS provides a completely different interpretation of the resulting control law, and reachable sets. Since the initial state and the disturbance are assumed to be normally distributed, and hence its PDF is defined on the whole state space, an attempt to compute the reachable set with the Minkowski sum fails. Instead, a confidence set is introduced, which contains the realizations of the stochastic variable with a certain probability/confidence. It is shown, that these confidence sets can be described by ellipsoids, such that the LMI formulation for the input constraints can be directly adopted from AS. Furthermore, by exploiting the fact, that the sum of two normally distributed variables is again normally distributed, an LMI for the next covariance matrix can be formulated. The formulation of the LMI for the next covariance matrix requires less optimization variables compared to the formulation of the Minkowski sum, and hence the lesser computational demand is beneficial. A feasible solution of the SDP in each iteration of the algorithms provides a time-varying control law, which leads to a desired closed-loop continuous dynamics of the corresponding sub-class. The reachable confidence sets contain the current state, which is a normally distributed stochastic variable, with a probability of δ at each time step. A successful termination of the presented control algorithms includes a feasible solution of the SDP at each time step, and the obtained control law transfers the initial reachable confidence set into the target set. It holds, that the target set contains the realization of the system state at least with confidence δ.

The introduction of confidence reachable ellipsoids motivates the consideration of state chance constraints, since strict state constraints can never be satisfied for normally distributed states. This thesis extends a promising approach, which makes use of Boole's inequality, and discusses different methods, known in literature, to incorporate chance constraints in an optimization problem. It is shown, that the suggested formulation is very efficient compared to the sampling based approaches, and less conservative compared to set-based approach. Although only addressed for APS, the chance constraints can be included in the controller synthesis for SAPS, and PWAPS, as well.

The computation of the confidence reachable sets for SAPS, and PWAPS is more challenging, due to the dynamics of the discrete mode z_k. This thesis proposes approaches for both, switched and switching systems, in which the change of the discrete mode is triggered externally by an additional input, or is triggered autonomously by the continuous system state. For both specifications of the discrete dynamics an algorithmic procedure is suggested. While the challenge for the switched dynamics (SAPS) is the determination of a feasible sequence of discrete modes, the challenge for switching dynamics (PWAPS) consists of a consistent computation of the reachable sets with ellipsoids. The possibility to choose a different discrete mode at each time step for switched systems motivates the consideration of a decision tree, and a feasible sequence is determined by the application of a tree search procedure. Since extensive literature for graph or tree search exists, this thesis suggests only a general search heuristic for the tree search. But it is exemplarily shown, that a well-balanced application of BFS and DFS provides a sufficiently good result.

In PWAPS, the state space is partitioned into a finite number of regions, and in each region a different continuous dynamics is valid for the system state. This is, first of all, not problematic as long as the confidence reachable set is completely contained in one of the regions. But as soon as a confidence reachable intersects with more than one region, caution is required. The SDP-based controller synthesis assumes one valid system dynamics for all states in the confidence set, but for the intersecting case the discrete mode can be chosen from a set of potential modes. The consecutive computation of the reachable sets is a big challenge, and this thesis proposes an approach, which allows a consistent computation with ellipsoids without loss of the probabilistic interpretation of the reachable sets. A naive approach could be to over-approximate each intersecting part of the confidence reachable set with a new ellipsoid, and to solve the optimization problem for each new ellipsoid. But this leads to a loss of the probabilistic interpretation, since the shape matrix of the added ellipsoids do not stem from a covariance matrix, as it is true for the confidence reachable sets. This motivates the introduced push-and-branch procedure. The increasing computational effort by each branching procedure is addressed with the merging procedure. This is approach is firstly reported in this thesis, and it merges two branches, if the distribution are very similar. The merging of different distributions results in a reduction of branches, and hence a reduction of the computational effort.

Although the consideration of stochastic effects may seem to impede the controller synthesis, it actually facilitates it. The control design is not only less demanding compared to the robust approach for AS, due to the simpler formulation of the next covariance matrix, but the design parameter δ for the confidence can also be used to influence of the outcome of the algorithm.

In general, the proposed algorithms in this thesis are supposed to be executed off-line to determine a robust control law for an on-line application. However, a guarantee for a successful termination of the algorithms cannot be provided, since the solution of the stated SDP might fail for a certain parametrization. For example, the linearization error might get to large, such that the convergence constraint cannot be satisfied by the SDP solver, or the parametrization of the flexible Lyapunov function might be ill posed. Each algorithm has a set of parameters, which has to be chosen before the execution of the algorithm, and, without mentioning all of them in detail, a bad choice can lead to a termination without success. But a failed execution of an algorithm does not consequently mean, that the set-to-target control problem is infeasible. It rather requires a new attempt with a new set of parameters, and an investigation of the results of the failed execution can be helpful for the new set. For example, the confidence parameter δ could be reduced, if admissible by the application, in order to obtain smaller confidence sets.

In summary, this thesis suggests an off-line control synthesis procedure based on reachability computations for different variants of SHS, using an efficient representation of the sets. Furthermore, the proposed methods do not require a complete discretization of the state and input space, which is huge reduction of the computational effort, compared to the mentioned approaches in Sec. 2.1.

Future Research Directions

This thesis proposes different control approaches for different sub-classes of SHS, and while some open question are closed, several new questions arose. These remaining open questions and promising research directions are as follows:

- As already mentioned, the over-approximation of the Lagrange remainder first by an hyper-interval and afterwards by an ellipsoid is conservative. A direct over-approximation by an ellipsoid would remove some conservatism and reduce the complexity of the optimization problem for NADS.

- For SAPS, the suggested tree search procedure uses only the value of the cost function in order to evaluate the corresponding node of the tree. An extension to this procedure could include some additional information of the continuous state, e.g. the position and size of the confidence ellipsoid. The tree search heuristic could process this additional information, and potentially reduce the number of nodes in the waiting list $\mathcal{O}_W$, and reduce the computational effort to solve the discrete optimization.
- The presented algorithms require a-priori sets of parameters for the execution, and for some parameters no general default values are available. For instance, the evaluation of the similarity of two distributions by the Bhattacharyya distance is certainly possible, but practically, a default value is not provided. It would be helpful to provide a procedure, which specifies each parameter required for the considered control algorithm.
- The assumption of normally distributed stochastic variables is fundamental for the formulation of the confidence reachable sets. The extension of the control approaches to arbitrary distributions is an interesting, but very challenging ambition.

List of Symbols

Abbreviations

APS	affine probabilistic system
AS	affine system
CDF	cumulative distribution function
HS	hybrid system
LLA-CC	Local linear approximation of chance constraint
LMI	linear matrix inequality
MPT	multi parametric toolbox
NADS	nonlinear affine disturbed system
ODE	ordinary differential equation
PDF	probability density function
PWAPS	piecewise affine probabilistic system
SAPS	switched affine probabilistic system
SB-CC	Scenario-based evaluation of chance constraint
SBE-CC	Set-based evaluation of chance constraint
SBMI-CC	Scenario-based evaluation of chance constraint by mixed-integer formulation
SHS	stochastic hybrid system

Functions

$\kappa : \mathbb{R}^n \times \mathbb{N}_0 \mapsto \mathbb{R}^m$	generic control law
$\Lambda()$	returns the set of eigenvalues for a matrix

$\phi_c : \mathbb{R}^n \times \mathbb{N}_0 \mapsto \mathbb{R}^m$	continuous control law
$\phi_d : \mathbb{N}_0 \mapsto Z$	switching logic for the discrete mode z_k
$\psi_i : \mathbb{R}^n_o \mapsto \mathbb{R}$	i-th constraint function
$centroid : 2^{\mathbb{R}^n} \mapsto \mathbb{R}^n$	the function provides the center point for an arbitrary set
D_B	Bhattacharyya distance measures the similarity of two probability distributions.
$D_H : 2^{\mathbb{R}^n} \times \mathbb{R}^n \mapsto \mathbb{R}$	shortest distance between a point and a half-plane
$f : \mathbb{R}^n \times \mathbb{R}^m \times Z \mapsto \mathbb{R}^n$	difference equation of nonlinear dynamic system
$F : \mathbb{R}^n \times \mathbb{R}^m \times Z \mapsto \mathbb{R}^n$	set valued evaluation of difference equation of nonlinear dynamic system
$F_{\chi^2} : \mathbb{R}_{\geq 0} \times \mathbb{N} \mapsto [0,1]$	cumulative distribution function of a χ^2 distribution
$f_{\chi^2} : \mathbb{R}_{\geq 0} \times \mathbb{N} \mapsto [0,1]$	probability density function of a χ^2 distribution
$f_\Gamma : \mathbb{R}^n \mapsto \mathbb{R}$	gamma function
$f_\gamma : \mathbb{R}^n \times \mathbb{R}^n \mapsto \mathbb{R}$	incomplete gamma function
$\bar{F}_{N,appr.} : \mathbb{R}^n \mapsto [0,1]$	Taylor approximation of a uni-variate Gaussian distribution
$F_{N,appr.} : \mathbb{R} \mapsto [0,1]$	approximation of the cumulative distribution function of a univariate Gaussian distribution
$F_N : 2^{\mathbb{R}^n} \mapsto [0,1]$	cumulative distribution function of a multivariate Gaussian distribution
$f_N : \mathbb{R}^n \mapsto [0,1]$	probability density function of a multivariate Gaussian distribution
$\tilde{g} : \mathbb{R}^{m\times m} \mapsto \mathbb{R}^{m\times m}$	linearization of non-convex constraint function $g(\cdot)$ for the ellipsoidal input constraint
$g : \mathbb{R}^{m\times m} \mapsto \mathbb{R}^{m\times m}$	non-convex constraint function for the ellipsoidal input constraint
$getAdjPart(x_k) : \mathbb{R}^n \mapsto 2^Z$	the function provides all adjacent partition elements to the continuous state vector x_k
$getIntReg : 2^{\mathbb{R}^n} \times \bar{\Theta} \mapsto 2^Z$	returns the intersecting regions for a probabilistic reachable set
$H : \mathbb{R}^n \mapsto [0,1]$	probability of violating a chance constraint

$intval : 2^{\mathbb{R}^n} \mapsto \mathcal{I}$	the function provides an n-dimensional interval for an arbitrary set
J_k	objective function in optimization problem
$L : \mathbb{R}^{n+m} \times \mathbb{R}^{n+m} \mapsto \mathbb{R}^n$	Lagrange remainder
$SetProb : \mathcal{E} \times \mathcal{P} \rightarrow [0,1]$	returns the current share ϵ_{RS} for a given confidence ellipsoid X_k^δ and region $\Theta^{(i)}$
$V : \mathbb{R}^n \mapsto \mathbb{R}$	quadratic Lyapunov function

General

$\lambda_{min}(\cdot), \lambda_{max}(\cdot)$	min. and max. eigenvalue of a matrix
τ	bounded time domain $\tau := \{0, 1, \dots, N\}$
k	discrete time k
$\mathcal{N}(q, Q)$	multivariate normal distribution with mean q and covariance matrix Q

Scalars and Constants

α_h	slack variable for evaluation of the Lagrange remainder
α_k	slack variable for the flexible Lyapunov function
α_P	slack variable for the linearization of the non-convex matrix inequality
β	confidence of sampled approximation of a distribution
δ	confidence level of reachable set
δ_x	probability value for a chance constraint
$\epsilon_{RS,\gamma_{k,i}}$	occurrence probability for the probabilistic reachable set X_k^δ in node $\gamma_{k,i}$
$\epsilon_{vio,i}$	probability of violating the $i-$th half plane in the state constraint $\mathbb{X}_k$
μ	weight for the value function in the optimization problem
ν	auxiliary weighting variable for the Minkowski addition of two ellipsoids:$\nu \in [0,1]$

ω	coupling vector for flexible Lyapunov function
π_k	decreasing rate of the mean vector $q_{x,k}$
π_{min}	lower bound for the decreasing rate of the mean vector $q_{x,k}$
ρ	threshold for decrease of the Lyapunov function
ϑ	quality of a sampled approximation of a distribution
c_ξ	scaling factor for the reachable set with confidence δ of a random variable ξ
k_π	first time step k, for which the decreasing criterion is not satisfied
m	dimension of input vector: $u_k \in \mathbb{R}^m$
N	finite horizon
n	dimension of state vector: $x_k \in \mathbb{R}^n$
n_c	number of constraints in an optimization problem
n_o	dimension of the optimization variable η
n_p	number of hyper spaces defining the H-representation of a polytope
n_r	number of vertices defining the V-representation of a polytope
n_T	number of nodes in a tree Γ: $n_T + 1$
n_T	number of nodes in a tree structure Γ
n_u	number of half-planes defining the polytopic input set U_P
n_v	dimension of the disturbance v
n_x	number of half-planes defining the state constraint $\mathbb{X}$
n_z	number of discrete modes z_k
$n_{\alpha,P}$	number of linearization points for the ellipsoidal input formulation
$N_{p,SBMI}$	number of scenarios of a random variable for SBMI-CC
$N_{p,SB}$	number of scenarios of a random variable for SB-CC

s	weighting variable for the Minkowski addition of two ellipsoids: $s \geq 0$

Sets

$\partial(\cdot)$	boundary of a set
$\{\,\}$	discrete set
$int(\cdot)$	interior of a set
$\lfloor \Psi \rceil$	n-dimensional interval of hybrid state and input set: $\Psi \subseteq \lfloor \Psi \rceil$
Ψ	hybrid state and input set: $\Psi = X_k \times U$
$\bar{\Theta}$	set of partition elements
$\Theta^{(i)}$	partition element: $\Theta^{(i)} \in \bar{\Theta}$
Ξ	countable set of stochastic events
$\mathcal{E}$	set of all ellipsoids
$\mathcal{I}$	set of all n-dimensional intervals
$\check{L}_E$	reduced over-approximation of the Lagrange remainder
L_{box}	over-approximation of the Lagrange remainder with a hyper-interval
L_E	over-approximation of the Lagrange remainder with an ellipsoid: $L_{box} \subseteq L_E$
$\mathbb{N}, \mathbb{N}_0$	set of natural numbers, set of natural numbers including 0
$\mathcal{P}$	set of all polytopes
$\mathbb{R}$	set of real numbers
$\mathbb{R}^n$	set of real vectors with n elements
$\mathbb{R}_{>0}, \mathbb{R}_{\geq 0}$	set of positive real numbers, set of non-negative real numbers
$\mathbb{T}$	terminal constraint: $\mathbb{T} \subset \mathbb{R}^n$
$\bar{U}_k$	feasible input: $\bar{U}_k \subseteq U_{P/E}$
$\check{U}_k$	over-approximation of feasible input: $\bar{U}_k \subseteq \check{U}_k \subseteq U$

$\lfloor \check{U}_k \rceil$ n-dimensional interval of over-approximation of feasible input: $\check{U}_k \subseteq \lfloor \check{U}_k \rceil$

U generic input constraint (polytopic or ellipsoidal): $U \subseteq \mathbb{R}^m$

V disturbance set

W arbitrary closed and compact set

$\hat{X}_k$ ellipsoidal reachable set at time step k

$\lfloor \hat{X}_k \rceil$ n-dimensional interval of ellipsoidal reachable set at time step k:$\hat{X}_k \subseteq \lfloor \hat{X}_k \rceil$

$\lfloor U \rceil$ hyper-interval of generic input constraint: $U \subseteq \lfloor U \rceil$

$\mathbb{X}_k$ state constraint at time k: $\mathbb{X}_k \subseteq \mathbb{R}^n$

X_k reachable set at time step k

X_k^δ ellipsoidal reachable set with confidence delta at time step k

Z set of discrete states

$Z_{adj,k}$ set of states with adjacent partition elements

$Z_{intsec,k}$ set of states intersecting with the confidence set at time k

Tree Structure

Γ tree structure, consisting of nodes γ_i, with $i = \{0, \ldots, n_T\}$

γ_i node of a tree

$activeNode : \Gamma \times \mathbb{N}_0 \times Z \to \Gamma$ returns the current active node γ_c for a given time step k and discrete mode z_k

$CostVal : \Gamma \to \mathbb{R}$ cost value of the SDP

$CtrlPrm : \Gamma \to \mathbb{R}^{m \times n} \times \mathbb{R}^m$ control tuple (K_k, d_k)

$Mode : \Gamma \to Z$ chosen mode z_{k-1}

$NgetIntReg : 2^\Gamma \to 2^Z \times \mathbb{R}^m$ set of discrete modes $Z_{int,k}$ of intersecting regions

$NSetProb : \Gamma \to [0, 1]$ returns the current share ϵ_{RS} for a given node γ_i

$Pre : 2^\Gamma \to 2^\Gamma$ predecessor node

$ReachSet : 2^\Gamma \to 2^{\mathbb{R}^n}$ current reachable set

$Succ : 2^{\Gamma} \to 2^{\Gamma}$ set of successor nodes

$TimeStep : \Gamma \to \mathbb{N}_0$ time step k

$TSH : \mathcal{O}_W \to \mathcal{O}_C$ tree search heuristic, which specifies the set of nodes used for exploration

$\mathcal{O}_C$ set of current nodes, to be used for further exploration of the tree in the current iteration

$\mathcal{O}_P$ set of past nodes, which are not further considered in the search procedure

$\mathcal{O}_W$ set of waiting nodes, to be considered for further exploration of the tree in consecutive search iteration

Vectors and Matrices

Δ arbitrary symmetric matrix

η optimization variable: $\eta \in \mathbb{R}^{n_o}$

ϕ_k feasible discrete mode $\phi_k \in Z$ at time step k

σ χ^2 random variable

θ variable for the definition of a unit ball

ξ multivariate random variable

$\bar{\zeta}$ linearization point for the extended system vector

ζ_k extended system vector containing the state and input vector x_k, and u_k

$A_{(k)}, B_{(k)}, G_{(k)}$ matrices defining affine dynamics (possibly time-varying, indicated by the subscript k)

$A_{cl,k}$ matrix for closed-loop dynamics

d_k affine control vector at time k

e_i i-th unit vector

h_b binary vector for big-M formulation mixed-integer program

h_L hybrid slack vector for evaluation of the Lagrange remainder: $h_L \in \lfloor \Psi_k \rceil$

I_n	identity matrix: $I_n \in \mathbb{R}^{n \times n}$
K_k	control feedback matrix at time k
M	positive semi-definite matrix, defining a Lyapunov function $V(\cdot)$
$\bar{P}_k$	over-approximating shape matrix of feasible input ellipsoid
$\bar{p}_{k,jh}$	matrix element of $\bar{P}_k$ in j-th row and h-th column
$\tilde{P}_k$	auxiliary matrix for ellipsoidal input constraint formulation
$\tilde{p}_{k,jh}$	matrix element of $\tilde{P}_k$ in j-th row and h-th column
p, P	center point and shape matrix of ellipsoidal input constraint:$U := \varepsilon(p, P)$
$p^{(i)}$	i-th vertex, defining a polytope.
P_L	linearization point for convexification of non-convex matrix inequality
$p_{L,jh}$	matrix element of P_L in j-th row and h-th column
$\bar{Q}$	bounding matrix for the covariance matrix $Q_{x,k}$
$\bar{q}$	bounding vector for the mean vector $q_{x,k}$
$\bar{Q}_y$	Linearization point of covariance matrix of the random variable y
$\bar{q}_y$	Linearization point of expected value of the random variable y
Q_y	covariance matrix of the random variable y
q_y	expected value of the random variable y
Q_ξ	covariance matrix of the random variable ξ
q_ξ	expected value of the random variable ξ
$Q_{L,k}$	shape matrix of ellipsoid containing the over-approximation of the Lagrange remainder $L_E(\bar{\zeta}_k)$
$q_{L,k}$	center point of ellipsoid containing the over-approximation of the Lagrange remainder $L_E(\bar{\zeta}_k)$
Q_T	shape matrix of ellipsoidal terminal set $\mathbb{T} \in \mathcal{E}$

q_T	center point of ellipsoidal terminal set $\mathbb{T} \in \mathcal{E}$
Q_v	covariance matrix of the disturbance set V
q_v	expected value of the random variable v
$Q_{x,k}$	covariance matrix of the random variable x_k at time k
$q_{x,k}$	expected value of the random variable x_k and center point of the reachable ellipsoid at time k
$Q^{\delta}_{x,k}$	shape matrix of the reachable ellipsoid with confidence δ
$R_{(\cdot)}, b_{(\cdot)}$	matrix and vector defining a polytope; the subscript determines whether it is state polytope (x_k) or input polytope (u_k)
r_i, b_i	i-th row in matrix R, and i-th entry in b
S_k	over-approximating matrix of the covariance matrix $Q_{x,k}$ at time k
$\bar{u}$	equilibrium point for the input vector
u_k	input of the system at time k
v_k	disturbance at time k
$v_k^{(i)}$	sampled scenario of the disturbance at time k
w	element of the closed and compact set W: $w \in W$
$\bar{x}$	equilibrium point for the continuous state
x_k	continuous state at time k
$x_k^{(i)}$	sampled scenario of the continuous state at time k
y_k	uni-variate auxiliary variable for the evaluation of the chance constraint
$\bar{z}$	equilibrium point for the discrete mode
z_k	discrete mode at time k

References

[1] A. Abate, “Probabilistic reachability for stochastic hybrid systems: Theory, computations, and applications,” Ph.D. dissertation, University of California, Berkeley, 2007.

[2] A. Abate, A. D’Innocenzo, and M. D. D. Benedetto, “Approximate abstractions of stochastic hybrid systems,” *IEEE Transactions on Automatic Control*, vol. 56, no. 11, pp. 2688–2694, Nov 2011.

[3] A. Abate, S. Amin, M. Prandini, J. Lygeros, and S. Sastry, “Probabilistic reachability and safe sets computation for discrete time stochastic hybrid systems,” in *Proceedings of the 45th IEEE Conference on Decision and Control*. IEEE, 2006, pp. 258–263.

[4] A. Abate, A. D’Innocenzo, M. Di Benedetto, and S. Sastry, “Markov set-chains as abstractions of stochastic hybrid systems,” *Hybrid Systems: Computation and Control*, pp. 1–15, 2008.

[5] A. Abate, J.-P. Katoen, J. Lygeros, and M. Prandini, “Approximate model checking of stochastic hybrid systems,” *European Journal of Control*, vol. 16, no. 6, pp. 624–641, 2010.

[6] A. Abate, M. Prandini, J. Lygeros, and S. Sastry, “An approximate dynamic programming approach to probabilistic reachability for stochastic hybrid systems,” in *47th IEEE Conference on Decision and Control*. IEEE, 2008, pp. 4018–4023.

[7] A. Abate, M. Prandini, J. Lygeros, and S. Sastry, “Probabilistic reachability and safety for controlled discrete time stochastic hybrid systems,” *Automatica*, vol. 44, no. 11, pp. 2724–2734, 2008.

[8] M. Abramowitz and I. A. Stegun, *Handbook of mathematical functions: with formulas, graphs, and mathematical tables*. Courier Corporation, 1964, vol. 55.

[9] J. Ackermann, *Robust control: the parameter space approach*. Springer Science & Business Media, 2012.

[10] E. E. Alessandro Soranzo, “Very simply explicit invertible approximations of normal cumulative and normal quantile function,” *Applied Mathematical Science*, vol. 8, no. 87, pp. 4323–4341, 2014.

[11] M. Althoff, "An introduction to CORA 2015," in *Proc. of the Workshop on Applied Verification for Continuous and Hybrid Systems*, 2015, pp. 120–151.

[12] M. Althoff, "Reachability analysis and its application to the safety assessment of autonomous cars," Ph.D. dissertation, Technische Universitaet Muenchen, 2010.

[13] M. Althoff, "On computing the minkowski difference of zonotopes," *Journal of Computational Geometry*, 2015.

[14] M. Althoff and B. H. Krogh, "Zonotope bundles for the efficient computation of reachable sets," in *50th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC),*. IEEE, 2011, pp. 6814–6821.

[15] M. Althoff, O. Stursberg, and M. Buss, "Reachability analysis of nonlinear systems with uncertain parameters using conservative linearization," in *47th IEEE Conf. on Decision and Control*, 2008, pp. 4042–4048.

[16] S. Amin, A. Abate, M. Prandini, J. Lygeros, and S. Sastry, "Reachability analysis for controlled discrete time stochastic hybrid systems," in *International Workshop on Hybrid Systems: Computation and Control.* Springer, 2006, pp. 49–63.

[17] T. M. Apostol, *Calculus, Volume 1. One-variable calculus with an introduction to linear algebra.* Wiley, New York, 1975.

[18] G. M. Appa, L. S. Pitsoulis, and H. P. Williams, *Handbook on modelling for discrete optimization.* Springer Science & Business Media, 2006, vol. 88.

[19] M. ApS, *The MOSEK optimization toolbox for MATLAB manual. Version 7.1 (Revision 53).*, 2016. [Online]. Available: http://docs.mosek.com/7.1/toolbox/index.html

[20] E. Asarin, O. Bournez, T. Dang, and O. Maler, "Approximate reachability analysis of piecewise-linear dynamical systems," in *Hybrid Systems: Computation and Control.* Springer, 2000, pp. 20–31.

[21] L. Asselborn, D. Gross, and O. Stursberg, "Control of uncertain nonlinear systems using ellipsoidal reachability calculus," in *9th IFAC Symp. on Nonlinear Control Systems*, 2013, pp. 50–55.

[22] L. Asselborn, M. Jilg, and O. Stursberg, "Control of uncertain hybrid nonlinear systems using particle filters," in *4th IFAC Conf. on Analysis and Design of Hybrid Systems*, 2012, pp. 436–441.

[23] L. Asselborn and O. Stursberg, "Probabilistic control of uncertain linear systems using stochastic reachability," *8th IFAC Symp. on Robust Control Design*, vol. 48, no. 14, pp. 167–173, 2015.

[24] L. Asselborn and O. Stursberg, "Robust control of uncertain switched linear systems based on stochastic reachability," in *5th IFAC Conf. on Analysis and Design of Hybrid Systems*, 2015.

[25] L. Asselborn and O. Stursberg, "Control of discrete-time piece-wise affine probabilistic systems using reachability analysis," in *IEEE Conference on Computer Aided Control System Design (CACSD)*, 2016, p. 661–666.

[26] L. Asselborn and O. Stursberg, "Probabilistic control of switched linear systems with chance constraints," in *15th European Control Conference*, 2016.

[27] E. August, J. Lu, and H. Koeppl, "Trajectory enclosures for nonlinear systems with uncertain initial conditions and parameters," in *American Control Conf.*, 2012, pp. 1488–1493.

[28] R. E. Bellman and S. E. Dreyfus, *Applied dynamic programming.* Princeton university press, 2015.

[29] A. Bemporad and S. Di Cairano, "Optimal control of discrete hybrid stochastic automata," in *International Workshop on Hybrid Systems: Computation and Control.* Springer, 2005, pp. 151–167.

[30] D. Bernardini and A. Bemporad, "Scenario-based model predictive control of stochastic constrained linear systems," in *Decision and Control, 2009 held jointly with the 2009 28th Chinese Control Conference. CDC/CCC 2009. Proceedings of the 48th IEEE Conference on.* IEEE, 2009, pp. 6333–6338.

[31] D. P. Bertsekas, *Dynamic programming and optimal control.* Athena Scientific Belmont, MA, 1995, vol. 1, no. 2.

[32] A. Bhattacharyya, "On a measure of divergence between two multinomial populations," *Sankhyā: The Indian Journal of Statistics (1933-1960)*, vol. 7, no. 4, pp. 401–406, 1946.

[33] L. Blackmore and M. Ono, "Convex chance constrained predictive control without sampling," in *Proceedings of the AIAA Guidance, Navigation and Control Conference*, 2009, pp. 7–21.

[34] L. Blackmore, M. Ono, A. Bektassov, and B. C. Williams, "A probabilistic particle-control approximation of chance-constrained stochastic predictive control," *IEEE Transactions on Robotics*, vol. 26, no. 3, pp. 502–517, 2010.

[35] L. Blackmore, M. Ono, and B. C. Williams, "Chance-constrained optimal path planning with obstacles," *IEEE Transactions on Robotics*, vol. 27, no. 6, pp. 1080–1094, 2011.

[36] L. J. C. Blackmore, "Robust execution for stochastic hybrid systems," Ph.D. dissertation, Massachusetts Institute of Technology, 2007.

[37] F. Blanchini and S. Miani, *Set-Theoretic Methods in Control.* Springer, 2008.

[38] H. A. Blom and J. Lygeros, *Stochastic hybrid systems: theory and safety critical applications.* Springer, 2006, vol. 337.

[39] O. Botchkarev and S. Tripakis, "Verification of hybrid systems with linear differential inclusions using ellipsoidal approximations," in *International Workshop on Hybrid Systems: Computation and Control.* Springer, 2000, pp. 73–88.

[40] O. Bournez, O. Maler, and A. Pnueli, "Orthogonal polyhedra: Representation and computation," in *Hybrid Systems: Computation and Control.* Springer, 1999, pp. 46–60.

[41] S. P. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan, *Linear matrix inequalities in system and control theory.* SIAM, 1994, vol. 15.

[42] S. Boyd and L. Vandenberghe, *Convex optimization.* Cambridge university press, 2004.

[43] L. Bujorianu, *Stochastic reachability analysis of hybrid systems.* Springer, 2012.

[44] G. Calafiore and M. C. Campi, "Uncertain convex programs: randomized solutions and confidence levels," *Mathematical Programming*, vol. 102, no. 1, pp. 25–46, 2005.

[45] G. C. Calafiore, "Random convex programs," *SIAM Journal on Optimization*, vol. 20, no. 6, pp. 3427–3464, 2010.

[46] C. G. Cassandras and J. Lygeros, *Stochastic hybrid systems.* CRC Press, 2006.

[47] M. Chen, C.-Y. Shih, and C. J. Tomlin, "Multi-vehicle collision avoidance via hamilton-jacobi reachability and mixed integer programming," *arXiv preprint arXiv:1603.05200*, 2016.

[48] X. Chen, E. Ábrahám, and G. Frehse, "Efficient bounded reachability computation for rectangular automata," in *Reachability Problems.* Springer, 2011, pp. 139–152.

[49] A. Chutinan and B. H. Krogh, "Verification of polyhedral-invariant hybrid automata using polygonal flow pipe approximations," in *International workshop on hybrid systems: computation and control.* Springer, 1999, pp. 76–90.

[50] E. A. Cross and I. M. Mitchell, “Level set methods for computing reachable sets of systems with differential algebraic equation dynamics,” in *American Control Conference, 2008.* IEEE, 2008, pp. 2260–2265.

[51] T. Dang, “Verification and synthesis of hybrid systems,” Ph.D. dissertation, Institut National Polytechnique de Grenoble-INPG, 2000.

[52] T. Dang and O. Maler, “Reachability analysis via face lifting,” in *Hybrid Systems: Computation and Control.* Springer, 1998, pp. 96–109.

[53] M. Dehghan and C.-J. Ong, “Characterization and computation of disturbance invariant sets for constrained switched linear systems with dwell time restriction,” *Automatica*, vol. 48, no. 9, pp. 2175–2181, 2012.

[54] M. Dehghan and C.-J. Ong, “Discrete-time switching linear system with constraints: Characterization and computation of invariant sets under dwell-time consideration,” *Automatica*, vol. 48, no. 5, pp. 964–969, 2012.

[55] J. Ding, “Methods for reachability-based hybrid controller design,” Ph.D. dissertation, University of California, Berkeley, 2012.

[56] J. Ding, A. Abate, and C. Tomlin, “Optimal control of partially observable discrete time stochastic hybrid systems for safety specifications,” in *2013 American Control Conference.* IEEE, 2013, pp. 6231–6236.

[57] J. Ding, M. Kamgarpour, S. Summers, A. Abate, J. Lygeros, and C. Tomlin, “A stochastic games framework for verification and control of discrete time stochastic hybrid systems,” *Automatica*, vol. 49, no. 9, pp. 2665–2674, 2013.

[58] G.-R. Duan and H.-H. Yu, *LMIs in Control Systems: Analysis, Design and Applications.* CRC Press, 2013.

[59] T. E. Faber, *Fluid dynamics for physicists.* Cambridge University Press, 1995.

[60] T. F. Filippova, “Estimates of trajectory tubes of uncertain nonlinear control systems,” in *International Conference on Large-Scale Scientific Computing.* Springer, 2009, pp. 272–279.

[61] T. F. Filippova, “Estimates of reachable sets of control systems with nonlinearity and parametric perturbations,” *Trudy Instituta Matematiki i Mekhaniki UrO RAN*, vol. 20, no. 4, pp. 287–296, 2014.

[62] T. Filippova, “Differential equations for ellipsoidal estimates for reachable sets of a nonlinear dynamical control system,” *Proceedings of the Steklov Institute of Mathematics*, vol. 271, pp. 75–84, 2010.

[63] B. A. Francis and P. P. Khargonekar, *Robust control theory.* Springer Science & Business Media, 2012, vol. 66.

[64] G. Frehse, C. Le Guernic, A. Donzé, S. Cotton, R. Ray, O. Lebeltel, R. Ripado, A. Girard, T. Dang, and O. Maler, "Spaceex: Scalable verification of hybrid systems," in *Proc. 23rd International Conference on Computer Aided Verification (CAV).* Springer, 2011, pp. 379–395.

[65] G. Frehse, C. Le Guernic, A. Donzé, S. Cotton, R. Ray, O. Lebeltel, R. Ripado, A. Girard, T. Dang, and O. Maler, "Spaceex: Scalable verification of hybrid systems," in *Proc. 23rd International Conference on Computer Aided Verification (CAV)*, ser. LNCS. Springer, 2011.

[66] K. Fukunaga, *Introduction to statistical pattern recognition.* Academic press, 2013.

[67] C. Gardiner, *Stochastic methods.* Springer Berlin, 2009.

[68] A. Genz and F. Bretz, *Computation of multivariate normal and t probabilities.* Springer Science & Business Media, 2009, vol. 195.

[69] A. Girard, "Reachability of uncertain linear systems using zonotopes," in *Hybrid Systems: Computation and Control.* Springer, 2005, pp. 291–305.

[70] A. Girard, C. Le Guernic *et al.*, "Efficient reachability analysis for linear systems using support functions," in *Proc. of the 17th IFAC World Congress*, 2008, pp. 8966–8971.

[71] A. Girard, C. Le Guernic, and O. Maler, "Efficient computation of reachable sets of linear time-invariant systems with inputs," in *Hybrid Systems: Computation and Control.* Springer, 2006, pp. 257–271.

[72] R. Goebel, R. Sanfelice, and A. Teel, "Hybrid dynamical systems," *Control Systems, IEEE*, vol. 29, no. 2, pp. 28–93, 2009.

[73] M. R. Greenstreet and I. Mitchell, "Integrating projections," in *Hybrid Systems: Computation and Control.* Springer, 1998, pp. 159–174.

[74] M. R. Greenstreet and I. Mitchell, "Reachability analysis using polygonal projections," in *Hybrid Systems: Computation and Control.* Springer, 1999, pp. 103–116.

[75] L. Habets, P. J. Collins, and J. H. van Schuppen, "Reachability and control synthesis for piecewise-affine hybrid systems on simplices," *IEEE Trans. on Automatic Control*, vol. 51, no. 6, pp. 938–948, 2006.

[76] L. C. Habets and J. H. van Schuppen, “Control of piecewise-linear hybrid systems on simplices and rectangles,” in *Hybrid Systems: Computation and Control*. Springer LNCS, 2001, vol. 2034, pp. 261–274.

[77] E. V. Haynsworth, “Determination of the inertia of a partitioned hermitian matrix,” *Linear algebra and its applications*, vol. 1, no. 1, pp. 73–81, 1968.

[78] E. V. Haynsworth, “On the schur complement.” DTIC Document, Tech. Rep., 1968.

[79] M. Henk, “Löwner-john ellipsoids,” *Documenta Math*, pp. 95–106, 2012.

[80] M. Herceg, M. Kvasnica, C. Jones, and M. Morari, “Multi-Parametric Toolbox 3.0,” in *Proc. of the European Control Conference*, Zürich, Switzerland, July 17–19 2013, pp. 502–510, http://control.ee.ethz.ch/~mpt.

[81] W. Hoeffding, “Probability inequalities for sums of bounded random variables,” *Journal of the American statistical association*, vol. 58, no. 301, pp. 13–30, 1963.

[82] J. Hu, J. Lygeros, and S. Sastry, “Towards a theory of stochastic hybrid systems,” in *Hybrid systems: Computation and Control*. Springer, 2000, vol. 1790, pp. 160–173.

[83] J. Hu, M. Prandini, and S. Sastry, “Aircraft conflict detection in presence of spatially correlated wind perturbations,” in *AIAA Guidance, Navigation and Control Conf., Austin, Texas, USA*, vol. 448, 2003.

[84] M. Kamgarpour, J. Ding, S. Summers, A. Abate, J. Lygeros, and C. Tomlin, “Discrete time stochastic hybrid dynamical games: Verification controller synthesis,” in *50th IEEE Conference on Decision and Control and European Control Conference*. IEEE, 2011, pp. 6122–6127.

[85] M. Kamgarpour, S. Summers, and J. Lygeros, “Control design for specifications on stochastic hybrid systems,” in *Hybrid systems: computation and control*. ACM, 2013, pp. 303–312.

[86] N. Kariotoglou, S. Summers, T. Summers, M. Kamgarpour, and J. Lygeros, “Approximate dynamic programming for stochastic reachability,” in *2013 European Control Conference (ECC)*. IEEE, 2013, pp. 584–589.

[87] D. A. Kendrick, *Stochastic control for economic models*. McGraw-Hill New York, 1981.

[88] E. C. Kerrigan, “Robust constraint satisfaction: Invariant sets and predictive control,” Ph.D. dissertation, Dept. Eng., Univ. Cambridge, 2000.

[89] E. C. Kerrigan, J. Lygeros, and J. M. Maciejowski, “A geometric approach to reachability computations for constrained discrete-time systems,” in *Proc. 15th IFAC World Congress on Automatic Control, Barcelona, Spain*, 2002.

[90] E. C. Kerrigan and D. Q. Mayne, “Optimal control of constrained, piecewise affine systems with bounded disturbances,” in *41st IEEE Conf. on Decision and Control*, 2002, pp. 1552–1557.

[91] E. Kostousovat, “State estimation for dynamic systems via parallelotopes optimization and parallel computations,” *Optimization Methods and Software*, vol. 9, no. 4, pp. 269–306, 1998.

[92] E. Kostousovat, “Control synthesis via parallelotopes: optimzation and parallel compuations,” *Optimization methods and software*, vol. 14, no. 4, pp. 267–310, 2001.

[93] X. Koutsoukos and D. Riley, “Computational methods for reachability analysis of stochastic hybrid systems,” *Hybrid Systems: Computation and Control*, vol. -, pp. 377–391, 2006.

[94] F. Kozin, “A survey of stability of stochastic systems,” *Automatica*, vol. 5, no. 1, pp. 95–112, 1969. [Online]. Available: http://dx.doi.org/10.1016/0005-1098(69)90060-0

[95] W. J. Krzanowski and F. H. C. Marriott, *Multivariate Analysis: Distributions, Ordination and Inference*. Wiley Interscience, 1994, vol. 2.

[96] W. Kühn, “Rigorously computed orbits of dynamical systems without the wrapping effect,” *Computing*, vol. 61, no. 1, pp. 47–67, 1998.

[97] W. Kühn, “Zonotope dynamics in numerical quality control,” in *Mathematical Visualization*. Springer, 1998, pp. 125–134.

[98] A. A. Kurzhanski and P. Varaiya, “Ellipsoidal techniques for reachability analysis of discrete-time linear systems,” *IEEE Transactions on Automatic Control*, vol. 52, no. 1, pp. 26–38, 2007.

[99] A. A. Kurzhanski and P. Varaiya, “Reach set computation and control synthesis for discrete-time dynamical systems with disturbances,” *Automatica*, vol. 47, no. 7, pp. 1414–1426, 2011.

[100] A. B. Kurzhanski and I. Valyi, “Ellipsoidal calculus for estimation and control,” *Systems & control*, 1997.

[101] A. B. Kurzhanski and P. Varaiya, “Ellipsoidal techniques for reachability analysis,” *Hybrid Systems: Computation and Control*, vol. –, pp. 202–214, 2000.

[102] A. B. Kurzhanski and P. Varaiya, "Ellipsoidal techniques for reachability analysis: internal approximation," *Systems & control letters*, vol. 41, no. 3, pp. 201–211, 2000.

[103] A. B. Kurzhanski and P. Varaiya, "On ellipsoidal techniques for reachability analysis. part i: External approximations," *Optimization methods and software*, vol. 17, no. 2, pp. 177–206, 2002.

[104] A. B. Kurzhanski and P. Varaiya, "On ellipsoidal techniques for reachability analysis. part ii: Internal approximations box-valued constraints," *Optimization Methods and Software*, vol. 17, no. 2, pp. 207–237, 2002.

[105] A. B. Kurzhanski and P. Varaiya, "Reachability analysis for uncertain systems-the ellipsoidal technique," *Dynamics of continuous discrete and impulsive systems*, vol. 9, pp. 347–368, 2002.

[106] A. B. Kurzhanski and P. Varaiya, "Reachability under state constraints-the ellipsoidal technique," in *IFAC World Congress*, 2002.

[107] A. B. Kurzhanski and P. Varaiya, "Ellipsoidal techniques for hybrid dynamics: the reachability problem," *New Directions and Applications in Control Theory*, pp. 193–205, 2005.

[108] A. B. Kurzhanski and P. Varaiya, "On verification of controlled hybrid dynamics through ellipsoidal techniques," in *44th IEEE Conference on Decision and Control, and European Control Conference.* IEEE, 2005, pp. 4682–4687.

[109] A. B. Kurzhanski and P. Varaiya, "Ellipsoidal techniques for reachability under state constraints," *SIAM Journal on Control and Optimization*, vol. 45, no. 4, pp. 1369–1394, 2006.

[110] A. A. Kurzhanskiy and P. Varaiya, "Ellipsoidal toolbox (et)," in *2006 45th IEEE Conference on Decision and Control.* IEEE, 2006, pp. 1498–1503.

[111] A. A. Kurzhanskiy and P. Varaiya, "Ellipsoidal toolbox," EECS Department, University of California, Berkeley, Tech. Rep. UCB/EECS-2006-46, May 2006. [Online]. Available: http://code.google.com/p/ellipsoids

[112] A. A. Kurzhanskiy and P. Varaiya, "Reach set computation and control synthesis for discrete-time dynamical systems with disturbances," *Automatica*, vol. 47, no. 7, pp. 1414–1426, 2011.

[113] H. Kushner and P. G. Dupuis, *Numerical methods for stochastic control problems in continuous time.* Springer Science & Business Media, 2013, vol. 24.

[114] M. Lazar, "Flexible control lyapunov functions," in *Proceedings of the 2009 conference on American Control Conference*, 2009, pp. 102–107.

[115] C. Le Guernic and A. Girard, "Reachability analysis of hybrid systems using support functions," in *Computer Aided Verification*. Springer, 2009, pp. 540–554.

[116] C. Le Guernic and A. Girard, "Reachability analysis of linear systems using support functions," *Nonlinear Analysis: Hybrid Systems*, vol. 4, no. 2, pp. 250–262, 2010.

[117] K. Lesser and M. Oishi, "Reachability for partially observable discrete time stochastic hybrid systems," *Automatica*, vol. 50, no. 8, pp. 1989–1998, 2014.

[118] D. Liberzon, *Switching in systems and control*. Birkhaeuser, 2003.

[119] H. Lin and P. J. Antsaklis, "Switching stabilization and l2 gain performance controller synthesis for discrete-time switched linear systems," in *45th IEEE Conf. on Decision Control*, 2006, pp. 2673–2678.

[120] H. Lin and P. J. Antsaklis, "Hybrid h∞ state feedback control for discrete-time switched linear systems," in *22nd Int. Symp. on Intelligent Control*, 2007, pp. 112–117.

[121] J. Löfberg, "Yalmip : A toolbox for modeling and optimization in MATLAB," in *Proceedings of the CACSD Conference*, Taipei, Taiwan, 2004. [Online]. Available: http://control.ee.ethz.ch/~{}joloef/yalmip.php

[122] J. Lygeros, C. Tomlin, and S. Sastry, "Hybrid systems: modeling, analysis and control," *preprint*, 1999.

[123] Y. Ma and F. Borrelli, "Fast stochastic predictive control for building temperature regulation," in *American Control Conference (ACC), 2012*. IEEE, 2012, pp. 3075–3080.

[124] I. Mitchell, "A toolbox of level set methods," UBC Department of Computer Science Technical Report TR-2007-11 (June 2007).

[125] I. Mitchell, A. M. Bayen, and C. J. Tomlin, "Validating a hamilton-jacobi approximation to hybrid system reachable sets," in *Hybrid Systems: Computation and Control*. Springer, 2001, pp. 418–432.

[126] I. Mitchell and C. J. Tomlin, "Level set methods for computation in hybrid systems," in *Hybrid Systems: Computation and Control*. Springer, 2000, pp. 310–323.

[127] I. M. Mitchell, A. M. Bayen, and C. J. Tomlin, "A time-dependent hamilton-jacobi formulation of reachable sets for continuous dynamic games," *IEEE Transactions on Automatic Control*, vol. 50, no. 7, pp. 947–957, 2005.

[128] I. M. Mitchell and Y. Susuki, "Level set methods for computing reachable sets of hybrid systems with differential algebraic equation dynamics," in *Hybrid systems: computation and control*. Springer, 2008, pp. 630–633.

[129] T. Motzkin, H. Raiffa, G. L. Thompson, and R. M. Thrall, "The double description method," in *Contributions to the Theory of Games II*, H. W. Kuhn and A. W. Tucker, Eds. Princeton University Press, 1953.

[130] A. Nemirovski, "Advances in convex optimization: conic programming," in *Proceedings oh the International Congress of Mathematicians: Madrid, August 22-30, 2006: invited lectures*, 2006, pp. 413–444.

[131] S. Osher and J. A. Sethian, "Fronts propagating with curvature-dependent speed: algorithms based on hamilton-jacobi formulations," *Journal of computational physics*, vol. 79, no. 1, pp. 12–49, 1988.

[132] P. Patrinos, P. Sopasakis, H. Sarimveis, and A. Bemporad, "Stochastic model predictive control for constrained discrete-time markovian switching systems," *Automatica*, vol. 50, no. 10, pp. 2504–2514, 2014.

[133] M. Prandini, S. Garatti, and R. Vignali, "Performance assessment and design of abstracted models for stochastic hybrid s+systems through a randomized approach," *Automatica*, vol. 50, no. 11, pp. 2852–2860, 2014.

[134] M. Prandini and J. Hu, "A stochastic approximation method for reachability computations," *Stochastic Hybrid Systems*, vol. 337, pp. 107–139, 2006.

[135] M. Prandini and J. Hu, "Stochastic reachability: Theory and numerical approximation," *Stochastic hybrid systems, Automation and Control Engineering Series*, vol. 24, pp. 107–138, 2006.

[136] M. Prandini and J. Hu, "Application of reachability analysis for stochastic hybrid systems to aircraft conflict prediction," *IEEE Transactions on Automatic Control*, vol. 54, 2009.

[137] A. Prékopa, "The use of discrete moment bounds in probabilistic constrained stochastic programming models," *Annals of Operations Research*, vol. 85, pp. 21–38, 1999.

[138] A. Prékopa, *Stochastic programming*. Springer Science & Business Media, 2013, vol. 324.

[139] S. Rakovic, "Robust control of constrained discrete time systems: Characterization and implementation," Ph.D. dissertation, Imperial College London (University of London), 2005.

[140] S. V. Raković, E. C. Kerrigan, and D. Q. Mayne, "Reachability computations for constrained discrete-time systems with state-and input-dependent disturbances," in *Decision and Control, 2003. Proceedings. 42nd IEEE Conference on*, vol. 4. IEEE, 2003, pp. 3905–3910.

[141] S. V. Rakovic, E. C. Kerrigan, and D. Q. Mayne, "Optimal control of constrained piecewise affine systems with state-and input-dependent disturbances," in *16th Int. Symp. on Mathematical Theory of Networks and Systems*, 2004, pp. MP8–01.25.

[142] S. V. Rakovic, 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, 2006.

[143] L. Ros, A. Sabater, and F. Thomas, "An ellipsoidal calculus based on propagation and fusion," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics,*, vol. 32, no. 4, pp. 430–442, 2002.

[144] E. Scholte and M. E. Campbell, "A nonlinear set-membership filter for on-line applications," *International Journal of Robust and Nonlinear Control*, vol. 13, no. 15, pp. 1337–1358, 2003.

[145] J. Schur, "Über potenzreihen, die im innern des einheitskreises beschränkt sind." *Journal für die reine und angewandte Mathematik*, vol. 147, pp. 205–232, 1917.

[146] M. J. Sobel *et al.*, "Noncooperative stochastic games," *The Annals of Mathematical Statistics*, vol. 42, no. 6, pp. 1930–1935, 1971.

[147] E. Sontag, "Nonlinear regulation: The piecewise linear approach," *IEEE Transactions on Automatic Control*, vol. 26, no. 2, pp. 346–358, 1981.

[148] S. E. Z. Soudjani and A. Abate, "Adaptive gridding for abstraction and verification of stochastic hybrid systems," in *Quantitative Evaluation of Systems (QEST), 2011 Eighth International Conference on*. IEEE, 2011, pp. 59–68.

[149] J. L. Stein, *Stochastic optimal control and the US financial debt crisis*. Springer, 2012.

[150] O. Stursberg and B. Krogh, "Efficient representation and computation of reachable sets for hybrid systems," in *Hybrid Systems: Computation and Control*. Springer, 2003, pp. 482–497.

[151] S. Summers, M. Kamgarpour, J. Lygeros, and C. Tomlin, "A stochastic reach-avoid problem with random obstacles," in *Proceedings of the 14th international conference on Hybrid systems: computation and control*. ACM, 2011, pp. 251–260.

[152] S. Summers, M. Kamgarpour, C. Tomlin, and J. Lygeros, "Stochastic system controller synthesis for reachability specifications encoded by random sets," *Automatica*, vol. 49, no. 9, pp. 2906–2910, 2013.

[153] S. Summers and J. Lygeros, "Verification of discrete time stochastic hybrid systems: A stochastic reach-avoid decision problem," *Automatica*, vol. 46, no. 12, pp. 1951–1961, 2010.

[154] K. K. Thulasiraman, S. Arumugam, A. Brandstädt, and T. Nishizeki, *Handbook of graph theory, combinatorial optimization, and algorithms.* Chapman and Hall/CRC, 2015.

[155] M. J. Todd and E. A. Yıldırım, "On khachiyan's algorithm for the computation of minimum-volume enclosing ellipsoids," *Discrete Applied Mathematics*, vol. 155, no. 13, pp. 1731–1744, 2007.

[156] C. J. Tomlin, J. Lygeros, and S. S. Sastry, "A game theoretic approach to controller design for hybrid systems," *Proceedings of the IEEE*, vol. 88, no. 7, pp. 949–970, July 2000.

[157] C. J. Tomlin, I. Mitchell, A. M. Bayen, and M. Oishi, "Computational techniques for the verification of hybrid systems," *Proceedings of the IEEE*, vol. 91, no. 7, pp. 986–1001, July 2003.

[158] C. Tomlin, J. Lygeros, and S. Sastry, "Computing controllers for nonlinear hybrid systems," in *Hybrid Systems: Computation and Control.* Springer, 1999, pp. 238–255.

[159] D. Van Hessem and O. Bosgra, "A conic reformulation of model predictive control including bounded and stochastic disturbances under state and input constraints," in *Proceedings of the 41st IEEE Conference on Decision and Control, 2002,*, vol. 4. IEEE, 2002, pp. 4643–4648.

[160] D. Van Hessem, C. Scherer, and O. Bosgra, "Lmi-based closed-loop economic optimization of stochastic process operation under state and input constraints," in *Proceedings of the 40th IEEE Conference on Decision and Control*, vol. 5. IEEE, 2001, pp. 4228–4233.

[161] P. Varaiya, "Reach set computation using optimal control," in *Proc. KIT Workshop*, 1997.

[162] M. P. Vitus and C. J. Tomlin, "Closed-loop belief space planning for linear, gaussian systems," in *IEEE Conf. on Robotics and Automation*, 2011, pp. 2152–2159.

[163] L. Wolsey and G. Nemhauser, *Integer and Combinatorial Optimization*, ser. Wiley Series in Discrete Mathematics and Optimization. Wiley, 1999. [Online]. Available: https://books.google.de/books?id=84trQgAACAAJ

[164] C. Yan, M. R. Greenstreet, and M. Laza, "A robust linear program solver for reachability analysis," in *Proceedings of the First International Conference on Mathematical Aspects of Computer and Information Sciences (MACIS)*, Beijing, China, 2006, pp. 231–242.

[165] F. Yang and Y. Li, "Set-membership fuzzy filtering for nonlinear discrete-time systems," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 40, no. 1, pp. 116–124, 2010.

[166] G. Zhai, H. Lin, and P. J. Antsaklis, "Quadratic stabilizability of switched linear systems with polytopic uncertainties," *Int. Journal of Control*, vol. 76, no. 7, pp. 747–753, 2003.

[167] K. Zhou and J. C. Doyle, *Essentials of robust control.* Prentice hall Upper Saddle River, NJ, 1998, vol. 104.

[168] G. M. Ziegler, *Lectures on polytopes.* Springer Science & Business Media, 1995, vol. 152.

ISBN 978-3-7376-0580-9

9 783737 605809 >