# Unifying Control and Verification of Cyber-Physical Systems (UnCoVerCPS)

*WP2 - Online Decision Making and Control*

*Deliverable 2.3: Report on Interleaving Online Control and Reachability Computation for Certified Behaviour of Cyber-Physical Systems*

| **WP2** | **D2.3 – Report on interleaving online control and reachability computation for certified behaviour of cyber-physical systems** |
|---|---|
| Authors | O. Stursberg, Z. Liu (UKS)<br>M. Prandini (PoliMi)<br>M. Althoff (TUM) |
| Short Description | Description of the results on combining methods for control synthesis and reachability analysis for cyberphysical systems, according to the investigations in Task 2.4 of the project |
| Deliverable Type | Report |
| Dissemination level | Public |
| Delivery Date | 31 Dec 2018 |
| Contributions by | UKS, PoliMi, TUM |
| Internally accepted by | M. Althoff |
| Date of acceptance | 30 Dec 2018 |

Document history:

| Version | Date | Authors | Description |
|---|---|---|---|
| 1.0 | Nov 19, 2018 | O. Stursberg et al. | Internal review |
| 2.0 | Dec 29/30, 2018 | O. Stursberg et al. | Revision |

# Contents

# 1 Introduction and Motivation

The main goal of UnCoVerCPS is to create control design techniques for cyber-physical systems (CPS) such that the satisfaction of safety properties is guaranteed, while at the same time controller adaption in response to changes of the CPS or its environment is enabled. If all conceivable changes can be foreseen and modeled before the CPS gets into operation, the natural solution is to design the controller a-priori as satisfying the system specifications robustly against the set of changes – this corresponds to the procedure established in the field of *robust control*. A possible approach is to determine all effects imposed on the systems by disturbances or the environment in terms of reachable sets, and to compute controllers that are robustly stabilizing for the complete reachable set. For this procedure, it can, however, occurr that the conservatism implies low control performance. In this case, or if it is not possible to determine the complete set of behaviors before operation, an alternative approach is commendable: The adaptation of a control law or the computation of a control strategy fitting to the momentary situation is necessary, as investigated in the established fields of *adaptive control* and *predictive control*. Since UnCoVerCPS targets systems, which are embedded in a changing environment, model predictive controllers (MPC) are a natural choice, as these solve online optimization problems subject to constraints. While several aspects of MPC have been investigated before (as, e.g., real-time efficiency and the robustness with respect to parametric uncertainty of the system to be controlled), the ensurance of safety properties was barely paid attention to so far.

For this purpose, one thread of research within UnCoVerCPS was to combine and intertwine model predictive control with reachability analysis. The objective is to use reachability computations either to compute the constraints imposed on the system over a prediction horizon starting from the current point of time, or to ensure that control inputs applied to the system can only lead to behaviors within the given specifications. Corresponding techniques for cyber-physical systems have to consider criteria for complexity of computation (thus leading to meet real-time requirements), as well as availability of information on interacting subsystems of the CPS. Along this line, this report describes three methods:

- A method called *reachset model predictive control* is proposed which embeds reachable set computations into standard MPC of nonlinear systems with disturbances and measurement disturbances to ensure that a terminal region is safely reached.
- For linear discrete-time systems with additive stochastic disturbances (with possibly unbounded support), an approach of stochastic MPC is proposed which considers input

and probabilistic state constraints, i.e. a scenario-based technique with constraint relaxation scheme determines the controlled reach set such that the state constraints are satisfied with a given high probability.

- For distributed CPS, an MPC approach is proposed in which the local controller of any subsystem takes into account the hybrid dynamics of the subsystems as well as state constraints obtained from reach set computation of interacting subsystems.

The following three sections describe the aforementioned techniques including numerical examples and discussions of the properties, before the deliverable closes with summarizing conclusions in Sec. 1. While the examples used for illustration in the sections 2 to 4 differ from the use cases investigated in WP5 of UnCoVerCPS, the techniques reported here have been applied for different instances of these use cases, in particular the case studies on autonomous driving and human-robot interaction.

# 2 Using Reachability Analysis in Model Predictive Control

In this section, we present a reachset model predictive control (MPC) approach: We combine reachability analysis with regular MPC in order to obtain provably safe controllers for disturbed nonlinear systems with constraints on states and inputs. The interested reader is referred to [70], where the following text was originally published.

## 2.1 Problem Formulation

We consider a continuous-time system with disturbed, nonlinear dynamics of the form

$$\dot{x}(t) = f\big(x(t), u(t), w(t)\big), \tag{1}$$

with states $x(t) \in \mathbb{R}^n$, inputs $u(t) \in \mathbb{R}^m$, and disturbances $w(t) \in \mathcal{W} \subset \mathbb{R}^d$ ($\mathcal{W}$ is compact, i.e., closed and bounded). We do not require any stochastic properties for $w(\cdot)$; we only assume that any possible disturbance trajectory is bounded at any point in time in the compact set $\mathcal{W}$. We denote this by $w(\cdot) \in \mathcal{W}$, which is shorthand for $w(t) \in \mathcal{W}, \forall t \in \mathbb{R}_0^+$. We use the same shorthand later for state and input constraints. We denote the solution of (1) with initial state $x(0)$, input $u(\cdot)$, and disturbance $w(\cdot)$ at time $t$ as $\xi(x(0), u(\cdot), w(\cdot), t)$. The measurement of the system is modeled by a function $h$, returning the measured state $\hat{x}(t)$ subject to a compact set of measurement errors $\mathcal{V} \subset \mathbb{R}^o$:

$$\hat{x}(t) \in \hat{\mathcal{X}}(t) = \{h(x(t), \eta(t)) \,|\, \eta(t) \in \mathcal{V}\}.$$

If not all states are measurable, $\hat{\mathcal{X}}(t)$ can also be obtained by a set-based observer [28, 46].

The goal is to find an MPC controller which steers the system from an initial state $x(0) \in \mathcal{X}$ in finite time into a goal set $\mathcal{X}_f$ while minimizing some cost function. At the same time, the controlled system must satisfy state and input constraints despite disturbances and measurement noise, i.e.,

$$\xi\big(x(0), u(\cdot), wt, \cdot\big) \in \mathcal{X}, \tag{2}$$

$$u(\cdot) \in \mathcal{U}, \tag{3}$$

where $\mathcal{X}$ and $\mathcal{U}$ are both convex sets in $\mathbb{R}^n$ and $\mathbb{R}^m$, respectively.

## 2.2 Reachset Model Predictive Control

In this subsection, we present our reachset model predictive control approach. After an overview, we provide required definitions and further detail our approach. In the end, we show all properties in the main theorem.

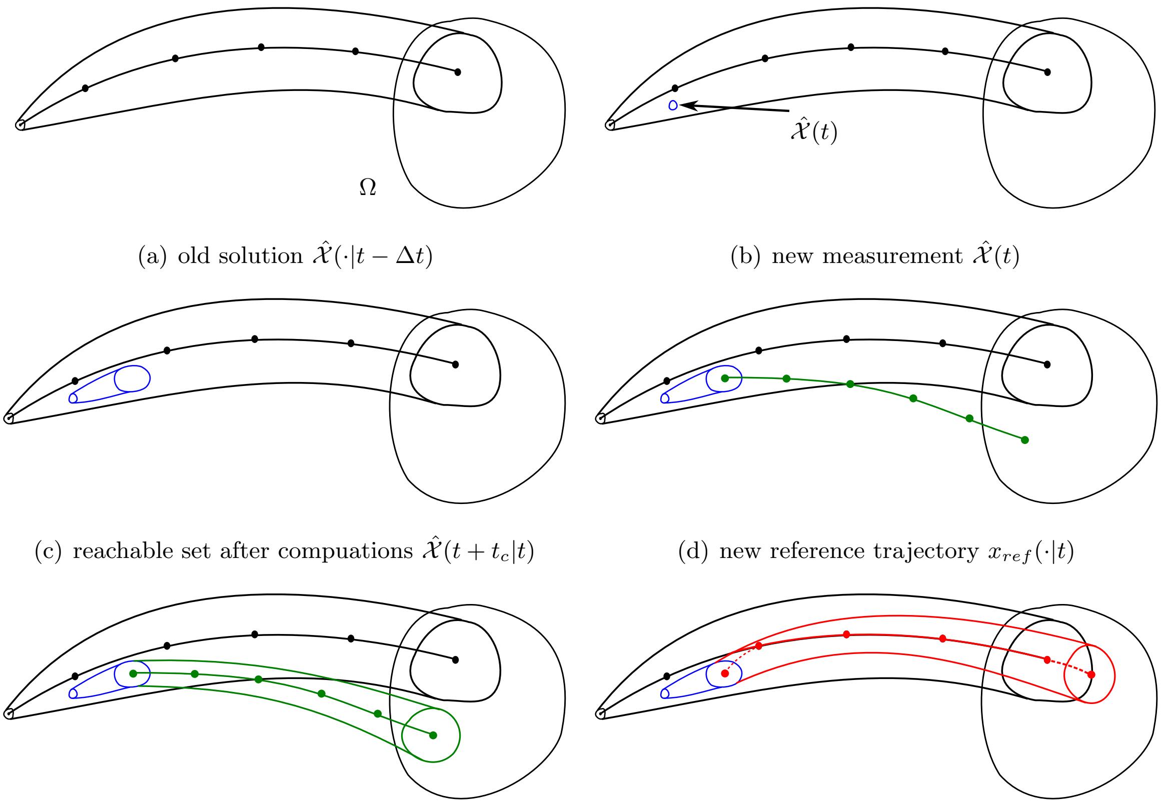
### Overview

The basic idea of our reachset MPC is shown in Fig. 1. Starting from the solution of the previous step (Fig. 1(a)), we obtain a measurement $\hat{x}(t)$ at time $t$ (Fig. 1(b)). As there might be measurement noise, we only know that we are in some uncertain set $\hat{\mathcal{X}}(t)$, which is a singleton when the state can be precisely measured. Based on this measurement, we are looking for the optimal controller which steers the system to the goal set $\mathcal{X}_f$. Since we cannot optimize for an infinite time horizon, we use a dual-mode MPC [53]. This means we consider a final prediction horizon of length $t_N$ and require that the prediction ends in a terminal region $\Omega$ (defined formally later in Def. 4), for which we know a safe and stabilizing controller.

Based on the obtained measurement, we optimize a new reference trajectory $x_{ref}(\cdot|t)$, which is tracked with a fixed feedback controller. To solve the optimization problem and to compute the reachable set, we need some time $t_c$, and we apply the controller from the previous prediction to the system during this time. Using reachability analysis, we predict where we end after the optimization and computation of the reachable set and use this set $\hat{\mathcal{X}}(t+t_c|t)$ as the initial set for our optimization problem (Fig. 1(c)). We use the notation $(t+t_c|t)$ to refer to the prediction for time $t+t_c$ made at time $t$. For efficiency reasons, we solve the optimization problem for the center trajectory only, but with tightened constraints (Fig. 1(d)). We then use reachability analysis to check if all possible solutions $\hat{\mathcal{X}}(\cdot|t)$ are guaranteed to satisfy all constraints (Fig. 1(e)). Only if this is the case, and if the computations finish in the allocated time $t_c$, we apply the new, guaranteed-safe solution. If not, we use a feasible solution which consists of the solution from the previous step, extended by the safe controller from the terminal region (Fig. 1(f)). Therefore, under the common assumption that we know a feasible trajectory at the initial time, we always know a feasible solution, which we can use as a backup if we cannot find a better feasible solution in the available time. We then apply the solution for time $\Delta t$ before we start the next optimization problem based on the new measurement. The feasible solution is defined as:

**Definition 1.** *The feasible solution is a possible non-optimal input trajectory, which leads to trajectories* $\xi\big(x(t), u(\cdot), w(\cdot), \cdot\big)$ *satisfying the constraints* (2)-(3) *and ends in the terminal region* $\Omega$ *after time* $t_N$*:* $\xi\big(x(t), u(\cdot), w(\cdot), t+t_N\big) \in \Omega$.

After defining reachable sets, we explain all steps of our approach in detail and discuss the guarantees at the end of this subsection.



(a) old solution $\hat{\mathcal{X}}(\cdot|t-\Delta t)$

(b) new measurement $\hat{\mathcal{X}}(t)$

(c) reachable set after compuations $\hat{\mathcal{X}}(t+t_c|t)$

(d) new reference trajectory $x_{ref}(\cdot|t)$

(e) new reachable set $\hat{\mathcal{X}}(\cdot|t)$

(f) fallback: feasible solution $\hat{\mathcal{X}}(\cdot|t)$

**Figure 1:** Illustration of our reachset MPC approach: Beginning with a feasible solution set $\hat{\mathcal{X}}(\cdot|t-\Delta t)$ from the previous time step (a), we obtain the measurement of the (possibly uncertain) state at time $t$ (b). Based on this set of possible states, we compute the reachable set $\hat{\mathcal{X}}(t+t_c|t)$ (blue) for the time $t_c$ which we need to solve the optimization problem (c). Starting with the center of this reachable set, we optimize the reference trajectory $x_{ref}(\cdot|t)$ (green) for the time horizon $t_N$ (d). After the optimization, we compute the corresponding reachable set $\hat{\mathcal{X}}(\cdot|t)$ (green) (e). If all constraints are satisfied for the reachable set, we use the new reference trajectory and continue with the next iteration at time $t+\Delta t$. If the solution is not feasible or is not computed in time, we follow the feasible solution (red) from the previous time step, which is extended by the auxiliary controller in the terminal region Ω (f).

### Reachability Analysis

To ensure the satisfaction of constraints despite disturbances and measurement noise, we use reachable sets:

**Definition 2.** *For a system (1), the reachable set $\mathcal{R}_{t,\mathcal{U},\mathcal{W}}(\mathcal{S}) \subset \mathbb{R}^n$ for a time $t$, inputs $u(\cdot) \in \mathcal{U} \subset \mathbb{R}^m$, disturbances $w(\cdot) \in \mathcal{W} \subset \mathbb{R}^d$, and a set of initial states $\mathcal{S} \subset \mathbb{R}^n$ is the set of end states of trajectories starting in $\mathcal{S}$ after time $t$, i.e.,*

$$\mathcal{R}_{t,\mathcal{U},\mathcal{W}}(\mathcal{S}) = \big\{x(t) \in \mathbb{R}^n | \exists x(0) \in \mathcal{S}, u(\cdot) \in \mathcal{U}, w(\cdot) \in \mathcal{W} : \xi\big(x(0), u(\cdot), w(\cdot), t\big) = x(t)\big\}.$$

*The reachable set over a time interval $[t_1, t_2]$ is the union of all reachable sets for these time points, i.e.,*

$$\mathcal{R}_{[t_1,t_2],\mathcal{U},\mathcal{W}}(\mathcal{S}) = \bigcup_{t\in[t_1,t_2]} \mathcal{R}_{t,\mathcal{U},\mathcal{W}}(\mathcal{S}).$$

If we consider the reachable set for a system with feedback $u_{fb}(\hat{x}(t))$, then we denote by $\mathcal{R}_{t,u_{fb},\mathcal{W}}(\mathcal{S})$ the reachable set obtained if we consider the closed-loop dynamics $\dot{x}(t) = f(x(t), u_{fb}(\hat{x}(t)), w(t))$ subject to disturbances and measurement errors. Since it is not possible to compute exact reachable sets for most systems [57], we compute over-approximations instead.

We represents sets by zonotopes due to their favorable properties for reachability analysis [4]:

**Definition 3.** *A set is called a zonotope if it can be written as*

$$\mathcal{Z} = \Big\{x \in \mathbb{R}^n \Big| x = c + \sum_{i=1}^{p} G\lambda, \lambda \in [-1, 1]^p\Big\}.$$

*Here, $c \in \mathbb{R}^n$ defines the center of the zonotope, and $G \in \mathbb{R}^{n\times p}$ its generator matrix. We use $\langle c, G\rangle$ as a more concise notation of $\mathcal{Z}$.*

### Dual Mode MPC

As is common in MPC, we use dual mode MPC [53] to limit the prediction horizon. We use the control law

$$u_\Omega(\hat{x}(t)) = K_\Omega \hat{x}(t) \tag{4}$$

to stabilize a terminal region $\Omega$ and the control law

$$u_{MPC}(\hat{x}(t)) = v(t) + K(\hat{x}(t) - x_{ref}(t)) \tag{5}$$

which controls the system into the terminal region. Here, $v(t)$ denotes the reference input, which is optimized online, and $x_{ref}$ refers to the corresponding state trajectory. The feedback matrices $K \in \mathbb{R}^{m\times n}$ and $K_\Omega \in \mathbb{R}^{m\times n}$ can be different from each other, and $K$ can even be time-varying as discussed at the end of this subsection. We use linear controllers for faster computation times; however, all concepts presented also work for nonlinear controllers. The terminal region $\Omega$ is defined as a region of attraction in which the state and input constraints are satisfied:

**Definition 4.** *Given a dynamical system of the form* (1) *and a terminal control law* (4)*. The terminal region* $\Omega$*,* $\mathcal{X}_f \subseteq \Omega \subseteq \mathcal{X}$*, is defined as*

$$\Omega = \Big\{x \Big| \forall \eta \in \mathcal{V} : h(x,\eta) \in \bar{\Omega}\Big\},$$

*with*

$$\begin{aligned}\bar{\Omega} = \Big\{x \Big| &\forall t \in \mathbb{R}_0^+, \forall \hat{x}(t) \in \hat{\mathcal{X}}(t), \forall w(t) \in \mathcal{W}, \exists t_f \in \mathbb{R}_0^+ : \\ &\xi(x, u_\Omega(\hat{x}(\cdot)), w(\cdot), t_f) \in \mathcal{X}_f, \\ &\xi(x, u_\Omega(\hat{x}(\cdot)), w(\cdot), t) \in \mathcal{X}, \\ &u_\Omega\big(\xi(x, u_\Omega(\hat{x}(\cdot)), w(\cdot), t)\big) \in \mathcal{U}\Big\}.\end{aligned}$$

Using a terminal region is standard in many MPC approaches and is required to provide guarantees beyond the finite prediction horizon [53]. It is computed before the controller is applied online. There exist different ways to compute an approximation of an invariant set of a controller; many of them use Lyapunov functions, which might be hard to find in practice. While a region of attraction can also be computed using Lyapunov functions, there also exist methods to compute them automatically and in many cases more efficiently using reachable sets [33]. The region of attraction is usually much larger than a positive invariant set, which provides more flexibility to our approach. In addition, by checking the satisfaction of the constraints during the execution of the algorithm from [33], we can automatically compute a safe region of attraction, i.e., a region of attraction for which the state and input constraints are satisfied despite disturbances.

As is common in dual mode MPC, we also use this terminal region to obtain the feasible solution as a backup plan by using the remainder of the previous solution:

$$v_f(\tau|t+\Delta t) = v(\tau|t) \text{ for } \tau = [t+\Delta t + t_c, t+t_N]. \tag{6}$$

Once we reach the terminal region at time $t+t_N$, we switch to the terminal controller (4).

During operation, we compute future reachable sets $\hat{\mathcal{X}}(t+\Delta t|t) = \mathcal{R}_{\Delta t, u_{MPC}, \mathcal{W}}(\hat{\mathcal{X}}(t))$ based on the current input trajectory $v(\cdot|t)$, with $\hat{\mathcal{X}}(t)$ composed of the measured state $\hat{x}(t)$

plus measurement uncertainty. Note that even though $\hat{\mathcal{X}}(t)$ might be partly outside of the reachable set, we know from the over-approximative nature of the reachability analysis that the real state $x(t)$ must lie inside the reachable set from the previous step, i.e., $x(t) \in \hat{\mathcal{X}}(t|t-\Delta t)$. Therefore, we only have to consider the intersection $\hat{\mathcal{X}}(t) \cap \hat{\mathcal{X}}(t|t-\Delta t)$ as the initial set for the next optimization. This is a common approach used in set-based observers[28, 46].

### Considering the Computation Time

When starting the optimization for a new measurement, we consider its computation time $t_c$. To be safe, we need to know the reachable set after $t_c$ due to uncertainties and disturbances:

$$\hat{\mathcal{X}}(t+t_c|t) = \mathcal{R}_{t_c,u_{MPC},\mathcal{W}}(\hat{\mathcal{X}}(t)).$$

By applying the reference trajectory plus feedback controller from the previous optimization, we know that the reachable set after the optimization time is inside the reachable set from the previous optimization.

The allowed computation time $t_c$ for the optimization and reachability analysis is a user-defined design parameter. Note that $t_c$ can be estimated quite well by restricting the iterations of the optimization algorithm and by considering the fact that the computation time for the reachability analysis scales approximately linear with the considered time horizon. However, inappropriate values of $t_c$ do not impede the desired properties in (2)-(3), as we can always go back to the feasible solution if $t_c$ is not sufficient to find a new solution. We compute the reachable set for this alloted time (see Fig. 1(c)). If the optimization algorithm finishes before that, we keep following the previous solution until the designated time, from which point on we apply the new solution. If we reach this point in time without a new feasible solution, we simply keep following the previous feasible solution and start a new optimization (see Fig. 1(f)).

### Contraction Constraint

An important consideration in MPC is to ensure the convergence to the goal set in a finite amount of time. While this could be done using Lyapunov functions, we use an approach similar to [12] which does not require a Lyapunov function. Through the construction of the terminal region using the approach from [33], we know that after reaching the terminal region, we converge in finite time to the desired goal set. Therefore, we only have to ensure that we converge in finite time to the terminal region. To do so, we introduce the distance operator from [12]:

**Definition 5.** *Given sets $\hat{\mathcal{X}}$ and $\Phi = 1/(1+\alpha)\Omega$, with $\alpha \in \mathbb{R}^+$, $\|\hat{\mathcal{X}}\|_\Phi$ is defined as*

$$\|\hat{\mathcal{X}}\|_\Phi = \min \beta, \ \ s.t. \ \hat{\mathcal{X}} \subseteq (1+\beta)\Phi, \, \beta \geq 0.$$

As mentioned in [12], $\|\hat{\mathcal{X}}\|_\Phi$ is equal to zero if and only if $\hat{\mathcal{X}} \subseteq \Phi$, and if $x \notin \Omega$, it follows that $\|x\|_\Phi > \alpha$. The authors also show that if $\Phi$ is a polyhedron defined by the intersection of half-spaces of the form $\Phi = \{x : d_i^T x \leq e_i, i \in \{1, \dots, p\}\}$ that contains the origin ($e_i > 0, i \in \{1, \dots, p\}$) and $\hat{\mathcal{X}} = \langle c, G \rangle$ is a zonotope, then $\|\hat{\mathcal{X}}\|_\Phi$ can be obtained from the equality

$$\|\hat{\mathcal{X}}\|_\Phi = \max \left\{ 0, \max_{i=1,\dots,p} \frac{d_i^T c - e_i + \|G^T d_i\|_1}{e_i} \right\},$$

where $\|G^T d_i\|_1$ denotes the sum of the absolute values of vector $G^T d_i$. By defining the distance with respect to the tighter set $\Phi$, we ensure a desired contraction rate, as shown later in Thm. 1.

### Optimal Control Problem

The optimization problem which is solved online at time $t$ is given by

$$\begin{aligned} &\min_{v(\cdot|t)} J\left(\hat{\mathcal{X}}(t+t_c|t), v(\cdot|t)\right) \\ =&\min_{v(\cdot|t)} \int_{t+t_c}^{t+t_N} L(x_{ref}(\tau|t), v(\tau|t)) d\tau + V(x_{ref}(t+t_N|t)) \end{aligned} \tag{7}$$

s.t.

$$x_{ref}(t+t_c|t) = \text{center}(\hat{\mathcal{X}}(t+t_c|t)), \tag{8}$$

$$\dot{x}_{ref}(t+\tau|t) = f(x_{ref}(t+\tau|t), v(t+\tau|t), 0), \forall \tau \in [t+t_c, t+t_N], \tag{9}$$

$$v(\tau|t) \in \bar{\mathcal{U}}(\tau|t), \quad \forall \tau \in [t+t_c, t+t_N], \tag{10}$$

$$x_{ref}(\tau|t) \in \bar{\mathcal{X}}(\tau|t), \quad \forall \tau \in [t+t_c, t+t_N], \tag{11}$$

$$x_{ref}(t+t_N|t) \in \bar{\Phi}, \tag{12}$$

$$\sum_{k=1}^{\bar{N}(t)-1} \|x_{ref}(t+k\Delta t|t)\|_\Phi - \sum_{k=1}^{\bar{N}(t-\Delta t)-1} \|x_{ref}(t-\Delta t+k\Delta t|t-\Delta t)\|_\Phi < -\bar{\alpha}, \tag{13}$$

where center$(\hat{\mathcal{X}}(t+t_c|t))$ refers to the center of the zonotope $\hat{\mathcal{X}}(t+t_c|t)$ and $\bar{N}(t) = \min_{k\in\mathbb{N}} x_{ref}(t+k\Delta t|t) \in \bar{\Phi}$.

We minimize the cost function $J(\cdot)$ in (7), consisting of a positive definite state cost $L(\cdot)$ and a positive definite terminal cost $V(\cdot)$, with respect to the center trajectory, which starts from the center of the reachable set (8) after $t_c$. To ensure the satisfaction of the constraints for the disturbed, closed-loop dynamics, we use tightened time-dependent input (10) and state constraints (11), $\bar{\mathcal{U}}(\cdot)$ and $\bar{\mathcal{X}}(\cdot)$, respectively, as discussed later. As is common in dual-mode MPC, we have a terminal constraint (12), which requires that the center trajectory ends in a tightened terminal region $\bar{\Phi}$. Finally, we have a contraction constraint (13) with parameter $\bar{\alpha}$ (not necessarily equal to $\alpha$), which ensures convergence to the terminal region $\Omega$.

**Tightened Constraints**

To be able to apply our MPC approach online, we only optimize the center trajectory without computing the reachable sets during this optimization. While it is possible to optimize over reachable sets[69], this is not possible in real-time for fast systems. The authors of [12] propose optimizing over the reachable sets; however, they do not discuss the computation times and their approach is rather conservative as demonstrated later in Sec. 2.3. Instead, we optimize only the center trajectory and tighten the constraint sets accordingly, such that state and input constraints are met. At the end of the optimization, we perform a reachability analysis to check if all constraints are actually satisfied. If this is not the case, we always have the feasible solution as a safe fallback. We initially guess the size of the reachable set and the resulting inputs from the controller based on the reachable set from the feasible solution and verify the solution later. This means we take the size of the reachable set of the feasible solution at the corresponding time step, scaled by a factor $\gamma \in \mathbb{R}^+$, and use this set to tighten the constraints sets. To do this in a set-based fashion, we introduce the Minkowski difference denoted by $\ominus$, i.e., the subtraction of two sets, as the complement of the Minkowski sum: for sets $\mathcal{X}, \mathcal{Y} \subset \mathbb{R}^n$ we define

$$\begin{aligned}
\mathcal{X} \oplus \mathcal{Y} &= \{x + y | x \in \mathcal{X}, y \in \mathcal{Y}\},\\
\mathcal{X} \ominus \mathcal{Y} &= \{z \subseteq \mathbb{R}^n | z \oplus \mathcal{Y} \subseteq \mathcal{X}\}.
\end{aligned}$$

This allows us to write the tightened constraints as

$$\begin{aligned}
\bar{\mathcal{X}}(t+\tau) &= \mathcal{X} \ominus \gamma\Big(\hat{\mathcal{X}}(t-\Delta t+\tau|t-\Delta t) \ominus x_{ref}(t-\Delta t+\tau|t-\Delta t)\Big), \forall \tau \in [t_c, t_N],\\
\bar{\mathcal{U}}(t+\tau) &= \mathcal{U} \ominus K\gamma\Big(\hat{\mathcal{X}}(t-\Delta t+\tau|t-\Delta t) \ominus x_{ref}(t-\Delta t+\tau|t-\Delta t)\Big), \forall \tau \in [t_c, t_N],\\
\bar{\Phi} &= \Phi \ominus \gamma\Big(\hat{\mathcal{X}}(t-\Delta t+t_N|t-\Delta t) \ominus x_{ref}(t-\Delta t+t_N|t-\Delta t)\Big).
\end{aligned}$$

As the reachable sets might change their size, the constraints become time-dependent. If this guess is too conservative, we only obtain a sub-optimal solution; if it is too optimistic, we have to go back to the feasible solution. In any case, we have a safe solution in the end.

### Guarantees Through Reachability Analysis

After obtaining the center trajectory, we use the pre-defined feedback controller to compute the reachable set for the closed-loop dynamics. We start from the reachable set $\hat{\mathcal{X}}(t+t_c|t)$ and compute it for the remaining prediction horizon (see Fig. 1(e)). Afterwards, we check if the reachable set satisfies the state and input constraints at all times, if the final reachable set is completely inside the terminal region, and if the contraction constraint is also satisfied for the reachable sets, i.e., we check if $\forall \tau \in [t_c, t_N]$ :

$$\hat{\mathcal{X}}(t+\tau|t) \subseteq \mathcal{X}, \tag{14}$$

$$v(t+\tau|t) \oplus K\left(\hat{\mathcal{X}}(t+\tau|t) \ominus x_{ref}(t+\tau|t)\right) \subseteq \mathcal{U}, \tag{15}$$

$$\hat{\mathcal{X}}(t+t_N|t) \subseteq \Phi, \tag{16}$$

$$\sum_{k=1}^{N(t)-1} \|\hat{\mathcal{X}}(t+k\Delta t|t)\|_\Phi - \sum_{k=1}^{N(t-\Delta t)-1} \|\hat{\mathcal{X}}\big(t+(k-1)\Delta t|t-\Delta t\big)\|_\Phi < -\alpha, \tag{17}$$

where we evaluate the contraction constraint (17) only at finitely many time points to obtain a finite cost and where

$$N(t) = \min_{k\in\mathbb{N}} \hat{\mathcal{X}}(t+k\Delta t|t) \subseteq \Phi. \tag{18}$$

To evaluate if the zonotope $\hat{\mathcal{X}}(t+\tau|t) = \langle c, G\rangle$ satisfies convex state and input constraints of the form $\mathcal{X} = \{x \in \mathbb{R}^n | Cx \leq d\}$, we simply have to check if the following inequality holds:

$$Cc + \sum_{i=1}^{p} |Cg^{(i)}| \leq d, \tag{19}$$

with $g^{(i)}$ denoting the $i$-th column of $G \in \mathbb{R}^{n\times p}$ and where the absolute value and less or equal operators are both performed element-wise. Using this formula and using the fact that the reachability analysis provides us with reachable sets for time intervals in the form of zonotopes, we can efficiently check if the constraints (14)-(17) are satisfied for the reachable sets at all times. If this is the case, we apply the new control input to the system and start with a new iteration step. If the solution does not satisfy all those constraints or if the computation takes longer than the pre-specified time, we apply the input from the feasible solution instead.

### Main Theorem

**Theorem 1.** *If we know an initial feasible solution at $t = 0$, then Alg. 1 remains feasible for all times and the system robustly converges to the goal set $\mathcal{X}_f$ in finite time. During the whole time, the system satisfies the state and input constraints* (2)-(3) *despite disturbances and uncertain measurements.*

**Algorithm 1** Reachset MPC Algorithm

```
Initialize: $t \leftarrow 0, v(\cdot|-\Delta t) \leftarrow$ initial feasible solution
while $\hat{x}(t) \notin \Omega$ do
    $u(\tau) \leftarrow v(\tau|t-\Delta t) + K(\hat{x}(\tau) - x_{ref}(\tau|t-\Delta t)),$
    $\tau \in [t, t+t_c]$
    $v_f(\cdot|t) \leftarrow$ feasible solution (6)
    $v^*(\cdot|t) \leftarrow$ solution of optimization problem (7)
    if Optimization problem feasible & solved in time & (14)–(17) satisfied then $v(\cdot|t) \leftarrow v^*(\cdot|t)$
    else $v(\cdot|t) \leftarrow v_f(\cdot|t)$
    end if
    $u(\tau) \leftarrow v(\tau|t) + K\big(\hat{x}(\tau) - x_{ref}(\tau|t)\big),$
    $\tau \in [t+t_c, t+\Delta t]$
    $t \leftarrow t + \Delta t$
end while
$u(\tau) \leftarrow K_{\Omega}\hat{x}(\tau), \tau \geq t$
```

*Proof.* We have to show three things: (i) The system remains recursively feasible, i.e., in each step we can find a feasible solution, (ii) the system reaches the goal set $\mathcal{X}_f$ in finite time, and (iii) the constraints are satisfied at all times despite disturbances and measurement noise. We keep the proof concise, as many parts follow standard robust MPC techniques, as used in [12].

(i) This can be shown by induction:

*Base Case:* For t=0, we know a feasible solution by assumption.

*Induction Hypothesis:* If we know a feasible solution at time $t$, then we can always get a feasible solution at $t + \Delta t$.

*Induction Step:* For every step at time $t + \Delta t$, we know from the over-approximative way of computing the reachable set, that we start inside the reachable set of the previous step, i.e., $\hat{\mathcal{X}}(t+\Delta t) \subseteq \hat{\mathcal{X}}(t+\Delta t|t)$, for which we know the remainder of the solution from the previous step, i.e., $v(t+\tau|t), \forall \tau \in [\Delta t, t_N]$. Since the solution at time $t$ is feasible, it ends in the terminal region, where we know by construction that the terminal controller provides a feasible solution, see Def. 4. Therefore, the previous solution extended by the terminal controller, see (6), is always feasible and can be applied if we do not find a better solution in time.

(ii) The terminal region $\Omega$ is computed such that any state inside $\Omega$ robustly converges to the goal set $\mathcal{X}_f$ in finite time despite disturbances and sensor noise. Therefore, we only have to ensure reaching the terminal region in finite time. From the contraction constraint (17), we enforce reaching the terminal region in at most $(1/\alpha)\sum_{k=1}^{N}\|\hat{\mathcal{X}}(t+k\Delta t|t)\|_\Phi$ steps. If we find a new solution, we know from (17) that this new solution satisfies the rate of at least $-\alpha$. Let us now show that the feasible solution is also guaranteed to have this convergence rate:

$$\begin{aligned}&\sum_{k=1}^{N(t)-1}\|\hat{\mathcal{X}}(t+k\Delta t|t-\Delta t)\|_\Phi - \sum_{k=1}^{N(t-\Delta t)-1}\|\hat{\mathcal{X}}(t+(k-1)\Delta t|t-\Delta t)\|_\Phi\\ &= -\|\hat{\mathcal{X}}(t|t-\Delta t)\|_\Phi < -\alpha,\end{aligned}$$

where we denote by $\hat{\mathcal{X}}(t+k\Delta t|t-\Delta t)$ the resulting reachable set from the feasible solution $v_f(\cdot|t)$. Since $\hat{\mathcal{X}}(t+(N(t-\Delta t)-1)\Delta t|t-\Delta t)\subseteq\Phi$, we know from (18) that $N(t-\Delta t)=N(t)+1$ and that $\|\hat{\mathcal{X}}(t+(N(t-\Delta t)-1)\Delta t|t-\Delta t)\|_\Phi=0$. Therefore, the difference is only the cost of $-\|\hat{\mathcal{X}}(t|t-\Delta t)\|_\Phi$. Because $\hat{\mathcal{X}}(t|t-\Delta t)\nsubseteq\Omega$, it follows from Def. 5 that $\|\hat{\mathcal{X}}(t|t-\Delta t)\|_\Phi>\alpha$, and therefore the last inequality holds. As we can always revert to the feasible solution, the convergence in finite time is guaranteed.

(iii) Before we apply the new solution, we check the constraints for the over-approximated reachable set of the disturbed system in (14)-(17). If they are satisfied, then the new solution is safe and can be applied. If they are violated, we apply the safe feasible solution; see (i). □

#### Extension

As mentioned before, we cannot guarantee that the solution resulting from the reference trajectory which is computed with the tightened constraints (10)-(13) will satisfy the actual constraints (14)-(17). While we are always safe, this might make our approach unnecessarily conservative. One way to overcome the problem without getting too conservative is to compute several possible solutions in parallel. Using different estimations of reachable sets and inputs applied by the feedback controller results in several optimization problems with different constraints. As they are completely independent, we can utilize modern multi-core processors by solving them and using reachability analysis in parallel and thus choose the best feasible solution.

### 2.3 Numerical Example

To compare our reachset MPC control algorithm with the approach from [12], we use the same nonlinear continuous stirred tank reactor (CSTR) system for our numerical example. The model of the reactor for an exothermic, irreversible reaction A → B with constant liquid volume is given by [12]:

$$\begin{aligned}\frac{d\,C_A}{dt} &= \frac{q}{V}\ (C_{Af} - C_A) - k_0\ \exp\left(-\frac{E}{RT}\right) \cdot C_A + w_1\ , \\ \frac{d\,T}{dt} &= \frac{q}{V}\ (T_f - T) - \frac{\Delta H \cdot k_0}{\rho\, C_p}\ \exp\left(-\frac{E}{RT}\right) \cdot C_A + \frac{U \cdot A}{V \cdot \rho \cdot C_p}\,(T_c - T) + w_2\ , \end{aligned} \tag{20}$$

where $C_A$ is the concentration of A in the reactor, $T$ is the temperature of the reactor and $T_c$ is the coolant stream temperature. The system state is defined as $x = \left[\left(C_A - C_A^0\right), \left(T - T^0\right)\right]^T$, and the system input as $u = T_c - T_c^0$, with the steady state $C_A^0 = 0.5\,mol/l$, $T^0 = 350\,K$, $T_c^0 = 300\,K$. The model parameters can be found in [12].

The set of inputs is $\mathcal{U} = [-20,\, 70]\,K$ and the uncertainty $w = [w_1,\, w_2]^T$ is bounded by $w_1 \in [-0.1, 0.1]\,mol/(l\,min)$ and $w_2 \in [-2, 2]\,K/min$. The example does not consider state constraints and assumes that the state can be precisely measured.

In order to determine a terminal region $\Omega$, we compute an LQR controller for the system linearized at the steady state $x_S = [0,\, 0]^T$, which results in $K_\Omega = [66.65, -4.86]$. We then use the approach from [33] to calculate $\Omega$ as explained before. The time step size of $\Delta t = 1.8\,s$ and a prediction horizon of $t_N = 19.8\,s$, which is equal to $N = 11$ time steps, are the same as in [12]. We keep the reference inputs constant in each time step. The cost functions $L(x, v) = v^T R_c\, v$ and $V(x) = x^T Q_c\, x$ are applied; $R_c = 10^{-12}$, and $Q_c$ is a diagonal matrix with 100 and 1 on the diagonal. Since no cost function is provided in [12], we use these parameters to best approximate their trajectory. We use $\alpha = \bar{\alpha} = 0.1$ for the contraction parameter and $\bar{\mathcal{U}} = [-18,\, 68]\,K$ for the tightened input constraints. For the control law $u_{MPC}(x)$ we apply a time-varying feedback matrix $K$, where at each time step $k$, we obtain a new $K$ as an LQR controller for the system linearized at $x^* = \big(x_{ref}\,(t + k\Delta t|t) + x_{ref}\,(t + (k+1)\Delta t|t)\,\big)/2$ and with input weighting matrix $R = 100$ and state weighting matrix $Q$ as the identity. In order to reproduce the behavior of the disturbed system during the execution of the algorithm, we simulate the model (20) with random values for the disturbances $w$. For the allocated optimization time we use the value $t_c = 0.54\,s$.

Our algorithm is implemented in MATLAB and we use the ACADO toolbox [43] to solve the optimal control problems with a multiple shooting algorithm. For the reachable set computation we use the CORA toolbox [5]. All computations are performed on a 2.9GHz quad-core i7 processor with 32GB memory and without using parallel computing.

The initial solution for the first numerical example with initial state $x_0 = [-0.15,\, -45]^T$ is displayed in Fig. 2. During Alg. 1, the maximum computation time for the optimization and reachability analysis is $0.51\,s < t_c$, which means that we are able to perform all computations in real time. As a comparison to our algorithm, Fig. 3 shows the initial solution of the

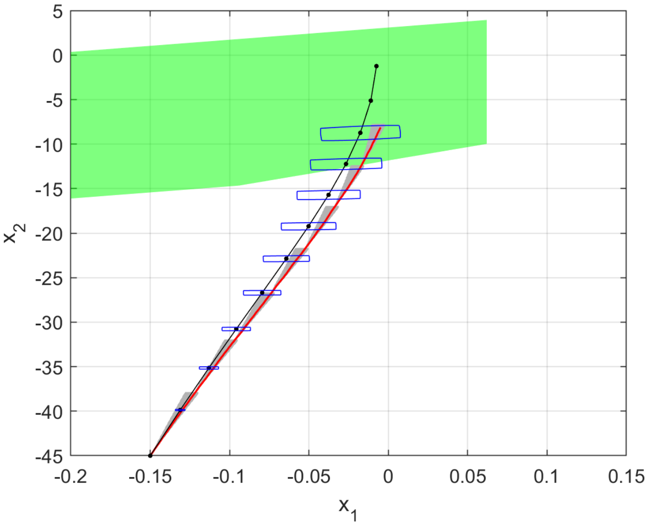


**Figure 2:** Center trajectory (black) and reachable sets at discrete time points (blue) of the initial solution for our approach. A resulting trajectory of the real system is shown in red, its reachable set in gray, and the terminal region $\Omega$ in green.

robust MPC (rMPC) approach from [12] for the same example. It is clearly visible from Fig. 2 and Fig. 3 that our reachable sets are smaller than the ones computed with rMPC. Small reachable sets are advantageous because they minimize the probability that the input or state constraints are violated. In addition, there is also a better chance that the sets are located inside the terminal region. Furthermore, the rMPC algorithm exhibits several major disadvantages that our approach is able to avoid: First, it does not provide formal safety guarantees for time-continuous systems, as it only considers time-discretized systems. Second, rMPC directly optimizes over the reachable sets, which leads to large computation times, because the reachable sets have to be calculated for each iteration of the optimization algorithm. To avoid this, we only optimize the center trajectory and compute the reachable sets only once after the optimization. Third, the technique that rMPC uses for reachability analysis results in larger over-approximations of the real reachable set of the system, as their technique is more conservative than our approach.

In order to compare our approach with the rMPC algorithm, we use the same parameters and same initial point as the authors in [12]. However, the example is not really suited for a good comparison of control approaches, because to stabilize the system from this initial point, the maximal available control input has to be applied for nearly the whole time horizon. This does not leave much room for the other objectives like minimization of the cost function or counteracting disturbances. Therefore, we provide a second example for the initial point $x_0 = [-0.3, -30]$. Compared to the case above, we changed the final prediction horizon to $t_N = 9\,s$ and the input weighting matrix to $R_c = 0.9$. The results are displayed in Fig. 4. For

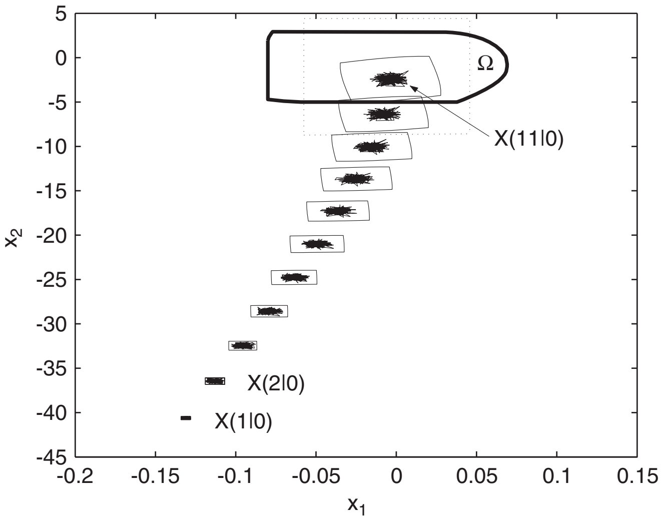


**Figure 3:** Initial solution with reachable sets for the rMPC approach, taken from [12].

this example, the maximum computation time for optimization and reachability analysis is $0.37\,s \leq t_c$ This example nicely demonstrates that our repeated optimization enables finding feasible trajectories that have a lower cost than the initial solution.

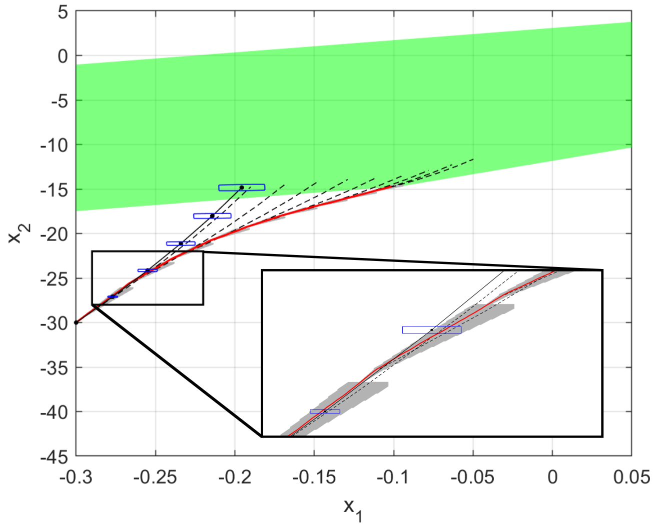


**Figure 4:** Our approach for a different initial point with terminal region Ω (green), center trajectory (solid black) and reachable sets at discrete time points (blue) of the initial solution, center trajectories for all iterations (dashed black), real system trajectory (red), and reachable set for the real system trajectory (gray). The resulting reachable sets can be seen better in the magnified section.

## 2.4 Discussion of the Algorithm

The computational complexity for our optimization is the same as for regular MPC. During operation, we solve the optimal control problem (8)-(13). Since we solve it only for a single state, we can use the same solvers which are developed for solving nonlinear programs and which are used for existing MPC. The only additional computation effort is the reachability analysis [4], which has a complexity of $\mathcal{O}(n^3)$, with $n$ denoting the dimension of the state space.

Because this computation only has to be performed once for the whole time horizon, we are able to do these computation in real-time, as shown in the numerical example. Since we do not optimize over the reachable sets and therefore are not able to obtain a global optimal solution (which is not feasible for nonlinear programs in general), we save a lot of computation time while still guaranteeing safety.

An advantage of our approach is that any kind of feedback controller can be used to track the center trajectory and counteract disturbances. It is also not necessary to compute the invariant set or some contraction set which has to hold everywhere in the state space. Instead, we compute the actual reachable set based on the predicted future situation, resulting in a less conservative solution.

# 3 Probabilistic Reach Sets in Stochastic Model Predictive Control

In this section, we address stochastic model predictive control (MPC) for a discrete time linear system affected by an additive stochastic disturbance with possibly unbounded support. At each time instant, a finite horizon constrained optimization problem needs to be solved so as to determine the control input to be applied at that time instant. Constraints generally involve state and input, and state constraints are necessarily enforced in probability if the disturbance has unbounded support and the state is required to remain in a bounded set. More precisely, the controlled reach set is required to keep within some constraint set for all disturbance realizations except for a set of probability $\varepsilon$ (probabilistic reach set of level 1-$\varepsilon$). By adopting the so-called scenario approach [23], an inner approximation – achieved through randomization – of the probabilistic reach sets of level 1-$\varepsilon$ is used for input design. The number of disturbance realizations used to approximate the reach sets can be appropriately set so as to guarantee that the resulting scenario solution satisfies constraints at any step with a probability that is smaller than $\varepsilon$ (chance-constrained solution), with high confidence.

The resulting policy can be implemented in a receding horizon fashion according to the MPC strategy. Such a possibility, however, is hampered by the fact that a feasibility issue may arise when recomputing the policy. Infeasibility indeed can occur if the disturbance has unbounded support and the state is required to remain in a bounded set.

In this section, we describe a solution to this issue that is based on the introduction of a constraint relaxation which is effective only when the original problem turns out to be unfeasible. This is obtained via a cascade of two probabilistically-constrained optimization problems where, in the first one, performance is neglected and the policy is designed to fully recover feasibility or –if this is not possible– to determine the minimum level of relaxation to get feasibility, and, in the second one, such a minimum relaxation level is imposed while optimally (re-)tuning the control policy parameters. Both problems are solved through a computationally tractable scenario-based scheme using the same finite number of disturbance realizations and providing an approximate solution that satisfies the original probabilistic constraints of the cascade, with high confidence. A simulation example shows the effectiveness of the proposed approach.

In [24, 26] stochastic uncertainty with bounded support is tackled by means of suitable probabilistic tubes, whereas in [48] constraint tightening is adopted to enforce recursive feasibility in MPC, always under the assumption of a bounded disturbance. In the case of systems affected by stochastic disturbance with *unbounded support*, control problems in presence of state constraints have been addressed in [6, 41, 8, 55, 60, 61, 10, 27]. In [6, 60, 61], state constraints are dealt with by means of a penalization term accounting for the state constraint violation so as to ensure feasibility. In [41, 55, 10, 27], an analytic convex relaxation of probabilistic constraints is proposed, whereas in [8] the problem is reformulated considering a bounded disturbance obtained by suitably cutting the tails of the disturbance distribution. In all these approaches, the disturbance is assumed to be a sequence of i.i.d. (independent and identically distributed) random variables. Many of them also assume that the disturbance has a Gaussian distribution, [6, 41, 8, 55, 10, 27].

Here, we address the unbounded disturbance case and, differently from the mentioned approaches, we do make no independence and Gaussianity assumptions. We adopt a scenario-based approach, which allows to address design in the presence of uncertainty, making solvable problems that were otherwise deemed computationally intractable, [73].

Our previous contribution [30] addresses the same set-up but recovers feasibility by either adding a term penalizing state constraint violation to the cost or introducing a certain predefined admissible deterioration of the system performance while relaxing the state constraints. The main advantage of the approach proposed here is that constraint relaxation is set to a minimal level needed to recover feasibility so as to avoid penalizing excessively performance. In particular, no relaxation is introduced if the randomized problem is feasible, which is not the case in [29, 30]. Other randomized approaches to constrained stochastic control for system (21) have been proposed in [16, 66, 17] but under the assumption that the noise has bounded support, whereas in [59] only input constraints are considered, and in [67, 78] recursive feasibility is assumed.

Scenario-based MPC was originally introduced for solving problems where achieving robustness is not feasible and a chance-constrained reformulation is needed because of the unboundedness of the disturbance support. However, given that the smaller the threshold $\varepsilon$ chosen by the user, the closer the scenario solution is to the robust one, the scenario approach could also be used as a heuristic method to find approximate (relaxed) solutions to robust MPC problems when the support of the disturbance is bounded. A comparison with computational approaches providing a solution that is robustly guaranteed shows that scenario-based may lead to an effective solution - whose robustness can be experimentally verified - even for tight constraints, whereas robust MPC methods may show some conservatism and require the constraints to be loose in order to be feasible. This enhances the use of randomized-based methods as a valid alternative to other approaches to robust MPC.

Notably, scenario-based MPC can be extended to nonlinear systems that can be feedback linearized and to set-ups with discrete inputs. The interested reader is referred to [31] and [13], respectively. In [31] and [13], filtering is adopted to account for observations and update the uncertainty distribution from which realizations are extracted so as to obtain a better tuned policy.

## 3.1 Problem Formulation

Consider the system

$$x_{t+1} = Ax_t + Bu_t + B_w w_t, \tag{21}$$

where $x_t \in \mathbb{R}^n$ is the state, $u_t \in \mathbb{R}^m$ is the control input and $w_t \in \mathbb{R}^{n_w}$ is an additive stochastic disturbance. Matrices $A$, $B$, and $B_w$ have appropriate dimensions so as to make (21) consistent. The probability distribution of $w_t$ is assumed to be known and it may have an unbounded support. Without loss of generality, we assume that $n_w \leq n$ and $B_w$ is full column rank. The state is accessible, i.e., at every $t$ a noise-free measurement of $x_t$ becomes available.

The following disturbance feedback parametrization for the control input is adopted:

$$u_t = \gamma_t + \sum_{\tau=0}^{t-1} \theta_{t,\tau} w_\tau, \tag{22}$$

where $\gamma_t \in \mathbb{R}^m$ represent open-loop terms, while $\theta_{t,\tau} \in \mathbb{R}^{m \times n_w}$ are the disturbance feedback gains. Note that the stochastic disturbance $w_\tau$ appearing in (22) can be recovered from the measurements of the state according to

$$w_\tau = B_w^\dagger (x_{\tau+1} - Ax_\tau - Bu_\tau), \tag{23}$$

where $B_w^\dagger$ denotes the pseudo-inverse of $B_w$. This expression reveals that the disturbance feedback control policy in (22) is in fact a state feedback control policy. Parametrization (22) was first proposed in [37], where it was shown that the family of policies in (22) is indeed equivalent to the family of affine state feedback policies $u_t = \tilde{\gamma}_t + \sum_{\tau=0}^{t} \tilde{\theta}_{t,\tau} x_\tau$. To be precise, for every choice of $\tilde{\gamma}_t, \tilde{\theta}_{t,\tau}$ there exists a parametrization $\gamma_t, \theta_{t,\tau}$ in (22) returning the same control action, and vice-versa. The great advantage of (22) is that, differently from other parameterizations, the input $u_t$ and the state $x_t$ are affine functions of the design parameters $\gamma_t$ and $\theta_{t,\tau}$, which yields clear computational benefits.

The objective is to design the parameters $\gamma_t$ and $\theta_{t,\tau}$ so as to minimize a cost function over a finite time horizon of length $M$, while accounting for constraints on the input and state variables. This problem may arise per-se in some applications (for instance, the positioning of the end-effector of an industrial robot equipped with a robot re-initialization device), but its significance mainly lies in the fact that it can be adopted in a Model Predictive Control (MPC) scheme, where it is repeatedly solved at every time step, [50, 18, 64, 45].

In our formulation, we admit as cost any strictly convex function $J$ of the parameters $\gamma_t$ and $\theta_{t,\tau}$ over the horizon $0, 1, \dots, M-1$. Plainly, the most common situations is when $J$ is defined as a function of the input and the state. A typical choice is the average quadratic cost

$$J = \mathbb{E}\left[\sum_{t=1}^{M} x_t^T Q x_t + \sum_{t=0}^{M-1} u_t^T R u_t\right], \tag{24}$$

where $Q$ and $R$ are symmetric and positive semi-definite matrices, and $\mathbb{E}$ denotes expectation with respect to the underlying probability distribution as induced by the (known) distribution of the disturbance. In this case, a sufficient condition for strict convexity to hold is that matrices $R$ and $\mathbb{E}[\mathbf{w}\mathbf{w}^T]$ are positive definite, see [30].

As for the input and state constraints, we assume that they are expressed as

$$f(u_0, \dots, u_{M-1}) \leq 0 \ \wedge \ g(x_1, \dots, x_M, u_0, \dots, u_{M-1}) \leq 0 \tag{25}$$

where $\wedge$ stands for “and”, $f : \mathbb{R}^{mM} \to \mathbb{R}^{p_u}$ and $g : \mathbb{R}^{(n+m)M} \to \mathbb{R}^{p_y}$ are continuous convex vector-valued functions, and the inequalities are meant component-wise. It is assumed that the admissible domain for $u_0, \dots, u_{M-1}$ and $x_1, \dots, x_M$ as given by (25) is nonempty with nonempty interior. For example, a typical requirement is that the norm of the input and of some output variable are kept within an admissible range at each time instant $t$ along the reference time horizon, i.e.,

$$f(u_0, \dots, u_{M-1}) = \begin{bmatrix} \|u_0\| \\ \vdots \\ \|u_{M-1}\| \end{bmatrix} - \bar{u}$$

$$g(x_1, \dots, x_M, u_0, \dots, u_{M-1}) = \begin{bmatrix} \|Cx_1\| \\ \vdots \\ \|Cx_M\| \end{bmatrix} - \bar{y},$$

where the vectors $\bar{u}$ and $\bar{y}$ defines the maximum allowed magnitude at each time instant and $\|\cdot\|$ denotes some norm of interest. Note that $g$ allows for joint state and input constraints along the temporal horizon of interest and the constraints expressed by $f$ could be incorporated in $g$. To ease further explanations, we however keep the constraints that depend on the input only separate from the others.

It should be noted that constraints (25) cannot be directly imposed since they miss to specify how to account for the presence of the stochastic disturbance affecting both the state and the input variables. Since the disturbance support is possibly unbounded, we assume that constraints are enforced probabilistically, namely, constraints (25) are required to hold with a certain (usually high) probability $1-\epsilon$, where $\epsilon \in (0,1)$ is a user-chosen parameter:

$$\begin{aligned} & \mathbb{P}\{f(u_0,\dots,u_{M-1}) \leq 0 \ \wedge g(x_1,\dots,x_M,u_0,\dots,u_{M-1}) \leq 0\} \\ & \geq 1-\epsilon. \end{aligned} \tag{26}$$

This probabilistic formulation of constraints is the most natural for many problems of interest and has actually become common in the recent literature on constrained stochastic control, [8, 55, 60, 61, 10, 27, 24, 35, 36, 48].

Altogether, the optimal design problem we are considering is as follows:

$$\min_{\gamma_i,\theta_{i,j}} \ J \text{ subject to (26).} \tag{27}$$

Note that the probabilistic constraint $\mathbb{P}\{f(u_0,\dots,u_{M-1}) \leq 0\} \geq 1-\epsilon$, where $g$ is not present, is always feasible, because, if needed, the disturbance feedback gains $\theta_{t,\tau} \in \mathbb{R}^{m \times n_w}$ in (22) can be set to zero, which makes $u_t$ deterministic. On the contrary, a feasibility issue arises precisely because of the presence of the requirement on $g(x_1,\dots,x_M,u_0,\dots,u_{M-1})$. As a matter of fact, the stochastic disturbance $w_t$ enters additively the system dynamics, and, since the input $u_t$ depends on the disturbance up to time $t-1$ at most, the dependence of $x_{t+1}$ on $w_t$ cannot be canceled. Since $w_t$ has possibly unbounded support and given the limitation imposed by the system dynamics and by the constraints on the input variable, it may then be that, depending on the system initialization $x_0$, no choice of $\gamma_t, \theta_{t,\tau}$ exists such that $g(x_1,\dots,x_M,u_0,\dots,u_{M-1}) \leq 0$ is attained with the required probabilistic level. In particular, when the noise is Gaussian and $B_w$ is the identity matrix, the state will exit any fixed bounded set with probability 1 as the time horizon length grows unbounded, [42].

The feasibility problem here discussed is severe because in many cases the designer has no direct control on the system initialization, which is indeed determined by exogenous causes. For example, in an MPC scheme where the optimization problem (27) is continuously repeated at each time step over a receding horizon and only the first calculated control action is actually implemented, the system initialization for a given time horizon is determined by the solutions at previous steps. At these previous steps, however, since constraints are only probabilistically enforced and since the disturbance has unbounded support, it may be that an unfortunate realization of $w_t$ drives the state far away in a region where the state constraint is strongly violated, so that no feasible control action exists to steer the state back in the region where $g(x_1,\dots,x_M,u_0,\dots,u_{M-1}) \leq 0$ holds with the required probability.

Our objective now is addressing the feasibility issue illustrated above by introducing a suitable relaxation of problem (27), which is conceived so as to adhere to the intent of the original problem formulation (27) as much as possible. Precisely, whenever the original constraint (26) is feasible, the original problem is maintained, while, otherwise, a new decision problem is formulated by relaxing the condition $g(x_1, \ldots, x_M, u_0, \ldots, u_{M-1}) \leq 0$ only for those components of the vector inequality that need to be relaxed to get feasibility. This reformulation leads to a cascade of two optimization problems with probabilistic constraints, which, admittedly, can be very difficult to solve in general, since problems involving probabilistic constraints can be NP-hard. The second contribution of this work is that of introducing a resolution scheme based on randomization in order to enhance computational tractability. Specifically, we resort to the so-called scenario approach, [14, 15, 20, 23], a recently introduced randomized method that can be used to provide approximate solutions to problems with probabilistic constraints establishing a precise link between the original problem and its approximation. Such a link is extended here to the scenario solution to the cascade of problems discussed above, which is a non-standard setup not fully covered by the available literature (see [51] for a contribution on cascading optimization).

### Compact Notation

In order to ease the notation we define:

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_M \end{bmatrix} \quad \mathbf{u} = \begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ u_{M-1} \end{bmatrix} \quad \mathbf{w} = \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_{M-1} \end{bmatrix}.$$

Then, the state vector can be calculated as:

$$\mathbf{x} = \mathbf{F}x_0 + \mathbf{G}\mathbf{u} + \mathbf{H}\mathbf{w}, \tag{28}$$

where matrices $\mathbf{F}$, $\mathbf{G}$ and $\mathbf{H}$ are given by

$$\mathbf{F} = \begin{bmatrix} A \\ A^2 \\ \vdots \\ A^M \end{bmatrix} \quad \mathbf{G} = \begin{bmatrix} B & 0_{n\times m} & \cdots & 0_{n\times m} \\ AB & B & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0_{n\times m} \\ A^{M-1}B & \cdots & AB & B \end{bmatrix}$$

$$\mathbf{H} = \begin{bmatrix} B_w & 0_{n\times n_w} & \cdots & 0_{n\times n_w} \\ AB_w & B_w & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0_{n\times n_w} \\ A^{M-1}B_w & \cdots & AB_w & B_w \end{bmatrix}.$$

Similarly, the disturbance feedback policy (22) can be rewritten in the following compact form

$$\mathbf{u} = \Gamma + \Theta \mathbf{w}, \tag{29}$$

where we let

$$\Gamma = \begin{bmatrix} \gamma_0 \\ \gamma_1 \\ \vdots \\ \gamma_{M-1} \end{bmatrix} \Theta = \begin{bmatrix} 0_{m\times n_w} & 0_{m\times n_w} & \cdots & 0_{m\times n_w} \\ \theta_{1,0} & 0_{m\times n_w} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0_{m\times n_w} \\ \theta_{M-1,0} & \cdots & \theta_{M-1,M-2} & 0_{m\times n_w} \end{bmatrix}.$$

By substituting the expression of the input in (29) into (28), the affine dependence of $\mathbf{x}$ on the design parameters $\Gamma$ and $\Theta$ becomes clear:

$$\mathbf{x} = \mathbf{F}x_0 + \mathbf{G}\Gamma + (\mathbf{G}\Theta + \mathbf{H})\mathbf{w}$$

Eventually, the nonzero components of $\Gamma$ and $\Theta$ are collected in the vector of optimization variables $\lambda$, so that the following notations can be adopted: $\mathbf{u} = \mathbf{u}_\lambda(\mathbf{w})$, $\mathbf{x} = \mathbf{x}_\lambda(\mathbf{w})$, and $J = J(\lambda)$, which point out the dependence of input, state, and cost on the optimization vector $\lambda$ and the disturbance realization $\mathbf{w}$. The constraints in (26) then become

$$\mathbb{P}\{f(\mathbf{u}_\lambda(\mathbf{w})) \leq 0 \ \wedge \ g(\mathbf{x}_\lambda(\mathbf{w}), \mathbf{u}_\lambda(\mathbf{w})) \leq 0\} \geq 1 - \epsilon.$$

### 3.2 State Constraint Relaxation to Ensure Feasibility

In order to recover feasibility, we introduce a relaxation of the condition $g(\mathbf{x}_\lambda(\mathbf{w}), \mathbf{u}_\lambda(\mathbf{w})) \leq 0$ by substituting its right-hand side with $h \in \mathbb{R}^{p_y}$, $h$ being a new optimization variable. By doing this, the constraint involving state variables turns out to be always feasible, since it is enough to take the variable $h$ large enough. On the other hand, large values for $h$ are clearly not desired since the bigger $h$ the larger the deviation from the original constraint. To stick to the original problem formulation as much as possible $h$ should be minimized component-wise. On the other hand, one should account for the minimization of the cost function $J(\lambda)$, which represents the system performance. To this purpose, the following cascade of optimization programs (two-step approach) is proposed, where $L(h)$ is an user-chosen strictly convex function of $h$, that is positive definite at $h = 0$ (i.e., $L(h) > 0$, $h \neq 0$ and $L(0) = 0$):

$$\min_{\lambda, h \geq 0} \ L(h) \text{ subject to:} \tag{30}$$

$$\mathbb{P}\{f(\mathbf{u}_\lambda(\mathbf{w})) \leq 0 \ \wedge \ g(\mathbf{x}_\lambda(\mathbf{w}), \mathbf{u}_\lambda(\mathbf{w})) \leq h\} \geq 1 - \epsilon,$$

$$\min_{\lambda} \ J(\lambda) \text{ subject to:} \tag{31}$$

$$\mathbb{P}\left\{f(\mathbf{u}_\lambda(\mathbf{w})) \leq 0 \wedge g(\mathbf{x}_\lambda(\mathbf{w}), \mathbf{u}_\lambda(\mathbf{w})) \leq h^o\right\} \geq 1 - \epsilon,$$

where $h^o$ is the optimal value for $h$ obtained in (30). The optimal value for $\lambda$ obtained from (31) is denoted by $\lambda^o$.

Problem (30) in the first step aims at determining the smallest value of $h$, according to the cost $L(h)$, that ensures the feasibility of the probabilistic constraint

$$\mathbb{P}\{f(\mathbf{u}_\lambda(\mathbf{w})) \leq 0 \ \wedge \ g(\mathbf{x}_\lambda(\mathbf{w}), \mathbf{u}_\lambda(\mathbf{w})) \leq h\} \geq 1 - \epsilon.$$

A possible choice for the cost function $L(h)$ is e.g. $L(h) = h^T T h$, which allows to assign a different importance to each component of $h$ by properly choosing the positive definite matrix $T$. Note that since the cost function $L(h)$ does not depend on $\lambda$, it may happen that the optimal cost $L(h^o)$ is achieved in correspondence of different choices for $\lambda$, each of them leading to a possibly different value of $J(\lambda)$. The second step optimization problem (31) then exploits this degree of freedom to minimize the performance cost. To this purpose, $J(\lambda)$ is minimized while the relaxed constraint $\mathbb{P}\left\{f(\mathbf{u}_\lambda(\mathbf{w})) \leq 0 \wedge g(\mathbf{x}_\lambda(\mathbf{w}), \mathbf{u}_\lambda(\mathbf{w})) \leq h^o\right\} \geq 1 - \epsilon$ is enforced. Since the bound on the state condition $g(\mathbf{x}_\lambda(\mathbf{w}), \mathbf{u}_\lambda(\mathbf{w}))$ is fixed to $h^o$ as computed in the previous step, problem (31) does not suffer from any feasibility issue.

The cascade of problems is conceived so that, when the probabilistic constraint in (27) is infeasible, the control action is basically designed according to (30) so as to recover feasibility (minimization of $L(h)$). In this case, (31) provides just a refinement of the solution. The requirement $h \geq 0$ in (30) ensures that the constraint relaxation in (31), component by component, cannot become tighter than the original constraint in (27), and for those components not requiring any relaxation (31) pursues the goal of minimizing $J(\lambda)$ as in (27). In particular, whenever (27) is already feasible, program (30) simply returns $h^o = 0$ and the original problem (27) is recovered in (31).

Overall, the cascade of problems (30) and (31) returns a solution given by the pair $(\lambda^o, h^o)$, where $\lambda^o$ determines the control action to be implemented and $h^o$ is the probabilistically guaranteed bound for the state constraint. Note that the value $h^o$ computed in the first step optimization problem can be inspected to evaluate the mismatch with respect to the original state constraint.

## 3.3 Scenario-based Resolution Scheme

As problems (30) and (31) are, in general, hard to solve because of the presence of a probabilistic constraint, we propose to tackle them by means of a sample-based scheme which is in the vein of the so-called scenario approach, [14, 15, 20, 23]. The proposed scheme allows to recover computational tractability at the price of introducing some approximation. However, by exploiting the scenario approach, which is here extended to the cascade of problems (30) and (31), precise probabilistic guarantees on the feasibility of the achieved solution are also provided.

The idea of the scenario approach is to consider $N$ disturbance realizations of length $M$:

$$\mathbf{w}^{(k)} = \left[ w_0^{(k)} \; w_1^{(k)} \; \dots \; w_{M-1}^{(k)} \right], \; k = 1, \dots, N,$$

each one extracted according to the disturbance probability distribution. Then, the probabilistic constraint in (30) and (31) are replaced by $N$ non-probabilistic constraints, one for each disturbance realizations. More precisely, we have the following cascade of problems that can be seen as a sample-based counterpart of the cascade of problems (30) and (31):

$$\begin{aligned} & \min_{\lambda, h \geq 0} L(h) \text{ subject to:} && (32) \\ & f(\mathbf{u}_\lambda(\mathbf{w}^{(k)})) \leq 0, \; k = 1, \dots, N, \\ & g(\mathbf{x}_\lambda(\mathbf{w}^{(k)})), \mathbf{u}_\lambda(\mathbf{w}^{(k)})) \leq h, \; k = 1, \dots, N, \\ & \min_{\lambda} J(\lambda) \text{ subject to:} && (33) \\ & f(\mathbf{u}_\lambda(\mathbf{w}^{(k)})) \leq 0, \; k = 1, \dots, N, \\ & g(\mathbf{x}_\lambda(\mathbf{w}^{(k)})), \mathbf{u}_\lambda(\mathbf{w}^{(k)})) \leq h^\star, \; k = 1, \dots, N, \end{aligned}$$

where $h^\star$ is the optimal value of $h$ obtained in (32). The optimal value for $\lambda$ obtained from (33) is denote by $\lambda^\star$.

Problems (32) and (33) are convex and have a finite number of constraints, hence they can be efficiently solved by means of standard solvers. Note that as the constraints are convex and the cost function $L(h)$ is strictly convex with respect to its argument, problem (32) uniquely determines the value of $h^\star$; similarly, thanks to the strict convexity of $J(\lambda)$, the solution to problem (33) is unique.

The same interpretation we had for the cascade of problems (30) and (31) in Section 3.2 applies to the cascade of problems (32) and (33): indeed, the solution of the latter cascade defined by the pair $(\lambda^\star, h^\star)$ is the empirical counterpart of the solution of the former. It is worth noticing that, as the pair $(\lambda^\star, h^\star)$ is feasible and optimal for (32), the second step optimization problem (33) can be regarded as a tie break rule by means of which the solution that minimizes the cost $J(\lambda)$ is chosen among the possible multiple solutions in $\lambda$ of the first step optimization problem (32).

We are now interested in studying the feasibility of the obtained scenario-based solution for the probabilistic constraint

$$\mathbb{P}\left\{ f(\mathbf{u}_\lambda(\mathbf{w})) \leq 0 \wedge g(\mathbf{x}_\lambda(\mathbf{w}), \mathbf{u}_\lambda(\mathbf{w})) \leq h \right\} \geq 1 - \epsilon, \tag{34}$$

so as to provide a connection between $(\lambda^\star, h^\star)$ and the original cascade of problems (30) and (31).

This question pertains to the theory of the scenario approach, which provides in a number of different setups guarantees on the feasibility of the scenario solution for the original probabilistic constraint as long as $N$ is suitably chosen, see e.g. [15, 20, 21, 19, 25, 22]. The tightest result is that of [20] which, however, does not directly apply to the cascade of problems (32) and (33). The results on cascading optimization in [51] apply to this context but the resulting bound on $N$ is conservative. The following theorem provides an extension of the result in [20] to the current framework.

**Theorem 2.** *Let $\beta \in (0,1)$ be a user-chosen confidence parameter. If the number of extracted disturbance realizations $N$ is chosen so as to satisfy*

$$\sum_{i=0}^{d-1} \binom{N}{i} \epsilon^i (1-\epsilon)^{N-i} \leq \beta, \tag{35}$$

*where $d$ is the dimensionality of $(\lambda, h)$, then it holds with confidence at least $1-\beta$ that*

$$\mathbb{P}\left\{f(\mathbf{u}_{\lambda^\star}(\mathbf{w})) \leq 0 \;\wedge\; g(\mathbf{x}_{\lambda^\star}(\mathbf{w}), \mathbf{u}_{\lambda^\star}(\mathbf{w})) \leq h^\star\right\} \geq 1-\epsilon,$$

*where $(\lambda^\star, h^\star)$ is the solution to the cascade of problems (32) and (33).*

**Proof:** For a given $(\lambda, h)$, define the violation probability of $(\lambda, h)$ as

$$\begin{aligned} V(\lambda,h) &:= \mathbb{P}\Big\{f(\mathbf{u}_\lambda(\mathbf{w})) > 0 \;\vee\; g(\mathbf{x}_\lambda(\mathbf{w}), \mathbf{u}_\lambda(\mathbf{w})) > h\Big\} \\ &= 1 - \mathbb{P}\Big\{f(\mathbf{u}_\lambda(\mathbf{w})) \leq 0 \;\wedge\; g(\mathbf{x}_\lambda(\mathbf{w}), \mathbf{u}_\lambda(\mathbf{w})) \leq h\Big\}, \end{aligned}$$

where $\vee$ stands for "or". Then, Theorem 2 amounts to showing that

$$\mathbb{P}^N\{V(\lambda^\star, h^\star) > \epsilon\} \leq \beta, \tag{36}$$

where $\mathbb{P}^N$ is the product probability underlying the independent extraction of the sample $\mathbf{w}^{(1)}, \dots, \mathbf{w}^{(N)}$ based on which the solution $(\lambda^\star, h^\star)$ is computed.

Consider the following auxiliary scenario programs

$$\begin{aligned} &\min_{\lambda, h\geq 0} L(h) + \frac{1}{n} J(\lambda) \text{ subject to:} \\ &f(\mathbf{u}_\lambda(\mathbf{w}^{(k)})) \leq 0,\; k = 1 \dots N, \\ &g(\mathbf{x}_\lambda(\mathbf{w}^{(k)}), \mathbf{u}_\lambda(\mathbf{w}^{(k)})) \leq h,\; k = 1 \dots N, \end{aligned} \tag{37}$$

for $n = 1, 2, \dots$, and denote by $(\lambda_n^\star, h_n^\star)$ its optimal solution, which exists and is unique, since: i. the cost function $L(h) + \frac{1}{n} J(\lambda)$ has compact level sets for every $n \geq 1$ thanks to its strict convexity; ii. the optimization feasibility domain defined by the constraints in (37) is close and nonempty.

The following two properties hold:

1. For every $n \geq 1$, it holds that
$$\mathbb{P}^N\{V(\lambda_n^\star, h_n^\star) > \epsilon\} \leq \beta. \tag{38}$$

2. For every multisample $\mathbf{w}^{(1)}, \ldots, \mathbf{w}^{(N)}$, the solution to (37) converges to the solution to (32) and (33), namely,
$$(\lambda_n^\star, h_n^\star) \to (\lambda^\star, h^\star) \text{ as } n \to \infty. \tag{39}$$

Formal proofs of (38) and (39) are given later on.

We now capitalize on (38) and (39) to show that (36) holds. To this purpose, start by fixing a sample $\mathbf{w}^{(1)}, \ldots, \mathbf{w}^{(N)}$ such that $V(\lambda^\star, h^\star) > \epsilon$, which, we recall, means that

$$\mathbb{P}\Big\{f(\mathbf{u}(\mathbf{w}, \lambda^\star)) > 0 \ \vee \ g(\mathbf{x}(\mathbf{w}, \lambda^\star), \mathbf{u}(\mathbf{w}, \lambda^\star)) > h^\star\Big\} > \epsilon.$$

By continuity of $f$ and $g$, this implies that

$$\mathbb{P}\Big\{f(\mathbf{u}(\mathbf{w}, \lambda)) > 0 \ \vee \ g(\mathbf{x}(\mathbf{w}, \lambda), \mathbf{u}(\mathbf{w}, \lambda)) > h\Big\} > \epsilon,$$

for all $(\lambda, h) : \|(\lambda, h) - (\lambda^\star, h^\star)\| \leq r$ for a radius $r$ small enough, and, since $(\lambda_n^\star, h_n^\star) \to (\lambda^\star, h^\star)$ so that $\|(\lambda, h) - (\lambda^\star, h^\star)\| \leq r$ for all $n$ bigger than a suitable $\bar{n}$, we can conclude that

$$V(\lambda_n^\star, h_n^\star) > \epsilon, \tag{40}$$

for $n > \bar{n}$. If we now let $\mathbf{w}^{(1)}, \ldots, \mathbf{w}^{(N)}$ vary and we consider the indicator function $\mathbb{I}_{\{\mathbf{w}^{(1)}, \ldots, \mathbf{w}^{(N)}:\, V(\lambda_n^\star, h_n^\star) > \epsilon\}}$, then (40) yields

$$\mathbb{I}_{\{V(\lambda^\star, h^\star) > \epsilon\}} \cdot \mathbb{I}_{\{V(\lambda_n^\star, h_n^\star) > \epsilon\}} \xrightarrow[n \to \infty]{} \mathbb{I}_{\{V(\lambda^\star, h^\star) > \epsilon\}},$$

for all possible realizations of $\mathbf{w}^{(1)}, \ldots, \mathbf{w}^{(N)}$. Applying the Lebesgue dominated convergence theorem gives

$$\begin{aligned}
&\lim_{n \to \infty} \mathbb{P}^N\{V(\lambda_n^\star, h_n^\star) > \epsilon\} \\
&= \lim_{n \to \infty} \int \mathbb{I}_{\{V(\lambda_n^\star, h_n^\star) > \epsilon\}} \mathbb{P}^N\{\mathrm{d}\mathbf{w}^{(1)}, \ldots, \mathrm{d}\mathbf{w}^{(N)}\} \\
&\geq \lim_{n \to \infty} \int \mathbb{I}_{\{V(\lambda^\star, h^\star) > \epsilon\}} \cdot \mathbb{I}_{\{V(\lambda_n^\star, h_n^\star) > \epsilon\}} \mathbb{P}^N\{\mathrm{d}\mathbf{w}^{(1)}, \ldots, \mathrm{d}\mathbf{w}^{(N)}\} \\
&= \int \mathbb{I}_{\{V(\lambda^\star, h^\star) > \epsilon\}} \mathbb{P}^N\{\mathrm{d}\mathbf{w}^{(1)}, \ldots, \mathrm{d}\mathbf{w}^{(N)}\} \\
&= \mathbb{P}^N\{V(\lambda^\star, h^\star) > \epsilon\}.
\end{aligned}$$

Hence, $\mathbb{P}^N\{V(\lambda^\star, h^\star) > \epsilon\} \leq \lim_{n\to\infty} \mathbb{P}^N\{V(\lambda_n^\star, h_n^\star) > \epsilon\}$, and since $\mathbb{P}^N\{V(\lambda_n^\star, h_n^\star) > \epsilon\} \leq \beta$, $\forall n$, relation (36) remains proven. ■

The theorem states that with high confidence $1-\beta$ the solution $(\lambda^\star, h^\star)$ achieved solving the scenario cascade of problems (32) and (33) is feasible for the original probabilistic constraint (34) in (30) and (31). Note that the presence of the confidence parameter $\beta$ is intrinsic and is related to fact that the obtained solution depends on the random extraction $\mathbf{w}^{(1)}, \dots, \mathbf{w}^{(N)}$: $\beta$ is needed to account for the possibility that a not representative enough sample $\mathbf{w}^{(1)}, \dots, \mathbf{w}^{(N)}$ is seen. However, by exploiting the results in [3], it can be shown that the number of required samples $N$ according to (35) scales logarithmically with $1/\beta$. Hence $\beta$ can be chosen to be very small such as $10^{-5}$ or $10^{-7}$ without affecting $N$ too much, so that the fact that the achieved solution $(\lambda^\star, h^\star)$ satisfies the probabilistic constraint (34) in (30) and (31) can be taken for granted.

**Proof of** (38) **and** (39)

**Proof of** (38)**:** By adding a slack variable $v \in \mathbb{R}$, problem (37) can be rewritten in epigraphic form as:

$$\begin{aligned}
&\min_{\lambda, h\geq 0, v} \; v \text{ subject to:} \qquad (41)\\
&f(\mathbf{u}_\lambda(\mathbf{w}^{(k)})) \leq 0,\ k = 1\dots N,\\
&g(\mathbf{x}_\lambda(\mathbf{w}^{(k)}), \mathbf{u}_\lambda(\mathbf{w}^{(k)})) \leq h,\ k = 1\dots N,\\
&L(h) + \frac{1}{n} J(\lambda) \leq v.
\end{aligned}$$

The solution to problem (41) is still unique, and the assumptions of Theorem 2.4 in [20] are satisfied. An application of this theorem gives

$$\mathbb{P}^N\{V(\lambda_n^\star, h_n^\star) > \epsilon\} \leq \sum_{i=0}^{d} \binom{N}{i} \epsilon^i (1-\epsilon)^{N-i},$$

where we have $d$ in place of $d-1$ because in (41) the number of optimization variables has been augmented by 1 and is equal to $d+1$. On the other hand, since the slack variable $v$ does not enter the expression defining it, the constraint

$$\{\lambda, h, v:\ f(\mathbf{u}_\lambda(\mathbf{w})) \leq 0 \ \wedge\ g(\mathbf{x}_\lambda(\mathbf{w}), \mathbf{u}_\lambda(\mathbf{w})) \leq h\}$$

is, irrespective of $\mathbf{w}$, a cylindroid infinitely extended along the $v$ direction. This entails that the family (with respect to the variability of $\mathbf{w}$) of constraints above has a so-called support rank equal to $d$, according to Definition 3.6 of [68] (see also [79]). The conclusion that

$$\mathbb{P}^N\{V(\lambda_n^\star, h_n^\star) > \epsilon\} \leq \sum_{i=0}^{d-1} \binom{N}{i} \epsilon^i (1-\epsilon)^{N-i}$$

then follows by invoking the observation made in [68] that Theorem 2.4 of [20] still applies by replacing the optimization domain dimensionality with the support rank (see Lemma 3.8). ■

**Proof of** (39)**:** To show that $(\lambda_n^\star, h_n^\star) \to (\lambda^\star, h^\star)$ as $n \to \infty$, consider the sets

$$\begin{aligned}\mathcal{H}_n = \Big\{(\lambda, h) :\ & (\lambda, h) \text{ is feasible for (37) and} \\ & L(h) + \frac{1}{n}J(\lambda) \le L(h^\star) + \frac{1}{n}J(\lambda^\star)\Big\},\end{aligned}$$

for $n = 1, 2, \ldots$. In words, $n$ by $n$, $\mathcal{H}_n$ is the set of all feasible points for (37) that also belong to the smallest level set of the cost function of (37) containing the solution $(\lambda^\star, h^\star)$ of (32) and (33). Note that, while the level set changes with $n$, the feasibility domain of (37) remains the same for all $n$ and it coincides with the feasibility domain of (32). This entails that $(\lambda^\star, h^\star)$ belongs to $\mathcal{H}_n$ for all $n$, showing also that $\mathcal{H}_n$ is nonempty. Moreover, $n$ by $n$, we have that

$$(\lambda_n^\star, h_n^\star) \in \mathcal{H}_n, \tag{42}$$

because $(\lambda_n^\star, h_n^\star)$ is feasible for (37), and, being also optimal, its cost value must be better than that of $(\lambda^\star, h^\star)$, which is the second condition defining $\mathcal{H}_n$.
A fundamental property of the family of sets $\mathcal{H}_n$ is that

$$\mathcal{H}_1 \supseteq \mathcal{H}_2 \supseteq \cdots \supseteq \mathcal{H}_n \supseteq \mathcal{H}_{n+1} \supseteq \cdots, \tag{43}$$

as pictorially depicted in Fig. 5. To show (43), suppose that a $(\bar\lambda, \bar h)$ belongs to $\mathcal{H}_{n+1}$. From

$$L(\bar h) + \frac{1}{n+1}J(\bar\lambda) \le L(h^\star) + \frac{1}{n+1}J(\lambda^\star)$$

it follows that $J(\bar\lambda) \le (n+1)(L(h^\star) - L(\bar h)) + J(\lambda^\star)$. Whence,

$$\begin{aligned}L(\bar h) + \frac{1}{n}J(\bar\lambda) &\le L(\bar h) + \frac{n+1}{n}(L(h^\star) - L(\bar h)) + \frac{1}{n}J(\lambda^\star) \\ &= L(h^\star) + \frac{1}{n}(L(h^\star) - L(\bar h)) + \frac{1}{n}J(\lambda^\star) \\ &\le L(h^\star) + \frac{1}{n}J(\lambda^\star),\end{aligned}$$

where the last inequality follows because $L(h^\star) - L(\bar h) \le 0$ being $L(h^\star)$ the lowest among feasible points by the definition of $h^\star$. This shows that $(\bar\lambda, \bar h) \in \mathcal{H}_n$ too, that is, (43) holds.
From (42) and (43), we have that $(\lambda_n^\star, h_n^\star) \in \mathcal{H}_1$, $\forall n$. Set $\mathcal{H}_1$ is compact, being the intersection of the feasibility domain of (32), which is close, with a level set of $L(h) + \frac{1}{n}J(\lambda)$, which is compact thanks to the assumptions of strict of convexity of $L$ and $J$. It then follows that the sequence $(\lambda_n^\star, h_n^\star)$ have limit points, which are feasible for (32). For simplicity, assume that there is just one, say $(\lambda_\infty^\star, h_\infty^\star)$, so that the sequence $(\lambda_n^\star, h_n^\star)$ is convergent to $(\lambda_\infty^\star, h_\infty^\star)$.

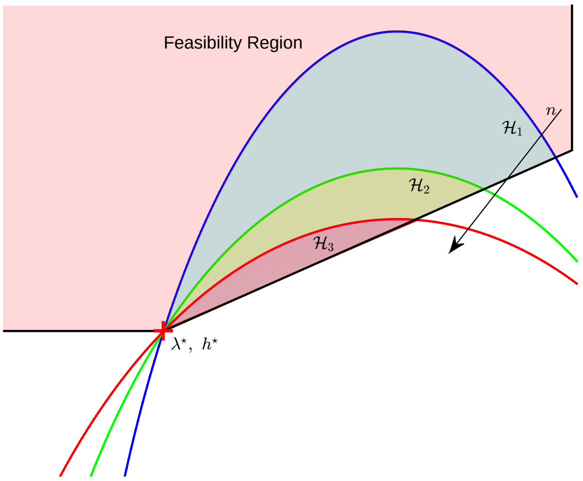


**Figure 5:** The sets $\mathcal{H}_n$'s in a simple case ($h, \lambda \in \mathbb{R}$, $L(h) = h^2$, $J(\lambda) = 3\lambda^2$).

If not, simply repeat the argument that follows to each limit point and the corresponding convergent subsequence.

From (42) and the definition of $\mathcal{H}_n$, we have that

$$L(h_n^\star) \leq L(h^\star) + \frac{1}{n}\left[J(\lambda^\star) - J(\lambda_n^\star)\right],$$

which in turn implies that

$$L(h_\infty^\star) \quad = \quad \lim_{n\to\infty} L(h_n^\star) \leq L(h^\star) + \lim_{n\to\infty} \frac{1}{n}\left[J(\lambda^\star) - J(\lambda_n^\star)\right] = L(h^\star).$$

Yet, being $L(h^\star)$ minimal, it cannot be that a strict inequality holds, so that eventually $L(h_\infty^\star) = L(h^\star)$. If $h_\infty^\star \neq h^\star$, then $(\frac{1}{2}\lambda^\star + \frac{1}{2}\lambda_\infty^\star, \frac{1}{2}h^\star + \frac{1}{2}h_\infty^\star)$ would be feasible for (32) thanks to the convexity of the feasible domain, while the strict convexity of $L(h)$ would give

$$L\left(\frac{1}{2}h^\star + \frac{1}{2}h_\infty^\star\right) \quad < \quad \frac{1}{2}L(h^\star) + \frac{1}{2}L(h_\infty^\star) = L(h^\star),$$

so contradicting the minimality of $L(h^\star)$. Hence, $h_\infty^\star = h^\star$.

From $(\lambda_n^\star, h_n^\star) \in \mathcal{H}_1$, we have that $J(\lambda_n^\star) \leq L(h^\star) - L(h_n^\star) + J(\lambda^\star)$ which, taking the limit, gives

$$J(\lambda_\infty^\star) \quad \leq \quad \lim_{n\to\infty} L(h^\star) - L(h_n^\star) + J(\lambda^\star) = J(\lambda^\star).$$

Plainly, it must be that $J(\lambda_\infty^\star) = J(\lambda^\star)$, for, otherwise, being $\lambda_\infty^\star$ feasible for (33), $J(\lambda_\infty^\star) < J(\lambda^\star)$ would contradict the minimality of $J(\lambda^\star)$. Moreover, if $\lambda_\infty^\star \neq \lambda^\star$, then $\frac{1}{2}\lambda^\star + \frac{1}{2}\lambda_\infty^\star$ would be feasible for (33), and, because of the strict convexity of $J(\lambda)$ we would have

$$J(\frac{1}{2}\lambda^\star + \frac{1}{2}\lambda_\infty^\star) < \frac{1}{2}J(\lambda^\star) + \frac{1}{2}J(\lambda_\infty^\star) = J(\lambda^\star),$$

contradicting again the minimality of $J(\lambda^\star)$. Hence, $\lambda_\infty^\star = \lambda^\star$, and this concludes the proof of Property 2. ■

### 3.4 Numerical Example

In this subsection we apply the proposed approach to a numerical example inspired by [27].

The mechanical system composed by 4 masses and 4 springs depicted in Fig. 6 is considered. Masses and stiffness coefficients of springs are all equal to 1. The state of the system is formed by the displacements of masses with respect to nominal positions and their derivatives, that is, $x = [d_1,\ d_2,\ d_3,\ d_4,\ \dot{d}_1,\ \dot{d}_2,\ \dot{d}_3,\ \dot{d}_4]^T$. The control inputs $u_1$, $u_2$, $u_3$ are instead the forces acting on the masses shown in Fig. 6. The continuous-time system equations are easily derived. The system dynamics is then discretized assuming that the input is kept constant in the interval $[t,\ t+T_s)$, with $T_s = 1$ s, so obtaining a system as in (21). A stochastic additive disturbance affecting both the masses displacements and speeds is supposed to be also present, resulting after discretization in $w \sim WGN(0,\ I_4)$ and $B_w = [0.5I_4\ I_4]^T$. The initial condition of the system is $x_0 = [10,\ -10,\ 10,\ -10,\ 0,\ 0,\ 0,\ 0]^T$.

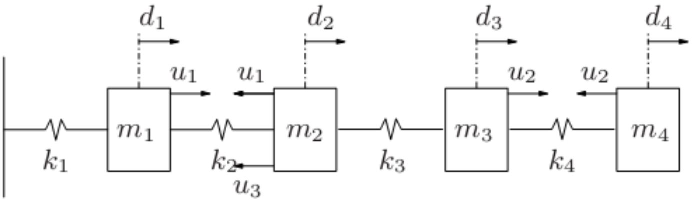


**Figure 6:** Scheme of the mechanical system.

The goal is to keep the masses close to their nominal position counteracting the action of the disturbance. We do not consider constraints on the input, but we enforce a state constraint requiring that the maximum speed of each mass keeps below a given bound.

To this purpose, we choose an average quadratic cost function as in (24) over a finite horizon $M = 8$, where the matrices $Q$ and $R$ are set so as to penalize deviations from the nominal positions:

$$Q = \begin{bmatrix} I_4 & 0_{4\times4} \\ 0_{4\times4} & 0_{4\times4} \end{bmatrix} \qquad R = 10^{-6} I_3,$$

The constraints on the speed of the masses are instead formulated as follows:

$$\|Cx_i\|_\infty \leq 10 \qquad i = 1, \ldots, M, \tag{44}$$

where $C = [0_{4\times4}\ I_4]$. To deal with the presence of the disturbance $w$, the constraint is enforced in probability with $\epsilon = 0.1$.

Following the approach of Section 3.2, the optimization variables $h_i$, $i = 1, \ldots, M$, are introduced so as to ensure feasibility of the probabilistic constraint. We set $\beta = 10^{-6}$ resulting in $N = 4614$ according to (35). Eventually, the cascade of problems (32) and (33) is solved with $L(h) = h^T h$.

Numerical results show that the bound on the state constraint cannot be enforced for the first 2 time steps. Indeed solving problem (32) gave that the smallest bound preserving feasibility is $h_1^\star = 1.62$, $h_2^\star = 1.08$, while for the other time steps we had $h_i^\star = 0$, $i = 3, \dots, M$. The cost $J(\lambda^\star)$ achieved solving problem (33) was 2305.55. Some Monte-Carlo simulations revealed that the probabilistic constraint (34) was satisfied by the achieved solution $(\lambda^\star, h^\star)$ as it was expected given Theorem 2.

In order to better evaluate the performance of the obtained scenario control policy, we compared it against a finite horizon LQ controller, which was designed according to the following cost function:

$$\begin{aligned} J_{LQ} &= \mathbb{E}\left[\sum_{t=1}^{M} x_t^T Q_{LQ} x_t + \sum_{t=0}^{M-1} u_t^T R_{LQ} u_t\right], \\ Q_{LQ} &= \begin{bmatrix} q_J I_4 & 0_{4\times 4} \\ 0_{4\times 4} & q_L I_4 \end{bmatrix} \qquad R_{LQ} = 10^{-6} I_3, \end{aligned}$$

where the weights $q_J$ and $q_L$ are degrees of freedom to tune the relative importance between displacements and speeds. In this way the LQ controller can partially account for the requirement on the masses speed.

The comparison of the performance of the scenario-based control policy and that of the LQ controller for different choices of $q_J$ and $q_L$ is displayed in Table 1, which reports the achieved cost $J$ and the actual probability of violation $\tilde{\epsilon}$ of the original state constraint (44) (computed via Monte Carlo simulations).

**Table 1**

| $q_J$ | $q_L$ | Approach | $J$ | $\tilde{\epsilon}$ |
|---|---|---|---|---|
| - | - | Scenario-based | 2305.55 | 0.1248 |
| 1 | 0 | LQ | 126.44 | 1 |
| 0 | 1 | LQ | 4347.20 | 0.9724 |
| 0.2 | 9 | LQ | 2318.50 | 0.9960 |

As it can be seen, by means of the scenario-based approach a good trade-off between the cost function $J$ and the violation $\tilde{\epsilon}$ can be achieved. In particular, though the required value of $\epsilon = 0.1$ is not achieved (as it turned out to be infeasible), the actual violation $\tilde{\epsilon}$ results quite close to the desired one, because $h_1^\star$ and $h_2^\star$ have been properly pushed toward 0 by the

the first program in the cascade of problems (32) and (33).

As for the LQ controller, instead, when only the mass displacements are accounted for ($q_J = 1$ $q_L = 0$) the achieved cost function $J$ is much improved, but, on the other hand, the state constraint is violated by a huge extent. When, instead, in the design of the LQ controller only the speeds of the masses are accounted for ($q_J = 0$ $q_L = 1$), the cost function $J$ turns out to be significantly increased with respect to the one obtained by the scenario-based controller. Moreover the speed constraint turns out to be violated with large probability, because the control action tends to excessively reduce the speed at subsequent time steps while it maintains a high speed at the first time instant. In a third design the LQ ($q_J = 0.2$ $q_L = 9$), weights are chosen so as to obtain a performance $J$ similar to the one obtained by the scenario-based controller. Also in this case, however, the probability of constraint violation is high, because the speed constraint is significantly violated in the first time instant, while the masses speeds are excessively reduced in the subsequent time instants.

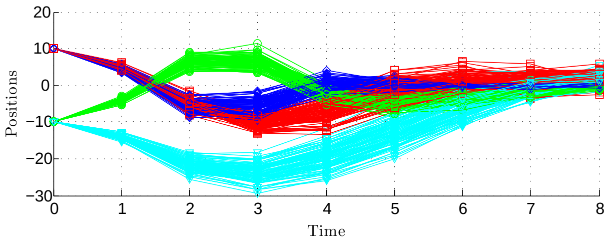


**Figure 7:** Scenario-based controller – Displacements of the masses: $d_1$ (blue diamonds), $d_2$ (green circles), $d_3$ (red squares), $d_4$ (cyan triangles).

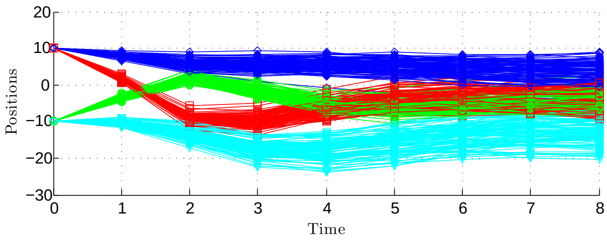


**Figure 8:** Displacements of the masses: $d_1$ (blue diamonds), $d_2$ (green circles), $d_3$ (red squares), $d_4$ (cyan triangles).

The different behaviors of the controllers can be also appreciated by analyzing the state trajectories corresponding to 100 disturbance realizations as depicted in Fig. 7 and 8 (displacements) and Fig. 9 and 10 (velocities). The scenario-based controller exploits the allowed speed to steer the masses close to their nominal position. On the contrary, the LQ control policy leads to the violation of the state constraint in the first time instant, while, in the other instants, the speed is kept conservatively small, and the masses are not steered toward the

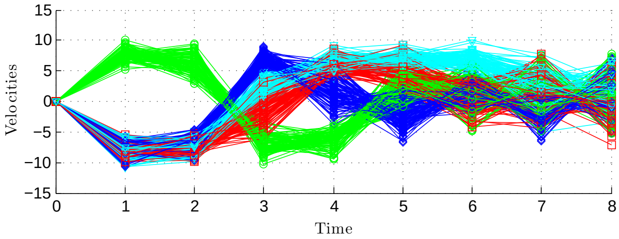


**Figure 9:** Scenario-based controller – Velocities of the masses: $\dot{d}_1$ (blue diamonds), $\dot{d}_2$ (green circles), $\dot{d}_3$ (red squares), $\dot{d}_4$ (cyan triangles).

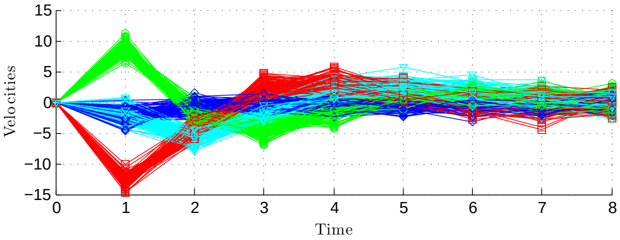


**Figure 10:** LQ controller $q_J = 0.2$, $q_L = 9$ – Velocities of the masses: $\dot{d}_1$ (blue diamonds), $\dot{d}_2$ (green circles), $\dot{d}_3$ (red squares), $\dot{d}_4$ (cyan triangles).

nominal position.

### 3.5 Reach Set Computations in Robust MPC

We consider the problem of setting up a robust MPC scheme for the uncertain discrete time linear system (21) where $w_t \in \mathbb{R}^n$ is an additive stochastic disturbance whose probability distribution has *bounded* support $W$ and $B_W = I$. To be more concrete, we shall consider the case that $W$ is a polytope. We assume that the state of the system is available.

Our aim is that of presenting a comparison between two types of computational approaches: those that provide a solution that is robustly guaranteed, and scenario-based method, which provides a solution with probabilistic guarantees only.

As for the finite horizon control problem to be solved at every time instant $t$, we suppose to minimize the average quadratic cost (24) with the constraint that the state and the input stay at every time instant in the prediction horizon $M$ in the polytopic sets $\mathcal{X}$ and $\mathcal{U}$:

$$x_{\tau+i} \in \mathcal{X} \ i = 1, \ldots, M \tag{45a}$$

$$u_{\tau+i} \in \mathcal{U} \ i = 0, \ldots, M-1. \tag{45b}$$

When the constraints (45) are enforced for every possible realization of the disturbance (robust MPC), the ensuing optimization problem turns out to be *semi-infinite*, i.e. a problem with a finite number of decision variables but an infinite amount of constraints. Semi-infinite problems require some care, since they may be very tough to solve, and in many cases they have been proven to be NP-hard.

A possible solution to this issue is that adopted in [52, 37, 62, 34], where the uncertain evolution of the system (21) is properly over-bounded so that an optimization problem that it is amenable of resolution at a relatively low computational burden can be formulated. In this case, the obtained solution is guaranteed to robustly satisfy the constraints. Yet, the introduced over-approximation may introduce some conservatism, so that feasibility is achieved only if the constraints to be satisfied are loose.

Alternatively, one can opt for an inner approximation of the uncertain evolution of the system through randomization, and adopt the scenario approach.

The advantage of a scenario-based solution is that it does not introduce any sort of conservatism in the constrained optimization problem resolution. It may hence lead to an effective solution even when constraints are tight and other approaches turn out to be unfeasible. This is shown next by performing a comparative analysis of the two types of approaches by means of prototype examples. The analysis highlights advantages and drawbacks of the various approaches, and all the results are substantiated by quantitative evaluations. As already anticipated, the performed analysis shows that scenario-based MPC may be a valid alternative to other approaches, enhancing its use to tackle robust MPC.

### 3.5.1 Approaches to Robust MPC

**Robust MPC based on Invariant Sets** This first approach was originally proposed in [52] and it relies on invariant sets to bound the uncertain dynamics of (21).

The control law is selected as

$$u_i = K(x_i - \bar{x}_i) + c_i, \tag{46}$$

where $K$ is an a-priori fixed gain (e.g. the optimal LQ gain of the infinite horizon problem without constraints), while the open loop term $c_i$ is the actual decision variable and $\bar{x}_i$ is the state variable of the following auxiliary nominal system

$$\bar{x}_{i+1} = A\bar{x}_i + Bc_i,$$

whose initialization $\bar{x}_0$ is a decision variable too. Given (46), the true system dynamics becomes

$$x_{i+1} = (A + BK)x_i - BK\bar{x}_i + Bc_i + w_i,$$

so that the difference between the actual and the nominal dynamics $\eta = x - \bar{x}$ satisfies the equation

$$\eta_{i+1} = (A + BK)\eta_i + w_i. \tag{47}$$

Now, let $Z$ be a disturbance invariant set for the system (47), that is, if $\eta_t \in Z$ then $\eta_{t+1}$ must belong to $Z$ too, for every realization of the disturbance $w_t \in W$. Based on the properties of invariant sets, it is trivial to verify that if $x_0 \in \bar{x}_0 + Z$ (note that $b + A = \{b + a,\ a \in A\}$), then $x_i \in \bar{x}_i + Z$, $i = 1, \ldots, M$, and $u_i \in c_i + KZ$ (note that $KA = \{Ka,\ a \in A\}$), $i = 0, \ldots, M-1$. In other words, $Z$ can be used to bound the uncertain dynamics of (21). The robust satisfaction of constraints (45) can be guaranteed by selecting $\bar{x}_0$ so that $x_\tau \in \bar{x}_0 + Z$, and $c_i$ so that $\bar{x}_i + Z \subseteq \mathcal{X}$ and $c_i + KZ \subseteq \mathcal{U}$. This leads to the following optimization problem:

$$\min_{c_i, \bar{x}_0} \mathbb{E}\left[\sum_{i=1}^{M} x_i^T Q x_i + \sum_{i=0}^{M-1} u_i^T R u_i\right] \text{subject to:} \tag{48}$$

$$\begin{cases} c_i \in \mathcal{U} \ominus KZ & i = 0 \ldots M-1 \\ \bar{x}_i \in \mathcal{X} \ominus Z & i = 1 \ldots M \\ x_0 - \bar{x}_0 \in Z & \end{cases},$$

where $\ominus$ denotes the Pontryagin difference ($A \ominus B = \{a : a + B \subseteq A\}$).

If the invariant set $Z$ is a polytope, then the sets $\mathcal{U} \ominus KZ$ and $\mathcal{X} \ominus Z$ are polytopes too and can be easily computed. In this case, problem (48) is thus convex and can be solved through standard optimization techniques like those used in CVX, [38], and YALMIP, [47].

The constraints in (48) can be seen as tightened versions of the original state and input constraints because of the difference with $Z$ and $KZ$. In order to achieved the widest feasibility for problem (48), the invariant set $Z$ should be as small as possible. The minimal invariant set, however, may not be a polytope, see [52], and moreover it is quite difficult to compute. Usually a polytopic outer approximation of the minimal invariant set is used, see e.g. [63].

**Tube-based Robust MPC** This second approach was developed in [34] for the case of multiplicative uncertainty. The case of additive uncertainty discussed here is obtained by means of straightforward modifications.

Let $z = \mathbb{E}[x]$ and $e = x - \mathbb{E}[x]$, and select the control law as

$$u_i = Kz_i + Le_i + c_i, \tag{49}$$

where $K$ and $L$ are a-priori fixed gains and the open loop term $c_i$ is the actual decision variable. The dynamics of $x = z + e$, then, is split into the dynamics of $z$ (nominal dynamics) and of $e$ (uncertainty dynamics):

$$z_{i+1} = (A + BK)z_i + Bc_i, \quad z_0 = x_0 \tag{50}$$

$$e_{i+1} = (A + BL)e_i + w_i, \quad e_0 = 0. \tag{51}$$

The idea is to find at each time step in the prediction horizon of length $M$ a polytope $P_i$ enclosing any possible evolution of $e$ at time $i$ as due to the disturbance $w$. Then, the evolution of $x$ is guaranteed to be contained in $z_i + P_i$, while $u_i$ is contained in $Kz_i + c_i + LP_i$ and the satisfaction of (45) is guaranteed by requiring that these polytopes are subsets of $\mathcal{X}$ and $\mathcal{U}$, respectively. The final optimization problem is

$$\min_{c_i} \mathbb{E}\left[\sum_{i=1}^{M} x_i^T Q x_i + \sum_{i=0}^{M-1} u_i^T R u_i\right] \text{subject to:} \tag{52}$$
$$\begin{cases} Kz_i + c_i \in \mathcal{U} \ominus LP_i & i = 0, \ldots, M-1 \\ z_i \in \mathcal{X} \ominus P_i & i = 1, \ldots, M \end{cases}.$$

As for the computation of the bounding polytopes, $P_i$ are selected in the form $\{e_i : Ve_i \leq \alpha_i\}$, where inequality is intended componentwise. The matrix $V$ has to be a-priori fixed so that the polytope facets have always the same orientation for every $t + i$. The bounding of the evolution of $e$ is achieved by suitably distancing the facets at each time instant as specified by vector $\alpha_i$. Technically speaking, the polytopes are recursively computed. Starting from $P_0 = \{0\}$, $P_{i+1}$ is obtained from $P_i$ by imposing that the set of points $e_i$ whose evolution $e_{i+1}$ according to (51) is contained in $P_{i+1}$ must include the points in $P_i$, that is

$$\begin{aligned} &\{e_i : Ve_i \leq \alpha_i\} \subseteq \\ &\{e_i : V(A + BL)e_i + Vw_i \leq \alpha_{i+1}, \forall w_i \in W\}. \end{aligned} \tag{53}$$

(53) can be enforced exploiting a corollary of Farkas's lemma, originally proven in [9], about the inclusion of polytopes: let $\mathcal{S}_1 = \{x : A_1 x \leq b_1\}$ and $\mathcal{S}_2 = \{x : A_2 x \leq b_2\}$; it holds that $\mathcal{S}_1 \subseteq \mathcal{S}_2$ if and only if there exists a matrix $H$ with non-negative entries such that $HA_1 = A_2$ and $Hb_1 \leq b_2$.

In view of this lemma, (53) is equivalent to

$$\begin{cases} HV = V(A + BL) \qquad h_{kl} \geq 0 \qquad \forall k, l & \text{(54a)} \\ H\alpha_i \leq \alpha_{i+1} - Vv \qquad \forall v \text{ vertex of } W & \text{(54b)} \end{cases}$$

from which the polytopes $P_i$ are eventually determined by first selecting the $H$ satisfying (54a) that gives the minimum value for the trace of $HH^T$, and then by selecting the $\alpha_i$ with minimal components that satisfies (54b). Note that the inequality in (54b) is posed for the vertices of $W$ only thanks to convexity.

### 3.5.2 Scenario-based MPC

In the considered scenario-based solution, the control law is expressed as

$$u_i = Kx_i + c_i, \tag{55}$$

where $K$ is an a-priori fixed gain and the open loop term $c_i$ is the actual decision variable.

In order to determine the open loop term $c_i$, a number $N$ of disturbance realizations of length $M$

$$w_0^{(k)}, w_1^{(k)}, \dots, w_{M-1}^{(k)}, \quad k = 1, 2, \dots, N,$$

are generated according to the underlying probability distribution of $w$. Then, a finite optimization problem, where the state and input constraints are posed in correspondence of the extracted realizations of the disturbance only, is considered.

To be precise, let $u_0^{(k)}, u_1^{(k)}, \dots, u_{M-1}^{(k)}$ be the control actions evaluated in correspondence of the $k$-th extracted realization of the disturbance, and $x_1^{(k)}, x_2^{(k)}, \dots, x_M^{(k)}$ the corresponding state trajectory (clearly, $u_i^{(k)}$ and $x_i^{(k)}$ still depend on the choice of $c_l$, $l = 1, \dots, i$). The scenario program to be solved at each step is

$$\min_{c_i} \mathbb{E} \left[ \sum_{i=1}^{M} x_i^T Q x_i + \sum_{i=0}^{M-1} u_i^T R u_i \right] \text{ subject to:} \tag{56}$$

$$\begin{cases} u_i^{(k)} \in \mathcal{U} & i = 0, \dots, M-1 \\ x_i^{(k)} \in \mathcal{X} & i = 1, \dots, M \end{cases}, \; k = 1, \dots, N.$$

Because of the finiteness of the considered realizations of the disturbance, problem (56) has a finite number of constraints only and is convex. Indeed, using the parametrization (55), both the input and the state depend linearly on the parameters $c_i$, and this entails that: *i)* the cost function is convex in the decision variables (actually, it is a quadratic function); and *ii)* for every fixed realization of the disturbance, the constraints (45) are convex as well (when $\mathcal{U}$ and $\mathcal{X}$ are polytopes, they are linear).

Despite the apparent naivety of the scenario approach, the obtained solution comes with some interesting guarantees about constraint feasibility, [14, 15, 20, 23, 1], which make it a sensible method to find an approximate solution to a robust problem, as discussed in the introduction. In the present convex set-up, the best available result is given by the following theorem.

**Theorem 3.** *Let $r$ be the total number of optimization variables in problem* (56)*. For any $\varepsilon \in (0,1)$ and $\beta \in (0,1)$, if*

$$N \geq \frac{r + 1 + \ln(1/\beta) + \sqrt{2(r+1)\ln(1/\beta)}}{\varepsilon},$$

*then, the probability that there exists a disturbance realization such that the solution to* (56) *does not satisfy the constraint*

$$\begin{cases} u_i \in \mathcal{U} & i = 0, \dots, M-1 \\ x_i \in \mathcal{X} & i = 1, \dots, M \end{cases}$$

*is no bigger than $\varepsilon$ with high confidence $1-\beta$.* □

The bound above is due to [2] and is an explicit expression of the implicit bound on $N$ given in [20].

In words, Theorem 3 says that the solution provided by the scenario approach is robust except for an $\varepsilon$ portion of the disturbance realizations ($\varepsilon$-robustness), as long as $N$ is suitably chosen. Note that $\varepsilon$-robustness is guaranteed with high confidence $1-\beta$ only. However, if one adopts small values of $\beta$ like $\beta = 10^{-6}$ or $\beta = 10^{-9}$, then, the $\varepsilon$-robustness is achieved beyond any reasonable doubt.

### 3.5.3 Comparative Analysis

The goal of this subsection is to make a comparative analysis of the approaches described in Sections 3.5.1 and 3.5.2, focusing in particular on their capability of providing an effective solution to the MPC problem as the state and input constraints become tighter and tighter. To this purpose, the approaches are applied in a receding horizon fashion to the following second order system:

$$x_{t+1} = \begin{bmatrix} 0.5 & -0.5 \\ 0.5 & 0.5 \end{bmatrix} x_t + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u_t + w_t, \tag{57}$$

where $w$ is a white noise uniformly distributed in $[-0.2\ 0.2]^2$. The reference finite horizon problem takes the following form:

$$\min \mathbb{E}\left[\sum_{i=1}^{M} x_i^T Q x_i + \sum_{i=0}^{M-1} u_i^T R u_i\right] \text{ subject to:} \tag{58}$$

$$\begin{cases} \|u_i\|_\infty \leq \bar{u} & i = 0,\dots,M-1 \\ \|Cx_i\|_\infty \leq \bar{y} & i = 1,\dots,M \end{cases},$$

where we set $M = 10$, $Q = I_2$, $R = 0.1$, and $C = I_2$.

As for the design parameters entering the solution of (58) according to the approaches of Sections 3.5.1 and 3.5.2, we have:

- the feedback gains $K$ in (46), $K$, $L$ in (49), and $K$ in (55) are set equal to the optimal LQ gain $K_{LQ}$;
- the shaping matrix $V$ of the tube-based approach is

$$V = \begin{bmatrix} 1 & 0 & -1 & 0 & 1 & 1 & -1 & -1 \\ 0 & 1 & 0 & -1 & 1 & -1 & 1 & -1 \end{bmatrix}^T$$

- the violation and confidence parameters of the scenario-based approach are $\varepsilon = 0.05$ and $\beta = 10^{-6}$;

We set the initial state equal to 0 and evaluate the threshold values $\bar{y}_T$ and $\bar{u}_T$ for $\bar{y}$ and $\bar{u}$ in (58) leading to unfeasibility of the optimization problems (48), (52), and (56). Our aim is to find how much the constraints can be tightened before incurring in unfeasibility so as to assess the possible conservativeness of the three approaches. Results are shown in Table 2.

**Table 2:** Estimate of the threshold values $\bar{y}_T$ and $\bar{u}_T$ for $\bar{y}$ and $\bar{u}$ before incurring in unfeasibililty.

| | $\bar{y}_T$ | $\bar{u}_T$ |
|---|---|---|
| Tube-based approach | 0.74 | 0.46 |
| Approach based on invariant sets | 0.53 | 0.35 |
| Scenario-based approach | 0.44 | 0.30 |

Note that, when the initial state is zero, the optimization problem (48) in the approach of Section 3.5.1 is feasible if and only if the invariant set $Z$ and its projection $KZ$ on the input space through the feedback gain $K$ are respectively contained in the constraint set $\mathcal{X}$ and $\mathcal{U}$. Hence, the thresholds $\bar{y}_T$ and $\bar{u}_T$ can be obtained based on $Z$ and $KZ$. Likewise, when the initial state is zero, problem (52) in the approach of Section 3.5.1 is feasible only if the tube sections $P_i$ and $LP_i$ are contained in the input and state constraint sets for every $i$. Since $P_i$ and $LP_i$ are increasing with $i$, $\bar{y}_T$ and $\bar{u}_T$ are obtained based on the tube sections $P_{t+M}$ and $LP_{t+M-1}$.

Computing $\bar{y}_T$ and $\bar{u}_T$ for the scenario-based approach is instead more tricky, since it is a randomized method. The values reported in Table 2 are heuristically determined by progressively reducing $\bar{y}$ and $\bar{u}$ and checking for each pair $(\bar{y}, \bar{u})$ whether problem (56) is feasible in 100 trials.

As it appears, the scenario-based approach outperforms the other approaches in terms of tightness allowed for the constraints. This can be justified by comparing the different approximations of the uncertain evolution of the state used by the three approaches. To this purpose, we take a bunch of 10000 disturbance realizations of length $M$ and simulate the state evolution from the initial condition $x_0 = 0$ when the control input is given by $u_i = K_{LQ}x_i$. Indeed, this is the solution to problems (48), (52), and (56) when constraints are feasible and $x_0 = 0$. We obtain for every time instant along the prediction horizon $[0, M]$ a cloud of possible states reached by the system and we superimpose them to the invariant set $Z$ and to the polytopes $P_i$ defining the tube section at that time instant.

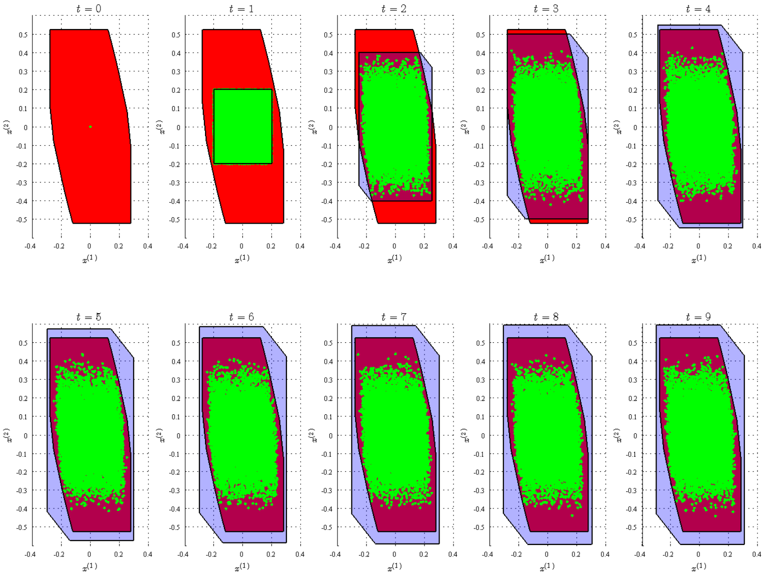


**Figure 11:** Realizations of the state of system (57) when the LQ control law is applied (green cloud), invariant set $Z$ (red) and sections of the tube (light blue). The green cloud is an inner approximation of the reach set.

As one can see in Fig. 11 the clouds of reachable states are smaller than their approximations as given by $Z$ and $P_{t+i}$. In particular note that the invariant set $Z$ has to contain the cloud at every time instant and it cannot adapt to the shape of the cloud. Furthermore, even if we superimpose all the clouds, some regions of the invariant set appear empty and hence are quite unlikely to be reached. Using tubes gives the possibility to shape the reachable set approximation so as to best fit the cloud, but it turns out that the tube approximation is well adapted to the cloud just for the first 4 time steps and then becomes even larger than the invariant set. The scenario-based approach, instead, is not affected by this over approximation effect. As a matter of fact, according to problem (56) the optimal control law is determined by considering $N$ disturbance realizations and the corresponding state and input values for defining the constraints. The resulting approximation of the uncertain state evolution is then tighter then that used by the other two approaches. Indeed, since we adopted the parametrization $u_{t+i} = K_{LQ}x_{t+i} + c_i$ for the scenario-based approach, the clouds plotted in Fig. 11 represent the reachable states for the scenario solution.

Note that the feasibility of problems (48), (52) and (56) depends on the choice of the feedback gains $K$ in (46), $K$, $L$ in (49), and $K$ in (55). It may be that choices other than $K_{LQ}$ lead to smaller thresholds than those reported in Table 2. However, in (48) and (52) $K$ and $(K, L)$ have to be fixed in advance, otherwise computational difficulties arise, and it is usually difficult to guess what is the right choice. In the scenario method, instead, the feedback gain $K$ can be easily optimized along with the open loop term $c$, so that $K$ can be automatically tuned towards the optimum. To be precise, to preserve convexity, one has to adopt the parametrization in (22)

$$u_{t+i} = \gamma_i + \sum_{j=0}^{i-1} \theta_{i,j} w_{t+j}, \tag{59}$$

where $w_{t+j}$ can be reconstructed from the state equation as $x_{t+j+1} - Ax_{t+j} - Bu_{t+j}$ and $\theta_{i,j} \in \mathbb{R}^{m \times n}$ and $\gamma_i \in \mathbb{R}^m$ are the design parameters. As previously discussed, (59) is equivalent to optimizing both the open loop term and the feedback gain, and by using it a further improvement in terms of tightness of the bounds can be achieved. For instance, when a reduced parametrization is adopted where the control input depends only on the previous 3 values of the disturbance, then, the threshold values $\bar{y}_T = 0.39$ and $\bar{u}_T = 0.28$ are obtained.

As for the receding horizon implementation of the three approaches, when constraints are loose and they are all feasible, the obtained performance are quite similar. More specifically, after some transient, they all converge to the LQ solution, also when using the control law parametrization (59) for the scenario-based approach. When constraints are tight and only the scenario-based approach with parametrization (59) provides a feasible solution, then such a solution does not necessarily converge to the LQ one. This appears to be the case if we set $\bar{u} = 0.28$ and $\bar{y} = 0.39$, as shown in Fig. 12, where a realization of the control input and the corresponding values as given by the LQ control law are plotted.

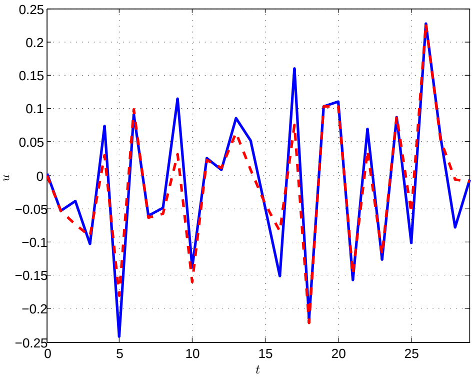


**Figure 12:** Input for system (57) controlled with the scenario-based approach applied in a receding horizon fashion (solid blue) and input given by the LQ control law (dashed red).

The results of this comparative analysis highlight some key features of the approaches in Sections 3.5.1 and 3.5.2. Specifically, the approaches in Section 3.5.1 may not be applicable to problems with tight constraints. Unfeasibility may in fact arise due to their use of *i)* an outer approximation of the uncertain system dynamics, and *ii)* a-priori fixed feedback gains in the control law parametrization. On the contrary, the scenario-based approach is able to tackle problems with tighter constraints because of the adopted inner approximation and, possibly, of the tuning of the state feedback gain. Both numerical examples reveal in fact that, though setting the feedback gain to the LQ gain is an optimal choice for what concerns the cost minimization, it may be not ‘optimal’ for the constraints satisfaction.

In turn, however, while the approaches in Section 3.5.1 are robust in that constraints are guaranteed to be satisfied for every and each disturbance realization, for the scenario-based approach only probabilistic guarantees are given. This drawback of the scenario approach can be only partly alleviated by setting the size $\varepsilon$ of the set of disturbance realizations that violate the constraints as small as possible, compatibly with the available computational resources.

# 4 Model Predictive Control for Interacting CPS with Hybrid Dynamics

In contrast to the sections before, this part considers a subsystem (as part of a larger CPS) which is embedded into an environment that imposes time-varying restrictions on the behavior of the subsystem. For its dynamics, we consider hybrid dynamics following the typical notion of hybrid automata [40, 49], in which guards set enable discrete transitions, and an invariant set have to satisfied while being in a discrete state. If interactions of several subsystems is considered, one possible option is to use the invariant sets of a hybrid automaton to model that the subsystem state must not enter a region currently occupied by a different subsystem. This situation is, e.g., relevant if an autonomous vehicle moves in the same space as other vehicles and collision-avoidance has to be ensured, as for the use cases of autonomous driving and human-robot interaction (investigated in WP5) . The perspective taken in this section is that the other subsystems have computed their planned trajectory over a prediction horizon, and have it communicated to the subsystem to be controlled. The latter can determine the occupied space from the planned trajectory by reachability analysis, i.e., by the using the methods developed in WP3 of UnCoVerCPS. From the reachable sets, it can construct a sequence of (typically polytopic) state sets, the invariants, which are free for its own control (see deliverable D2.1 for details). Computing the control strategy is, also here, accomplished by MPC, i.e., an online optimization problem is solved which is subject to the hybrid dynamics and the invariant sets. If a feasible strategy can be determined, it is executed, otherwise an emergency procedure is initiated. See Fig. 13 for an overview of the method.

For the hybrid automaton constructed online, the optimzation problem determines a control strategy as a sequence of mixed inputs, i.e., of continuous inputs together with discrete inputs for triggering enabled transitions. The transition dynamics introduces conditional constraints on the continuous states, leading to integer variables in the problem formulation. The sequence of transitions as well as the discrete control inputs represent two sources of combinatorial complexity for the optimization, that typically lead to large numbers of value combinations for the integer variables, and thus large computation times [74].

Schemes to transform the hybrid dynamics into linear inequalities for integer and continuous variables within the context of optimal and predictive control have been proposed in the past, e.g. with respect to mixed logical dynamic systems [7], or for hybrid automata with linear continuous dynamics [72]. The obtained reformulated problems can then be solved by tools for mixed-integer programming. A common objective for such reformulations is certainly to

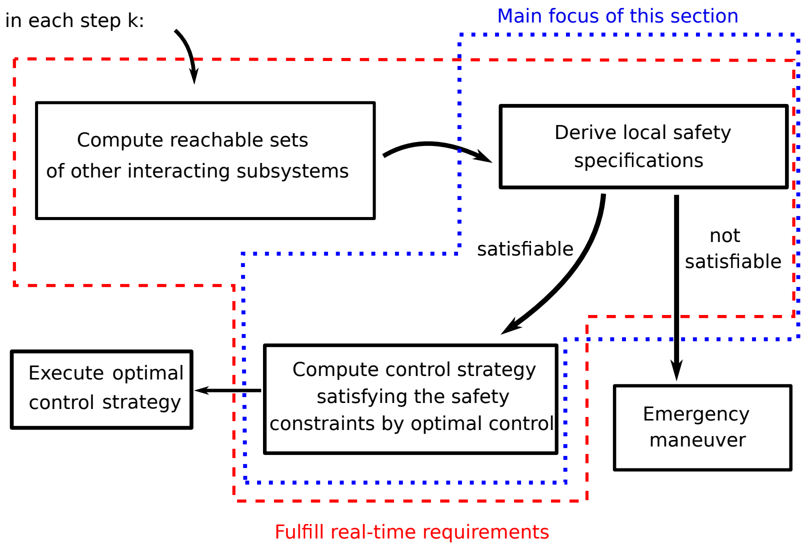

in each step k:
Main focus of this section
Compute reachable sets of other interacting subsystems
Derive local safety specifications
satisfiable
not satisfiable
Execute optimal control strategy
Compute control strategy satisfying the safety constraints by optimal control
Emergency maneuver
Fulfill real-time requirements

**Figure 13:** Steps of computation to be carried out in any time step $k$. The determination of reachable sets and the conversion in safety constraints uses the techniques reported in D2.1. The computation of control strategies as described in the following relies on the model format for hybrid systems as introduced in D1.2.

keep the number of binary variables small. An issue which has not been addressed and solved satisfactory up to-date is how the number of value combinations of the necessary integer variables can be limited to the extent which refers to the set of admissible executions of the hybrid systems – this is the objective of the present section.

It should be mentioned for completeness that a larger variety of direct and indirect methods exist to solve hybrid optimal control problems without the use of mixed-integer programming, e.g. [11, 56, 65, 77, 39, 71, 32]. More important for this investigation are those approaches, however, that aim at finding sequences of discrete states (and binary variables) which encode a particular goal-attaining temporal execution of the hybrid system. The work in [76, 44, 54] use, e.g. linear temporal logic [58] as a task-specification tool to force the obtained state trajectory satisfying the desired property of the task. However, the encoding of the LTL formula is often elaborate. As indicated in [76], the number of the binary variables used for encoding a single *Until* operator is quadratic in the time horizon. This work also aims at determining task specifications by encoding each attained discrete state as well as the guard set with binary variables. But instead of directly encoding the LTL formula as mixed-integer linear constraints, a matrix of binary variables is determined to formulate the trajectories leading from the initial to the goal state. Constraints are formulated for this matrix to impose a particular structure, leaving only value combinations of the binary variables that correspond to admissible executions.

## 4.1 Problem Formulation

The control approach proposed in this section targets the discrete model definition for networked CPS with hybrid dynamics, as introduced in D1.2. To ease notation, we here moderately simplify the model definition from D1.2: in particular, we abstain from using subsystem indices, which is justified, since after converting the subsystem interaction into constraints, a local constrained control problem results, which does not require explicit reference to the interacting subsystems. Thus, let the subsystem of the CPS under consideration be modeled by a hybrid system with mixed inputs acording to $HA = (T, U, X, Z, I, \mathcal{T}, G, V, r, f)$, consisting of:

- the discrete time-domain $T = \{t_k \mid k \in \mathbb{N} \cup \{0\}, \Delta \in \mathbb{R}^{>0} : t_k := k \cdot \Delta\}$, where $k$ is used in the following to refer to $t_k$;
- the continuous input space $U \subseteq \mathbb{R}^{n_u}$ with the continuous input $u \in U$;
- the continuous state space $X \subseteq \mathbb{R}^{n_x}$ on which the state vector $x$ is defined;
- the finite set of discrete states $Z = \{1, \cdots, n_z\}$;
- a set $I = \{I_1, \ldots, I_{n_z}\}$ of invariants where the invariant of any discrete state $i$ is a polytope $I_i = \{x \mid n_{p_i} \in \mathbb{N}, C_i \in \mathbb{R}^{n_{p_i} \times n_x}, d_i \in \mathbb{R}^{n_{p_i}}, x \in X : C_i \cdot x \leq d_i\}$;
- the finite set of transitions $\mathcal{T} \subseteq Z \times Z$, in which a transition from $i \in Z$ to $j \in Z$ is denoted by the ordered pair $(i, j) \in \mathcal{T}$;
- the set $G$ of guard sets contains one polytopic set $G_{(i,j)} = \{x \mid C_{(i,j)} \in \mathbb{R}^{n_{G(i,j)} \times n_x}, d_{(i,j)} \in \mathbb{R}^{n_{G(i,j)}}, x \in I_i : \ C_{(i,j)} \cdot x \leq d_{(i,j)}\}$ for any transition $(i, j) \in \mathcal{T}$; let for any pair of the outgoing transitions from the discrete state $i$ the corresponding guard sets be disjoint, i.e. $G_{(i,j)} \cap G_{(i,l)} = \emptyset$, $\forall j \neq l$;
- the finite set $V$ of discrete input variables, where any element $v_{(i,j)}$ in $V$ refers to one transition $(i, j) \in \mathcal{T}$; the variable $v_{(i,j)}$ is a binary one, and for $v_{(i,j)} = 1$ the transition $(i, j)$ is triggered if $x \in G_{(i,j)}$ applies; if $v_{(i,j)} = 0$ or $x \notin G_{(i,j)}$, the transition cannot occur;
- a reset function $r$: $\mathcal{T} \times X \to \mathbb{X}$ which updates the state vector $x$ upon a transition $(i, j) \in \mathcal{T}$ according to $x' = E_{(i,j)} \cdot x + e_{(i,j)}$;
- and the function $f : X \times U \times Z \to X$ defining the discrete-time continuous dynamics according to $x_{k+1} = A_i \cdot x_k + B_i \cdot u_k$ with $x_{k+1} := x(t_{k+1})$, $i \in Z$, $x_k \in I_i$.

The execution of $HA$ is defined as follows: assume a finite time set $T_N = \{0, 1, \ldots, N\} \subset T$, and let the initial states $(x_0, z_0)$ satisfy $x_0 \in I_{z_0}$ and $x_0 \notin G_{(z_0,j)}$ for each $(z_0, j) \in \mathcal{T}$ for $j \in Z$. For given input sequences $\phi_u = \{u_0, u_1, \ldots, u_{N-1}\}$ and $\phi_v = \{v_0, v_1, \ldots, v_{N-1}\}$, the pair of state sequences $\phi_x = \{x_0, x_1, \cdots, x_N\}$ and $\phi_z = \{z_0, z_1, \cdots, z_N\}$ is *admissible*, if and only if for any $k \in \{0, \ldots, N\}$ the pair $(x_{k+1}, z_{k+1})$ follows from $(x_k, z_k)$ according to the following steps:

1.) $x' := A_i \cdot x_k + B_i \cdot u_k \in I_i$,

2.) if $x' \in G_{(i,j)}$ and $v_k = 1$, then $x_{k+1} := r((i, j), x') \in I_j$ and $z_{k+1} := j$, else $x_{k+1} := x'$, $z_{k+1} := i$.

The second step makes obvious that a transition is bound to the condition that a discrete control decision is imposed in addition to the fact that the intermediate state $x'$ is contained in a guard set.

In order to introduce the control task, assume now that a set of hybrid goal states $(X_g, z_g)$ is defined by one $z_g \in Z$ and $X_g = \{x \mid n_{p_g} \in \mathbb{N}, C_g \in \mathbb{R}^{n_{pg} \times n_x}, d_g \in \mathbb{R}^{n_{pg}}, x \in I_g : C_{X_g} \cdot x \leq d_{X_g}\}$. Furthermore, let a state $x_c \in X_g$ be specified (e.g. the volumetric center of $X_g$ ) to later define a distance to the goal region in a computationally easy way.

If $(x_0, z_0)$, $(X_g, z_g)$, and $T_N$ are specified, the control objective is to find admissible state sequences $\phi_x$ and $\phi_z$, or corresponding input sequences $\phi_u$ and $\phi_v$ respectively, which minimize an appropriate cost functional. Hereto, we define:

$$\begin{aligned}\mathcal{J}(x_0, x_f, N) = &\sum_{k=1}^{N} \{(x_k - x_{c,k}^{i,j})^{\mathrm{T}} Q (x_k - x_{c,k}^{i,j}) \\ &+ (u_{k-1} - u_g)^{\mathrm{T}} R (u_{k-1} - u_g)\} + q_g \cdot N_g\end{aligned} \tag{60}$$

where $Q$ and $R$ are positive-definite weighting matrices, and $q_g \in \mathbb{R}^{\geq 0}$. The variable $N_g := \min\{k \in \{1, \ldots, N\} \mid x_k \in X_g, z_k = z_g\}$ encodes the first point of time at which the continuous state has reached the goal set. We assume that $(u_k, v_k)$ exists for $k \in \{N_g, \ldots, N\}$ to hold the system in the goal set. For any $k \in \{1, \ldots, N\}$ with $z_k \neq z_g$, the state $x_c^{i,j}$ encodes the center of the guard set $G_{(i,j)}$, if $x_k \in I_i$ and if $z_k$ is left through the transition $(i, j) \in \mathcal{T}$. Thus, any term of the sum in (60) encodes the weighted distance to the guard set, which can be seen as a *temporary goal set* while $HA$ is in the discrete state $z_k$. For $z_k = z_g$, we require $x_c^{i,j} = x_c$.

The overall control problem can then be defined as:

**Problem 1.** *For $HA$ initialized to $(x_0, z_0)$, let a time set $T_N$ and a goal $(X_g, z_g)$ be given.*

*Then, determine input sequences $\phi_u^*$ and $\phi_v^*$ as the solution of:*

$$
\begin{aligned}
&\min_{\phi_u,\phi_v} \mathcal{J}(x_0, x_f, N) \qquad (61)\\
s.t.:\ &\phi_u \textit{ with } u_k \in U, k \in \{0,\dots,N-1\}\\
&\phi_v \textit{ with } v_k \in \{0,1\}, k \in \{0,\dots,N-1\}\\
&\phi_x, \phi_z \textit{ admissible for } HA, x_N \in X_g, z_N = z_g.
\end{aligned}
$$

The solution of this problem is difficult for large values of $N$ (a parameter for which a sufficiently high value to reach $(X_g, z_g)$ is not known a-priori), due to the combinatorics in $\phi_z$ and $\phi_v$. Note that expressing the conditions for $x_k$ being contained in invariants and guard sets for certain discrete states or transitions implies to use binary variables, when converting Problem 1 into a form that can be processed by available solvers. In addition, solving the problem also includes to decide (by the discrete inputs $v_k$) whether taking a transition upon reaching a guard set is better in terms of feasibility and costs than continuing to stay in the current discrete state. This is different from most other settings in existing literature on optimal control of hybrid systems. The following sections propose a new approach to approximate the optimal solution to the MIQP problem formulated by Prob. 1 efficiently in many cases.

## 4.2 Representation of Admissible Trajectories by Algebraic Programs

This section introduces a particular format to encode Prob. 1 as algebraic program with a number of binary variables that is relatively small compared to other formulations. It is well-known that implications like $(x_k \in I_i) \Leftrightarrow (b = 1)$ for mapping invariant set containment of $x_k$ into a binary variable $b$ can be accomplished by rules as those explained in [75] (often referred to the *Big-M-approach*). Such mechanisms have been re-used in different work on hybrid system optimization, e.g. [7] and [72], but the particular challenge is to use an as small as possible number of binary variables and constraints on these variables for low computational times. This issue is addressed in the following for Prob. 1. To facilitate the description and understanding of the procedure, we first refer to the simplified case that a *phase sequence* is known: let the order of the discrete states $Z$ by which $HA$ passes through be known, but the times in $T_N$ at which the discrete states are left or are reached still have to be determined. Hence, the remaining task is to determine the transition times as well as $\phi_u$ and $\phi_v$ such that $\phi_x$ is led (if possible) through the appropriate series of invariants and guards. Formally, a phase sequence is denoted by $\phi_p = \{p_0, \dots, p_L\}$, where $p_l$ with $l \in \{0, \dots, L\}$ is set to the index of the discrete state which is invariant in the $l-th$ phase (i.e. $\phi_p \subset \phi_z$ is obtained from eliminating consecutive equal elements in $\phi_z$).

The phases are now important to identify the number of binary variables required to encode the execution of $HA$ within the optimization problem: consider a phase $p_i$, as shown in Fig. 14, from a hybrid state $(x_k, z_k)$ with $z_k = i$ (reached by a preceding transition) up to the state $(x_{k+5}, z_{k+5})$ with $z_{k+5} = j$, reached through the transition $(i, j)$. Note that $x' \in G_{(i,j)}$ is an intermediate state, which is immediately transferred into $x_{k+5} := r((i, j), x') \in I_j$ by the transition with reset upon $v_{k+4} = 1$, according to the definition of an admissible run above. Two points are obvious from this figure: (1.) for any of the states $\{x_k, \dots, x_{k+4}, x'\}$ the same invariant constraint (element of $I_i$) applies, i.e. one binary variable per phase is sufficient to express this fact; (2.) the state $x'$ must be associated with an additional binary variable to encode $x' \in G_{(i,j)}$ for $p_i$. Since $x'$ must be treated separately, we use an extended index set for the states to be considered: $\tilde{k} \in \{0, N + L\}$. Within this set, the following assignments correspond to an admissible run of $HA$:

- $\tilde{k} = 0$ refers to $x_0$;
- $L$ values indicate intermediate states $x'$, and thus an exit from a discrete state;
- $L$ values belong to the entry into a newly reached discrete state;
- and one value encodes the entry into $X_g$.

Next, the constraints on the continuous states $x_{\tilde{k}}$ have to be formulated suitably. Recall that all invariants, guard sets, and $X_g$ are given as polytopic sets. Exemplarily for an invariant set $I_i$, the efficient algebraic encoding is explained: using the principles proposed in [75], the constraint $C_i \cdot x_{\tilde{k}} \leq d_i$ can be modeled equivalently by:

$$C_i \cdot x_{\tilde{k}} \leq d_i + b_{i,\tilde{k}} \cdot M_i \tag{62}$$

if $M_i \in \mathbb{R}^{n_{p_i} \times 1}$ is a vector of large constants, and $b_{i,\tilde{k}} \in \{0, 1\}$ one binary variable. If $b_{i,\tilde{k}} = 0$, the invariant constraint is enforced, while $b_{i,\tilde{k}} = 1$ relaxes the constraint. Likewise, a guard

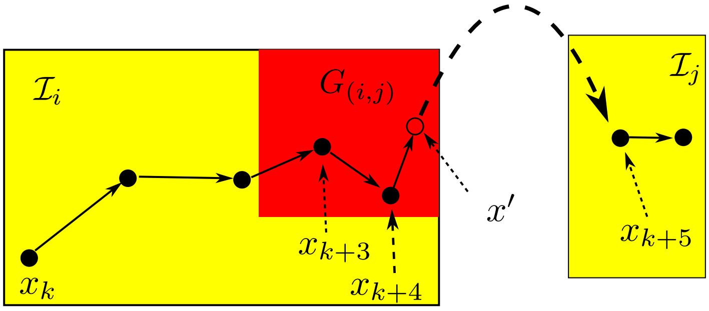


**Figure 14:** Execution of $HA$ within one phase.

constraint $x_{\tilde{k}} \in G_{(i,j)}$ results in:

$$C_{(i,j)} \cdot x_{\tilde{k}} \leq d_{(i,j)} + b_{(i,j),\tilde{k}} \cdot M_{(i,j)}. \tag{63}$$

Consider that two binary variables are required per phase (one for the invariant conditions, and one for the guard condition (or the terminal set, respectively)), we introduce a vector of $2 \cdot (L+1)$ binary variables:

$$\mathbf{b}_{\tilde{k}} = [b_{0,\tilde{k}},\, b_{(0,1),\tilde{k}},\, b_{1,\tilde{k}},\, \ldots,\, b_{L,\tilde{k}},\, b_{(L,X_g),\tilde{k}}]^{\mathrm{T}} \tag{64}$$

for each $\tilde{k} \in \{0, N+L\}$. The last entry represents containment in the goal set $X_g$. For $\tilde{k} = 0$, the numeric values of this vector are, $\mathbf{b}_0 = [0, 1, \ldots, 1]^{\mathrm{T}}$, and for the transition from phase $i$ to $i+1$ we have: (a) $\mathbf{b}_{\tilde{k}} = [1, \ldots, 1, \underbrace{0}_{2i+1}, \underbrace{0}_{2i+2}, 1, \ldots, 1]^{\mathrm{T}}$ corresponding to the intermediate state $x'$, and (b) $\mathbf{b}_{\tilde{k}} = [1, \ldots, 1, \underbrace{0}_{2i+3}, 1, \ldots, 1]^{\mathrm{T}}$ for the entry in the next invariant. For $\tilde{k} = N+L$, the vector is: $\mathbf{b}_{N+L} = [1, \ldots, 1, 0, 0]^{\mathrm{T}}$, and all of these vectors are collected in a matrix:

$$\mathcal{B}_m = [\mathbf{b}_0, \mathbf{b}_1, \ldots \mathbf{b}_{N+L}] = \tag{65}$$

$$\begin{bmatrix} \begin{bmatrix}0\\1\\1\\\vdots\\1\\1\\1\end{bmatrix} \cdots \underbrace{\begin{bmatrix}0\\1\\1\\\vdots\\1\\1\\1\end{bmatrix}}_{\tilde{k}_0^{out}-1} \underbrace{\begin{bmatrix}0\\0\\1\\\vdots\\1\\1\\1\end{bmatrix}}_{\tilde{k}_0^{out}} \underbrace{\begin{bmatrix}1\\1\\0\\1\\\vdots\\1\\1\end{bmatrix}}_{\tilde{k}_1^{in}} \cdots \underbrace{\begin{bmatrix}1\\\vdots\\1\\0\\0\\1\\1\end{bmatrix}}_{\tilde{k}_{L-1}^{out}} \underbrace{\begin{bmatrix}1\\\vdots\\1\\1\\1\\0\\1\end{bmatrix}}_{\tilde{k}_L^{in}} \cdots \begin{bmatrix}1\\\vdots\\1\\1\\1\\0\\0\end{bmatrix} \begin{bmatrix}1\\1\\1\\\vdots\\1\\0\\0\end{bmatrix} ] \end{bmatrix}$$

The last line refers to the time indexing, where $\tilde{k} = \tilde{k}_0^{out}$ refers to the instance in which the first invariant $I_0$ is left, and $\tilde{k} = \tilde{k}_1^{in}$ to the instance in which the second invariant of $\phi_z$ is reached. The following holds by construction:

**Lemma 1.** *If $\phi_x$ and $\phi_z$ determine an admissible run of $HA$ with $z_N = z_g$ and $x_N \in X_g$, then a matrix $\mathcal{B}_m \in \{0,1\}^{(2L+2)\times(N+L+1)}$ exists according to the rules* (62) *to* (65)*, and each column in $\mathcal{B}_m$ uniquely determines which constraints apply to $x_{\tilde{k}}$ for $\tilde{k} \in \{0, N+L\}$.* □

Let all constraints of the form (62) and (63) be collected in the order of the indexing of $x_{\tilde{k}}$ in:

$$\mathcal{C} \cdot x_{\tilde{k}} \leq \mathcal{D} + diag(\mathcal{B}_m(:, \tilde{k}+1)) \cdot \mathcal{M}. \tag{66}$$

The search for an admissible run $\phi_x$ and $\phi_z$ thus means to satisfy (66) for all $\tilde{k} \in \{0, \ldots, N+L\}$. While $(2L+2) \times (N+L)$ binary variables ($b_{0,\tilde{k}}$ is known) encode in principle $2^{(2L+2)\times(N+L)}$ combinations (prohibitively many for larger $N$ and $L$), the particular structure of $\mathcal{B}_m$ reduces the number of possible combinations (and thus of $\phi_z$) considerably. The following section proposes a scheme to efficiently exploit this structure in searching for an optimal $\phi_x$ and $\phi_z$.

### 4.3 Formulation of the Optimization Problem

In order to explain how $\mathcal{B}_m$ enables to search only over those value combinations of binary variables that represent admissible runs of $HA$, we first focus on the first two rows of the matrix. They represent the values of the binary variables $b_{0,\tilde{k}}$, $b_{(0,1),\tilde{k}}$ over $\tilde{k} \in \{0, N+L\}$, and these variables model that $x_{\tilde{k}}$ is contained in the invariant of the first discrete state (value 0), and respectively, that the first transition is triggered (again value 0):

$$\begin{bmatrix} \mathcal{B}_m(1,:) \\ \mathcal{B}_m(2,:) \end{bmatrix} = \begin{bmatrix} 0 & 0 & \cdots & 0 & 0 & 1 & 1 & \cdots & 1 \\ 1 & 1 & \cdots & 1 & 0 & 1 & 1 & \cdots & 1 \end{bmatrix}. \tag{67}$$

Note that the column in which $\mathcal{B}_m(1,:)$ changes from 0 to 1 is not yet determined. Let the value of $\mathcal{B}_m(1,\tilde{k}+1)$ depend on an auxiliary vector $\mathbf{d}^{\mathrm{T}}_{1,\tilde{k}+1} = [\mathcal{B}_m(1,\tilde{k}), \mathcal{B}_m(2,\tilde{k})]$ according to:

$$\mathcal{B}_m(1,\tilde{k}+1) = \begin{Bmatrix} 0 \\ 1 \end{Bmatrix} \text{ if } \mathbf{d}^{\mathrm{T}}_{1,\tilde{k}+1} = \begin{Bmatrix} [0,1] \\ [0,0] \text{ or } [1,1] \end{Bmatrix}. \tag{68}$$

Now, define two parameter/vectors $\alpha_1 \in \mathbb{R}^{3\times1}$ and $\beta_1 \in \mathbb{R}^{3\times1}$ satisfying the following conditions:

$$\begin{aligned} \begin{bmatrix} -\infty \\ 0 \\ 0 \end{bmatrix} &< \begin{bmatrix} 0 & 1 \\ 0 & 0 \\ 1 & 1 \end{bmatrix} \cdot \alpha_1(1:2) + \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \cdot \alpha_1(3) < \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}, \\ \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} &< \begin{bmatrix} 0 & 1 \\ 0 & 0 \\ 1 & 1 \end{bmatrix} \cdot \beta_1(1:2) + \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \cdot \beta_1(3) < \begin{bmatrix} 1 \\ \infty \\ \infty \end{bmatrix}, \end{aligned} \tag{69}$$

where the matrices in front of the vectors $\alpha_1(1:2)$ and $\beta_1(1:2)$ encode the possible values of $\mathbf{d}^{\mathrm{T}}_{1,\tilde{k}+1}$ in (68). Then the relation (68) can be algebraically and equivalently formulated as:

$$\begin{aligned} \mathcal{B}_m(1,\tilde{k}+1) &\geq \alpha_1^{\mathrm{T}}(1:2) \cdot \mathbf{d}_{1,\tilde{k}+1} + \alpha_1(3), \\ \mathcal{B}_m(1,\tilde{k}+1) &\leq \beta_1^{\mathrm{T}}(1:2) \cdot \mathbf{d}_{1,\tilde{k}+1} + \beta_1(3). \end{aligned} \tag{70}$$

While this encoding relates to the first phase, the principle can be transferred to the subsequent phases. For a phase with index $l \in \{1, \cdots, L-1\}$, the $(2l+1)$st row of $\mathcal{B}_m$ is relevant. It refers to the binary variable $b_{l,\tilde{k}}$, and the value of $\mathcal{B}_m(2l+1,\tilde{k}+1)$ is written depending on an auxiliary vector $\mathbf{d}^{\mathrm{T}}_{2l+1,\tilde{k}+1} = [\mathcal{B}_m(2l,\tilde{k}), \mathcal{B}_m(2l+1,\tilde{k}), \mathcal{B}_m(2l+2,\tilde{k})]$:

$$\begin{aligned} \mathcal{B}_m(2l+1,\tilde{k}+1) = &\begin{Bmatrix} 0 \\ 1 \end{Bmatrix} \\ &\text{if } \mathbf{d}^{\mathrm{T}}_{2l+1,\tilde{k}+1} = \begin{Bmatrix} [0,1,1] \text{ or } [1,0,1] \\ [1,1,1] \text{ or } [1,0,0] \end{Bmatrix}. \end{aligned} \tag{71}$$

If parameter vectors $\alpha_l \in \mathbb{R}^{4\times 1}$ and $\beta_l \in \mathbb{R}^{4\times 1}$ are defined similarly to (69), the assignment (71) can be equivalently formulated as:

$$\begin{aligned} \mathcal{B}_m(2l+1,\tilde{k}+1) &\geq \alpha_l^{\mathrm{T}}(1:3)\cdot \mathbf{d}_{2l+1,\tilde{k}+1} + \alpha_l(4), \\ \mathcal{B}_m(2l+1,\tilde{k}+1) &\leq \beta_l^{\mathrm{T}}(1:3)\cdot \mathbf{d}_{2l+1,\tilde{k}+1} + \beta_l(4). \end{aligned} \tag{72}$$

With respect to the penultimate row of $\mathcal{B}_m$, which refers to $b_{L,\tilde{k}}$, the value of $\mathcal{B}_m(2L+1,\tilde{k}+1)$ depends likewise on an auxiliary vector $\mathbf{d}^{\mathrm{T}}_{2L+1,\tilde{k}+1} = [\mathcal{B}_m(2L,\tilde{k}), \mathcal{B}_m(2L+1,\tilde{k})]$ with:

$$\mathcal{B}_m(2L+1,\tilde{k}+1) = \begin{Bmatrix} 0 \\ 1 \end{Bmatrix} \text{ if } \mathbf{d}^{\mathrm{T}}_{2L+1,\tilde{k}+1} = \begin{Bmatrix} [0,1] \text{ or } [1,0] \\ [1,1] \end{Bmatrix}. \tag{73}$$

Using parameter vectors $\alpha_g \in \mathbb{R}^{3\times 1}$ and $\beta_g \in \mathbb{R}^{3\times 1}$, (73) is translated into:

$$\begin{aligned} \mathcal{B}_m(2L+1,\tilde{k}+1) &\geq \alpha_g^{\mathrm{T}}(1:2)\cdot \mathbf{d}_{2L+1,\tilde{k}+1} + \alpha_g(3), \\ \mathcal{B}_m(2L+1,\tilde{k}+1) &\leq \beta_g^{\mathrm{T}}(1:2)\cdot \mathbf{d}_{2L+1,\tilde{k}+1} + \beta_g(3). \end{aligned} \tag{74}$$

For any $2l$-th row of $\mathcal{B}_m$ (with $l \in \{1,\cdots,L\}$), which refers to $b_{(l-1,l),\tilde{k}}$, only one entry equals 0 (indicating that the reset is only triggered once), what can be enforced by:

$$\sum_{\tilde{k}=0}^{N+L} \mathcal{B}_m(2l,\tilde{k}+1) = N+L, \quad \forall l \in \{1,\cdots,L\}. \tag{75}$$

Finally, for the last row, referring to $b_{(L,X_g),\tilde{k}}$, only the last entry $\mathcal{B}_m(2L+2,N+L+1)$ is forced to 0, modeling $x_N \in X_g$. This is translated into:

$$\mathcal{B}_m(2L+2,N+L+1) = 0. \tag{76}$$

The condition that $x_{\tilde{k}} \in X_g$ if $x_{\hat{k}} \in X_g$ for $\tilde{k} \geq \hat{k}$ is modeled by:

$$\mathcal{B}_m(2L+2,\tilde{k}) \geq \mathcal{B}_m(2L+2,\tilde{k}+1). \tag{77}$$

Note that the options considered for $\mathbf{d}^{\mathrm{T}}_{1,\tilde{k}+1}$ in (68), for $\mathbf{d}^{\mathrm{T}}_{2l+1,\tilde{k}+1}$ in (71), and for $\mathbf{d}^{\mathrm{T}}_{2L+1,\tilde{k}+1}$ in (73) are sufficient to encode the part of $\mathcal{B}_m$ which corresponds to the change of phases. Using this fact, and the constructive rules provided above to determine the linear inequalities formulated for elements of $\mathcal{B}_m$, the following fact can be established:

**Lemma 2.** *If a binary matrix* $\mathcal{B}_m \in \{0,1\}^{(2L+2)\times(N+L+1)}$ *with first column* $\mathcal{B}_m(:,1) = \mathbf{b}_0$ *and last column* $\mathcal{B}_m(:, N+L+1) = \mathbf{b}_{N+L}$ *satisfies the constraints* (70)*,* (72)*, and* (74) *to* (77)*, then it has the same structure as in* (65)*.* □

Lemma 1 and 2 together also imply that these constraints encode the set of admissible trajectories of $HA$. All constraints introduced for the matrix $\mathcal{B}_m$ can be collected in the set of linear constraints:

$$\mathcal{Q}\cdot\mathcal{B}_m \leq \mathcal{W} + \mathcal{N}, \tag{78}$$

where the matrices $\mathcal{Q}$, $\mathcal{W}$, and $\mathcal{N}$ depend on the various parameter vectors $\alpha$ and $\beta$. The constraints in (78) reduce the value combinations of the respective binary variables in $\mathcal{B}_m$ from $2^{(2L+2)\times(N+L)}$ to $\binom{N+L}{2L}$.

The search for an admissible run $\phi_x$ and $\phi_z$ of $HA$ now means to let $\mathcal{B}_m$ satisfy (66) and (78), i.e. the transformed problem is:

**Problem 2.** *For a given phase sequence* $\phi_p$*, determine input sequences* $\phi_u^*$ *and a matrix* $\mathcal{B}_m^*$ *as solution to:*

$$\min_{\phi_u,\mathcal{B}_m} \sum_{\tilde{k}=0}^{N+L} \{(\overline{x}_{\tilde{k}+1} - \overline{x}_{c,\tilde{k}+1})^{\mathrm{T}} Q (\overline{x}_{\tilde{k}+1} - \overline{x}_{c,\tilde{k}+1}) \tag{79a}$$

$$+ (u_{\tilde{k}} - u_g)^{\mathrm{T}} R (u_{\tilde{k}} - u_g)\} + q_g \cdot \sum_{\tilde{k}=0}^{N+L} \mathcal{B}_m(2L+2, \tilde{k}+1)$$

$$\textit{s.t.: } \mathcal{Q}\cdot\mathcal{B}_m \leq \mathcal{W} + \mathcal{N}; \tag{79b}$$

$$\textit{for } \tilde{k} \in \{1,\ldots,N+L\}:$$

$$\overline{x}_{\tilde{k}} \leq x_{\tilde{k}} + \lambda_x \cdot (L - \sum_{i=1}^{L} \mathcal{B}_m(2i, \tilde{k}+1)), \tag{79c}$$

$$\overline{x}_{\tilde{k}} \geq x_{\tilde{k}} - \lambda_x \cdot (L - \sum_{i=1}^{L} \mathcal{B}_m(2i, \tilde{k}+1)), \tag{79d}$$

$$\overline{x}_{\tilde{k}} \leq \lambda_x \cdot (\sum_{i=1}^{L} \mathcal{B}_m(2i, \tilde{k}+1) + 1 - L), \tag{79e}$$

$$\overline{x}_{\tilde{k}} \geq -\lambda_x \cdot (\sum_{i=1}^{L} \mathcal{B}_m(2i, \tilde{k}+1) + 1 - L); \tag{79f}$$

$$\overline{x}_{c,\tilde{k}} = \sum_{i=1}^{L} (1 - \mathcal{B}_m(2i, \tilde{k})) \cdot x_c^{i-1,i}; \tag{79g}$$

$$x_{\tilde{k}} = \sum_{i=0}^{L} [A_i \cdot \xi_{\tilde{k},i} + B_i \cdot \pi_{\tilde{k},i}] + \sum_{i=0}^{L-1} \xi_{\tilde{k},(i,i+1)}; \tag{79h}$$

$$\mathcal{C} \cdot x_{\tilde{k}} \leq \mathcal{D} + diag(\mathcal{B}_m(:, \tilde{k}+1)) \cdot \mathcal{M}, \;\; u_{\tilde{k}-1} \in U; \tag{79i}$$

$$for \;\; i \in \{0, \cdots, L-1\}:$$

$$\xi_{\tilde{k},i} \leq \Theta_i^+ \cdot (\mathcal{B}_m(2i+2, \tilde{k}) - \mathcal{B}_m(2i+1, \tilde{k})), \tag{79j}$$

$$\xi_{\tilde{k},i} \geq \Theta_i^- \cdot (\mathcal{B}_m(2i+2, \tilde{k}) - \mathcal{B}_m(2i+1, \tilde{k})), \tag{79k}$$

$$\xi_{\tilde{k},i} \leq x_{\tilde{k}-1} + \lambda_x \cdot (1 - \mathcal{B}_m(2i+2, \tilde{k}) + \mathcal{B}_m(2i+1, \tilde{k})), \tag{79l}$$

$$\xi_{\tilde{k},i} \geq x_{\tilde{k}-1} - \lambda_x \cdot (1 - \mathcal{B}_m(2i+2, \tilde{k}) + \mathcal{B}_m(2i+1, \tilde{k})), \tag{79m}$$

$$\pi_{\tilde{k},i} \leq \Theta_u^+ \cdot (\mathcal{B}_m(2i+2, \tilde{k}) - \mathcal{B}_m(2i+1, \tilde{k})), \tag{79n}$$

$$\pi_{\tilde{k},i} \geq \Theta_u^- \cdot (\mathcal{B}_m(2i+2, \tilde{k}) - \mathcal{B}_m(2i+1, \tilde{k})), \tag{79o}$$

$$\pi_{\tilde{k},i} \leq u_{\tilde{k}-1} + \lambda_u \cdot (1 - \mathcal{B}_m(2i+2, \tilde{k}) + \mathcal{B}_m(2i+1, \tilde{k})), \tag{79p}$$

$$\pi_{\tilde{k},i} \geq u_{\tilde{k}-1} - \lambda_u \cdot (1 - \mathcal{B}_m(2i+2, \tilde{k}) + \mathcal{B}_m(2i+1, \tilde{k})), \tag{79q}$$

$$\xi_{\tilde{k},(i,i+1)} \leq \Theta_{i+1}^+ \cdot (1 - \mathcal{B}_m(2i+2, \tilde{k})), \tag{79r}$$

$$\xi_{\tilde{k},(i,i+1)} \geq \Theta_{i+1}^- \cdot (1 - \mathcal{B}_m(2i+2, \tilde{k})), \tag{79s}$$

$$\xi_{\tilde{k},(i,i+1)} \leq E_{(i,i+1)} \cdot x_{\tilde{k}-1} + e_{(i,i+1)} + \lambda_x \cdot \mathcal{B}_m(2i+2, \tilde{k}), \tag{79t}$$

$$\xi_{\tilde{k},(i,i+1)} \geq E_{(i,i+1)} \cdot x_{\tilde{k}-1} + e_{(i,i+1)} - \lambda_x \cdot \mathcal{B}_m(2i+2, \tilde{k}). \tag{79u}$$

The cost function (79a) is an equivalent reformulation of the one in Prob. 1, where $\overline{x}_{c,\tilde{k}}$ depends on the guard set relevant for $\tilde{k}$, according to (79g). The sum in the last term of (79a) counts the total number of steps in which $x_{\tilde{k}}$ is not in $X_g$.

The constraints (79c) to (79f) ensure that the costs induced by the intermediate states $x'$ are not recorded in the cost function. The conditions (79b) and (79i) force the resulting trajectory $\phi_x$ to comply to $\phi_p$. The equations and inequalities (79h) and (79j) to (79u) refer to standard reformulations of the hybrid dynamics by introducing auxiliary variables $\xi_{\tilde{k},i}$, $\pi_{\tilde{k},i}$, and $\xi_{\tilde{k},(i,i+1)}$. Details of such reformulations can be found in [72]. In addition, the following parameters have to be determined:

$$\begin{aligned} \Theta_i^+ &= \begin{bmatrix} \max\limits_{x \in I_i} x_1 & \cdots & \max\limits_{x \in I_i} x_{n_x} \end{bmatrix}^{\mathrm{T}}, \\ \Theta_u^+ &= \begin{bmatrix} \max\limits_{u \in U} u_1 & \cdots & \max\limits_{u \in U} u_{n_u} \end{bmatrix}^{\mathrm{T}}, \end{aligned} \tag{80}$$

and likewise for minimal values in $\Theta_i^-$ and $\Theta_u^-$. The relaxation vectors $\lambda_x \in R^{n_x}$, $\lambda_u \in R^{n_u}$ are selected, to have for all $x \in X$ and $u \in U$:

$$\begin{aligned} x + \lambda_x \gg 0^{n_x \times 1}, \;\; x - \lambda_x \ll 0^{n_x \times 1}, \\ u + \lambda_u \gg 0^{n_u \times 1}, \;\; u - \lambda_u \ll 0^{n_u \times 1}. \end{aligned} \tag{81}$$

Since all constraints in Prob. 2 are linear, the optimization represents an MIQP problem, which can be solved by existing solvers. The constraints (79b) reduce the possible combinations of values for the binary variables significantly. The obtained $\mathcal{B}_m^*$ determines $\phi_v^*$ straightforwardly. Furthermore, since (79b) admits all possible values of $\mathcal{B}_m$ corresponding to the structure in (65) and since no approximation is involved, the following applies:

**Corollary 1.** *If no feasible solution exists to Problem 2, then there exists no* admissible *trajectory corresponding to the given phase sequence $\phi_p$.* □

Thus, Prob. 2 can be used to verify the existence of an *admissible* trajectory satisfying Prob. 1 for the considered $\phi_p$.

**Theorem 4.** *If the solution of Problem 2 returns a feasible solution $\phi_u^*$ and $\mathcal{B}_m^*$, then it represents the optimal solution of Problem 1 for the given phase sequence $\phi_p$.* □

This result follows from the relation between Prob. 2 and Prob. 1 for the given $\phi_p$ as established by Lemma 1 and 2, and from the fact that solvers for MIQP problem terminate with the optimal solution if the search tree is fully explored.

If now Prob. 1 is addressed without restriction to certain single $\phi_p$, the solution is obtained by solving one instance of Prob. 2 for any possible phase sequence connecting $z_0$ with $z_g$. If the number of possible phase sequences connecting the initial discrete state $z_0$ and the target state $z_g$ is not very high[1], the search can be carried out by enumeration.

### 4.4 Numeric Examples

To illustrate the procedure, we consider the example of an $HA$ with $x \in \mathbb{R}^3$ and 5 discrete states $Z = \{z_0, z_1, z_2, z_3, z_g\}$. The invariant sets of these states are marked by yellow regions, and the guard sets by orange regions in the following figures. The continuous dynamics, reset functions, and input constraints are parametrized suitably (but not shown here for brevity), and the set of transitions follows from the adjacency of the invariant sets. The initial state is $x_0 = [12, -7, 0]^{\mathrm{T}} \in I_0$, and the terminal state is set to $x_g = [-2, -12, -2]^{\mathrm{T}} \in I_g$. The terminal region $X_g$ is marked as a green region in the figures, and $N$ is first selected to be 15, which leads to a number of 102 binary variables to be employed in Problem 2. Three different phase sequences are possible, and the respective trajectories are shown in Figs. 15 and 16. Only for $\phi_p = \{z_0, z_2, z_g\}$ and $\phi_p = \{z_0, z_3, z_g\}$ optimal admissible trajectories are found with $N = 15$, leading to costs of 3135.18 and 3429.26, and requiring computation times of 0.080$s$ and 0.096$s$ on a 3.4GHz processor using Matlab 2015a and the solver CPLEX. Through constraint (78),

[1] As applies not seldomly for hybrid models

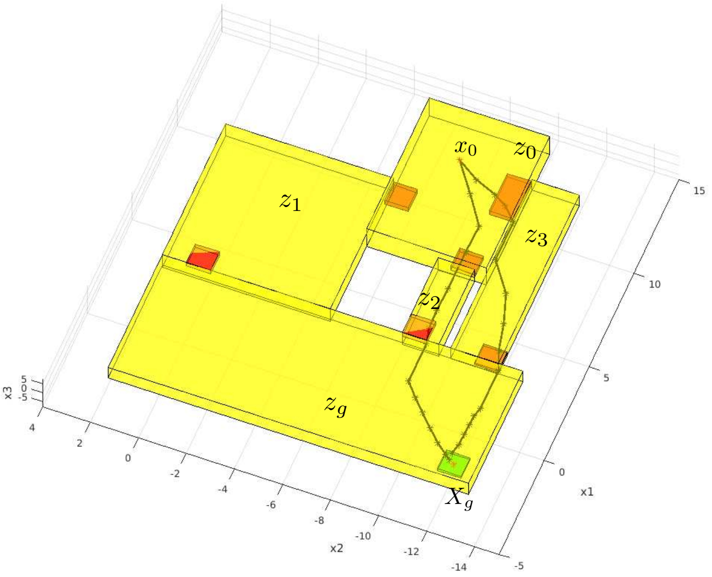


**Figure 15:** Optimal trajectory for $\phi_p = \{z_0, z_2, z_g\}$ and $\phi_p = \{z_0, z_3, z_g\}$.

the relevant combinations of the binary variables are reduced from $2^{102}$ to $\binom{17}{4} = 2380$, and the time to verify the infeasibility of $\phi_p = \{z_0, z_1, z_g\}$ for $N = 15$ is about $0.01sec$. If, for the latter $\phi_p$, the time horizon is increased to $N = 25$, then the admissible trajectory shown in Fig. 16 is obtained with optimal cost of 6160.51 computed in $0.717sec$. A further test with a

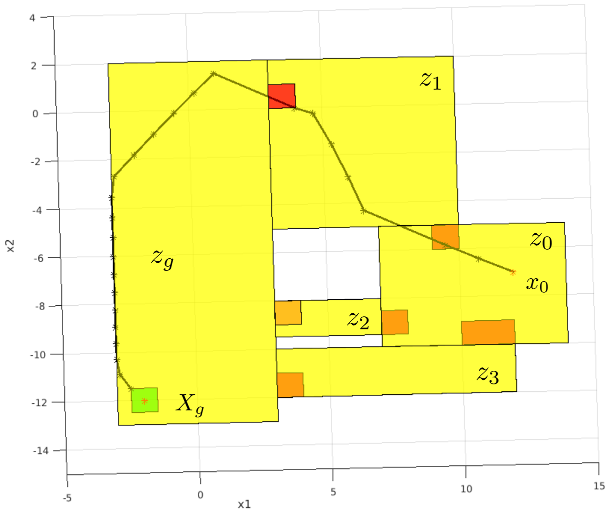


**Figure 16:** Optimal trajectory for $\phi_p = \{z_0, z_1, z_g\}$ with $N = 25$.

longer $\phi_p$ using $L = 3$ and a horizon $N = 24$ is illustrated in Fig. 17, obtained in $1.06sec$.

In summary, this section has described a new method for efficient trajectory optimization of

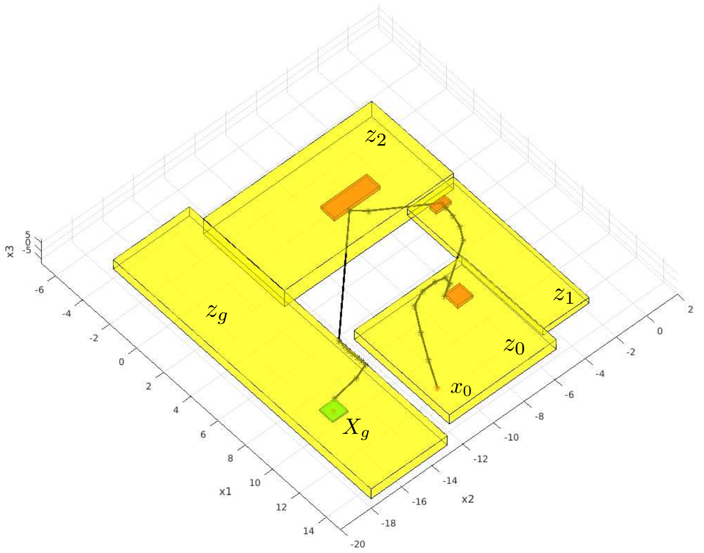


**Figure 17:** Optimal trajectory for $\phi_p = \{z_0, z_1, z_2, z_g\}$ with $N = 24$.

CPS with hybrid dynamics. The key aspect of the method is to cast the semantics of admissible trajectories into a tailored set of linear constraints which reduce the value combinations of binary variables required to formulate the transition dynamics. The significant reduction of the number of value combinations, also reduces the search space of the underlying MIP, and thus increases the computational efficiency. The procedure does not involve approximation and thus ensures that the globally optimal solution is found.

# 5 Summarizing Conclusions

This document has reported on mechanisms of combining (the outcome of) reachable set computations with online controller synthesis in terms of model predictive control. The contributions are as follows:

- Taking advantage from reachable set computations, a dual-mode model predictive control algorithm was proposed which leads to provably safe behavior even in presence of disturbances and uncertain measurements. Due to the online computation of reachable sets, the proposed method is less conservative than those often seen in literature. In addition, the computation time of the method is relatively low, and this time is considered when computing the reachable sets.
- Using the principle particle filters, probabilistic reach sets with arbitrary probability distributions was embedded into a version of stochastic model predictive control for system with disturbances without bounded suport. The method relaxes the state constraints when the original problem is not feasible. The adoption of a scenario-based approach leads to computational tractability where alternative methods are not applicable.
- A method for trajectory optimization of hybrid systems within MPC was proposed, which cast the semantics of admissible trajectories into a tailored set of linear constraints of a mixed-integer optimization problems. The constraints involve a relatively small number of binary variables required to formulate the transition dynamics. This significantly reduces the search space of the optimization and contributes to good computational efficiency. The procedure does not involve approximation and thus ensures that the globally optimal solution is found.

Overall, the proposed methods all combine techniques of online reachable set comptations into synthesizing control strategies or laws online by predictive control. This combination ensures properties of safety and robustness against disturbances or environment effects while the control strategies or laws are adapted to the current situation encountered by the CPS.

Note that the breadth of techniques described in this report covers relatively different model classes of CPS, from continuous nonlinear dynamics with bounded disturbances, over affine systems with stochastic disturbances, to distributed hybrid dynamics with affine continuous-valued dynamics. This set of techniques enables one to select the method which is most suited for a given application. If, an application requires a model in between the three classes listed above, the combination of the presented ideas appears to be possible.

The presented techniques are all implemented in Matlab (calling dedicated additional optimization packages). The implementations establish the unit for controller synthesis within the tool chain of UnCoVerCPS, as developed in WP4 of the project.

Since the techniques all rely on MPC, they are particularly useful for applications for which a control strategy has to be computed online in response to current specifications and constraints imposed by the environment. With respect to the case studies from WP5, this applies for the autonmous driving use case (in which a planned trajectory has to circumvent the actual position and planned trajectory of other autonomous cars), and for the use case of human-robot interaction (when the trajectory of the robot arm has to be adapted to avoid collision with the human operator), see D5.3. The MPC approaches have been successfully applied to these two use-cases.

# References

[1] T. Alamo, R. Tempo, and E. F. Camacho. A randomized strategy for probabilistic solutions of uncertain feasibility and optimization problems. *IEEE Transactions on Automatic Control*, 54:2545–2559, 2009.

[2] T. Alamo, R. Tempo, and A. Luque. On the sample complexity of randomized approaches to the analysis and design under uncertainty. In *Proc. of the 2010 Amer. Contr. Conf.*, pages 4671–4676, Baltimore, MD, USA, June 2010.

[3] T. Alamo, R. Tempo, A. Luque, and D. Ramirez. Randomized methods for design of uncertain systems: Sample complexity and sequential algorithms. *Automatica*, 52:160 – 172, 2015.

[4] M. Althoff. *Reachability Analysis and its Application to the Safety Assessment of Autonomous Cars*. PhD thesis, Technische Universität München, 2010.

[5] M. Althoff. An introduction to CORA 2015. In *Proc. of the Workshop on Applied Verification for Continuous and Hybrid Systems*, pages 120–151, 2015.

[6] I. Batina, A. Stoorvogel, and S. Weiland. Optimal control of linear, stochastic systems with state and input constraints. In *Proc. of the 41st IEEE Conference on Decision and Control*, Dec. 2002.

[7] A. Bemporad and M. Morari. Control of systems integrating logic, dynamics, and constraints. *Automatica*, 35(3):407–427, 1999.

[8] D. Bertsimas and D. Brown. Constrained stochastic LQC: A tractable approach. *Automatic Control, IEEE Transactions on*, 52(10):1826–1841, 2007.

[9] G. Bitsoris. On the positive invariance of polyhedral sets for discrete-time systems. *Systems & control letters*, 11(3):243–248, 1988.

[10] L. Blackmore and M. Ono. Convex chance constrained predictive control without sampling. *Proceedings of the AIAA Guidance, Navigation and Control Conference*, 2009.

[11] S. Branicky, S. Borkar, and K. Mitter. A unified framework for hybrid control: Model and optimal control theory. *IEEE Trans. Automatic Control*, 43(1):31–45, 1998.

[12] J. Bravo, T. Alamo, and E. Camacho. Robust MPC of constrained discrete-time nonlinear systems based on approximated reachable sets. *Automatica*, 42(10):1745 – 1751, 2006.

[13] C. Brocchini, A. Falsone, G. Manganini, O. Holub, and M. Prandini. A chance-constrained approach to the quantized control of a heat ventilation and air conditioning system with prioritized constraints. In *Proceedings of the 22nd International Symposium on Mathematical Theory of Networks and Systems (MTNS 2016), Minneapolis, Minnesota, USA*, pages 137–144, July 2016.

[14] G. Calafiore and M. Campi. Uncertain convex programs: randomized solutions and confidence levels. *Mathematical Programming*, 102(1):25–46, 2005.

[15] G. Calafiore and M. Campi. The scenario approach to robust control design. *IEEE Transactions on Automatic Control*, 51(5):742–753, 2006.

[16] G. Calafiore and L. Fagiano. Robust model predictive control via random convex programming. In *Decision and Control and European Control Conference (CDC-ECC), 2011 50th IEEE Conference on*, pages 1910–1915, 2011.

[17] G. Calafiore and L. Fagiano. Robust model predictive control via scenario optimization. *Automatic Control, IEEE Transactions on*, 58(1):219–224, 2013.

[18] E. Camacho and C.Bordons. *Model Predictive Control.* Springer, London, 2004.

[19] M. Campi and A. Carè. Random convex programs with $l_1$-regularization: sparsity and generalization. *SIAM Journal on Control and Optimization*, 51(5):3532–3557, 2013.

[20] M. Campi and S. Garatti. The exact feasibility of randomized solutions of uncertain convex programs. *SIAM Journal on Optimization*, 19(3):1211–1230, 2008.

[21] M. Campi and S. Garatti. A sampling-and-discarding approach to chance-constrained optimization: Feasibility and optimality. *Journal of Optimization Theory and Applications*, 148(2):257–280, 2011.

[22] M. Campi and S. Garatti. Wait-and-judge scenario optimization. *Mathematical Programming*, 167(1):155–189, 2018.

[23] M. Campi, S. Garatti, and M. Prandini. The scenario approach for systems and control design. *Annual Reviews in Control*, 33(2):149–157, 2009.

[24] M. Cannon, B. Kouvaritakis, S. Rakovic, and Q. Cheng. Stochastic tubes in model predictive control with probabilistic constraints. *IEEE Transactions on Automatic Control*, 56(1):194–200, 2011.

[25] A. Carè, S. Garatti, and M. Campi. Scenario min-max optimization and the risk of empirical costs. *SIAM Journal on Optimization*, 25(4):2061–2080, 2015.

[26] Q. Cheng, M. Cannon, B. Kouvaritakis, and M. Evans. Stochastic MPC for systems with both multiplicative and additive disturbances. In *Proceedings of the 19th IFAC World Congress*, Cape Town, South Africa, 2014.

[27] E. Cinquemani, M. Agarwal, D. Chatterjee, and J. Lygeros. Convexity and convex approximations of discrete-time stochastic control problems with constraints. *Automatica*, 47(9):2082–2087, 2011.

[28] C. Combastel. A state bounding observer based on zonotopes. In *Proc. of the European Control Conference*, pages 2589–2594, 2003.

[29] L. Deori, S. Garatti, and M. Prandini. Stochastic constrained control: Trading performance for state constraint feasibility. In *Proceedings of the 2013 European Control Conference*, pages 2740–2745, Zurich, Switzerland, 2013.

[30] L. Deori, S. Garatti, and M. Prandini. Trading performance for state constraint feasibility in stochastic constrained control: a randomized approach. *Journal of the Franklin Institute*, 354(354):501–529, 2017.

[31] L. Deori, S. Garatti, and M. Prandini. 4-d flight trajectory tracking: A receding horizon approach integrating feedback linearization and scenario optimization. *IEEE Transactions on Control Systems Technology*, pages 1–16, 2018.

[32] M. Egerstedt, Y. Wardi, and F. Delmotte. Optimal control of switching times in switched dynamical systems. In *Proc. 42nd IEEE Conf. on Decision and Control*, volume 3, pages 2138–2143, 2003.

[33] A. El-Guindy, D. Han, and M. Althoff. Estimating the region of attraction via forward reachable sets. In *Proc. of the American Control Conference*, pages 1263–1270. IEEE, 2017.

[34] M. Evans, M. Cannon, and B. Kouvaritakis. Robust mpc for linear systems with bounded multiplicative uncertainty. In *Proceedings of the 51st IEEE Conference on Decision and Control*, 2012.

[35] M. Farina, L. Giulioni, L. Magni, and R. Scattolini. An mpc approach to output-feedback control of stochastic linear discrete-time systems. *IEEE Transactions on Automatic Control*, 55:140–149, 2015.

[36] M. Farina, L. Giulioni, and R. Scattolini. Stochastic linear model predictive control with chance constraints - a review. *Journal of Process Contro*, 44:53–67, 2016.

[37] P. Goulart, E. Kerrigan, and J. Maciejowski. Optimization over state feedback policies for robust control with constraints. *Automatica*, 42(4):523–533, April 2006.

[38] M. Grant and S. Boyd. CVX: Matlab software for disciplined convex programming, version 2.1. `http://cvxr.com/cvx`, Mar. 2014.

[39] S. Hedlund and A. Rantzer. Optimal control of hybrid systems. In *Proc. IEEE Conf. on Decision and Control*, pages 3972–3977, 1999.

[40] T. Henzinger. The Theory of Hybrid Automata. In *Proc.* $11^{th}$ *IEEE Symp. on Logic in Comp. Science*, pages 278–292, 1996.

[41] D. V. Hessem and O. Bosgra. Stochastic closed-loop model predictive control of continuous nonlinear chemical processes. *Journal of Process Control*, 16(3):225 – 241, 2006.

[42] P. Hokayem, E. Cinquemani, D. Chatterjee, F. Ramponi, and J. Lygeros. Stochastic receding horizon control with output feedback and bounded controls. *Automatica*, 48(1):77 – 88, 2012.

[43] B. Houska, H. Ferreau, and M. Diehl. ACADO Toolkit – An Open Source Framework for Automatic Control and Dynamic Optimization. *Optimal Control Applications and Methods*, 32(3):298–312, 2011.

[44] S. Karaman, G. Sanfelice, and E. Frazzoli. Optimal control of mixed logical dynamical systems with linear temporal logic specifications. In *Proc. 47th IEEE Conf. on Decision and Control*, pages 2117–2122, 2008.

[45] B. Kouvaritakis and M. Cannon. *Model Predictive Control.* Springer, 2016.

[46] V. T. H. Le, C. Stoica, T. Alamo, E. F. Camacho, and D. Dumur. Zonotopic guaranteed state estimation for uncertain systems. *Automatica*, 49(11):3418–3424, 2013.

[47] J. Löfberg. Yalmip: a toolbox for modeling and optimization in MATLAB. In *Proc. of the CACSD Conference*, Taipei, Taiwan, 2004.

[48] M. Lorenzen, F. Dabbene, R. Tempo, and F. Allgöwer. Constraint-tightening and stability in stochastic model predictive control. *IEEE Transactions on Automatic Control*, 62(7):3165–3177, July 2017.

[49] N. Lynch, R. Segala, and F. Vaandrager. Hybrid I/O Automata. *Information and Computation*, 185:105–157, 2003.

[50] J. Maciejowski. *Predictive Control with constraints.* Prentice Hall, Harlow, England, 2002.

[51] K. Margellos, M. Prandini, and J. Lygeros. On the connection between compression learning and scenario based single-stage and cascading optimization problems. *IEEE Transactions on Automatic Control*, 60(10):2716–2721, Oct 2015.

[52] D. Q. Mayne, M. M. Seron, and S. Raković. Robust model predictive control of constrained linear systems with bounded disturbances. *Automatica*, 41(2):219–224, 2005.

[53] H. Michalska and D. Q. Mayne. Robust receding horizon control of constrained nonlinear systems. *IEEE Transactions on Automatic Control*, 38(11):1623–1633, 1993.

[54] V. Nenchev, C. Belta, and J. Raisch. Optimal motion planning with temporal logic and switching constraints. In *Proc. European Control Conf.*, pages 1141–1146, 2015.

[55] M. Ono and B. Williams. Iterative risk allocation: A new approach to robust model predictive control with a joint chance constraint. In *Proc. of the 47th IEEE Conference on Decision and Control*, Dec. 2008.

[56] B. Passenberg, M. Kröninger, G. Schnattinger, M. Leibold, O. Stursberg, and M. Buss. Initialization concepts for optimal control of hybrid systems. *IFAC Proceedings Volumes*, 44(1):10274–10280, 2011.

[57] A. Platzer and E. M. Clarke. The image computation problem in hybrid systems model checking. In *International Workshop on Hybrid Systems: Computation and Control*, pages 473–486, 2007.

[58] A. Pnueli. The temporal logic of programs. In *18th IEEE Symp. on Foundations of Computer Science*, pages 46–57, 1977.

[59] M. Prandini, Garatti, S., and J. Lygeros. A Randomized Approach to Stochastic Model Predictive Control. In *IEEE Conference on Decision and Control*, Maui, Hawaii, USA, Dec. 2012.

[60] J. Primbs. A soft constraint approach to stochastic receding horizon control. In *Proc. of the 46th IEEE Conference on Decision and Control*, Dec. 2007.

[61] J. Primbs and H. Chang. Stochastic receding horizon control of constrained linear systems with state and control multiplicative noise. *Automatic Control, IEEE Transactions on*, 54(2):221–230, 2009.

[62] S. Rakovic, B. Kouvaritakis, M. Cannon, C. Panos, and R. Findeisen. Parameterized tube model predictive control. *IEEE Transactions on Automatic Control*, 57:2746–2761, 2012.

[63] S. V. Rakovic, E. C. Kerrigan, K. I. Kouramas, and D. Q. Mayne. Invariant approximations of the minimal robust positively invariant set. *IEEE Transactions on Automatic Control*, 50(3):406–410, 2005.

[64] J. Rawlings and D. Mayne. *Model Predictive Control Theory and Design.* Nob Hill Publishing, Madison, WI, 2009.

[65] P. Riedinger, F. Kratz, C. Iung, and C. Zanne. Linear quadratic optimization for hybrid systems. In *IEEE Conf. on Decision and Control*, pages 3059–3064, 1999.

[66] G. Schildbach, G. Calafiore, L. Fagiano, and M. Morari. Randomized Model Predictive Control for Stochastic Linear Systems. In *American Control Conference*, pages 417–422, Montreal, Canada, June 2012.

[67] G. Schildbach, L. Fagiano, C. Frei, and M. Morari. The Scenario Approach for Stochastic Model Predictive Control with Bounds on Closed-Loop Constraint Violations. *Automatica*, 50(12):3009–3018, 2014.

[68] G. Schildbach, L. Fagiano, and M. Morari. Randomized solutions to convex programs with multiple chance constraints. *SIAM Journal on Optimization*, 23(4):2479–2501, 2013.

[69] B. Schürmann and M. Althoff. Optimal control of sets of solutions to formally guarantee constraints of disturbed linear systems. In *Proc. of the American Control Conference*, pages 2522–2529, 2017.

[70] B. Schürmann and M. Althoff. Reachset model predictive control of disturbed nonlinear systems. In *Proc. of the 57th Conference on Decision and Control*, 2018.

[71] M. S. Shaikh and P. E. Caines. On the hybrid optimal control problem: Theory and algorithms. *IEEE Trans. on Automatic Control*, 52:1587–1603, 2007.

[72] O. Stursberg and S. Panek. Control of switched hybrid systems based on disjunctive formulations. *Hybrid systems: Computation and Control*, pages 820–871, 2002.

[73] R. Tempo, G. Calafiore, and F. Dabbene. *Randomized Algorithms for Analysis and Control of Uncertain Systems, with Applications*. Springer-Verlag, London, UK, 2013.

[74] J. Till, S. Engell, S. Panek, and O. Stursberg. Applied Hybrid System Optimization - An Empirical Investigation of Complexity. *Control Engineering Practice*, 12(10):1269–1278, 2004.

[75] H. Williams. *Model Building in Mathematical Programming*. Wiley, 1st edition, 1978.

[76] E. M. Wolff, U. Topcu, and R. M. Murray. Optimization-based trajectory generation with linear temporal logic specifications. In *Proc. of the International Conference on Robotics and Automation*, pages 5319–5325, 2014.

[77] X. Xu and P. J. Antsaklis. Optimal control of switched systems based on parameterization of the switching instants. *IEEE Trans. on Automatic Control*, 49:2–16, 2004.

[78] X. Zhang, S. Grammatico, K. Margellos, P. Goulart, and J. Lygeros. Randomized nonlinear MPC for uncertain control-affine systems with bounded closed-loop constraint violations. In *Proceedings of the 19th IFAC World Congress*, Cape Town, South Africa, 2014.

[79] X. Zhang, S. Grammatico, G. Schildbach, P. Goulart, and J. Lygeros. On the sample size of random convex programs with structured dependence on the uncertainty. *Automatica*, 60:182–188, 2015.