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

WP1 Modelling and conformance testing

D1.2 – Report on modelling of networked cyber-physical system for verification and control

WP1	**D1.2 – Report on modelling of networked cyber-physical system for verification and control**
Authors	Olaf Stursberg, Damian Kontny, Zonglin Liu - UKS Alexander Rausch, Jens Oehlerking - Bosch Maria Prandini - PoliMi Goran Frehse - UJF
Short Description	This deliverable reports on the model basis used in UnCoVerCPS, as developed in the Tasks 1.1 and 1.2. Starting from a general definition of cyber-physical systems, the report describes a set of techniques for deriving abstracted or approximated model variants tailored to procedures of conformance testing, verification, and control design.
Deliverable Type	Report
Dissemination level	Public
Delivery Date	30 Jun 2017
Contributions by	UKS, Bosch, PoliMi, UJF
Internal review by	Maria Prandini, Joshue Perez, Matthias Althoff
External review by	
Internally accepted by	
Date of acceptance	

Document history:

Version	Date	Author	Description
1.0	April 1, 2017	O. Stursberg et al.	Version for 1st internal review
1.1	May 31, 2017	O. Stursberg et al.	Version for 2nd internal review
1.2	June 20, 2017	O. Stursberg et al.	Version for 3rd internal review
1.3	June 22, 2017	O. Stursberg et al.	Version for review by the project coordinator
1.3	June 28, 2017	O. Stursberg et al.	Version in response to the latest review

Contents

1 Introduction and Motivation

The application domains of cyber-physical systems (CPS) as envisaged in UnCoVerCPS exhibit three main modeling challenges: 1) Their structure is networked either by coupling of the dynamic behavior of subsystems, or by communication of joint goals or specifications. 2) The interaction among the subsystems, or the interaction of the CPS with the environment evolves over time, such that not all conceivable situations can be considered at design time, but the procedures of control and verification must rely on models which are adapted at runtime based on current measurements. 3) The dynamics is only suitably modeled if the intertwined behavior of continuous or hybrid plant dynamics with the digital and discrete-time signal processing and algorithms for control and verification is considered.

Given the project ambition to provide online procedures for control and verification, the requirement of being applicable in real-time calls for specific routines to handle the complexity of a general CPS model which accounts for all these challenges. According to the general philosophy of UnCoVerCPS, the use of approximated models (e.g. to enable online control synthesis) is justified, if they are complemented by analysis techniques relying on the original or abstracted models, such that desired system properties like stability or safety can be guaranteed.

These considerations led us to: (1.) introducing a general model class for CPS, which comprises model components to account for the above challenges and thus serves as a comprehensive modeling framework for the developments in UnCoVerCPS, (2.) covering a set of methods to modularize, abstract, or approximate the general model format to the purpose of conformance testing, verification, and control design, each one of these objectives calling for an appropriate treatment so as to obtain a model that is consistent with its intended use, (3.) evaluating the various modeling and model transformation procedure for different instances of the project case studies.

By these developments, the state-of-the-art in modeling CPS is advanced with respect to:

- a new model definition which combines hybrid dynamics with a communication structure and time-varying state constraints (invariants and guards),
- a new modeling procedure for conformance monitors which enable the verification of conformance between abstract and concrete systems,
- a new technique for reducing the dimension of the continuous state space for continuous-time switched affine systems (in the context of probabilistic verification),

- a model reduction scheme for discrete-time piecewise affine systems such that the input-ouput behavior is preserved for verification,
- and techniqes of model-approximations to represent CPS with switched inputs, or time-varying state constraints respectively, in a form which is amenable to techniques of online optimizing control.

The deliverable is organized as follows: The general model structure for hybrid networked CPS is presented in Section 2, including a paragraph describing how the model format can be translated into the language of *SpaceEx*, one of the central verification tools of the project. In Section 3, a conformance monitor for model-to-model conformance testing is presented, which relates two models of the same system on different abstraction levels. Model reduction techniques of affine systems for probabilistic verification and of piecewise affine systems for preserving the input/output relation are described and illustrated via some numerical examples in Section 4. Model approximation of the general CPS into modular nonlinear or hybrid automata with time-varying constraints and affine dynamics as a suitable basis for online control synthesis is covered in Section 5 (including also examples of application to robot modeling and automated vehicles). The deliverable concludes with a summary in Section 6.

2 General Model Structure for Hybrid Networked CPS

With the ubiquity of information technology, the complexity of systems has dramatically increased due to the number of embedded computing, communication, and sensing devices. The term cyber-physical systems (CPS) has become a common one in this context [41], whereas the definitions of this model class differs among different sources. The common understanding is, however, that CPS combine a physical system part (exhibiting continuous or hybrid dynamics) with digital computing devices, and networking aspects arising from communication networks or coupling of subsystem dynamics. The model definition used in this section stems from the interpretation that CPS can be understood as the extension of hybrid dynamic systems [5, 4, 35, 44] to a notion of inputs/outputs to model the interaction and communication of subsystems – this point of view is, of course, not unique but in line with a series of previous work. The closest definition in this context is the one in [46], which defines hybrid I/O automata. We extend this model to a more general definition which:

- comprises time-varying components, in particular time-varying invariants and guard (as suitable to account for changing restrictions imposed by connected subsystems),
- includes a network of time-varying topology to represent the flow of information between the local controllers of the subsystems.

Due to the different methodical developments based on this model, we describe versions in discrete and continuous time, and show for a simple system [48] how the continuous time version can be represented in the verification tool *SpaceEx* [25].

2.1 Model Definition

The overall model structure considered in this section is shown in Fig. 1. The model comprises N subsystems (with index $i \in \{1, \ldots, N\}$), each of which consists of a plant part and a controller part. The plant parts can affect each other through input and output variables, while the subsystem controllers can exchange information through a communication network. The transmitted information may refer to momentary information of the states or control actions, or to predicted / planned behavior over a future time horizon. In general, the interaction of controlled subsystems in a CPS can be classified into and modeled by:

- coupling of the dynamics of the plants given by the occurrence of output (and/or input) variables in the differential / difference equations of another subsystem;

- restriction of a subsystem behavior through constraints which are derived from the outputs or states of another subsystem (e.g. an autonomous vehicle must not enter a region around another vehicle);
- the control objectives, which may establish a competition (if the subsystems follow only their own local goals) or cooperation (e.g. if a global cost function is minimized, as in the case of distributed MPC).

The set of case studies considered in the project require that the subsystem interaction can be modeled in a time-varying manner. For each of the three named types of subsystem interaction, a time-varying directed graph is an appropriate means to represent the presence and direction of an effect between any pair of subsystems. Let $G(t_k) = (V(t_k), E(t_k))$ denote a graph established at time t_k taken from a discrete time domain $T_G = \{t_0, t_1, \ldots\}$, modeling times in which the graph is modified. The set of vertices $V(t_k) = \{1, \ldots, N(t_k)\}$ represents the subsystems existing in the overall structure from time t_k up to t_{k+1}. The set $E(t_k)$ of directed edges $(i, j) \in E(t_k)$ for $i \neq j$, $\{i, j\} \subseteq V(t_k)$ models the existence of a link from subsystem i to subsystem j in this direction. Such a graph is suitable to establish the system topology by considering an edge (i, j) to model the effect of an output of subsystem i on the dynamics of subsystem j, or to represent that subsystem i imposes constraints on the behavior of the subsystem j, or to indicate that a path of communication exists from i to j. In order to encode the information transmitted along a link, an edge $(i, j) \in E(t_k)$ can be annotated with the specific information transmitted (or imposed) by the sending (or affecting) subsystem.

If an edge $(i, j) \in E(t_k)$ represents the transmission of information, the communication may itself be subject to dynamic behavior modeling imperfect exchange of information. In this case, the delay of communication, or the loss of packets are typically considered, i.e. an edge (i, j) is attributed by a function modeling how sent information is transformed into the received information over time. Such modeling concepts have been extensively studied in the field of *networked control systems* in the past decade, but are not the focus of UnCoVerCPS.

The modeling and design of the subsystem controllers is content of WP2, and are thus described in detail in the corresponding deliverables of that work package. The focus of the following description is instead the modeling of the plant parts of the subsystems. As mentioned above, the objective is to provide a modeling scheme for hybrid, time-varying, interacting subsystem dynamics. To comply with the requirements for the various case studies and methods for control and verification, definitions with continuous and discrete time domain

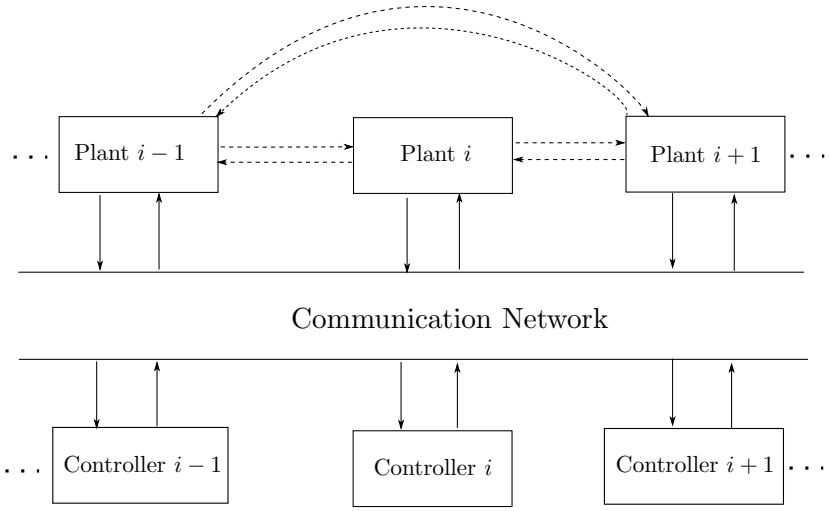

Figure 1: Networked system structure of the CPS.

are covered. Throughout the definitions, the index $i \in \{1, \ldots, N\}$ indicates the subsystem.

Definition 1. *Continuous-time plant of a CPS subsystem*

The continuous-time dynamics of the plant of subsystem i consists of the following elements:

- *a domain of continuous time $T \subset \mathbb{R}^{\geq 0}$ on which the time variable t is defined;*
- *a domain $T_k = \{t_0, t_1, t_2, ...\}$ of discrete points of time $t_k \in \mathbb{R}^{\geq 0}$ in which the discrete state changes;*
- *the hybrid state space $\mathcal{X}^i := Z^i \times X^i$ is the Cartesian product of:*
  - *the set Z^i of discrete states $z_k^i \in Z^i$,*
  - *and the continuous state space $X^i \subseteq \mathbb{R}^{n_c,i}$ on which $x^i(t) \in X^i$ is defined;*

  the hybrid state is denoted by $s^i = \begin{bmatrix} z^i \\ x^i \end{bmatrix}$;
- *the hybrid input space $\mathcal{U}^i := V^i \times U^i$ as the Cartesian product of:*
  - *the set V^i of discrete inputs $v_k^i \in V^i$,*
  - *and the space $U^i \subseteq \mathbb{R}^{m_c,i}$ of continuous inputs $u^i(t) \in U^i$;*

  the hybrid input vector is denoted by as $w^i = \begin{bmatrix} v^i \\ u^i \end{bmatrix} \in \mathcal{U}^i$;
- *an output space $\mathcal{Y}^i \subseteq \mathbb{R}^{q,i}$ on which $y^i(t)$ is defined;*
- *a time-dependent invariant set $I_{z^i}^i(t) \subseteq X^i$ for any discrete state $z^i \in Z^i$;*

- *a discrete state transition function* $f_d^i: T_k \times Z^i \times V^i \times X^i \rightarrow Z^i$;
- *a time-dependent guard set* $G^i_{z^i,\bar{z}^i}(t) \subseteq X^i$ *for any transition between a pair* $z^i, \bar{z}^i \in Z^i$ *of discrete states;*
- *a continuous state jump function* $j_c^i: T_k \times X^i \times Z^i \times Z^i \rightarrow X^i$;
- *the continuous state dynamics* $f_c^i: T \times X^i \times U^i \times Z^i \rightarrow X^i$, *defining the right-hand side of a set of ordinary differential equations for the continuous state* $x^i(t)$;
- *an output function* $g^i: T \times X^i \times Z^i \rightarrow Y^i$.

Definition 2. *Semantics of the continuous-time plant of a CPS subsystem*
For the model according to Def. 1, assume that:

- *the time domains* T *and* T_k *are bounded with the same limits* t_0 *and* t_f,
- *an initial state* $s_0^i = \begin{bmatrix} z_0^i \\ x^i(t_0) \end{bmatrix}$, $z_0^i \in Z^i$, $x^i(t_0) \in X^i$ *is given,*
- *a discrete input sequence* $\phi_v^i = \{v_0^i, ..., v_k^i, ...\}$ *with* $v_k^i = v^i(t_k) \in V^i$, $t_k \in T_k$ *and a continuous input trajectory* $u^i(t) \in U^i$, $t \in T$ *is specified.*

Then, the trajectories $x^i(t)$, $y^i(t)$ *for* $t \in T$ *and* $z^i(t_k)$ *for* $t_k \in T_k$ *are called an 'admissible run', iff the following applies:*

1. *continuous state progress for* $t \in [t_k, t_{k+1}[$ *by solution of* $\dot{x}^i(t) = f_c^i(t, x^i(t), u^i(t), z_k^i)$ *starting from* $x(t_k)$*:*
$$x^i(t_{k+1}^-) := x^i(t_k) + \int_{t_k}^{t_{k+1}} f_c^i(\tau)\,\mathrm{d}\tau$$
where $x^i(\tau) \in I^i_{z^i(t_k)}(t)$ *must apply for all* $\tau \in [t_k, t_{k+1}[$ *and* $x^i(t_{k+1}^-)$ *denotes the left-hand time limit* $\lim_{\epsilon \to 0} x^i(t_{k+1} - \epsilon)$;
2. *discrete state transition: if* $x(t_{k+1}^-) \in G^i_{z^i(t_k), z^i(t_{k+1})}(t_{k+1}^-)$ *applies at time* t_{k+1}, *the hybrid state* $s^i(t_{k+1}) = (z^i(t_{k+1}), x^i(t_{k+1}))^T$ *follows from:*
   - $z^i(t_{k+1}) := f_d^i(t_{k+1}, z_k^i, v_{k+1}^i, x^i(t_{k+1}^-))$
   - $x^i(t_{k+1}) := j_c^i(t_{k+1}, x^i(t_{k+1}^-), z_k^i, z_{k+1}^i) \in I^i_{z^i(t_{k+1})}(t_{k+1})$
3. *system output:* $y^i(t) = g^i(t, x^i(t), z_k^i)$ *for* $t \in [t_k, t_{k+1}]$.

The input and output vectors in this model definition can be related to a graph $G(t_k)$, as introduced above. The graph specifies which components the vectors $w^i(t)$ and $y^i(t)$

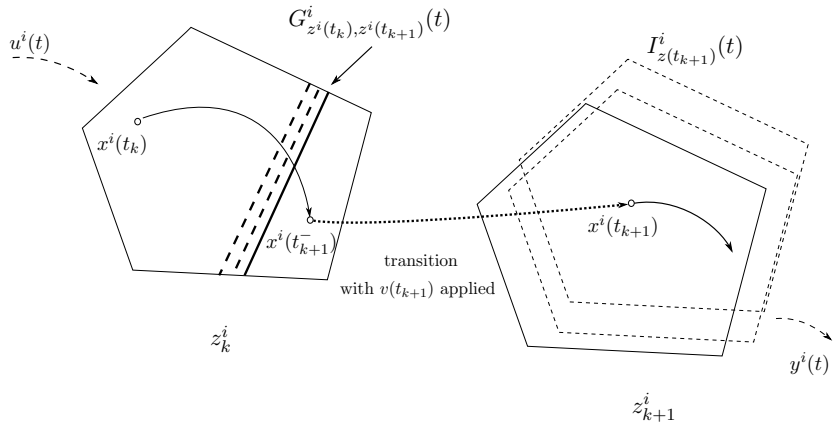

Figure 2: Admissible Run of a continuous time CPS.

include within the time interval $[t_k, t_{k+1}]$. For simplicity of exposition, this functionality is not explicitely formulated in Def. 1. Note that the discrete input $v^i(t_k)$ selects the discrete state which is reached upon a transition. Figure 2 shows exemplarily the evolution of the subsystem around one transition.

As a frequently-used step in control design (for instance when resorting to model predictive control), the approximation of a continuous-time model by a discrete-time substitute is important. Thus, we next define a variant of the CPS, in which all time-depending components are defined on a common discrete time-domain. Procedures for approximating the ordinary differential equations in Def. 1 by difference equations for a given (fixed or variable) sampling time are ample.

Definition 3. *Discrete-time plant of a CPS subsystem*
The discete-time model consists of the following elements:

- *the discrete time-domain* $T_k = \{t_0, t_1, t_2, ...\}$ *with* $t_k \in \mathbb{R}^{\geq 0}$*;*
- *the hybrid state space* $\mathcal{X}^i := Z^i \times X^i$ *as in Def. 1, but* $s^i = \begin{bmatrix} z^i \\ x^i \end{bmatrix}$ *defined only for the time in* T_k*;*
- *the hybrid input space* $\mathcal{U}^i := V^i \times U^i$ *also as in Def. 1, but* $w^i = \begin{bmatrix} v^i \\ u^i \end{bmatrix} \in \mathcal{U}^i$ *defined on* T_k*;*
- *an output space* $\mathcal{Y}^i \subseteq \mathbb{R}^{q,i}$ *on which* $y^i(t_k)$ *is defined;*
- *a time-dependent invariant set* $I^i_{z^i}(t_k) \subseteq X^i$ *for all states* $z^i \in Z^i$*;*
- *a discrete state transition function* $f^i_d : T_k \times Z^i \times V^i \times X^i \rightarrow Z^i$ *as in Def. 1;*

- *a time-dependent guard set* $G^i_{z^i,\tilde{z}^i}(t_k) \subseteq X^i$ *for any transition between two discrete states;*
- *a continuous state jump function* $j^i_c : T_k \times X^i \times Z^i \times Z^i \rightarrow X^i$*;*
- *the continuous state dynamics* $f^i_c : T_k \times X^i \times U^i \times Z^i \rightarrow X^i$*, specifying the right-hand side of a set of difference equations for the continuous state* $x^i(t_k)$*;*
- *the output function* $g^i : T_k \times X^i \times Z^i \rightarrow Y^i$*.*

The dynamic evolution of this model is:

Definition 4. *Semantics of a discrete-time plant of a CPS subsystem*
Given the system according to Def. 3 and:

- *a bounded domain* $T_k = \{t_0, \ldots, t_f\}$*,*
- *an initial state* $s^i(t_0) = \begin{bmatrix} z^i_0 \\ x^i(t_0) \end{bmatrix}$, $z^i_0 \in Z^i$, $x^i(t_0) \in X^i$*,*
- *a input sequences* $\phi^i_v = \{v^i_0, ..., v^i_k, ...\}$ *with* $v^i_k = v^i(t_k) \in V^i$, $t_k \in T_k$ *as well as* $\phi^i_u = \{u^i_0, ..., u^i_k, ...\}$ *with* $u^i(t_k) \in U^i$*.*

The discrete-time trajectories $\phi^i_x = \{x^i_0, ..., x^i_k, ...\}$ *with* $x^i_k = x^i(t_k)$*,* $\phi^i_z = \{z^i_0, ..., z^i_k, ...\}$ *with* $z^i_k = z^i(t_k)$*, and* $\phi^i_y = \{y^i_0, ..., y^i_k, ...\}$ *with* $y^i_k = y^i(t_k)$ *determine an 'admissible run', iff for any* $t_k \in T_k$ *applies:*

1. *continuous state progress:* $\tilde{x}^i_{k+1} = f^i_c(k, x^i_k, u^i_k, z^i_k)$ *such that* $\tilde{x}^i_{k+1} \in I^i_{z^i}(t_k)$*;*
2. *discrete state transition: if* $\tilde{x}(t_{k+1}) \in G^i_{z^i(t_k),z^i(t_{k+1})}(t_k)$ *applies at time* t_{k+1}*, the hybrid state* $s^i(t_{k+1}) = (z^i(t_{k+1}), x^i(t_{k+1}))^T$ *follows from:*
   - $z^i(t_{k+1}) := f^i_d(t_{k+1}, z^i_k, v^i_{k+1}, \tilde{x}^i(t_{k+1}))$*,*
   - $x^i(t_{k+1}) := j^i_c(t_{k+1}, \tilde{x}^i(t_{k+1}), z^i_k, z^i_{k+1}) \in I^i_{z^i(t_{k+1})}(t_{k+1})$*;*
3. *system output:* $y^i(t_k) = g^i(t_k, x^i(t_k), z^i_k)$*.*

The selection of components of $u^i(t_k)$ and $y^i(t_k)$ by a time-depending graph to model time-dependent subsystem interaction applies likewise as commmented for the continuous-time version of the CPS model.

2.2 Realization in SpaceEx

The model definitions provided before establish a quite general class of networked, time-varying, and hybrid models. The specific models of the case studies used in UncoVerCPS determine (and the methods developed in the project require) different sub-classes of this general model. In order to illustrate the definition for a simple example, and in particular to sketch how one model instance can be represented in the main verification tool of the project (SpaceEx, see [20]), a particular version of a bouncing ball model is briefly described in this subsection. SpaceEx, which was reported on in the earlier project deliverable D5.1, is an efficient and scalale tool to compute reachable sets for monolithic hybrid systems. Coupling to other subsystems can be considered in terms of disturbances of the subsystem dynamics.

We first consider a case of just one subsystem which is not affected by other, coupled subsystems. A bouncing ball obviously has hybrid dynamics, since a bouncing event resets the velocity ($v(t)$) of the ball discontinuously, while the height over ground ($h(t)$) changes continuously over time. Let the ball ($i = 1$) be modeled by a continuous state vector $x^1(t) := (h^1(t), v^1(t))^{\mathrm{T}}$, with uncertain initialization of the height $h^1(0) \in [10, 10.2]$ and $v^1(0) = 0$. Let the flight phase be modeled by one discrete state (z_1^1) with the normalized continuous dynamics:

$$\dot{h}^1(t) = v^1(t),\ \dot{v}^1(t) = -1,$$

and an invariant $I_{z^1}^1 = [0, 2] \times [-10, 10]$. The discrete state has a self-loop transition (modeling the bouncing) which is bound to the guard set $G_{z_1,z_1}^1 = [0, 0] \times [-10, 0]$ and to which the following jump is associated:

$$h^1(t_k) := h^1(t_k^-),\ v^1(t_k) := -0.75 \cdot v^1(t_k^{-1}).$$

This model can be straighforwardly represented in SpaceEx, and the left part of Fig. 3 shows the reachable set computed by the tool for the named configuration.

Next, assume that the ball is affected by an input $w^1(t) \in [-0.05, 0.05]$, interpreted as a disturbance for wich the bounds are known. For this case, the continuous dynamics is extended to:

$$\dot{h}^1(t) = v^1(t),\ \dot{v}^1(t) = -1 + w^1(t), \tag{1}$$

and the corresponding reachable set for the height as obtained by SpaceEx is shown in the right part of Fig. 3.

Now, in order to consider a model with time-varying components, assume a variant of the model with time-varying guard set. As sketched in Fig. 4, this scenario may be obtained

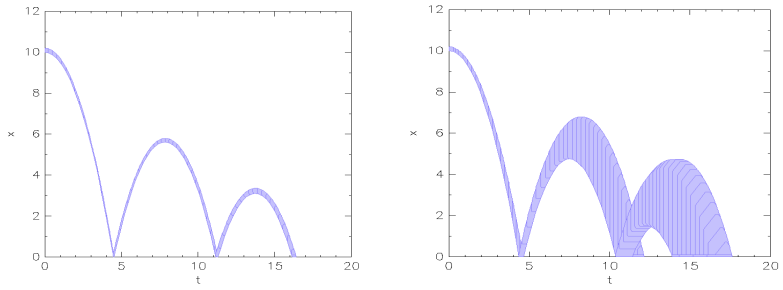

Figure 3: Reachable set of the height $h^1(t)$ of the bouncing ball over time for the case without disturbance (left) and with disturbance (right).

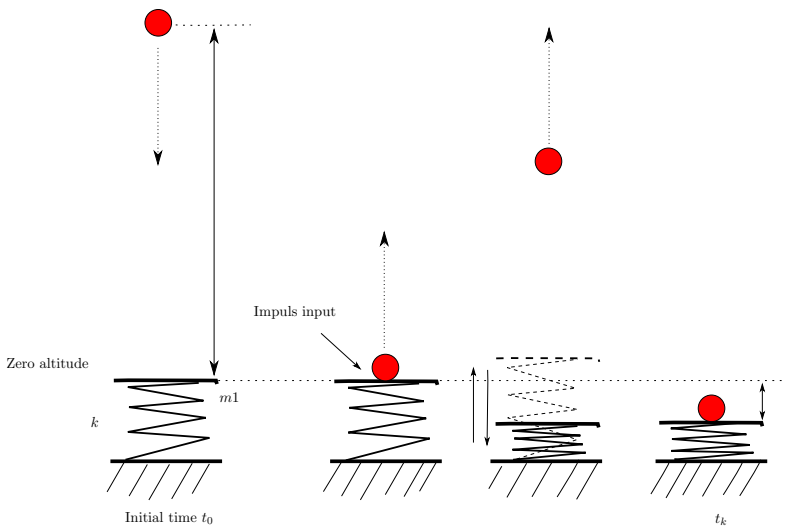

Figure 4: Ball bouncing on a ground with varying height over time.

when the ball hits a plate fixed on top of a spring. This system is naturally modeled by two subsystems, one representing the ball and one the plate with spring. When bouncing, the ball imposes an impulse on the plate, leading to its oscillation. On the other hand, the plate height determines a time-varying guard for the self-loop transition in the subsystem modeling the ball. For the case that the jump function associated to this transition is specified as $v^1(t_k) := -0.75 * v^1(t_k^-)$, the following figures 5 and 6 show the evolution of the height of the ball and the plate.

2.3 Conclusion

The type of model introduced in Sec. 2.1 combines hybrid dynamics with a modular and networked system structure, and time-variance of several system components. Thus, it includes the model characteristics of the case studies of UnCoVerCPS, and can be seen as an

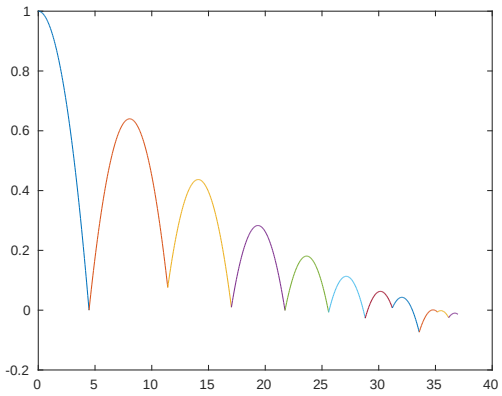

Figure 5: Height $h^1(t)$ of the ball over time.

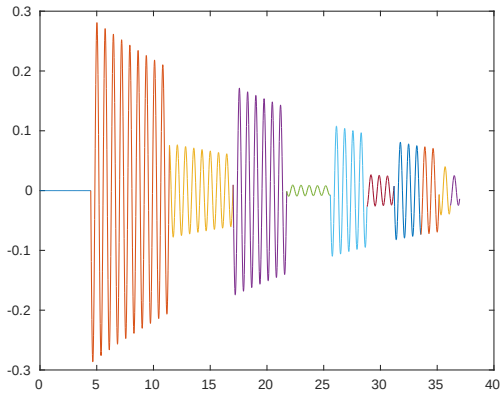

Figure 6: Height $h^2(t)$ of the board over time.

envelope of the types of models underlying the different methods developed in the project. With respect to the techniques of model abstraction and approximation to be reported in the upcoming sections, the following specific instances of the general model are considered:

- Sec. 3 refers to a concrete model with continuous-time continuous dynamics which can be considered either as the composition of a set of plant subsystems ($i \in \{1, \ldots, N\}$), or as a single subsystem without coupling, but subject to inputs u^i; the same section furthermore considers abstracted models (the *conformance monitors*) with hybrid hybrid time-invariant dynamics, where differential equations are replaced by differential inclusions;

- Sec. 4.2 uses again monolithic systems with hybrid dynamics specified as continuous-time switched affine systems, in which switching is triggered endogenously in time-invariant manner;

- Sec. 4.3 is focused on hybrid dynamics given as mixed-logical dynamic systems (i.e. an alternative representation of piecewise affine systems) formulated in discrete-time;

- Sec. 5.1 considers networked CPS, in which the subsystem plant dynamics is modeled by nonlinear differential equations and is subject to time-varying state constraints communicated by interacting subsystems;

- finally, Sec. 5.2 also starts from nonlinear continuous-time dynamics subject to varying constraints imposed by other subsystems, and it derives time-varying discrete-time linearizations for different modes.

These model instances as well as the abstracted / approximated substitutes are defined in more detail in the respective sections and illustrated for examples.

3 Conformance Monitors for Model-To-Model Conformance

3.1 Introduction

In this section, we present results from Task 1.2: "Abstraction and refinement of hybrid system models." Here, the goal is to relate two models of the same system on different abstraction levels: we will in the following refer to an *abstract model* and a *concrete model*. In an industrial development process, abstract models are first built to verify high-level properties and these models are then concretized step by step into implementations.

For example, abstract models of controllers are typically continuous-time and continuous-value models, abstracting away from the execution platform. Also, for a first analysis of a controller, certain aspects of the physical or software environment of the system under design may be neglected. In practice, these abstract models are usually simulated, and also sometimes used for systematic controller design. Since, in the end, an actual implementation of the controller is needed, it is highly desirable to be able to transfer properties shown on the abstract system to the concrete one.

Within Task 1.2, we looked at this problem from a formal standpoint, doing *conformance verification*. In particular, we deal with the question how conformance between models on different abstraction levels can be shown *formally*, while, in contrast, the preceding Deliverable 5.2 used *conformance testing* only for the two use cases on automated driving and wind turbine control. Since, for these use cases, we do not have models on different abstraction levels at our disposal which are amenable to formal methods, the methods presented in the following were validated on a Bosch example: the experimental *electro-mechanical brake (EMB) system*, which is described in detail in [62, 24]. For this system, there are two controller abstraction levels available: an abstract level with a time-continuous idealized controller and a concrete level with a fixed-rate discrete-time controller.

As a tool for the formal conformance verification, we employed SpaceEx. The basic idea of the approach is to model the conformance relation as a *conformance monitor* which runs in parallel in SpaceEx with the concrete model of the system. The monitor checks if the generated traces are also permitted under the abstract system, and goes to an error state if this is not the case. The verification task then becomes to show that these error states are unreachable, proving conformance between the models. These conformance monitors complement the requirement monitors that were defined in Deliverable 1.1: instead of monitoring the fulfillment of a temporal property, they monitor the fulfillment of a conformance property to another model. In order to show conformance, we in fact show that the abstract system

can trace every step taken by the concrete system.

3.2 Related Work

The work provided in this section is in part also inspired by the ModelPlex approach by Mitsch et al. [49]. In contrast to ModelPlex, which concentrates on conformance monitoring for *online monitoring*, our approach is presented here for the *offline* case, i.e., the monitor is run in SpaceEx, but not in the actual control software. Therefore, we can not only check the currently observed concrete trajectory, but instead all possible trajectories at once, via reachability analysis.

A common approach to related formal models of the same system in a formal manner is based on the notion of *simulation*, for example as described in [63]. A generalization of this is the notion of *approximate simulation*, as proposed in [29, 30], where simulation relations are only required to hold with some tolerance parameters. This is particularly useful if safety properties are to be shown. Since, in this section, we are not only interested in safety properties in the context of this deliverable, we choose a different approach, showing trace inclusion instead of approximate simulation. For the same reason, we do not employ our own concept of *reachset conformance* [59], as it is also tailored to showing safety properties of systems.

A detailed survey of different conformance notions and conformance checking approaches will be included in Deliverable 1.3 "Conformance Testing in the Development Process" by the conclusion of UnCoVerCPS.

3.3 Model-To-Model Conformance

Let C be the concrete (continuous) model, given by an initial set $X_C(0)$ and a differential equation:

$$\dot{x}(t) = f(x(t), u(t)), \tag{2}$$

and let A be the abstract (continuous) model, given by an initial set $X_A(0)$ and a differential inclusion:

$$\dot{x}(t) \in g(x(t), u(t)) \oplus \mathcal{V}(t). \tag{3}$$

Here, $u(t)$ is the time-dependent input to both models, assumed to be taken from some set U of possible input signals, i.e., $\forall t : u(t) \in \mathcal{U}$. The time-variant set $\mathcal{V}(t)$ is added to the right hand side of a differential equation, turning it into a differential inclusion with several possible state derivatives per input-state pair. This non-determinism will be used to enclose

the behaviors of the concrete model in the set of behaviors of the abstract model. Here, the operator $\oplus$ represents the *Minkowski sum*:

$$x \oplus \mathcal{V} := \{x + v \mid v \in \mathcal{V}\}. \tag{4}$$

Note that we are restricting ourselves to a time-dependent additive uncertainty $\mathcal{V}(t)$ here, because this is what is currently supported by the tool SpaceEx. In the general case, we could write $\dot{x}(t) \in G(x(t), u(t))$ as a general differential inclusion. To simplify the discussion, we first restrict ourselves to purely continuous models. In the following discussions, we will then also look at the case when the concrete system has a discrete-time controller instead. The example application in Sec. 3.4 then also has a discrete-time controller, turning the concrete system into a sampled-data control loop.

Given two models above, the task is to show that C is trace conformant to A. We define trace conformance as follows:

Definition 5 (Trace conformance)**.** *Let $Trace_C(u)$ be the set of traces generated by C under input signal u, and let $Trace_A(u)$ be the set of traces generated by A under input signal u. C is* trace conformant *to A, if for all $u \in \mathcal{U}$ we have $Trace_C(u) \subseteq Trace_A(u)$.*

This means that all behaviors of C can also be observed in A, guaranteeing the transference of linear temporal logic (LTL, [56]) properties from A to C, i.e. all LTL properties that hold on all traces of A can also be shown on to hold on all C. In general, LTL covers the vast majority of functional requirements that are relevant in industrial practice. This enables us to use a (possibly simpler or more general) model A for verifying properties of the system. These properties are then guaranteed to also hold for C.

3.3.1 Conformance Relation as a Hybrid Automaton

For continuous systems, trace conformance can be proven by:

1. showing that $X_C(0) \subseteq X_A(0)$ (Property 1), and
2. showing that the condition $f(x(t), u(t)) \in g(x(t), u(t)) \oplus \mathcal{V}(\mathrm{t})$ holds for all times t and all pairs $(x(t), u(t))$, $x(t) \in Reach_t(C, u)$ (Property 2),

where $Reach_t(C, u)$ is the set containing all reachable states x of C at time t, under the input signal u. The conditions above can be viewed as a special kind of simulation relation between the two systems: all trajectory segments of C can be generated by A as well. For the purpose of this deliverable we assume that the state variables and inputs of A and C are identical – this restriction can also be lifted by providing a mapping between the input

and state spaces of A and C. The first property is a set inclusion and can be easily checked. The second property can be checked by successively checking the condition on the computed reachable sets $Reach_I(C, u)$ for time intervals $I = [t_i, t_{i+1}]$. This check can also be translated to a monitor automaton consisting of two discrete states:

- one state "conf" with no invariant
- one fault state "notconf" with no invariant

and one transition from "conf" to "notconf" with guard $f(x, u) \notin g(x, u) \oplus \mathcal{V}(t)$. This monitor automaton is meant to be composed in parallel with C, checking the conformance of C and A via reachable set computation. In this case, the absence of traces leading to "notconf" would prove conformance. This is sound, but due to the overapproximative reachable set computation not complete.

3.3.2 Example

In the following, we give a simple example to illustrate the approach. Its application to a more complex system is then described in Section 3.4.

Consider the following differential equations for the concrete model C

$$\dot{x}(t) = f(x(t)) = a_C \cdot x(t) \tag{5}$$

with $a_C = -1$, and the abstract model A

$$\dot{x}(t) \in g(x(t)) \oplus \mathcal{V} = a_A \cdot x(t) \oplus [\tilde{v}_{min}, \tilde{v}_{max}] \tag{6}$$

with $a_A = -0.99$. The analytical solution of C is $x(t) = x_0 \cdot e^{a_C t}$. Furthermore we assume that, in both models, x is limited by $-1 \leq x \leq 1$. Thus, the abstract model $\dot{x}(t) = a_A \cdot x(t) \oplus \mathcal{V}(\mathrm{t})$ is trace conformant to the concrete model C if the uncertainty $\mathcal{V}(t) = [\tilde{v}_{min}, \tilde{v}_{max}] = [-0.01, 0.01]$ for all t. Figure 7 shows the conformance monitor as a hybrid automaton in SpaceEx. The two transitions from state "conf" to state "notconf" capture the case that $f(x) - g(x) \notin \mathcal{V}(t)$ (negation of Property 2). For numerical reasons, the theoretical limits $\tilde{v}_{min}$ and $\tilde{v}_{max}$ of the conformance monitor have to be extended by the parameter ϵ, arriving at $v_{min} = \tilde{v}_{min} - \epsilon$ and $v_{max} = \tilde{v}_{max} + \epsilon$.

By choosing $\epsilon = 10^{-9}$ and constraining the initial state to $-1 \leq x(0) \leq 1$ for our example, the state "notconf" cannot be reached. Figure 8 shows the reachable set (initial state $-1 \leq x(0) \leq 1$) of the concrete model C and the abstract model $\dot{x}(t) = a_A \cdot x(t) \oplus \mathcal{V}(t)$ for a time horizon of 10s.

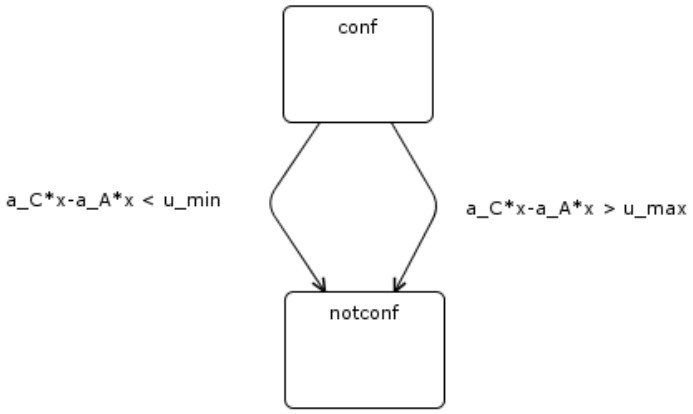

Figure 7: Conformance monitor as a hybrid automaton in SpaceEx of the example

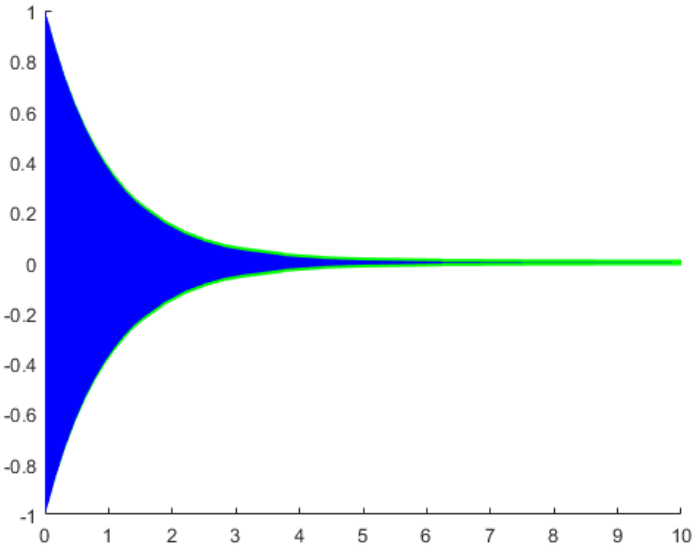

Figure 8: Overlay of the reachable set of the concrete model C (blue) and the abstract model A (green)

In a next step, we extend the conformance monitor by a time-discretization of the concrete model. This is supposed to reflect the fact that the physical quantities of the concrete system can only be observed at discrete points in time, with a certain fixed rate. As a consequence, we also only execute the monitor when new data is available. To this end, we use an estimated derivative $\dot{x}$ based on the observations, by using linear interpolation. Figure 9 shows the discretized version of the conformance monitor for our example. The transition to the state *conf* updates the variable $x_{previous}$ with respect to the synchronization label *tic*, thus reflecting the discrete update of values in concrete implementations of a system.

Here, a parallel automaton generates the synchronization label *tic* with a strict period *sampling* (not shown in Figure 9). The transitions from state "conf" to state "notconf"

capture the negation of Property 2. In contrast to the continuous conformance monitor, the guards of both transitions contain the interpolated derivative with respect to the period *sampling*.

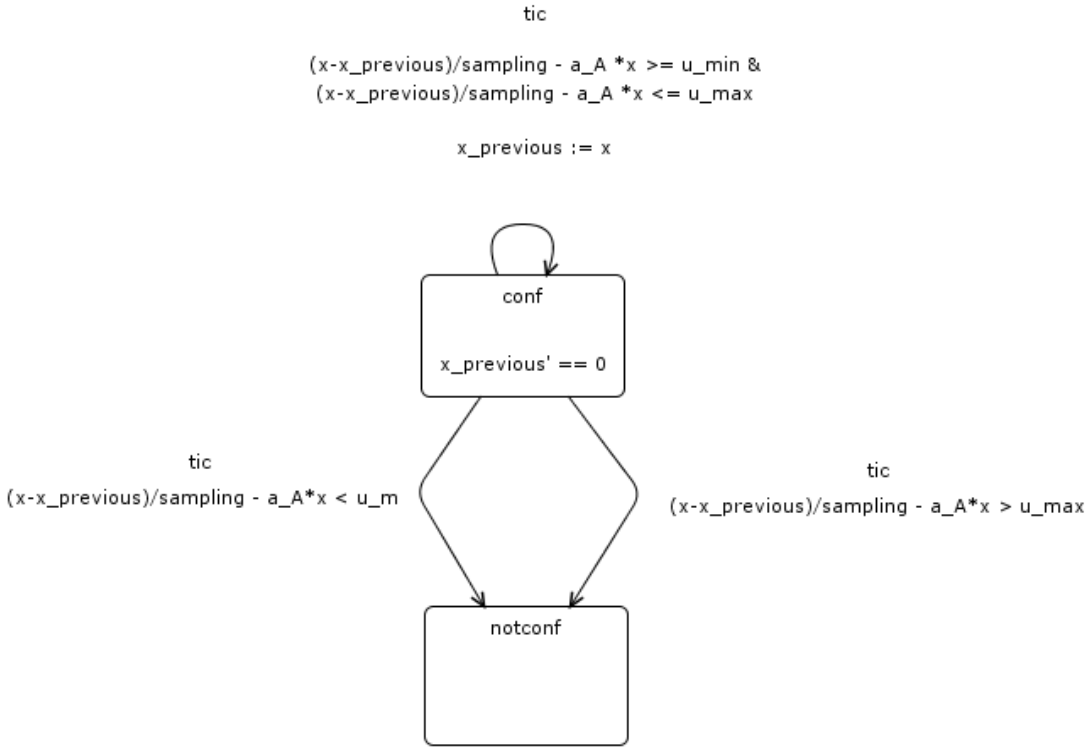

Figure 9: Time discretized version of the example's conformance monitor

3.4 Conformance Monitors for the Experimental Electro-Mechanical Brake Example

The system under consideration is an experimental *electro-mechanical brake (EMB)* with a closed-loop controller, which has been described in detail in [62]. We use a simplified model of physics and controller, which has also been used in [24]. This model consists of a brake system with a brake caliper, which is brought into contact with a brake disk via an electrical engine. This engine is controlled by applying an appropriate current to the engine. The goal here is to study the effects of time discretization on controller performance. We do this by deriving an abstract model A that does not explicitly contain a discretized controller, but instead over-approximates possible discretization effects by an uncertainty $\mathcal{V}(t)$, and by relating it to a concrete model C containing a discrete-time controller.

The challenges faced are twofold: a) deriving an abstract model A for which the system requirement still holds, and b) implementing a conformance monitor of the trace conformance relation and using it to prove that C is indeed trace conformant to A. As outlined above, this

proof is conducted by showing that non-conformant states are unreachable in C. Together, this implies that C also fulfills the requirement.

3.4.1 Plant Dynamics and Controller Model

The idealized plant model is as follows:

$$\dot{I}(t) = f(I(t), U(t)) = \frac{1}{L} \cdot \left(-R + \frac{K \cdot K}{d_{rot}} \right) \cdot I(t) + \frac{1}{L} \cdot U(t) \tag{7}$$

$$\dot{x}(t) = h(I(t)) = \frac{K}{i \cdot d_{rot}} \cdot I(t) \tag{8}$$

Here, x is the position of the brake caliper, I is the electric current in the electrical engine moving the caliper, and U is the voltage that is set by the controller. All other symbols represent physical constants which concrete values are summarized in Table 1 (see [62] for a detailed description). The controller we use is a simple P-controller with

$$U(t) = k_P(x_0 - x(t)), \tag{9}$$

where k_P is the controller parameter, and the target position x_0 is an input to the system. x_0 is the desired position of the brake caliper, representing the point where contact is made with the brake disk. After a brake request has been registered, controller (9) is supposed to control the position, smoothly steering it towards the position were brake disk and caliper make contact. As soon we have reached a certain distance to this point, a second controller (not modeled here) would take over, slowly moving the caliper onto the disk and subsequently controlling the brake force.

The requirement we focus on is the following condition: $x(t)$ should reach $x_0 = 0.05dm$ within $20ms$ with a precision of 4% (i.e., $x = 0.048dm$). This means that our controller needs at most $20ms$ until it can hand over to the second controller, with a position within 4% of x_0.

Both the concrete model C and the abstract model A are derived from the dynamics above. Note that both plant models and the continuous controller are hybrid systems in the sense of Definition 2 of Section 2.1. Since we use SpaceEx, which only allows for continuous-time modeling, the discrete-time controller used in C is also modeled as a continuous-time hybrid system. It can also be expressed as a discrete-time system according to Definition 3 of Section 2.1. The concrete closed-loop model C is then a sampled-data system: the controller equation is computed every Δ seconds, whereas the dynamics for x and I are still time-continuous. As can be seen in the following, this leads to a faster response as in the idealized model above. This is due to the strong controller response in the beginning, which is held for a longer time due to the sample-hold controller.

Parameter	Value
sampling conformance monitor	$0.2ms$
sampling controller	$1.0ms$
x_0	$0.05dm$
k_P	75
R	0.5
L	10^{-3}
K	0.02
d_{rot}	0.1
i	113.1167

Table 1: Overview of the concrete parameter values used in the EMB system

Formally, the concrete model is:

$$\dot{I}(t) = f(I(t), U(t)) = \frac{1}{L} \cdot \left(-R + \frac{K \cdot K}{d_{rot}}\right) \cdot I(t) + \frac{1}{L} \cdot U(t) \tag{10}$$

$$\dot{x}(t) = h(I(t)) = \frac{K}{i \cdot d_{rot}} \cdot I(t) \tag{11}$$

$$U(t) = k_P(x_0 - x(\tilde{t})), \tag{12}$$

where $\tilde{t} = \max\{\Delta n \mid n \in \mathbb{N}, \Delta n <= t\}$ is the time of the last discrete sampling and with sampling time Δ.

Figures 10 and 11 show simulation runs of the idealized model without uncertainties versus the concrete model C. Figure 10 gives the caliper position over time, while Figure 11 shows the electric current. As can be seen here, the sampled-data controller in C results in faster convergence of $x(t)$ to the target position $x_0 = 0.05$, while the signal $I(t)$ is close to being piecewise constant, with fast changes at the sampling points and near constant behavior in between.

The abstract model is derived from the idealized model by including additive, time-variant uncertainties to the differential equations. The model then becomes:

$$\dot{I} = f(I, U) = \frac{1}{L} \cdot \left(-R + \frac{K \cdot K}{d_{rot}}\right) \cdot I + \frac{1}{L} \cdot U \oplus \mathcal{V}_I(t) \tag{13}$$

$$\dot{x} = h(I) = \frac{K}{i \cdot d_{rot}} \cdot I \oplus \mathcal{V}_x(t) \tag{14}$$

$$U(t) = k_P(x_0 - x(t)) \tag{15}$$

The intervals $\mathcal{V}_I(t)$ and $\mathcal{V}_x(t)$ are given in Table 2. Note that the uncertainties are chosen to become progressively smaller over time, as the effect of the time discretization error on

the derivatives decreases as the control error gets smaller. This means that the uncertainty actually has the characteristic of a multiplicative error – as SpaceEx can only deal with additive errors at present, we model the uncertainty as a time dependent, decreasing additive error instead. The SpaceEx plant model for the system A is given in Figure 12.

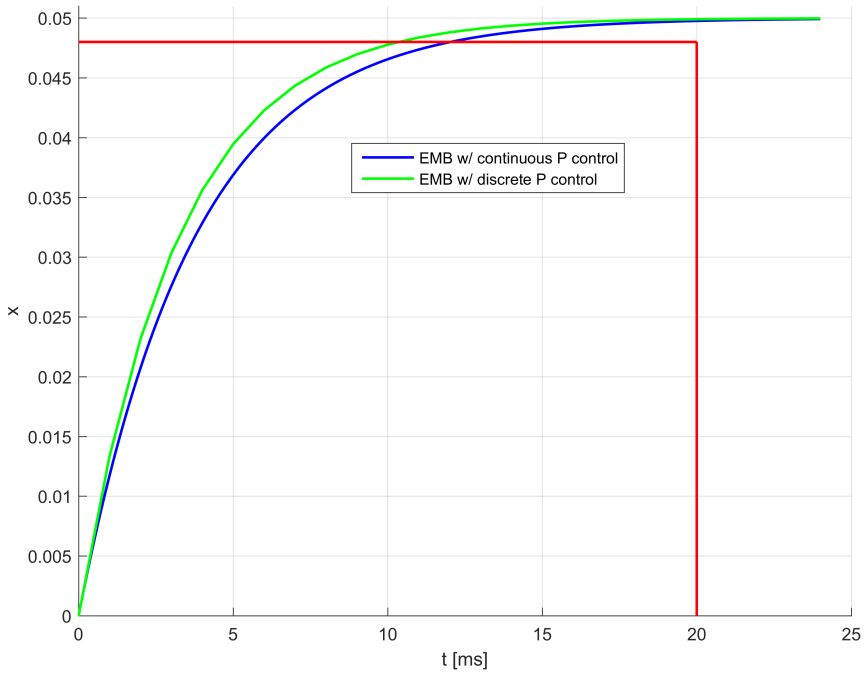

Figure 10: x-position over time. The red line indicates the parameters of the related requirement.

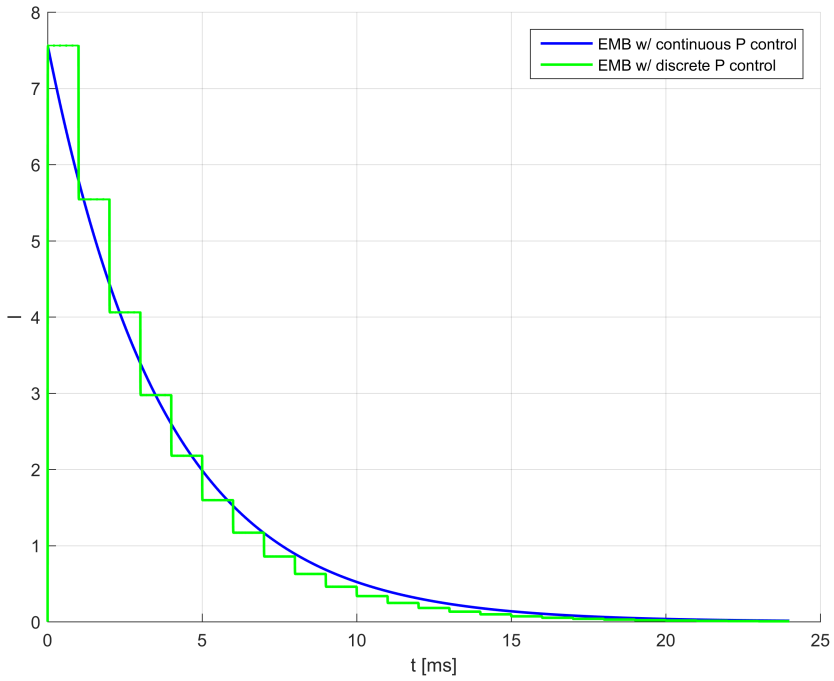

Figure 11: Current I over time.

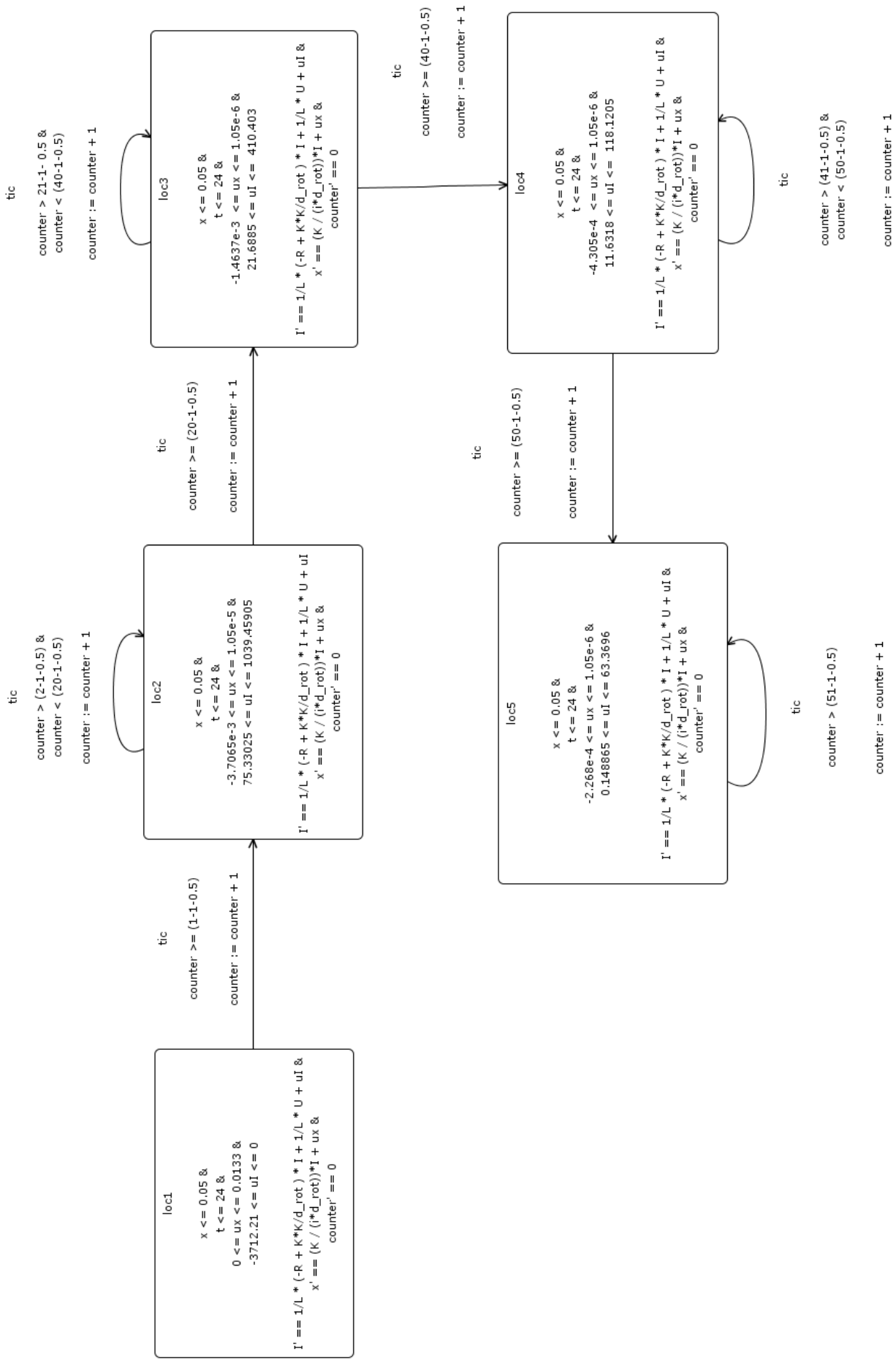

Figure 12: Conformant plant model of the EMB with switched uncertainties.

Time interval for t in ms	Interval $\mathcal{V}_x(t)$	Interval $\mathcal{V}_I(t)$
$[0, 0.2]$	$[0, 1.33 \cdot 10^{-2}]$	$[-3712.21, 0]$
$(0.2, 4.0]$	$[-3.7065 \cdot 10^{-3}, 1.05 \cdot 10^{-5}]$	$[75.33025, 1039.45905]$
$(4.0, 8.0]$	$[-1.4637 \cdot 10^{-3}, 1.05 \cdot 10^{-6}]$	$[21.6885, 410.403]$
$(8.0, 10]$	$[-4.305 \cdot 10^{-4}, 1.05 \cdot 10^{-6}]$	$[11.6318, 118.1208]$
$(10, \infty)$	$[-2.268 \cdot 10^{-4}, 1.05 \cdot 10^{-6}]$	$[0.148865, 63.3696]$

Table 2: Resulting uncertainty intervals of the EMB with continuous P control.

The interval bounds were determined by a combination of random search and interval bound minimization. First, we identified the range for the candidate interval bounds based on simulations. We then used random search combined with binary search to explore the Pareto front of possible uncertainties with successive SpaceEx calls, checking whether the "notconf" state of the conformance monitor (see Figure 13) was reachable. A call to SpaceEx using a small flowpipe tolerance took around $150s$ on an Intel Xeon Workstation, whereas for initial exploration higher tolerance are reasonable. The result is then a set of possible Pareto-optimal uncertainties for which "notconf" is not reachable. Note there is a Pareto front of uncertainties because the distribution of uncertainties between $\mathcal{V}_x(t)$ and $\mathcal{V}_I(t)$ is not unique – these are degrees of freedom in defining the abstract model A. Pareto-optimal here then means that there is no other possible abstract system with strictly smaller intervals that is also conformant. Which model on the Pareto front to choose is then dependent on the properties to be shown on abstract model A. We picked one point on the Pareto front, leading to the uncertainties given in Table 2. Computing the reachable set for the resulting model shown in Figure 12 took around $8500s$ with the result shown in Figure 14. Figure 14 shows the upper and lower bounds of the reachable set on the position $x(t)$ as black lines. What can be seen here is that under this abstraction, the requirement can not quite be shown ($x = 0.048$ is reached around $22.5ms$ instead of $20ms$). The reason for the slower dynamics on $x(t)$ that are also included in the reachset lies in the fact that the dynamics for $I(t)$ also need to be conservatively over-approximated. The over-approximation for $I(t)$ also contains uncertainties that can make the dynamics of $I(t)$ slower and consequently also the dynamics of $x(t)$. One solution to this may be an automatic exploration of the Pareto front, searching for an abstract model A for which the requirements is fulfilled. Due to the long computation times in SpaceEx we did not run a systematic analysis of this type. Failing this, either the requirement or the controller itself must be changed. The controller could be made more aggressive by increasing the constant k_P. The requirement could also be relaxed

by increasing the time bound or by increasing the tolerance band around x_0. This second changed would mean that the force controller that takes over once we are close to x_0 must be prepared to deal also with slightly smaller positions $x(t)$ upon being activated.

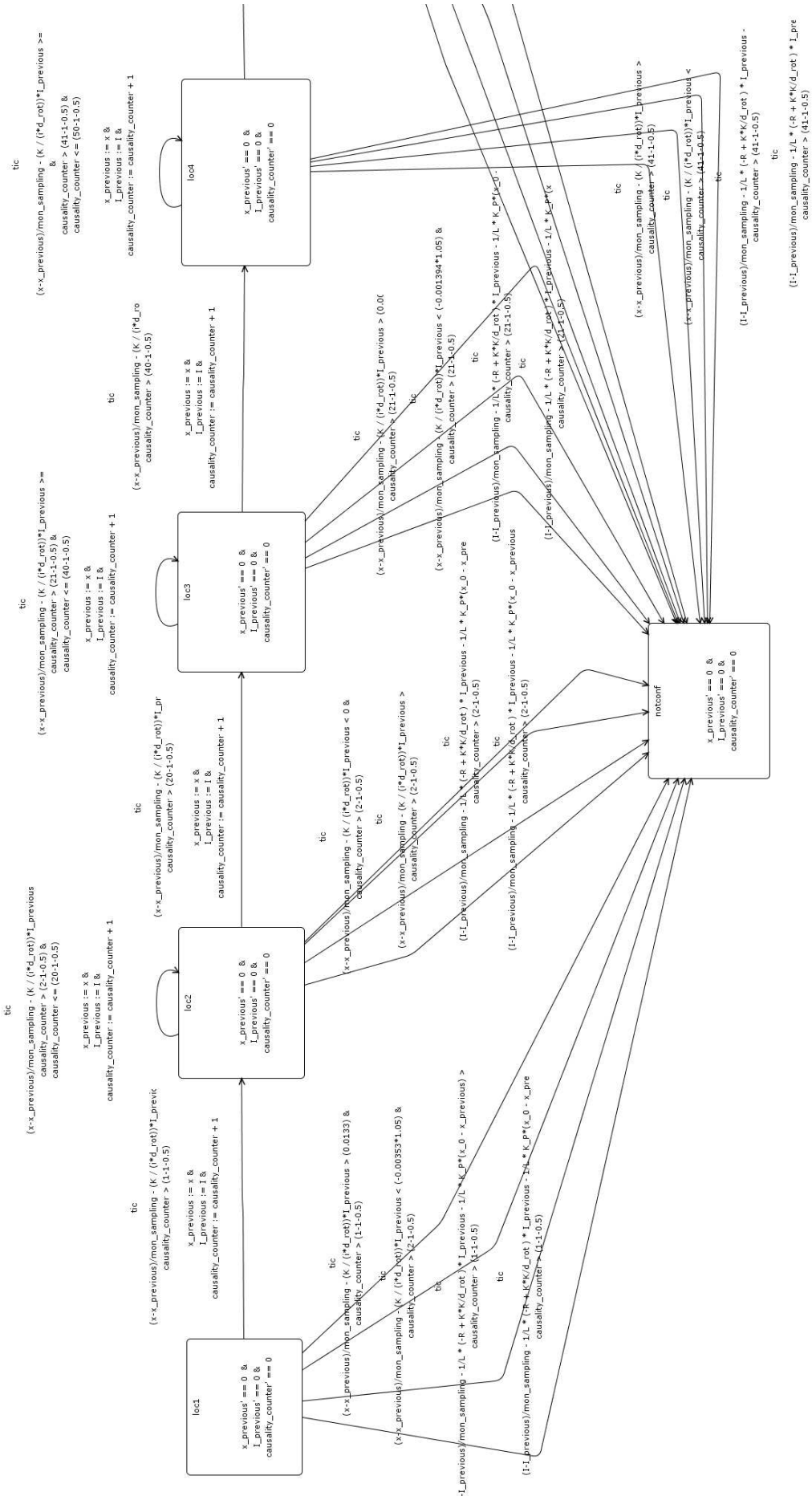

Figure 13: Excerpt of the switched conformance monitor for the EMB in SpaceEx.

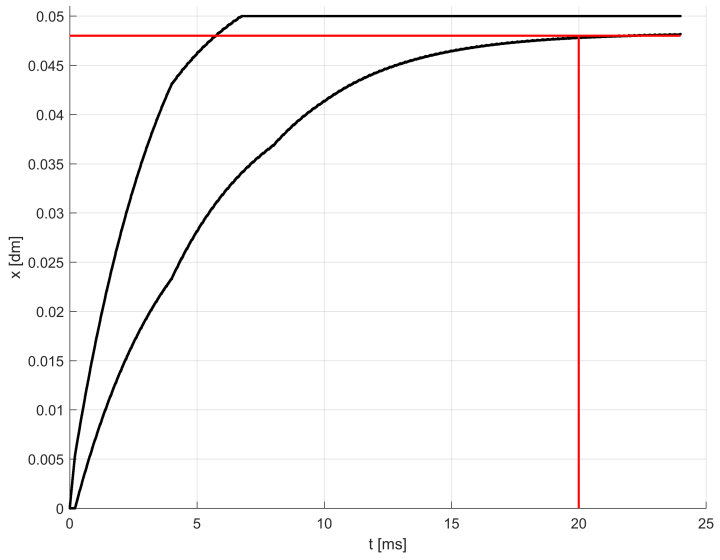

Figure 14: Upper and lower bound of the reachable set of the EMB with uncertainties (cf. Figure 12). The red line indicates the parameters of the related requirement.

3.5 Conclusions

In this work, we showed that SpaceEx can in principle be used to relate controller models on different abstraction levels, containing different implementation effects. By showing non-reachability of non-conformant states, conformance can be shown. As an extension to the monitor automata for temporal requirements, as proposed in Deliverable 1.1, it is now also possible to treat conformance to an abstract model as a requirement within SpaceEx. While this has proven a difficult task – meaningful abstractions for cyber-physical systems that are also formally stringent are hard to find – we succeeded in applying this method to an example exhibiting some characteristics of automotive control loops. In practice, the presented work can reduce the time to verify a variant of a system considerably since one only has to check conformance between the abstract and the concrete model of the new variant, rather than checking all requirements on the new variant again. For instance, a controller solutions can be instantiated with different sampling rates or embedded into different scheduling algorithms, without the need to re-verify all requirements if conformance to an abstract model is shown. On top of this, conformant abstract models of control loops are also useful for analyzing interactions between different controllers that for example have been designed independently by different departments, and where the complexity of the concrete models would be too difficult to handle.

Note that the work on conformance testing in Deliverable 5.2 does, in contrast, focus on

the testing aspect, when the concrete system can only be observed through measurements. Similar ideas about how to identify a suitable abstract model were applied there, replacing the formal analysis of SpaceEx by individual tests.

4 Model Reduction and Approximation for Verification

4.1 Introduction

This section addresses the design of an approximate model for a hybrid system (see e.g. [47, 38, 60, 31, 32, 57, 29]). The study of hybrid systems is typically challenging since they are characterized by intertwined continuous and discrete dynamics, [43]. Indeed, many problems that have been solved for purely discrete or purely continuous systems still lack an effective solution for hybrid systems. In particular, this is the case for the design of a reduced model.

We shall consider the problem of approximating a system so as to mimic its input/out put behavior. This is of interest when dealing with verification of properties that depend on the behavior of the system output. Verification of properties related to the system response, like, e.g., safety and reach/avoid properties, is typically addressed in the literature through reachability analysis methods in both the deterministic, [65, 28, 23, 40], and the stochastic, [1, 2], settings. These methods scale unfavorable with the dimension of the continuous state space component, with an exponential dependence if they do resort to gridding like [1, 2]. One can then conceive a two-step procedure where an approximate abstraction with a reduced order continuous state space component is built first, and then a numerical verification method is applied to this abstraction in place of the original system.

We do address model reduction in this section for the class of hybrid systems that are characterized by different affine dynamics, each one associated with a mode that depends on the value taken by the state of the system. This class of systems naturally arise as models in various application contexts and can also be adopted as approximate models for classical nonlinear systems, given that a smooth nonlinear function can be approximated with arbitrary accuracy by a piecewise linear function if an appropriately fine partition of its domain is taken, [61]. Various analysis and control problems have been studied for this class of systems, which are characterized by significant modeling capabilities, despite their simple description via affine equations and constraints, which typically simplifies analysis and design problems. Here, we shall focus on the recent developments within UnCoVerCPS on model reduction in both continuous and discrete time in Subsections 4.2 and 4.3, respectively. The presented work has appeared in [52] and [70].

4.2 Model Reduction of Switched Affine Systems for Probabilistic Verification

We focus on continuous-time Switched Affine (SA) systems with endogenous switching, and address the problem of obtaining a model that is simpler to analyze than the original system, and that is able to mimic its output behavior over a finite horizon $\mathcal{T}$. When the input signal of the system is stochastic, the notion of approximate simulation introduced in [38] for stochastic hybrid systems [45] can be used to quantify the model performance over the output realizations. A randomized approach for assessing the performance of a given abstracted model according to this notion was proposed in [57]. The approach also extends to model design. However, no constructive procedure is given on how to select and parameterize the model class. On the contrary, in this paper we provide a constructive procedure to build an approximate model of a SA system in the form of a reduced order Switched Linear (SL) system with appropriately defined state reset maps. The SA system is first rewritten as a SL one with state reset, and then Balanced Truncation (BT) [6] is adopted for reducing the order of the linear dynamics governing the evolution of the continuous state component in each mode. State reset maps are suitably redefined accounting for the mismatch in the continuous state vectors associated with different modes. A randomized method is also proposed to determine the order of the reduced linear dynamics in each mode, while accounting for the effect of discrete transitions and state resets on the hybrid system evolution. The overall methodology is extended to the case when a Dwell Time (DT) is present.

Note that BT is applied to switched linear systems in [55] which however deals with the case of externally induced switching. Our approach is inspired by [47] which uses BT for hybrid systems with linear dynamics and endogenous switching. The main advances with respect to [47] are: 1) the extension to the class of SA systems, 2) the introduction of novel state reset maps that provide better performance than the one adopted in [47], and of variants of these maps able to preserve continuity. Correspondingly, different initializations of the approximate model are derived based on the same logic underlying the reset maps definition, 3) the introduction of a randomized approach to select the order of the reduced linear dynamics in each mode, when the input is stochastic, and 4) the extension to the case of SA systems with DT. As a matter of fact, mode transitions and resets may strongly affect the system evolution. Indeed, the state reset map determines the new value of the continuous state after a discrete transition between modes has just occurred; while for a linear asymptotically stable system the contribution of the initial state becomes negligible in the long run, in a SA system this is generally not the case. One would in fact need to

guarantee that the time between discrete transitions is sufficiently large to make the zero-input response (ZIR) vanish, which cannot be guaranteed a-priori, unless a suitable DT triggering the discrete transitions is enforced.

The choice of the order of the approximate model should then account for the influence of the state reset map on the quality of the approximation. Hence it cannot be based only on the analysis of the Hankel Singular Values (HSVs) of the linear dynamics in each mode, as suggested in [47]. The proposed randomized approach serves this purpose, since it accounts for the hybrid evolution of the candidate approximate model including mode transitions and resets. The quality of the approximation is determined also by the domains triggering the mode transitions of the SA system. Notably, redesigning the domains is quite a complex issue, [27], and it is not addressed in this paper but left for further investigation.

The scope of this work does not include the problem of minimal realization. To the best of our knowledge, minimal realization theory has been mainly developed for linear and bilinear switched and hybrid systems with externally induced switching, while it is still an open problem for continuous-time hybrid systems with endogenous switching [53].

4.2.1 Switched affine systems modeling framework

A SA system is an instance of a hybrid system, whose dynamics are characterized through a discrete state component q_a (mode) taking values in $Q = \{1, 2, \ldots, m\}$ and a continuous component $\xi_a \in \Xi_a = \mathbb{R}^n$ evolving according to affine dynamics that depend on the value taken by q_a. The output $y_a \in Y_a = \mathbb{R}^p$ of the systems is an affine function of the state and of the input $u \in U = \mathbb{R}^m$ that depends on q_a as well. The continuous dynamics of a SA system within a given mode $q_a \in Q$ are given by

$$\mathcal{S}_a : \begin{cases} \dot{\xi}_a(t) = \mathcal{A}_{q_a}\xi_a(t) + \mathcal{B}_{q_a}u(t) + f_{q_a} \\ y_a(t) = \mathcal{C}_{q_a}\xi_a(t) + g_{q_a}. \end{cases} \tag{16}$$

Assumption 1. *For any $i \in Q$, matrix $\mathcal{A}_i$ is Hurwitz, $(\mathcal{A}_i, \mathcal{B}_i)$ is controllable, and $(\mathcal{A}_i, \mathcal{C}_i)$ is observable.* □

As for the discrete state evolution, a collection of polyhedra $\{Dom_{a,i} \subseteq Y_a \times U, i \in Q\}$ is given, which covers the whole set $Y_a \times U$, i.e., $\cup_{i\in Q} Dom_{a,i} = Y_a \times U$. $Dom_{a,i}$ is defined through r_i linear inequalities, i.e., $Dom_{a,i} = \{(y_a, u) \in Y_a \times U : G_i^{y_a} y_a + G_i^u u \leq G_i\}$, with $G_i^{y_a} \in \mathbb{R}^{r_i \times p}$, $G_i^u \in \mathbb{R}^{r_i \times m}$ and $G_i \in \mathbb{R}^{r_i}$.

Mode $i \in Q$ is active as long as (y_a, u) keeps evolving within $Dom_{a,i}$ and a transition to mode $j \neq i \in Q$ occurs as soon as (y_a, u) exits $Dom_{a,i}$ and enters into $Dom_{a,j}$ (endogenous switching).

Assumption 2. *The switched affine system* (16) *admits a unique solution from any initial state.* □

Note that the considered switched system can be rephrased in the hybrid automata framework described in [64], where a precise notion of execution is given and conditions for well-posedness (existence and uniqueness) are mentioned. Moreover, if the collection $\{Dom_{a,i}, i \in Q\}$ is a polyhedral subdivision of $Y_a \times U$[1], then the SA system reduces to a standard piecewise affine system.

Remark 1. *If the transition condition depends on the state ξ_a, then one can include ξ_a as output variable to get back to the considered modeling framework where domains are defined as a function of the output (and input).*

4.2.2 System reduction based on balanced truncation

The proposed procedure unfolds into the following steps: 1) the SA system is rewritten as a SL system with state reset; 2) a reduced order model of the SL system is introduced by first applying BT to the continuous dynamics in each mode, and then introducing appropriate maps for the reset of the reduced continuous state component when a mode transition occurs; 3) the output of the SA system is reconstructed based on the reduced SL system output.

1) Reformulation as a SL system with state reset: We next build a SL system with state reset that is equivalent to the original SA system, in that (ξ_a, q_a) and y_a can be recovered exactly from the state and output variables of the SL system.

Let $y \in Y = Y_a$, and $\xi \in \Xi = \Xi_a$ evolve according to linear dynamics that depend on the operating mode $q \in Q$ as follows:

$$\mathcal{S} : \begin{cases} \dot{\xi}(t) = \mathcal{A}_q \xi(t) + \mathcal{B}_q u(t) \\ y(t) = \mathcal{C}_q \xi(t) \end{cases} \tag{17}$$

Set $\bar{y}_{a,q} = \mathcal{C}_q \bar{\xi}_{a,q} + g_q$, where $\bar{\xi}_{a,q} = -\mathcal{A}_q^{-1} f_q$, with $\mathcal{A}_q$ invertible by Assumption 1. A transition from mode $i \in Q$ to mode $j \in Q$ occurs as soon as $(y + \bar{y}_{a,i}, u)$ exits Dom_i and enters Dom_j, where $Dom_q = Dom_{a,q}$, $q \in Q$.

When a discrete transition from mode $i \in Q$ to mode $j \in Q$ occurs at time t^-, then, ξ is reset as follows

$$\xi(t) = \xi(t^-) + \Delta_{ji}^{\xi}, \quad \text{with } \Delta_{ji}^{\xi} = \bar{\xi}_{a,i} - \bar{\xi}_{a,j}. \tag{18}$$

[1]This requires that each polyhedron $Dom_{a,i}$ is of dimension $p + m$, and the intersection $Dom_{a,i} \cap Dom_{a,j}$, $i \neq j$, is either empty or a common proper face of both polyhedra.

Proposition 1. *Suppose that the SA and SL systems are initialized with initial conditions* $\xi_a(0)$, $q_a(0)$, *and* $\xi(0) = \xi_a(0) - \bar{\xi}_{a,q_a(0)}$, $q(0) = q_a(0)$, *respectively, and are both fed by the same input* $u(t)$, $t \in [0, \mathcal{T}]$. *Then, the executions of* ξ_a, q_a *and* y_a *over* $[0, \mathcal{T}]$ *can be recovered from those of* ξ, q *and* y *as follows:*

$$\begin{aligned} q_a(t) &= q(t), \\ \xi_a(t) &= \xi(t) + \bar{\xi}_{a,q(t)}, \\ y_a(t) &= y(t) + \bar{y}_{a,q(t)}. \end{aligned} \tag{19}$$

Remark 2. *The reset condition* (18) *is such that* ξ_a *reconstructed from* ξ *according to* (19) *is continuous. Continuity of* ξ_a *is generally not guaranteed if* ξ *is approximated through a reduced order model of the SL system.*

2) Reduction of the SL system: A reduced order model of the SL system with state reset defined above can be obtained by applying BT with the state residualization approach [6], to each single linear dynamics in (17). If the mode of the system were fixed, then, BT would be effective in reproducing the response y, at least in the long run, when the ZIR has vanished.

We associate with each mode $q \in Q$ a reduced model of order $n_{r,q} < n$:

$$\mathcal{S}_r : \begin{cases} \dot{x}_{r,q}(t) = A_{r,q} x_{r,q}(t) + B_{r,q} u(t) \\ \hat{y}(t) = C_{r,q} x_{r,q}(t) + D_{r,q} u(t) \end{cases} \tag{20}$$

and define transitions between modes, say from mode i to mode j, by evaluating when $(\hat{y} + \bar{y}_{a,i}, u)$ exits from domain Dom_i and enters into Dom_j. Indeed, $\hat{y} + \bar{y}_{a,i}$ represent the output y_a reconstructed using (19). As for the state reset map (18) associated with a mode transition from $i \in Q$ to $j \in Q$, we shall reformulate it as

$$x_{r,j}(t) = L_{ji} x_{r,i}(t^-) + M_{ji} u(t^-) + N_{ji} \Delta^{\xi}_{ji}, \tag{21}$$

where $x_{r,i}(t^-) \in \mathbb{R}^{n_{r,i}}$ is the value of the reduced state in mode i, prior to the transition to mode j, $x_{r,j}(t) \in \mathbb{R}^{n_{r,j}}$ is the updated reduced state value, and L_{ji}, M_{ji}, N_{ji} are matrices of appropriate dimensions. In Section 4.2.3, we present different methods to define them.

3) Reconstruction of the SA system output: The output of the SA system is reconstructed based on (19) using the output $\hat{y}$ of the SL reduced system as an estimate of the output y of the SL system. This leads to

$$\hat{y}_a(t) = \hat{y}(t) + \bar{y}_{a,q(t)}. \tag{22}$$

4.2.3 State reset maps: alternative choices

In this section we introduce different reset maps that can be used for the approximate model. The choice of the reset map is of utter importance, since it strongly affects the quality of the approximated solution.

Preliminary definitions: Consider a transition from mode $i \in Q$ to mode $j \in Q$. One can determine an expression for $\hat{\xi}_j$, representing the SL system state associated with mode j as reconstructed from the reduced state $x_{r,i}$.

Recall first that $\hat{\xi}_i$ can be obtained by applying the balanced transformation matrix T_i to the reconstructed continuous state $\hat{x}_i$ of the SL system, i.e., $\hat{\xi}_i = T_i^{-1}\hat{x}_i$. In turn, $\hat{x}_i$ can be reconstructed as $\hat{x}_i = \left[x'_{r,i}\ x'_{nr,i}\right]'$, where $x_{nr,i}$ is the part of the state that is neglected in the reduced model (20), and that can be recovered as a function of $x_{r,i}$ and u by assuming an equilibrium condition in the original not-reduced linear dynamics (BT with state residualization) [6]. This leads to:

$$\hat{x}_i = H_i x_{r,i} + K_i u,$$

where H_i and K_i are suitable defined matrices [51]. Plugging the expressions of $\hat{\xi}_i$ and $\hat{x}_i$ into (18), yields

$$\begin{aligned} \hat{\xi}_j(t) &= \hat{\xi}_i(t^-) + \Delta_{ji}^{\xi} \\ &= T_i^{-1} H_i\, x_{r,i}(t^-) + T_i^{-1} K_i\, u(t^-) + \Delta_{ji}^{\xi}. \end{aligned} \tag{23}$$

We next shall define the reset maps for the state of the reduced SL system when a mode transition occurs from $i \in Q$ at time t^- to $j \in Q$ at time t.

SR map – a reset map based on state reconstruction: The State Reconstruction-based reset map (SR map for brevity) was proposed in [47] and relies on the following idea: reconstruct the whole state $\hat{x}_j(t)$ in balanced form and then extract its first $n_{r,j}$ components corresponding to the reduced order state in mode j. In formulas, $x_{r,j}(t) = E_{n_{r,j}}\hat{x}_j(t)$, where $E_{n_{r,j}}$ is a matrix that extracts the first $n_{r,j}$ rows from $\hat{x}_j(t)$, $n_{r,j}$ being the dimension of $x_{r,j}$ in mode j. Now, $\hat{x}_j(t)$ can be obtained as $\hat{x}_j(t) = T_j\hat{\xi}_j(t)$. Plugging the expression of $x_{r,j}(t)$ into the expression of $\hat{x}_j(t)$, and using (23), we finally obtain

$$x_{r,j}(t) = E_{n_{r,j}} T_j \left(T_i^{-1} H_i x_{r,i}(t^-) + T_i^{-1} K_i u(t^-) + \Delta_{ji}^{\xi} \right). \tag{24}$$

Matrices L_{ji}, M_{ji}, and N_{ji} can be obtained by direct comparison with (21). According to a similar reasoning, the system is initialized as follows

$$q_r(0) = q_a(0) = q_0,\ x_{r,q_0}(0) = E_{n_{r,q_0}} T_{q_0} \left(\xi_a(0) - \bar{\xi}_{a,q_0} \right),$$

with the understanding that $(y_a(0), u(0))$ is an interior point of Dom_{a,q_0}, for any admissible $u(0)$.

OG map – a reset map to reproduce the output ZIR: Model reduction techniques for asymptotically stable linear systems aim at finding a model that best reproduce the forced response of the system, while neglecting the ZIR. However, in SA systems, the system response depends on the mode transitions, which, in turn, depends on the continuous output behavior (forced plus ZIR). We here introduce a reset map that minimizes the L^2-norm of the error when reproducing the ZIR of the output y. As we shall see next, its expression depends on the Observability Gramians (OG) of the linear systems associated with the different modes.

In formulas, we set $x_{r,j} = \Psi_j \hat{\xi}_j$ and choose Ψ_j so as to minimize

$$J(\Psi_j) = \int_0^{+\infty} \|y_{zir,j}(t) - \hat{y}_{zir,j}(t)\|^2 \, \mathrm{d}t, \tag{25}$$

where $y_{zir,j}$ and $\hat{y}_{zir,j}$ respectively denote the ZIR of the original linear dynamics (17) initialized with $\hat{\xi}_j$ and that of the reduced order dynamics (20) initialized with $x_{r,j} = \Psi_j \hat{\xi}_j$. The solution to this optimization problem can be found analytically as shown in Proposition 2, which proof can be found in [51].

Proposition 2. *Suppose that the reduced order model* (20) *with $q = j$ is observable. Then, matrix $\Psi_j^\star$ minimizing* (25) *for any $\hat{\xi}_j$ is given by $\Psi_j^\star = \mathcal{W}_{r,o,j}^{-1} \mathcal{W}_{\times,j}$, where*

$$\mathcal{W}_{r,o,j} = \int_0^{+\infty} (e^{A_{r,j}t})' C_{r,j}' C_{r,j} e^{A_{r,j}t} \, \mathrm{d}t$$
$$\mathcal{W}_{\times,j} = \int_0^{+\infty} (e^{\mathcal{A}_j t})' \mathcal{C}_j' C_{r,j} e^{A_{r,j}t} \, \mathrm{d}t.$$

Remark 3. *Note that the observability assumption in Proposition 2 is satisfied under mild conditions as detailed in [6].*

Matrix $\mathcal{W}_{r,o,j}$ can be obtained by solving the Lyapunov equation

$$A_{r,j} \mathcal{W}_{r,o,j} + \mathcal{W}_{r,o,j} A_{r,j}' + C_{r,j}' C_{r,j} = 0,$$

while matrix $\mathcal{W}_{\times,j}$ is the solution to the Sylvester equation

$$A_{r,j}' \mathcal{W}_{\times,j} + \mathcal{W}_{\times,j} \mathcal{A}_j + C_{r,j}' \mathcal{C}_j = 0.$$

Now, plugging the expression (23) for $\hat{\xi}_j(t)$ into $x_{r,j} = \Psi_j \hat{\xi}_j$ and setting $\Psi_j = \Psi_j^\star$, we get

$$x_{r,j}(t) = \Psi_j^\star \left(T_i^{-1} H_i x_{r,i}(t^-) + T_i^{-1} K_i u(t^-) + \Delta_{ji}^\xi \right) \tag{26}$$

Matrices L_{ji}, M_{ji}, and N_{ji} can be obtained by direct comparison of with (21). As for the system initialization, we set

$$q_r(0) = q_a(0) = q_0, \; x_{r,q_0}(0) = \Psi_{q_0}^\star \left(\xi_a(0) - \bar{\xi}_{a,q_0} \right). \tag{27}$$

Instead of considering an infinite horizon when evaluating the ZIR output error, one can take into account the switching nature of the system and consider the error only during the finite horizon $[0, \tau]$. Correspondingly, the error function to be minimized becomes

$$J_\tau(\Psi_j^\tau) = \int_0^\tau \|y_{zir,j}(t) - \hat{y}_{zir,j}(t)\|^2 \, \mathrm{d}t.$$

The resulting optimal $\Psi_j^{\tau\star}$ matrix is given by $\Psi_j^{\tau\star} = \mathcal{W}_{r,o,j}^{-1}(\tau)\mathcal{W}_{\times,j}(\tau)$, where

$$\begin{aligned}\mathcal{W}_{r,o,j}(\tau) &= \int_0^\tau (e^{A_{r,j}t})' C_{r,j}' C_{r,j} e^{A_{r,j}t} \, \mathrm{d}t \\ \mathcal{W}_{\times,j}(\tau) &= \int_0^\tau (e^{\mathcal{A}_j t})' \mathcal{C}_j' C_{r,j} e^{A_{r,j}t} \, \mathrm{d}t.\end{aligned}$$

The proof of this result is analogous to that in the infinite horizon case. Still, observability of the reduced order model (20) with $q = j$ is required for $\mathcal{W}_{r,o,j}$ to be invertible and Remark 3 applies.

The finite horizon quantities involved in the expression of $\Psi_j^{\tau\star}$ can be computed as

$$\begin{aligned}\mathcal{W}_{r,o,j}(\tau) =& \mathcal{W}_{r,o,j} - \int_\tau^{+\infty} (e^{A_{r,j}t})' C_{r,j}' C_{r,j} e^{A_{r,j}t} \, \mathrm{d}t \\ =& \mathcal{W}_{r,o,j} - \mathcal{W}_{r,o,j}^{(\tau,\infty)}, \\ \mathcal{W}_{\times,j}(\tau) =& \mathcal{W}_{\times,j} - \int_\tau^{\infty} (e^{\mathcal{A}_j t})' \mathcal{C}_j' C_{r,j} e^{A_{r,j}t} \, \mathrm{d}t \\ =& \mathcal{W}_{\times,j} - W_{\times,j}^{(\tau,\infty)},\end{aligned}$$

where $\mathcal{W}_{r,o,j}^{(\tau,\infty)}$ and $\mathcal{W}_{\times,j}^{(\tau,\infty)}$ can be obtained as the solution of the Lyapunov and Sylvester equations

$$\begin{aligned}A_{r,j}\mathcal{W}_{r,o,j}^{(\tau,\infty)} + \mathcal{W}_{r,o,j}^{(\tau,\infty)} A_{r,j}' + \left(e^{A_{r,j}\tau}\right)' C_{r,j}' C_{r,j} e^{A_{r,j}\tau} = 0, \\ A_{r,j}'\mathcal{W}_{\times,j}^{(\tau,\infty)} + \mathcal{W}_{\times,j}^{(\tau,\infty)} \mathcal{A}_j + \left(e^{A_{r,j}\tau}\right)' C_{r,j}' \mathcal{C}_j e^{\mathcal{A}_j\tau} = 0.\end{aligned}$$

Note that well-posedness of the above equations is guaranteed by the fact that A_j and $A_{r,j}$ are Hurwitz.

The matrices in the reset map (21) and the system initialization are analogous to the case of infinite horizon, but with $\Psi_j^{\tau\star}$ in place of $\Psi_j^\star$.

The choice for τ depends on the settling times of the different mode dynamics. A sensible choice is to set τ equal to the settling time of the neglected dynamics.

To distinguish between the two OG reset maps, we shall refer to the one with the infinite horizon as OG_∞ and the one with finite horizon $[0, \tau]$ as OG_τ.

Variants that preserve the output continuity In certain application contexts, it may be desirable to preserve the continuity of the output of the original system. This is not guaranteed when adopting the reset maps defined above and motivates the derivations hereafter.

To get continuity, the value of the output $\hat{y}_a(t)$ reconstructed based on (22) before and after the reset should be identical. This leads to the following equation

$$C_{r,j}x_{r,j}(t) + D_{r,j}u(t) + \bar{y}_{a,j} = C_{r,i}x_{r,i}(t^-) + D_{r,i}u(t^-) + \bar{y}_{a,i}.$$

Under the assumption that the input u is a continuous signal, and letting $\Delta^y_{ji} = \bar{y}_{a,i} - \bar{y}_{a,j}$, this simplifies to

$$C_{r,j}x_{r,j}(t) = C_{r,i}x_{r,i}(t^-) + (D_{r,i} - D_{r,j})u(t^-) + \Delta^y_{ji}.$$

The values of $x_{r,j}(t)$ that satisfy the above condition can be expressed as $x_{r,j}(t) = \tilde{x}_{r,j}(t) + w_j$, with

$$\tilde{x}_{r,j}(t) = C^\dagger_{r,j}\left(C_{r,i}x_{r,i}(t^-) + (D_{r,i} - D_{r,j})u(t^-) + \Delta^y_{ji}\right)$$

where $C^\dagger_{r,j}$ is the pseudo-inverse of $C_{r,j}$ and $w_j \in \mathbb{R}^{n_{r,j}}$ is in the null space of $C_{r,j}$, here denoted as $\ker(C_{r,j})$. If $\ker(C_{r,j}) \neq \{0\}$, we have some degrees of freedom to spend and we can choose w_j so that the resulting value for $x_{r,j}(t)$ best matches some given reference value $\bar{x}_{r,j}(t)$. If instead $\ker(C_{r,j}) = \{0\}$, then, $w_j = 0$, and the reset matrices are derived by a direct comparison with (21).

Let us consider now the case when $\ker(C_{r,j}) \neq \{0\}$. If we let $\{v_1, v_1, \ldots, v_{n_{v,j}}\}$ be a basis of $\ker(C_{r,j})$, and set $V_j = \begin{bmatrix} v_1 & v_2 & \ldots & v_{n_{v,j}} \end{bmatrix}$, then, $w_j = V_j\alpha$ with $\alpha \in \mathbb{R}^{n_{v,j}}$ and we can select α by solving the least squares problem

$$\alpha^\star = \arg\min_\alpha \|\tilde{x}_{r,j}(t) + V_j\alpha - \bar{x}_{r,j}(t)\|,$$

which leads to $\alpha^\star = V_j^\dagger \bar{x}_{r,j}(t)$, since it holds that $V_j^\dagger C^\dagger_{r,j} = 0$. We then finally have:

$$x_{r,j}(t) = \tilde{x}_{r,j}(t) + V_j V_j^\dagger \bar{x}_{r,j}(t), \tag{28}$$

which, depending on the chosen $\bar{x}_{r,j}(t)$ leads to different expressions for the matrices L_{ji}, M_{ji}, and N_{ji} in the reset map (21).

If we adopt the expression in the SR map (24) for $\bar{x}_{r,j}(t)$, then we can define the Continuous State Reconstruction-based reset map (CSR map). If we instead set $\bar{x}_{r,j}(t)$ equal to the OG_∞ map expression (26), we obtain the Continuous Observability Gramian-based map with infinite horizon (COG_∞ map). Analogously, we can define the Continuous Observability Gramian-based map with finite horizon $[0, \tau]$ (COG_τ).

As for the initialization, $q(0) = q_a(0) = q_0$, whereas the value for $x_{r,q_0}(0)$ is obtained by setting the value of the output $\hat{y}_a(0)$ reconstructed based on (22) equal to that of $y_a(0)$ obtained based on the system initialization. This leads to the following equation

$$C_{r,q_0}x_{r,q_0}(0) + D_{r,q_0}u(0) + \bar{y}_{a,q_0} = y_a(0),$$

where $y_a(0)$ is given by the initial conditions of the system, i.e., $y_a(0) = \mathcal{C}_{q_a}\xi_a(0) + g_{q_0}$.

From this equation, by following similar steps than those used for deriving (28), we get that

$$x_{r,q_0}(0) = \begin{cases} \tilde{x}_{r,q_0}(0), & \ker(C_{r,j}) = \{0\} \\ \tilde{x}_{r,q_0}(0) + V_{q_0}V_{q_0}^{\dagger}\bar{x}_{r,q_0}(0), & \ker(C_{r,j}) \neq \{0\}, \end{cases}$$

where we set $\tilde{x}_{r,q_0}(0) = C_{r,q_0}^{\dagger}(-D_{r,q_0}u(0) - \bar{y}_{a,q_0} + y_a(0))$, and $\bar{x}_{r,q_0}(0)$ is the initialization of the SR, OG_∞, or OG_τ reset map.

4.2.4 A randomized method for order selection

In [47], following an approach that is quite standard for linear systems [6], a threshold value γ is chosen, and the order of the reduced SL system (20) in mode $q \in Q$ is set equal to

$$n_{r,q} = \min\{i \in \{1, 2, \ldots, n\} : \psi_q(i) < \gamma\}, \tag{29}$$

where $\psi_q : \{1, 2, \ldots n\} \to [0, 1)$ is given by $\psi_q(i) = 1 - \sum_{j=1}^{i}\sigma_{j,q}/\sum_{j=1}^{n}\sigma_{j,q}$, $\sigma_{1,q} \geq \sigma_{2,q} \geq \cdots \geq \sigma_{n,q}$ being the HSVs of the SL system dynamics (17) in mode q.

Our goal here is to introduce a sound method for making an appropriate selection of the threshold value γ, when the input u is stochastic and one has to verify a property that depends on the behavior of the SA system output y_a along a finite time horizon $\mathcal{T}$. For the resulting stochastic hybrid system and its executions to be well-defined according to the notion in [37], we shall assume in the following that input u is a white noise with a given power spectral density.

A randomized method for order selection is proposed, which involves feeding the candidate reduced order models and the system with the same realizations of the stochastic input and comparing their outputs over $\mathcal{T}$. If the number of realizations is appropriately chosen, then the quality of the model assess over them generalizes to the unseen instances, except for a set of a-priori defined probability ϵ. Notably, this can be reinterpreted as an ϵ-robust assessment result.

Let us denote by Γ the (finite) set of possible threshold values γ, those that result in a different choice for $\{n_{r,q},\ q \in Q\}$, and by $\hat{y}_a^{\gamma}$ the estimate of y_a obtained through the reduced SL system when the threshold value is set equal to γ.

The approximation error can be quantified through a function $d_{\mathcal{T}}(\cdot,\cdot)$ that maps each pair of trajectories $y_a(t)$, $t \in \mathcal{T}$, and $\hat{y}_a^{\gamma}(t)$, $t \in \mathcal{T}$, into a positive real number $d_{\mathcal{T}}(y_a, \hat{y}_a^{\gamma})$ that represents the extent to which the output y_a of the SA system differs from its estimate $\hat{y}_a^{\gamma}$

along the time horizon $\mathcal{T}$. Function $d_{\mathcal{T}}(\cdot,\cdot)$ satisfies $d_{\mathcal{T}}(y_a, \hat{y}_a^\gamma) = 0$ if $\gamma = 0$, since in that case no reduction is performed and, hence, $\hat{y}_a^\gamma(t) = y_a(t)$, $t \in \mathcal{T}$.

In order to make an appropriate selection of γ, we adopt the notion of approximate simulation in [38, 3, 26, 57] to assess the quality of the reduced order model with threshold value γ. This involves computing the maximal value $\rho_\gamma^\star$ taken by $d_{\mathcal{T}}(y_a, \hat{y}_a^\gamma)$ over all realizations of the stochastic input $u(t)$ and the (possibly) stochastic initialization $\xi_a(0)$ of the SA system, except for a set of probability at most $\epsilon \in (0,1)$. An 'optimal' value for γ can then be chosen by inspecting the values of $\rho_\gamma^\star$ as a function of $\gamma \in \Gamma$ and selecting the appropriate compromise between quality of the approximation and tractability of the resulting reduced order model.

More precisely, we introduce the following family of chance-constrained optimization problems (CCPs) parametrized by $\gamma \in \Gamma$:

$$\begin{aligned} CCP_\gamma : \min_{\rho} \ & \rho \\ \text{subject to: } & \mathbb{P}\{d_{\mathcal{T}}(y_a, \hat{y}_a^\gamma) \leq \rho\} \geq 1 - \epsilon. \end{aligned} \tag{30}$$

By directly inspecting the solution of (30) as a function of γ, one can then select the appropriate compromise between accuracy and simplicity of the model, respectively expressed through $\rho_\gamma^\star$, and $n_{r,q}$, $q \in Q$, in (29).

Remark 4. *As argued in [3], the directional Hausdorff distance $d_{\mathcal{T}}(y_a, \hat{y}_a^\gamma) = \sup_{t\in\mathcal{T}} \inf_{\tau\in\mathcal{T}} \|y_a(t) - \hat{y}_a^\gamma(\tau)\|$ is a sensible choice for $d_{\mathcal{T}}(y_a, \hat{y}_a^\gamma)$ when performing probabilistic verification, e.g., when estimating of the probability that y_a will enter some set within $\mathcal{T}$.*

Solving CCPs like (30) is known to be difficult, and even NP-hard in some cases, [15]. We then head for an approximate solution where instead of considering all the possible realizations for the stochastic uncertainty, we consider only a finite number N of them called "scenarios", extracted at random, and treat them as if they were the only admissible uncertainty instances. This leads to the formulation of Algorithm 1, where the chance-constrained solution is determined based on the extracted scenarios and a empirical violation parameter $\eta \in (0, \epsilon)$. Notably, in Proposition 3 it is proven that, if the number N of extractions is appropriately chosen, the obtained estimate of $\rho_\gamma^\star$ is chance-constrained feasible, uniformly with respect to $\gamma \in \Gamma$, with a-priori specified (high) probability. The proof of Proposition 3 can be found in [51], and rests on results from the *scenario approach* [16, 15].

Proposition 3. *Select a confidence parameter $\beta \in (0,1)$, and an empirical violation parameter $\eta \in (0, \epsilon)$. If N satisfies*

$$\textstyle\sum_{i=0}^{\lfloor \eta N \rfloor} \binom{N}{i} \epsilon^i (1-\epsilon)^{N-i} \leq \frac{\beta}{|\Gamma|}, \tag{31}$$

Algorithm 1

1: extract N realizations of the stochastic input $u^{(i)}(t)$, $t \in \mathcal{T}$, $i = 1, 2, \ldots, N$, and N samples of the initial condition $\xi_a(0)^{(i)}$, $i = 1, 2, \ldots, N$, and let $k = \lfloor \eta N \rfloor$;

2: for all $\gamma \in \Gamma$ do

2.1: determine the N realizations of the output signals $y_a^{(i)}(t)$ and $\hat{y}_a^{\gamma,(i)}(t)$, $t \in \mathcal{T}$, $i = 1, 2, \ldots, N$, when the SL system and the reduced order model with parameter γ are fed by the extracted $u^{(i)}(t)$;

2.2: compute $\hat{\rho}^{(i)} := d_{\mathcal{T}}(y_a^{(i)}, \hat{y}_a^{\gamma,(i)})$, $i = 1, 2, \ldots, N$, and determine the indexes $\{h_1, h_2, \ldots h_k\} \subset \{1, 2, \ldots, N\}$ of the k largest values of $\{\hat{\rho}^{(i)}, i = 1, 2, \ldots, N\}$

2.3: set $\hat{\rho}_\gamma^\star = \max_{i \in \{1,2,\ldots,N\} \setminus \{h_1, h_2, \ldots, h_k\}} \hat{\rho}^{(i)}$.

where $|\Gamma|$ *denotes the cardinality of* Γ*, then, the solution* $\hat{\rho}_\gamma^\star$, $\gamma \in \Gamma$*, to Algorithm 1 satisfies* $\mathbb{P}\{d_{\mathcal{T}}(y_a, \hat{y}_a^\gamma) \leq \hat{\rho}_\gamma^\star\} \geq 1 - \epsilon, \ \forall \gamma \in \Gamma$, *with probability at least* $1 - \beta$. □

If we discard the confidence parameter β for a moment, this proposition states that for any $\gamma \in \Gamma$, the randomized solution $\hat{\rho}_\gamma^\star$ obtained through Algorithm 1 is feasible for the chance-constrained problem (30). As η tends to ϵ, $\hat{\rho}_\gamma^\star$ approaches the desired optimal chance constrained solution $\rho_\gamma^\star$. In turn, the computational effort grows unbounded since N scales as $\frac{1}{\epsilon - \eta}$, [15], therefore, the value for η depends in practice from the available computational resources. As for β, one should note that $\hat{\rho}_\gamma^\star$ is a random quantity that depends on the randomly extracted input realizations and initial conditions. It may happen that the extracted samples are not representative enough, in which case the size of the violation set will be larger than ϵ. Parameter β controls the probability that this happens and the final result holds with probability $1 - \beta$. N satisfying (31) depends logarithmically on $|\Gamma|/\beta$, [15], so that β can be chosen as small as 10^{-10} (and, hence, $1 - \beta \simeq 1$) without growing significantly N.

Interestingly, the guarantees provided by Proposition 3 are valid irrespectively of the underlying probability distribution of the input, which may even not be known explicitly, e.g., when running Algorithm 1 with collected time series as realizations of the stochastic input u.

Remark 5. *Note that even in the case of stable continuous dynamics, switching can produce unstable behaviors. However, if some reduced order model presents an unstable behavior, which makes the distance between* y_a *and* $\hat{y}_a^\gamma$ *large, that model is not selected.*

4.2.5 Numerical example

In this section, a multi-room heating system with a switching control policy is presented. The example is inspired to a benchmark for hybrid system verification presented in [21].

Consider the problem of controlling the temperature in a number of rooms of a building. Each room has one heater, but there is a constraint on the number of heaters in the building that can be "active" (i.e., that can be used and turned on if needed) at the same time. Differently from the original benchmark in [21], we model also the dynamics of the heaters.

The temperature T_i in a room $i \in \{1, \ldots, N_r\}$ depends on T_i itself, on the temperature of the adjacent rooms T_j with $j \neq i$, on the outside temperature T_{ext}, and on h_i, a boolean variable that is 1 when the heater is on in room i, and 0 otherwise. The heat transfer coefficient between room i and room j is k_{ij}, and the one between room i and the external environment is $k_{e,i}$. We assume that the heat exchange is symmetric, i.e., $k_{ij} = k_{ji}$. Rooms i and j are adjacent when $k_{ij} > 0$, otherwise $k_{ij} = 0$.

The volume of the room is V_i, and the wall surface between room i and room j is $S_{r,ij}$, while that between room i and the environment is $S_{e,i}$. Air density and heat capacity are $\rho_a = 1.225\,\text{kg/m}^3$ and $c = 1005\,\text{J/(kg K)}$, respectively. Letting $\phi_i = \rho_a c V_i$, we can formulate the following dynamic model for room i and its heater:

$$\phi_i \dot{T}_i = \sum_{j \neq i} S_{r,ij} k_{ij} (T_j - T_i) + S_{e_i} k_{e,i} (T_{\text{ext}} - T_i) + \kappa_i \theta_i$$

$$\tau_{h,i} \dot{\theta}_i = -\theta_i + h_i \cdot p_i - \chi_i T_{\text{ext}}$$

which is an affine system, with κ_i representing the maximum heat flow rate that the heater can provide, while $p_i \in \{0, 1\}$ is a binary variable indicating if the heater is active in room i ($p_i = 1$) or not ($p_i = 0$). The heater dynamics is represented by a first-order system with a time constant $\tau_{h,i}$. If we neglect the term $-\chi_i T_{\text{ext}}$ in the heater dynamics and set $h_i = p_i = 1$, the heater state variable θ_i will tend to 1 so that the heater will provide its maximum heat flow rate κ_i to the room when it is active and on. The term $-\chi_i T_{\text{ext}}$ is introduced to account for the influence of the external temperature on the heating system. Notice that $p_i = 1$ just indicates that the heater is active in room i, while h_i is the variable that indicates whether it is actually turned on ($h_i = 1$).

The physical nature of the considered system is not switching. However, the switching control policy presented in [21] is used to control the temperature in the rooms.

A *room policy* decides whether to switch the heater on in the room: each room has a thermostat that switches the heater on if $T_i \leq \text{on}_i$, and off when $T_i \geq \text{off}_i$.

A *building policy* decides and limits the number of heaters that are jointly active, by

setting the constraint $\sum_{i=1}^{N_r} p_i = \overline{P}$, with $\overline{P} \leq N_r$. The heater of room i is turned active, and the heater of room j becomes not active when: 1) the heater of some room, say room i, is not active, i.e., $p_i = 0$, 2) room j is adjacent to room i and has an active heater, i.e., $p_j = 1$, 3) temperature $T_i \leq \text{get}_i$, and 4) the difference $T_j - T_i \geq \text{dif}_i$.

Each room is identified by an integer index, and whenever a room has more than one adjacent room fulfilling the above condition, the heater is always set active in the room with higher index.

In the following we consider $N_r = 4$ adjacent rooms, with the constraint that only $\overline{P} = 3$ heaters can be active at the same time. The values of the physical system parameters for the considered instance of the problem are reported in Table 3. The external temperature T_{ext} is modeled as a sinusoidal source of period 24 hours with an offset of 4°C, affected by an additive white noise. Note that the resulting stochastic hybrid system and its execution are still well-defined (see [39]).

We assume deterministic initial conditions, i.e., $T_i(0) = 20$, $\theta_i(0) = 0$, $i = 1, \ldots, N_r$, $h(0) = p(0) = [0\ 1\ 1\ 1]'$. The condition that only 3 out of 4 heaters are active at the same time is satisfied by $p(0)$. As for the control policy parameters, we set $\text{off}_i = 21$, $\text{on}_i = 20$, $\text{get}_i = 18$, $\text{dif}_i = 1$, with $i = 1, \ldots, N_r$. Due to the switching policy, the control system can be described as a SA system with continuous state $\xi_a = [T'\,\theta']'$, input $u = T_{\text{ext}}$, and output $y_a = T$:

$$\begin{cases} \dot{\xi}_a = \mathcal{A}\,\xi_a + \mathcal{B}\,u + f_{q_a} \\ y_a = \mathcal{C}\xi_a. \end{cases} \tag{32}$$

As for the mode q_a, it is identified by the values of the binary variables h_i and p_i, which determine the affine term f_{q_a} entering the dynamics of ξ_a. The polyhedral sets Dom_{a,q_a} are determined by the building and room control policies through the chosen thresholds. Note that only the affine term f_{q_a} in (32) depends on the discrete mode $q_a \in Q$, while the state-space matrices $(\mathcal{A}, \mathcal{B}, \mathcal{C})$ are constant. Therefore, the BT can be computed only once, and applied identically for each discrete mode. Still, when selecting the order of the reduced model

Table 3: The multi-room physical system parameters.

$S_{r,ij}$	$12\,\text{m}^2$	$k_{e,i}$	$1\,\text{W/(m}^2\text{K)}$	$\tau_{h,3}$	45.00s
$S_{e,i}$	$24\,\text{m}^2$	κ_1	0.373	$\tau_{h,4}$	47.25s
V_i	$48\,\text{m}^3$	κ_2	0.395	χ_1	1.0×10^{-4}
ϕ_i	59094 J/K	κ_3	0.417	χ_2	2.0×10^{-4}
k_{12}	$2\,\text{W/(m}^2\text{K)}$	κ_4	0.439	χ_3	3.0×10^{-4}
k_{23}	$5\,\text{W/(m}^2\text{K)}$	$\tau_{h,1}$	40.50s	χ_4	4.0×10^{-4}
k_{34}	$2\,\text{W/(m}^2\text{K)}$	$\tau_{h,2}$	42.75s		

one should consider the impact of the selected order on the switched system approximation, which involves also mode transitions. Using standard approaches for the order selection, as the one used in [47] relying on classical HSV analysis, can be misleading. Indeed, the obtained HSVs are $\sigma_1 = 0.993$, $\sigma_2 = 0.026$, $\sigma_3 = 0.001$, $\sigma_4 = 4.514 \times 10^{-5}$, $\sigma_5 = 1.897 \times 10^{-6}$, $\sigma_6 = 6.995 \times 10^{-7}$, $\sigma_7 = 1.805 \times 10^{-8}$, $\sigma_8 = 3.534 \times 10^{-10}$. The HSV analysis suggests that most of the dynamics can be caught by reducing the continuous dynamics of the SA system to a first-order one. Indeed, computing the distance $\psi(n_r)$ used in [47] results in $\psi(1) \cdot 100 = 2.64\%$.

Care has to be taken when applying HSV analysis to the context of SA systems. In fact, classical BT techniques are typically based on the assumption that the ZIR of the system can be neglected since it vanishes in an asymptotically stable linear system, a fact that notoriously does not always hold when dealing with switching systems. Moreover, HSV analysis does not take into account the impact of the reset map.

The multi-room control system is next reduced by means of the constructive methodology proposed in this paper, and the randomized approach for order selection based on the directional Hausdorff distance evaluated over a finite horizon $\mathcal{T} = [0, 200]$ min is applied. In particular, we set $\epsilon = 0.1$ in the CCP (30) and solve it via Algorithm 1. The number of extractions in Algorithm 1 is $N = 778$ and is obtained through the implicit formula (31) with $\eta = 0.05$, $\beta = 10^{-6}$ and $|\Gamma| = 7$.

Since we adopt the same order for the reduced dynamics in each mode, 7 model order reductions are examined, and, according to Proposition 3, the results on the quality assessment of the reduced order models hold with probability $1 - 10^{-6}$.

The length τ of the finite horizon $[0, \tau]$ adopted in OG_τ and COG_τ is set to the settling time of the neglected dynamics. Equation (29) maps each threshold value $\gamma \in \Gamma$ into the order $n_{r,q}$ of the reduced dynamics within mode $q \in Q$ of the SL system with state reset. In this example, we adopt the same order for the reduced dynamics in each mode. Hence, we can simplify the notation to n_r, dropping the dependence from mode q. The values for $\hat{\rho}^\star_\gamma$ obtained with the different reset methods are presented in Figure 15 as a function of n_r. Some interesting considerations can be made by analyzing the results presented in Figure 15. First of all, one can compare the reset maps that do preserve continuity with those that do not. The plots in Figure 15 show that preserving continuity leads to worse performance in terms of accuracy of the approximation. This holds despite of the fact that, for the maps that do not preserve output continuity, a drastic order reduction may yield discontinuities in the state reset that possibly produce chattering behaviors. Furthermore, Figure 15 shows that the OG reset maps exhibit better performance with respect to the SR maps. In particular,

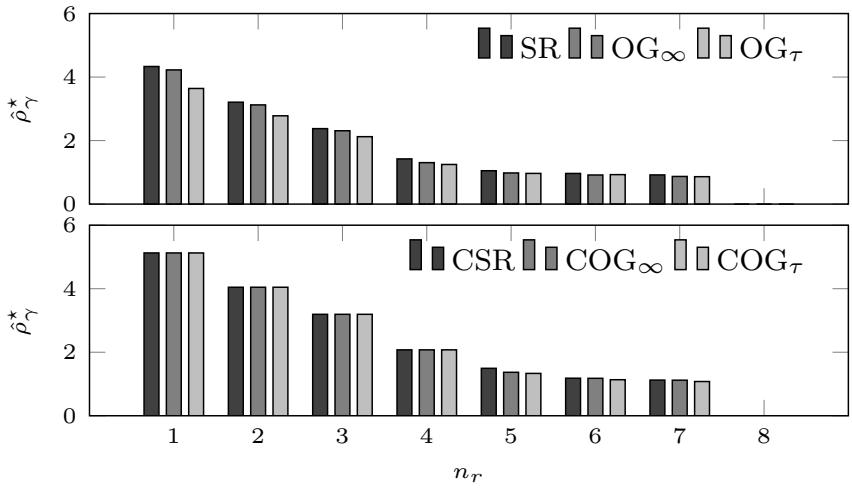

Figure 15: Performance of different reduced models as a function of the order n_r and of the adopted reset maps.

for the OG$_\tau$ map $\hat{\rho}^\star_\gamma$ is reduced on average over n_r by 10.03% in the discontinuous map case, and by 2.68% in the continuous map case.

Notice also that when a reduced order $n_r \leq p = 4$ is used and the output continuity is enforced, then, the same results are obtained with the different reset maps. This is due to the fact that whenever $n_r \leq p$, there are no degrees of freedom left by the continuity constraint to match the originally introduced SR or OG reset maps (see the derivations in Section 4.2.3), so that all maps just enforce continuity and become identical.

From the randomized analysis in Figure 15, it appears that one can push the reduction up to a fifth order without significantly deteriorating the accuracy of the model when the goal of the approximation is the analysis of reachability properties for which the directional Hausdorff distance is a suitable accuracy measure. Reducing the system to a first-order approximation as suggested by the analysis based on the HSV only would instead result in a quite significant degradation of the reduced model performance.

4.2.6 Extension to switched affine systems with dwell time

The approach that we proposed in Section 4.2.5 for model order reduction can also be applied to the case when the mode transitions of the SA system are subject to a DT constraint, which means that a transition from mode $i \in Q$ to mode $j \neq i \in Q$ is enabled when (y_a, u) exits $Dom_{a,i}$ and enters into $Dom_{a,j}$, but can actually occur only if a certain minimum amount of time $\bar{\delta}_i \in \mathbb{R}^+$ (the so-called *dwell time*) has elapsed. Note that DT can be present in a system for two different reasons: either is due to an intrinsic characteristic of the system that presents some delay/inertia when commuting, or it is introduced when designing a control strategy, as in DT switching control, see e.g. [36, 42].

An extension of the SA modeling framework is needed if a DT constraint is present. If we start from a SA system of the form (16), we can introduce the DT constraint as described next. DT can be accounted for by adding to each mode a continuous state variable $\delta \in \mathbb{R}$ that represents a clock with the dynamics of an integrator. The dynamics (16) then is augmented as follows:

$$\begin{bmatrix} \dot{\xi}_a(t) \\ \dot{\delta}(t) \end{bmatrix} = \begin{bmatrix} \mathcal{A}_{q_a} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \xi_a(t) \\ \delta(t) \end{bmatrix} + \begin{bmatrix} \mathcal{B}_{q_a} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u(t) \\ v(t) \end{bmatrix} + \begin{bmatrix} f_{q_a} \\ 0 \end{bmatrix}$$
$$\begin{bmatrix} y_\delta(t) \\ y_a(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ \mathcal{C}_{q_a} & 0 \end{bmatrix} \begin{bmatrix} \xi_a(t) \\ \delta(t) \end{bmatrix} + \begin{bmatrix} 0 \\ g_{q_a} \end{bmatrix}$$

with $v(t) = \bar{v} = 1 \, \forall t \geq 0$, and the extended domain of a discrete mode $q_a \in Q$ is modified as

$$Dom^e_{a,q_a} = \mathbb{R}^+ \times Dom_{a,q_a} \times \{1\} \cup \left[0, \bar{\delta}_{q_a}\right] \times \mathbb{R}^{p \times m} \times \{1\}$$

so as to impose the DT constraint.

Within this extended framework, mode $i \in Q$ is active as long as $([y_\delta \; y_a']', [u' \; v]')$ keeps evolving within $Dom^e_{a,i}$, and a transition to mode $j \neq i \in Q$ occurs as soon as $([y_\delta \; y_a']', [u' \; v]')$ exits $Dom^e_{a,i}$, and enters into $Dom^e_{a,j}$. The reset map $\delta(t) = 0$ needs to be added as soon as a mode transition occurs at time t^-.

Note that the augmented dynamics within each mode is still affine. However, the resulting dynamic matrix is not Hurwitz due to the presence of the clock. Yet, under Assumption 1, the procedure in Section 4.2.2 for model order reduction can be still adopted, in that it can be applied to the original SA system. The clock dynamics and its reset can be considered separately, and only affect the mode transitions of the reduced system via the extended domains definition.

As a consequence to the introduction of the DT, dynamics that decay in a time scale that is larger than the DT itself will be unlikely to be removed when selecting the model order through the proposed randomized approach: This is because of their contribution at the switching times when the state is reset. Finally, the length τ of the time horizon in OG_τ and COG_τ can be tailored to the DT value.

Numerical example: the multi-room heating system: We consider the example of the multi-room heating system in Subsection 4.2.5 and introduce a DT to the switching policy. This means that, we require that the time elapsing between two subsequent switches (heater activated/de-activated and heater turned on/off when active) must be greater than or equal to the DT. We thus increase the state vector with a clock $\delta(t)$ with dynamics $\dot{\delta} = 1$, that is reset to 0 whenever a switch occurs.

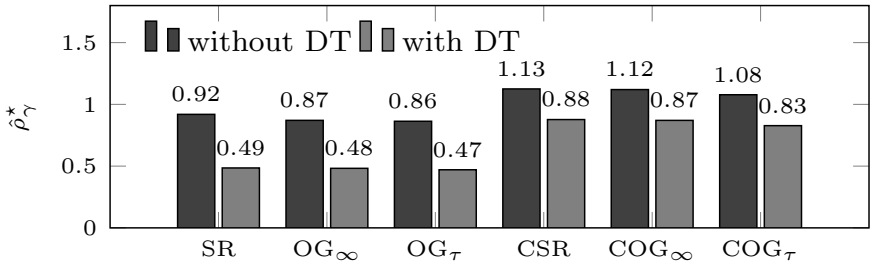

Figure 16: Quality of the reduced order model when a DT is introduced in the control policy (light bars) and when it is not adopted (dark bars).

Note that since the unstable dynamics of the clock do not affect the dynamics of the remaining state variables, one can apply BT to the original system without the clock. Therefore, even if the (augmented) continuous state variable of the multi-room heating system has dimension 9, the reduction must be performed only on the original state of dimension 8 as in the example of Section 4.2.5.

The results obtained when the neglected dynamics has order 1 are reported in Figure 16. The value of the optimal directional Hausdorff distance $\hat{\rho}^\star_\gamma$ is computed through Algorithm 1 for the different reset maps, but just for a value of γ corresponding to an unitary order reduction of the asymptotically stable part of the system. The same parameter values of Section 4.2.5 are here adopted. The DT is set equal to 5 minutes (which is also the settling time of the continuous dynamics) and the time horizon length τ in the OG_τ reset map is set equal to the DT. The OG_τ map gives the best performance in terms of ISE with respect to the other reset maps (see Figure 16). Indeed, the DT is long enough to let the ZIR of the asymptotically stable continuous component vanish.

Not surprisingly, a comparative analysis with the values of the directional Hausdorff distance obtained without the adoption of the DT in the switching policy (see Figure 16) reveals that the quality of the reduced order model deteriorates when the DT is present, and this occurs irrespectively of the adopted reset map.

4.2.7 Conclusions

In this work, we proposed to extend BT to the model reduction of SA systems with endogenous switching. This involved introducing appropriate state reset maps and integrating the reduced order model design with a randomized procedure for model order selection. A comparative analysis of different maps, possibly preserving the output continuity, was performed on a benchmark example of a multi-room heating system. The approach was extended to the case of switched affine systems with DT.

The proposed order selection is based on the discrepancy between the real and approximated output trajectories. If the obtained discrepancy is zero, then the reduced order model exactly reproduces the input-output behavior of the system, and it is possibly a minimal realization. A rigorous approach to exact model reduction for piecewise-affine hybrid systems is proposed in [54].

The considered class of switched systems is characterized by an endogenous switching signal. However, our method can be applied also to the case when transitions are determined by some exogenous switching signal, possibly probabilistic as in the case of Markov jump linear systems.

4.3 Model reduction preserving the input/output behavior for discrete time piecewise affine systems

In this section, we consider the problem of model reduction for discrete time hybrid systems. We focus on the class of Mixed Logical Dynamical (MLD) systems originally introduced in [11]. MLD systems are equivalent to various classes of hybrid models [34, 9], and, in particular, to PieceWise Affine (PWA) systems commuting between a finite set of affine dynamics (the modes), each one associated with a polyhedral region in the partitioned state cross input space. Various analysis and design problems have been addressed for this class of systems using an optimization-based perspective with a mixed integer programming formulation, see e.g. [12, 66, 10, 13, 68, 69]

The goal is to simplify the structure of the system while preserving its input/output behavior. This is particularly useful when addressing a reachability problem where the input has to be designed so as to satisfy some specification expressed in terms of the output evolution, or, more generally, when addressing analysis or design problems that concern the output.

To achieve our goal, we introduce a structural approach based on observability-like analysis. The notion of observability for MLD systems has been treated extensively in [9], where the concept of incrementally observable MLD system is introduced. Possible impact of observability analysis on model reduction is mentioned in the conclusions of the related paper [22]. Here, we propose an approach to model reduction that rests on the Kalman canonical decomposition into observable and unobservable part of the affine dynamics appearing in the MLD model description, which can be isolated by neglecting the discrete component of the hybrid dynamics. The so-obtained seemingly unobservable components may actually affect the discrete mechanism underlying the hybrid system evolution and, hence, they may become observable. We then introduce a sufficient condition to determine if the unobservable com-

ponents of the affine dynamics remains unobservable in the hybrid system dynamics. The approach applies to MLD systems and their equivalent PWA counterpart. If the obtained reduced MLD system is mapped into a PWA system (e.g., via the approach in [8]) that has the same dynamics in adjacent regions of the state cross input space, a mode aggregation procedure can be applied to further simplify the PWA model.

The proposed approach is conceptually simple and easy to implement, since it is based on the standard notion of observability for linear systems. Model reduction methods that preserve the input/output behavior of a PWA system have been proposed in the literature but in a continuous time setting, [54]. These approaches are, hence, not directly comparable with our discrete time method.

It is worth noticing that the work in this section strictly relates to minimal realization theory in that the MLD systems is simplified while preserving exactly its input/output behavior. In the literature, minimal realization theory has been mainly developed for linear and bilinear switched and hybrid systems with externally induced switching. Apparently, it remains an open problem when considering hybrid systems with endogenous switching (see [53]). Our work can hence be seen as a preliminary step in this direction.

The rest of the section is structured as follows. In Subsection 4.3.1, we describe the modeling context, recalling the equivalence between MLD and PWA systems that was proven in [9]. We then illustrate the proposed approach for model reduction based on observability-like analysis in Subsection 4.3.2. We describe the mode reduction procedure in Subsection 4.3.3. We present some numerical examples in Subsection 4.3.4 and conclude the section with some remarks in Subsection 4.3.5.

4.3.1 Modeling framework

We consider a Mixed Logical Dynamical (MLD) system described by the following equalities and inequalities:

$$\begin{aligned} x(k+1) &= Ax(k) + B_u u(k) + B_\delta \delta(k) + B_z z(k) + B_{aff} \\ y(k) &= Cx(k) + D_u u(k) + D_\delta \delta(k) + D_z z(k) + D_{aff} \\ E_x x(k) &+ E_u u(k) + E_\delta \delta(k) + E_z z(k) \leq E_{aff} \end{aligned} \tag{33}$$

where $x \in \mathbb{R}^{n_c} \times \{0,\ 1\}^{n_l}$ is the state composed of both continuous and binary variables, $u \in \mathbb{R}^{m_c} \times \{0,\ 1\}^{m_l}$ is the input vector comprising a continuous and a discrete component. As for δ and z, they are binary and continuous-valued auxiliary variables: $\delta \in \{0,\ 1\}^{r_l}$ and $z \in \mathbb{R}^{r_c}$.

We assume that some reachability specification is given in terms of the behavior in time of the output $y \in \mathbb{R}^{p_c} \times \{0,\ 1\}^{p_l}$.

For an MLD system to be well-defined, the solution to the inequalities in (33) must be unique, i.e., given a state-input pair there exists a unique value for the auxiliary variables δ and z satisfying such inequalities.

Without loss of generality, we shall assume next that the affine terms B_{aff} and D_{aff} are both zero. Indeed, if this were not the case, one can introduce $\bar{x}(k)$ and $\bar{y}(k)$ given by the solution to the system

$$\bar{x}(k+1) = A\bar{x}(k) + B_{aff} \tag{34}$$

$$\bar{y}(k) = C\bar{x}(k) + D_{aff} \tag{35}$$

and replace x and y in (33) with $x + \bar{x}$ and $y + \bar{y}$. As a result, the affine terms will cancel out and the right hand side of the last inequality in (33) will become $E_{aff} - E_x\bar{x}(k)$. If $I - A$ is invertible and one can choose $\bar{x}(0) = (I - A)^{-1}B_{aff}$, then, the solution $\bar{x}(k)$ and $\bar{y}(k)$ to (34) keep constant and, hence, E_{aff} in (33) is replaced by a time invariant term $E_{aff} - E_x\bar{x}(k) = E_{aff} - E_x(I - A)^{-1}B_{aff}$.

Let us consider a PieceWise Affine (PWA) systems governed by

$$\begin{aligned} x(k+1) &= A_i x(k) + B_i u(k) + f_i \\ y(k) &= C_i x(k) + D_i u(k) + g_i \end{aligned} \quad \text{for} \begin{bmatrix} x(k) \\ u(k) \end{bmatrix} \in \mathcal{A}_i, \tag{36}$$

where $x \in \mathcal{X} \subseteq \mathbb{R}^n$ is the state, $u \in\ \mathcal{U} \subseteq \mathbb{R}^m$ is the input, and $y \in \mathbb{R}^p$ is the output, whereas f_i, g_i are constant vectors. Suppose that the collection of sets $\{\mathcal{A}\}_{i=1}^s$ forms a *polyhedral subdivision* of the space $\mathcal{X} \times \mathcal{U}$, that is if $\cup_{i=1}^s \mathcal{A}_i = \mathcal{X} \times \mathcal{U}$, each $\mathcal{A}_i$ is of dimension $n+m$, and the intersection $\mathcal{A}_i \cap \mathcal{A}_j$, $i \neq j$, is either empty or a common proper face of both polyhedra. Then, system (36) is well posed and if $\mathcal{X}$ and $\mathcal{U}$ are bounded, it can be converted in an equivalent MLD system. The idea behind the conversion of a PWA system into the MLD form is to introduce auxiliary variables δ and z that respectively capture which of the original PWA mode is active, and the dynamics associated to that mode. This is done by means of *big-M* techniques that lead to the linear inequalities in (33) (see [9]). The MLD form is typically more convenient when performing optimization-based analysis and design. Notably, the opposite implication also holds true, i.e., MLD systems have an equivalent PWA form.

4.3.2 Structural reduction

Our aim is to detect whether there exists some part of the MLD system (33) that can be neglected without affecting the output behavior.

We start by considering the simple case when the MLD system (33) reduces to a standard linear system, i.e.,

$$\begin{aligned} x(k+1) &= Ax(k) + B_u u(k) \\ y(k) &= Cx(k) + D_u u(k). \end{aligned} \tag{37}$$

In this setting, we just need to determine the non-observable part of the system and then remove it, which entails neglecting those inputs that do not affect the observable part (*non-influential inputs*). This is achieved via a three-steps procedure:

1. Rewrite the system in its observable canonical form by means of an appropriate similarity transformation T_o:

$$\begin{bmatrix} x_{no}(k+1) \\ x_o(k+1) \end{bmatrix} = \tilde{A} \begin{bmatrix} x_{no}(k) \\ x_o(k) \end{bmatrix} + \tilde{B}_u u(k),$$
$$y(k) = \tilde{C} \begin{bmatrix} x_{no}(k) \\ x_o(k) \end{bmatrix} + D_u u(k),$$

   where $\begin{bmatrix} x_{no} \\ x_o \end{bmatrix} = T_o x$, $x_{no} \in \mathbb{R}^{\nu_{no}}$, $x_o \in \mathbb{R}^{\nu_o}$, $\tilde{B} = T_o B_u$, $\tilde{C} = C\ T_o^{-1} = [0 \quad C_o]$ and $\tilde{A}$ has the following upper triangular structure:

$$\tilde{A} = \begin{bmatrix} A_{no} & A_{12} \\ 0 & A_o \end{bmatrix}.$$

2. Remove the non-observable state component x_{no}, i.e., remove the first ν_{no} rows of $\tilde{A}, \tilde{B}_u$ and the first ν_{no} columns of $\tilde{C}$. The resulting system is given by:

$$\begin{aligned} x_o(k+1) &= A_o x_o(k) + \tilde{B}_{u,o} u(k), \\ y(k) &= C_o x_o(k) + D_u u(k), \end{aligned} \tag{38}$$

   where $\tilde{B}_{u,o}$ is the matrix obtained by extracting the last ν_o rows of $\tilde{B}_u$.

3. Check if there exists an index $j \in \{1, \ldots, m\}$ such that the j-th column of both $\tilde{B}_{u,o}$ and D_u are null; if that is the case, input u_j is non-influential and can be removed.

Note that, by construction, the evolution of the output of the reduced order system (38) coincides with the evolution of the output of the original system (37), for any initial condition, and for any assignment of the input.

Our aim now is to detect and remove the *non-observable* part of the system in the case when some discrete dynamics is present. For sake of simplicity, we consider MLD systems

without logic states, i.e., $x \in \mathbb{R}^{n_c}$. We rewrite the MLD system (33) with $B_{aff} = 0$ and $D_{aff} = 0$ and the time index dropped for convenience:

$$\begin{aligned} x^+ &= Ax + B_u u + B_\delta \delta + B_z z \\ y &= Cx + D_u u + D_\delta \delta + D_z z \\ E_x x &+ E_u u + E_\delta \delta + E_z z \leq E_{aff}. \end{aligned} \tag{39}$$

We start focusing on matrices A, B_u, C, D_u, as if the system were linear, and compute the similarity transformation T_o as in the linear case. The system become:

$$\begin{bmatrix} x_{no} \\ x_o \end{bmatrix}^+ = \tilde{A} \begin{bmatrix} x_{no} \\ x_o \end{bmatrix} + \tilde{B}_u u + \tilde{B}_\delta \delta + \tilde{B}_z z \tag{40}$$

$$y = \tilde{C} \begin{bmatrix} x_{no} \\ x_o \end{bmatrix} + D_u u + D_\delta \delta + D_z z \tag{41}$$

$$\tilde{E}_x \begin{bmatrix} x_{no} \\ x_o \end{bmatrix} + E_u u + E_\delta \delta + E_z z \leq E_{aff} \tag{42}$$

where $\tilde{A}$ and $\tilde{C}$ are defined as in the previous section and $\tilde{B}_u = T_o B_u$, $\tilde{B}_\delta = T_o B_\delta$, $\tilde{B}_z = T_o B_z$, $\tilde{E}_x = E_x T_o^{-1}$.

Despite the structure of matrices $\tilde{A}$ and $\tilde{C}$, before possibly removing x_{no}, we need first to check if x_{no} affects the output via the inequalities (42). To understand why this might be the case, suppose that x_{no} affects the value of δ via the inequalities (42), then, x_{no} is indirectly influencing the output via the term $D_\delta \delta$. However, it may be also the case that x_{no} affects only those elements of δ that are "hidden" by matrix D_δ, so that in the end x_{no} does not affect the output. For this reason we should check if x_{no} affects $\tilde{\tilde{\delta}} = D_\delta \delta$ instead of δ. This consideration applies also to variables u and z, so that before analyzing the dependencies introduced by the inequalities (42) we need first to set some changes of variables.

To make the discussion as general as possible we consider the general case when D_u, D_δ and D_z may be rank deficient matrices, i.e.: $\text{rank}(D_u) = \mathbf{r}_u \leq \min\{p, m\}$, $\text{rank}(D_\delta) = \mathbf{r}_\delta \leq \min\{p, r_l\}$, $\text{rank}(D_z) = \mathbf{r}_z \leq \min\{p, r_c\}$.

The full rank factorization (see [14]) of D_u, D_δ and D_z: $D_u = D_{u,L} D_{u,R}$, $D_\delta = D_{\delta,L} D_{\delta,R}$, $D_z = D_{z,L} D_{z,R}$, where $D_{u,L}$, $D_{\delta,L}$, $D_{z,L}$ have, respectively, $\mathbf{r}_u$, $\mathbf{r}_\delta$, $\mathbf{r}_z$ columns, can be used to introduce the following change of variables:

$$\begin{bmatrix} \tilde{u} \\ \tilde{u}^\perp \end{bmatrix} = \begin{bmatrix} D_{u,R} \\ F_u \end{bmatrix} u, \quad \begin{bmatrix} \tilde{\delta} \\ \tilde{\delta}^\perp \end{bmatrix} = \begin{bmatrix} D_{\delta,R} \\ F_\delta \end{bmatrix} \delta, \quad \begin{bmatrix} \tilde{z} \\ \tilde{z}^\perp \end{bmatrix} = \begin{bmatrix} D_{z,R} \\ F_z \end{bmatrix} z, \tag{43}$$

where each row of matrix F_i, $i \in \{u, \delta, z\}$, is orthogonal to each row of the corresponding matrix $D_{i,R}$ (i.e., the rows of F_i form a basis of the null space of $D_{i,R}$). Note that the resulting matrices P_i defined as $P_i = [D'_{i,R}\ F'_i]'$, $i \in \{u, \delta, z\}$ are square and invertible by construction.

In view of the change of variables in (43), the system can be rewritten as:

$$\begin{bmatrix} x_{no} \\ x_o \end{bmatrix}^+ = \tilde{A} \begin{bmatrix} x_{no} \\ x_o \end{bmatrix} + \begin{bmatrix} B^*_{u,L} & B^*_{u,R} \end{bmatrix} \begin{bmatrix} \tilde{u} \\ \tilde{u}^\perp \end{bmatrix} + \begin{bmatrix} B^*_{\delta,L} & B^*_{\delta,R} \end{bmatrix} \begin{bmatrix} \tilde{\delta} \\ \tilde{\delta}^\perp \end{bmatrix} + \begin{bmatrix} B^*_{z,L} & B^*_{z,R} \end{bmatrix} \begin{bmatrix} \tilde{z} \\ \tilde{z}^\perp \end{bmatrix}$$

$$y = \tilde{C} \begin{bmatrix} x_{no} \\ x_o \end{bmatrix} + D_{u,L}\tilde{u} + D_{\delta,L}\tilde{\delta} + D_{z,L}\tilde{z}$$

$$\tilde{E}_x \begin{bmatrix} x_{no} \\ x_o \end{bmatrix} + \begin{bmatrix} E^*_{u,L} & E^*_{u,R} \end{bmatrix} \begin{bmatrix} \tilde{u} \\ \tilde{u}^\perp \end{bmatrix} + \begin{bmatrix} E^*_{\delta,L} & E^*_{\delta,R} \end{bmatrix} \begin{bmatrix} \tilde{\delta} \\ \tilde{\delta}^\perp \end{bmatrix} + \begin{bmatrix} E^*_{z,L} & E^*_{z,R} \end{bmatrix} \begin{bmatrix} \tilde{z} \\ \tilde{z}^\perp \end{bmatrix} \leq E_{aff},$$

where we set

$$\begin{aligned} \begin{bmatrix} B^*_{u,L} & B^*_{u,R} \end{bmatrix} &= P_u^{-1}\tilde{B}_u & \begin{bmatrix} E^*_{u,L} & E^*_{u,R} \end{bmatrix} &= P_u^{-1}E_u \\ \begin{bmatrix} B^*_{\delta,L} & B^*_{\delta,R} \end{bmatrix} &= P_\delta^{-1}\tilde{B}_\delta & \begin{bmatrix} E^*_{\delta,L} & E^*_{\delta,R} \end{bmatrix} &= P_u^{-1}E_\delta \\ \begin{bmatrix} B^*_{z,L} & B^*_{z,R} \end{bmatrix} &= P_z^{-1}\tilde{B}_z & \begin{bmatrix} E^*_{z,L} & E^*_{z,R} \end{bmatrix} &= P_z^{-1}E_z. \end{aligned}$$

Finally, by defining variables $\tilde{\tilde{u}}$, $\tilde{\tilde{\delta}}$, $\tilde{\tilde{z}}$ as:

$$\tilde{\tilde{u}} = D_{u,L}\tilde{u}, \quad \tilde{\tilde{\delta}} = D_{\delta,L}\tilde{\delta}, \quad \tilde{\tilde{z}} = D_{z,L}\tilde{z}, \tag{44}$$

the system can be rewritten as:

$$\begin{bmatrix} x_{no} \\ x_o \end{bmatrix}^+ = \tilde{A} \begin{bmatrix} x_{no} \\ x_o \end{bmatrix} + \begin{bmatrix} \bar{B}_u & B^*_{u,R} \end{bmatrix} \begin{bmatrix} \tilde{\tilde{u}} \\ \tilde{u}^\perp \end{bmatrix} + \begin{bmatrix} \bar{B}_\delta & B^*_{\delta,R} \end{bmatrix} \begin{bmatrix} \tilde{\tilde{\delta}} \\ \tilde{\delta}^\perp \end{bmatrix} + \begin{bmatrix} \bar{B}_z & B^*_{z,R} \end{bmatrix} \begin{bmatrix} \tilde{\tilde{z}} \\ \tilde{z}^\perp \end{bmatrix} \tag{45}$$

$$y = \tilde{C} \begin{bmatrix} x_{no} \\ x_o \end{bmatrix} + \tilde{\tilde{u}} + \tilde{\tilde{\delta}} + \tilde{\tilde{z}} \tag{46}$$

$$\tilde{E}_x \begin{bmatrix} x_{no} \\ x_o \end{bmatrix} + \begin{bmatrix} \bar{E}_u & E^*_{u,R} \end{bmatrix} \begin{bmatrix} \tilde{\tilde{u}} \\ \tilde{u}^\perp \end{bmatrix} + \begin{bmatrix} \bar{E}_\delta & E^*_{\delta,R} \end{bmatrix} \begin{bmatrix} \tilde{\tilde{\delta}} \\ \tilde{\delta}^\perp \end{bmatrix} + \begin{bmatrix} \bar{E}_z & E^*_{z,R} \end{bmatrix} \begin{bmatrix} \tilde{\tilde{z}} \\ \tilde{z}^\perp \end{bmatrix} \leq E_{aff} \tag{47}$$

where

$$\begin{aligned} \bar{B}_u &= B^*_{u,L}D^\dagger_{u,L} & \bar{E}_u &= E^*_{u,L}D^\dagger_{u,L} \\ \bar{B}_\delta &= B^*_{\delta,L}D^\dagger_{\delta,L} & \bar{E}_\delta &= E^*_{\delta,L}D^\dagger_{\delta,L} \\ \bar{B}_z &= B^*_{z,L}D^\dagger_{z,L} & \bar{E}_z &= E^*_{z,L}D^\dagger_{z,L} \end{aligned}$$

and $Q^\dagger$ denotes the *left Moore-Penrose pseudoinverse* of Q, i.e., $Q^\dagger = (Q'Q)^{-1}Q'$. Note that the transformations (43) and (44) can be combined, thus leading to:

$$\begin{bmatrix} \tilde{\tilde{u}} \\ \tilde{u}^\perp \end{bmatrix} = T_u u, \quad \begin{bmatrix} \tilde{\tilde{\delta}} \\ \tilde{\delta}^\perp \end{bmatrix} = T_\delta \delta, \quad \begin{bmatrix} \tilde{\tilde{z}} \\ \tilde{z}^\perp \end{bmatrix} = T_z z,$$

where matrices T_u, T_δ and T_z are given by

$$T_u = \begin{bmatrix} D_u \\ F_u \end{bmatrix}, \quad T_\delta = \begin{bmatrix} D_\delta \\ F_\delta \end{bmatrix}, \quad T_z = \begin{bmatrix} D_z \\ F_z \end{bmatrix}$$

and have all full column rank by construction. Note that the above transformation highlights $\tilde{\tilde{u}}$, $\tilde{\tilde{\delta}}$, $\tilde{\tilde{z}}$, which represent the linear combinations of elements of the original vectors u, δ, z that affect the output of system (39).

Based on (45), (46), (47), we can now carry out the removal of those parts of the system that do not affect the output. To this purpose, we propose the following procedure.

1. Construct the undirected graph $\mathcal{G}$ of dependencies among the components of x_{no}, x_o, $\tilde{\tilde{u}}$, $\tilde{u}^\perp$, $\tilde{\tilde{\delta}}$, $\tilde{\delta}^\perp$, $\tilde{\tilde{z}}$, $\tilde{z}^\perp$ induced by inequalities (47). In particular, define as the nodes of $\mathcal{G}$ such components and draw an arc between two nodes if there is a scalar inequality in (47) involving the corresponding variables.

2. Build vector $\hat{x}_{no}$ with the components of x_{no} that are not connected via a path of $\mathcal{G}$ to any component of x_o, $\tilde{\tilde{u}}$, $\tilde{\tilde{\delta}}$, $\tilde{\tilde{z}}$.

3. Collect in $\hat{\tilde{u}}^\perp$ the components of $\tilde{u}^\perp$, whose corresponding column in $B^*_{u,R}$ is null and that are not connected via a path of $\mathcal{G}$ to any component of x_o, $\tilde{\tilde{u}}$, $\tilde{\tilde{\delta}}$, $\tilde{\tilde{z}}$. Similarly, define $\hat{\tilde{\delta}}^\perp$ and $\hat{\tilde{z}}^\perp$.

4. Remove from (45) all state equations corresponding to the elements of $\hat{x}_{no}$. Accordingly, remove also the corresponding columns of $\tilde{A}$, $\tilde{C}$, $\tilde{E}_x$.

5. Remove from $\tilde{u}^\perp$ the components in $\hat{\tilde{u}}^\perp$ and remove the corresponding columns in $B^*_{u,R}$ and $E^*_{u,R}$. Proceed in the same way for the components of $\hat{\tilde{\delta}}^\perp$ and $\hat{\tilde{z}}^\perp$.

6. Remove from the transformation matrix T_u the rows corresponding to the components in $\hat{\tilde{u}}^\perp$. If the resulting matrix has a column j which is identically 0, then the associated original input u_j is non-influential, and, hence, can be neglected.

Note that the procedure described above can be carried out with very little computational effort, since it only requires the computation of the paths on a graph, which is an operation for which extremely efficient methods exist. Also, it is not affected by E_{aff}, so that the fact that E_{aff} may be time varying is not an issue.

4.3.3 Removal of redundant modes

As mentioned in Section 4.3.1, if an MLD system is well-posed, then, it can be converted in an equivalent PWA system.

It may be the case that, after the model reduction performed on the MLD system, some modes in the PWA form share the same dynamics. In these cases it may be convenient to *merge* them, so as to reduce the total number of modes in the PWA model. The PWA representation (36) requires the sets $\mathcal{A}_i$ to form a polyhedral subdivision of the state-input space. For this reason, in the proposed mode merging approach, we first detect the subsets of modes that share the same dynamics, then we check if there exists a pair of modes such that their union is convex and, if so, we merge them. The resulting set becomes a new element of the subset of modes that share that same dynamics, and the exploration continues iteratively. Note that the order followed in the merging of the modes matters, as it is shown in Figure 17. One can opt for a *greedy* exploration which is sub-optimal in terms of number of modes merged but it is less time consuming, or an exhaustive exploration, which merges the maximum number of modes but it is more time consuming.

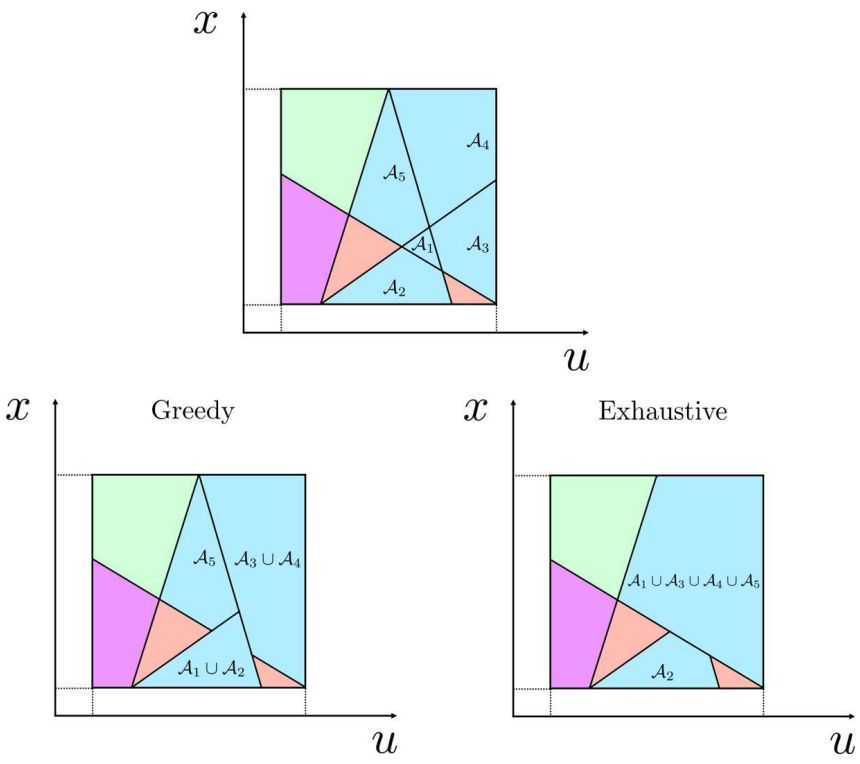

Figure 17: Pictorial view of the difference between a *greedy* merging routine and an exhaustive one. In the greedy routine we merge a mode with the first mode found that makes the union convex. Merging $\mathcal{A}_1$ with $\mathcal{A}_2$ generates a region that can not be merged neither with $\mathcal{A}_5$ nor with the union of $\mathcal{A}_3$ and $\mathcal{A}_4$. Thus, the total number of obtained regions that share the same dynamics is 3. On the other hand, an exhaustive exploration is able to construct only two regions.

4.3.4 Numerical examples

We next show the effectiveness of the approach described in Section 4.3.2 via a numerical example. Consider the MLD system described by:

$$\begin{aligned} x(k+1) &= Ax(k) + B_u u(k) + B_z z(k) \\ y(k) &= Cx(k) + D_u u(k) + D_z z(k) \\ E_x x(k) &+ E_u u(k) + E_\delta \delta(k) + E_z z(k) \leq E_{aff} \end{aligned} \tag{48}$$

where

$$A = \begin{bmatrix} 1 & 0 & -2 & 0 \\ 0 & 1 & 0 & 0 \\ -3 & 0 & -4 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \end{bmatrix},$$

$$B_u = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 2 & 0 & 0 \\ 1 & 3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix}, \quad B_z = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$D_u = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad D_z = \begin{bmatrix} -1 & 0 & 1 & 0 \\ 0 & 0 & 2 & 0 \end{bmatrix}$$

and the matrices of the linear inequalities in (48) are defined according to the following relations (we refer to the HYSDEL notation, see [67])

$$\begin{aligned} \delta_1 = u_1 \leq 5 \quad & \text{if } \delta_1 \text{ then } x_1 \text{ else } x_1 + x_3 \\ \delta_2 = u_2 \leq 5 \quad & \text{if } \delta_2 \text{ then } 2x_2 \text{ else } -x_1 - x_2 \\ \delta_3 = u_3 \leq 5 \quad & \text{if } \delta_3 \text{ then } x_1 - x_3 \text{ else } -x_3 \\ \delta_4 = u_4 \leq 5 \quad & \text{if } \delta_4 \text{ then } -x_4 \text{ else } 2x_4 \end{aligned}$$

We aim at obtaining a reduced order system, that preserves the input/output behavior of (48). To this end, we apply the procedure described in Section 4.3.2 and obtain:

`1 State variables eliminated:` x_4

`2 Non-influential input variables found:` u_4, u_5

`2 Auxiliary variables eliminated:` δ_4, z_4

so that the resulting system is described by:

$$A_{\text{red}} = \begin{bmatrix} 1 & 0 & -2 \\ 0 & 1 & 0 \\ -3 & 0 & -4 \end{bmatrix}, \quad C_{\text{red}} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \end{bmatrix},$$

$$B_{u,\text{red}} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 2 \\ 1 & 3 & 0 \end{bmatrix}, \quad B_{z,\text{red}} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

$$D_{u,\text{red}} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad D_{z,\text{red}} = \begin{bmatrix} -1 & 0 & 1 \\ 0 & 0 & 2 \end{bmatrix}$$

with matrices $E_{x,\text{red}}$, $E_{u,\text{red}}$, $E_{\delta,\text{red}}$, $E_{z,\text{red}}$, $E_{aff,\text{red}}$ defined by:

$$\begin{aligned} \delta_1 = u_1 \leq 5 \qquad & \text{if} \quad \delta_1 \quad \text{then } x_1 \text{ else } x_1 + x_3 \\ \delta_2 = u_2 \leq 5 \qquad & \text{if} \quad \delta_2 \quad \text{then } 2x_2 \text{ else } -x_1 - x_2 \\ \delta_3 = u_3 \leq 5 \qquad & \text{if} \quad \delta_3 \quad \text{then } x_1 - x_3 \text{ else } -x_3. \end{aligned}$$

Note that u_4, and u_5 were found to be non-influential inputs. This means that they will not affect the output behavior and hence can be removed.

We now illustrate some results of the modes merging algorithm in Subsubsection 4.3.3.

Consider the following PWA system:

$$\begin{bmatrix} x_1^+ \\ x_2^+ \end{bmatrix} = \begin{cases} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u, & \begin{bmatrix} x_1 \\ x_2 \\ u \end{bmatrix} \in \cup_{i=1}^{6} \mathcal{A}_i \\ \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 2 \end{bmatrix} u, & \begin{bmatrix} x_1 \\ x_2 \\ u \end{bmatrix} \in \mathcal{A}_7 \cup \mathcal{A}_8, \end{cases} \tag{49}$$

where $(x_1,\ x_2) \in \mathcal{X} = [-100,\ 100]^2$, $u \in \mathcal{U} = [-10,\ 10]$ and the sets $\mathcal{A}_i$, $i = 1, \ldots, 8$ are the elements of the partition of the space $\mathcal{X} \times \mathcal{U}$ defined by the following inequalities (see Figure 18):

$$x_1 \leq 0, \quad x_2 - x_1 \leq 2, \quad u \leq 2. \tag{50}$$

We now exploit the procedure described in Section 4.3.3 to merge the modes associated to the same dynamics. We group the modes in the two sets $\{\mathcal{A}_1,\ \mathcal{A}_2,\ \mathcal{A}_3,\ \mathcal{A}_4,\ \mathcal{A}_5,\ \mathcal{A}_6\}$ and

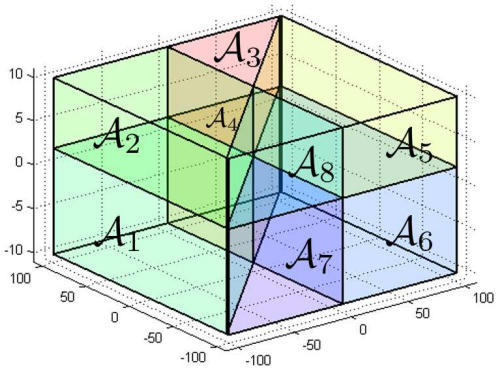

Figure 18: Partition of space $\mathcal{X} \times \mathcal{U}$ defined by inequalities (50)

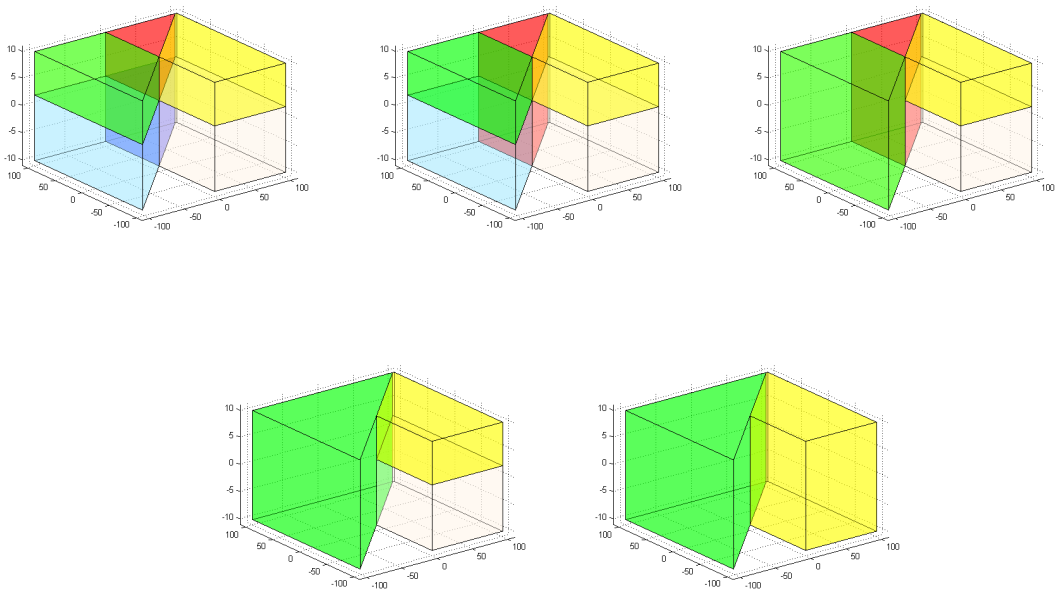

Figure 19: Merging of modes $\mathcal{A}_1$, $\mathcal{A}_2$, $\mathcal{A}_3$, $\mathcal{A}_4$, $\mathcal{A}_5$, $\mathcal{A}_6$ associated to the first dynamics

$\{\mathcal{A}_7, \mathcal{A}_8\}$ and perform the merging on each of them. The results are depicted in Figure 18 and Figure 20.

Starting from a total of 8 modes we have obtained a reduced system with just 3 modes. The results have been obtained by applying a greedy exploration, that, in this case, perform as well as the exhaustive exploration. The case of the greedy exploration performing worse than the exhaustive one is shown in Figure 21, where we associated mode $\mathcal{A}_8$ to the first dynamics and mode $\mathcal{A}_6$ to the second dynamics. In this case the reduction returns a total of 5 modes.

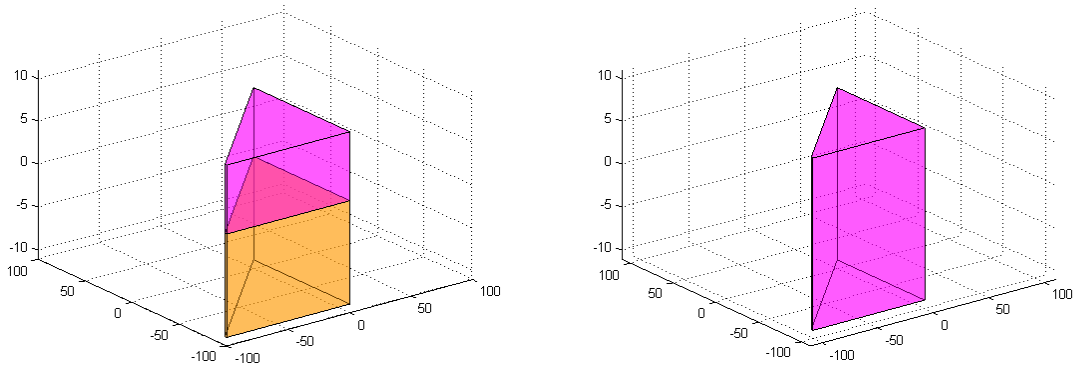

Figure 20: Merging of modes $\mathcal{A}_7$, $\mathcal{A}_8$ associated to the second dynamics

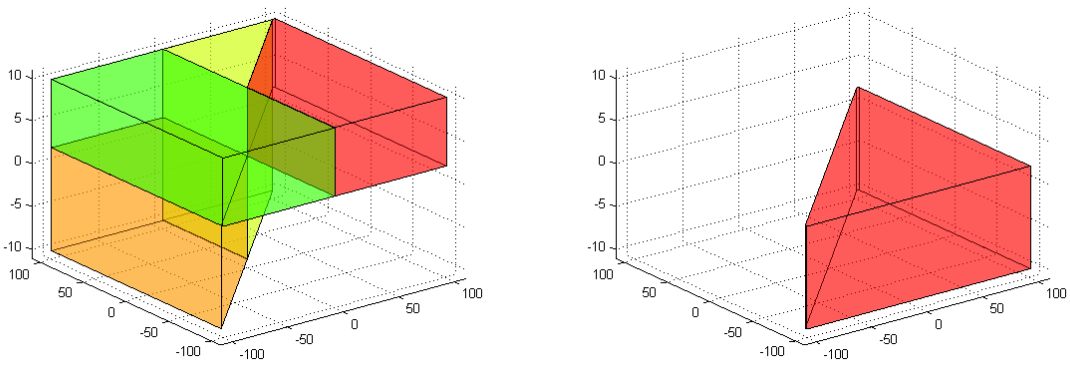

Figure 21: Greedy exploration: the modes associated to the first dynamics (on the left) are merged in a total of 4 regions

4.3.5 Conclusions

In this section, we introduced an approach to model reduction of discrete time hybrid systems that preserves the input/output behavior. The proposed approach rests on a sufficient condition for the unobservable part of the affine dynamics entering the MLD model description to remain unobservable when accounting for the hybrid system evolution. Our aim was reproducing the input/ouput behavior, irrespectively of the system initialization. If the system initialization were exactly known or confined to some region, the model reduction procedure could account for this additional information and, possibly, further reduce the model. This would be the case for linear systems. In the MLD systems framework, some combinations of the δ auxiliary variable that define the switching between modes in the PWA form might be pruned out because not admissible, which will possibly simplify the inequalities in the MLD representation. This requires further work.

5 Model Approximation for Control

This section addresses the question of how model approximations of the general CPS model from Sec. 2 can be obtained such that, in particular, optimization-based techniques of online control become applicable in real-time. This is one key objective in WP 2 of the project, where predominantly variants of model-predictive control are investigated. Typically such techniques are formulated in discrete time, and time-discretization is often used also to restrict the optimization to finite-dimensional problems. The techniques to be proposed in Sec. 5.1 and Sec. 5.2 follows this line, and they consider CPS, in which for any subsystem the effects of other subsystems are cast into time-varying constraints. This is possible, if the behavior of interacting subsystems was communicated, and is transformed (e.g.) into a subset of the state space in which the local controller can plan the state trajectory of the subsystem plant. The optimization can then be carried out in a decentralized fashion, if the control objectives are decoupled, or in a cooperative scheme, if the results of local optimization problems are iteratively exchanged. Another aspect of model approximation to be covered in Sec. 5.1 is that of approximating nonlinear dynamics by on-the-fly linearization. Using linearized models in computing controls is justified by the fact that UnCoVerCPS proposes techniques in which verification complements control – the following parts illustrate the modeling procedures for control with reference to two use cases of the project (robot control and autonomous driving).

5.1 Approximations To Reach-Avoid Problems in Human-Robot-Interaction

With respect to the CPS model of Sec. 5.2, this section focuses on one CPS subsystem in which environments uncertainties are modeled by establishing constraints, determined such that the state is restricted to regions in which the uncertainties may not cause unsafe behavior. We consider time-varying state space constraints, as well as time-varying goal states for control. The state constraints can be immediately referred to the invariant functions $I_z(t)$ on Definition 1 (Sec. 2) of the continuous-time subsystem of a CPS. The continuous dynamics of the subsystem (for which the subsystem index is omitted for ease of notation) is first modeled by a set of nonlinear differential equations:

$$\dot{x}(t) = f(x(t), u(t)), \tag{51}$$

with time $t \in \mathbb{R}$, state vector $x(t) \in \mathbb{R}^{n_x}$, and input vector $u(t) \in \mathcal{U} \subseteq \mathbb{R}^{n_u}$. Time discretization with zero-order-hold (ZOH) approximation yields a discrete-time system:

$$x_{k+j+1|k} = f(x_{k+j|k}, u_{k+j|k}), \tag{52}$$

with time indices $j \in \mathcal{J}_{\mathcal{N}} := \{0, ..., N\}$, $k \in \mathbb{N}_0$, state vector $x_{k+j|k} \in \mathbb{R}^{n_x}$, and input vector $u_{k+j|k} \in \mathcal{U} \subseteq \mathbb{R}^{n_u}, \forall j \in \mathcal{J}_{\mathcal{N}}$. The index of $x_{k+j|k}$ denotes the state x at point of time $k+j$ determined by the information available at time k. The goal state is denoted by $x^0_{f|k}$, $u^0_{f|k}$, and may, as mentioned above, change over time k.

A typical reach-avoid problem for such a system is to control the nonlinear system from teh current state x_s into a goal state, while avoiding collisions with a polytopic moving obstacle $\mathcal{P}_{x,k+j}$ (modeling the uncertain environment of the subsystem, e.g. the space occupied by another subsystem). The obstacle position is assumed to be known over the prediction horizon $j \in \mathcal{J} := \{0, ..., H\}$, which is justified if it is obtained from communicated information, or from estimation using an appropriate model. The goal is then to control the system by minimizing given cost functional while satisying all relevant state and input constraints. This problem can be formulated as:

$$\begin{aligned} \min_{x_{k+j|k},\ u_{k+j|k}} & (x_{k+H|k} - x_{f|k})^T Q_{end} (x_{k+H|k} - x_{f|k}) + \dots \\ & \dots + \sum_{j=0}^{H-1} (x_{k+j|k} - x_{f|k})^T Q (x_{k+j|k} - x_{f|k}) + (u_{k+j|k} - u_{f|k})^T R (u_{k+j|k} - u_{f|k}) \\ \text{s.t.} \quad & x_{k+j+1} = f(x_{k+j|k}, u_{k+j|k}),\ u_{k+j|k} \in \mathcal{U},\ x_{k+j|k} \notin \mathcal{P}_{x,k+j}. \forall j \in \mathcal{J},\ x_{k|k} = x_s, \end{aligned} \tag{53}$$

The solution of this problem yields the optimized trajectories $\hat{x}^*_{:|k}, \hat{u}^*_{:|k}$ for the prediction horizon H. The considered performance function is quadratic for positive-definite weighting matrices $Q \in \mathbb{R}^{n_x \times n_x}$, and $R \in \mathbb{R}^{n_u \times n_u}$. In model predictive control (MPC, see e.g. [33]), the solution to the control problem (53) is implemented in a receding horizon fashion, where only in the first entry $u^*_{k|k}$ of the input sequence is applied to the system, and the calculation then repeated for incremented k.

As an application example, consider a CPS modeling human-robot cooperation in an industrial process, i.e. human worker and a robotic manipulator working in the same space. The task is to control the manipulator such that collisions with the human worker are excluded while, at the same time, the robot maintains a close-to-optimal operation in accomplishing its task.

This problem can be addressed by model predictive control (MPC) using mixed integer programming (MIP), as shown already in [17, 18]. A problem of these techniques is that the integer variables needed to encode the collision avoidance together with the nonlinear dynamics typically slows down may the solution of the optimization problems considerably – this is the very motivation of the developments in the tasks 2.2/2.3 on fast online-optimizing control. As a preparing step, the space occupied by the human worker is over-approximated

by convex polytopes (see [19] for a technique to accomplish this step):

$$\mathcal{P}_{x,k+j} := \{x_{k+j} |\ C_{k+j} x_{k+j} \leq d_{k+j}\} \subseteq \mathbb{R}^{n_x}, \tag{54}$$

with $C \in \mathbb{R}^{c \times n_x}$, and $d \in \mathbb{R}^c$. Furthermore, the nonlinear dynamics (51) of the robot system has to be simplified. This is done by time-varying linearization around the current state x_k.

5.1.1 Robot Modeling and Abstraction

In this part, a simple model of a 2-D robotic manipulator is introduced, and the steps of model transformation are illustrated. The obtained approximation is tailored to be used for validation of a fast online control method based on homotopies, as described in deliverable D2.2.

The model consdierd here is a robotic manipulator with two joints and two links, in a 2-D Cartesian space $x(t) = (x_1(t), x_2(t))^T \in \mathbb{R}^2$, see Fig. 22. The manipulator configuration is described by the angles $\theta = (\theta_1(t), \theta_2(t))^T \in \mathbb{R}^2$, and the angular velocities $\dot{\theta}(t) = (\dot{\theta}_1(t), \dot{\theta}_2(t))^T \in \mathbb{R}^2$ for time t. The combined vector is denoted by $\Theta(t) = (\theta_1(t), \dot{\theta}_1(t), \theta_2(t), \dot{\theta}_2(t))^T$. The two links have the lengths $l_1,\ l_2$. The masses $m_1,\ m_2$ are assumed as mass points which are centered at the end of the first and second links, hence on positions $r_i(t) \in \mathbb{R}^2$, $i = \{1, 2\}$. The end effector $r_2(t)$ of the manipulator is also described as the tool center point (TCP). Furthermore, friction depending on the angular velocities is modeled by the coefficients c_1, c_2, and the gravitational constant is denoted by g. The torques applied to the joints are $f(t) = (\tau_1(t), \tau_2(t))^T$.

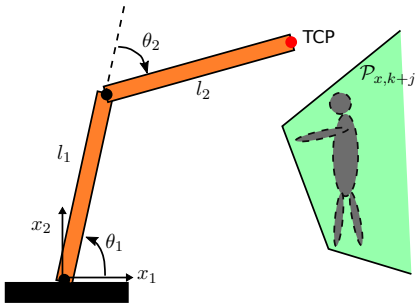

Figure 22: Illustration of a robotic manipulator with two links and joints working in the same area with a human. The robot is operating in a two dimensional Cartesian space $x = (x_1, x_2)$. The safety region of the human worker is approximated by a convex time-varying polytope $\mathcal{P}_{x,k+j}$

By means of Lagrange mechanics, the nonlinear differential equation can be derived as:

$$\frac{\partial L(t)}{\partial \theta_i(t)} - \frac{d}{dt}\frac{\partial L(t)}{\partial \dot{\theta}_i(t)} = -\tau_i, \quad i = \{1, 2\}, \tag{55}$$

where the Lagrangian $L(t)$ for a system of particles is:

$$L(t) = T_{kin}(t) - V_{pot}(t), \tag{56}$$

with $T_{kin}(t)$ as the total kinetic energy:

$$T_{kin}(t) = \sum_{i=1}^{2} \frac{1}{2} m_i \dot{r}_i(t)^2, \tag{57}$$

and the potential energy:

$$V_{pot}(t) = \sum_{i=1}^{2} \frac{1}{2} m_i g \dot{r}_i(t)^2. \tag{58}$$

To determine the two energy equations, the positions $r_i(t)$ of each mass point m_i are modeled as functions of the generalized coordinates $\theta_i(t)$ by means of the Denavit-Hartenberg convention. There, a rotation (by an angle $\theta_i(t)$) around the $x_{3,[i-1]}$-axis from the x_3-axis of the $i-1$ to the i-th coordinate system is given by:

$$Rot(x_{3,[i-1]}, \theta_i(t)) = \begin{pmatrix} \cos\,\theta_i(t) & -\sin\theta_i(t) & 0 & 0 \\ \sin\,\theta_i(t) & \cos\,\theta_i(t) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \tag{59}$$

A translation of length l_i along the $x_{1,[i]}$-axis of coordinate system i is modeled by:

$$Trans(x_{1,[i]}, l_i) = \begin{pmatrix} 1 & 0 & 0 & l_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \tag{60}$$

With these transformations, the position $r_i(t)$ of mass m_i can be obtained from the matrix:

$$Q_i = \left(\begin{array}{ccc|c} & O & & r_i(t) \\ & & & 0 \\ \hline 0 & 0 & 1 & 1 \end{array}\right), \tag{61}$$

where $O \in \mathbb{R}^{3\times 3}$ is a submatrix describing the rotation. Therefore, the position $r_1(\theta_1(t))$ is

obtained from:

$$Q_1 = Rot(x_{3,[0]}, \theta_1(t)) \cdot Trans(x_{1,[1]}, l_1) = \left(\begin{array}{ccc|c} & O & & \begin{matrix} l_1 \cos\theta_1(t) \\ l_1 \sin\theta_1(t) \\ 0 \end{matrix} \\ \hline 0 & 0 & 1 & 1 \end{array}\right) \tag{62}$$

$$\Rightarrow r_1(\theta_1(t)) = \begin{pmatrix} l_1 \cos\theta_1(t) \\ l_1 \sin\theta_1(t) \end{pmatrix}, \tag{63}$$

and position $r_2(\theta_1(t), \theta_2(t))$ of mass m_2:

$$Q_2 = Rot(x_{3,[0]}, \theta_1(t)) \cdot Trans(x_{1,[1]}, l_1) \cdot Rot(x_{3,[1]}, \theta_2(t)) \cdot Trans(x_{1,[2]}, l_2) = \left(\begin{array}{ccc|c} & O & & \begin{matrix} l_1 \cos\theta_1(t) + l_2 \cos(\theta_1(t) + \theta_2(t)) \\ l_1 \sin\theta_1(t) + l_2 \sin(\theta_1(t) + \theta_2(t)) \\ 0 \end{matrix} \\ \hline 0 & 0 & 1 & 1 \end{array}\right) \tag{64}$$

$$\Rightarrow r_2(\theta_1(t), \theta_2(t)) = \begin{pmatrix} l_1 \cos\theta_1(t) + l_2 \cos(\theta_1(t) + \theta_2(t)) \\ l_1 \sin\theta_1(t) + l_2 \sin(\theta_1(t) + \theta_2(t)) \end{pmatrix}. \tag{65}$$

With these equations, the evaluation of (55) yields a system of two differential equations, each of order two, describing the manipulator dynamics:

$$\begin{aligned} \tau_1(t) - c_1\dot{\theta}_1 &= (l_1^2 m_1 + l_1^2 m_2 + l_2^2 m_2 + 2 l_1 l_2 m_2 \cos\theta_2(t)) \cdot \ddot{\theta}_1(t) + (l_2^2 m_2 + l_1 l_2 m_2 \cos\theta_2(t)) \cdot \ddot{\theta}_2(t) \\ &\quad + g \cdot (l_2 m_2 \cos(\theta_1(t) + \theta_2(t)) + l_1 m_1 \cos\theta_1(t) + l_1 m_2 \cos\theta_1(t)) \end{aligned} \tag{66}$$

$$\begin{aligned} \tau_2(t) - c_2\dot{\theta}_2 &= (l_2^2 m_2 + l_2 m_2 l_1 \cos\theta_2(t)) \cdot \ddot{\theta}_1(t) + l_2^2 m_2 \ddot{\theta}_1(t) \\ &\quad + l_2 m_2 l_1 \sin\theta_2(t) \, (\dot{\theta}_1(t))^2 + l_2 m_2 \dot{\theta}_2(t) l_1 \sin\theta_2(t) \, \dot{\theta}_1(t) + g \cdot (l_2 m_2 \cos(\theta_1(t) + \theta_2(t))). \end{aligned} \tag{67}$$

The nonlinear manipulator dynamics has the form:

$$M(\theta(t))\ddot{\theta}(t) + C(\theta(t), \dot{\theta}(t))\dot{\theta}(t) + G(\theta(t)) - f = 0, \tag{68}$$

with $M(\theta(t))$ representing the inertial forces due to acceleration of the joints, $C(\theta(t), \dot{\theta}(t))$ modeling the Coriolis and centrifugal forces, $G(\theta(t))$ the gravitational forces, and $f = (\tau_1, \tau_1)^T \in \mathbb{R}^2$ are the applied torques.

The next step is to transform the second order system of differential equations into a system of four equations, each of order one, by substituting the state vector $\Theta(t)$ by $Z(t) =$

$(z_1(t), z_2(t), z_3(t), z_4(t))^T \in \mathbb{R}^4$:

$$\begin{pmatrix} \theta_1(t) \\ \dot{\theta}_1(t) \\ \theta_2(t) \\ \dot{\theta}_2(t) \end{pmatrix} = \begin{pmatrix} z_1(t) \\ z_2(t) \\ z_3(t) \\ z_4(t) \end{pmatrix}, \text{ and } \begin{pmatrix} \dot{\theta}_1(t) \\ \ddot{\theta}_1(t) \\ \dot{\theta}_2(t) \\ \ddot{\theta}_2(t) \end{pmatrix} = \begin{pmatrix} \dot{z}_1(t) \\ \dot{z}_2(t) \\ \dot{z}_3(t) \\ \dot{z}_4(t) \end{pmatrix}. \tag{69}$$

With this substitution, the nonlinear differential equations (68) become:

$$\begin{pmatrix} \dot{z}_1(t) \\ \dot{z}_3(t) \\ \dot{z}_2(t) \\ \dot{z}_4(t) \end{pmatrix} = \begin{pmatrix} z_2(t) \\ z_4(t) \\ (M(z_1(t), z_3(t)))^{-1}[f - C(Z(t)) \begin{pmatrix} z_2(t) \\ z_4(t) \end{pmatrix} - G(z_1(t), z_3(t))] \end{pmatrix} \tag{70}$$

Linearization of this dynamics by a first order Taylor expansion around the current state vector $\bar{Z}$ and input vector $\bar{f}$, followed by a zero-order hold (ZOH) discretization with step dT leads to:

$$\begin{aligned} Z_{k+j+1|k} &= A_{Z,:|k} Z_{k+j|k} + B_{Z,:|k} f_{k+j|k} \\ &+ \underbrace{g(\bar{Z}, \bar{f}) - A_{Z,:|k} \bar{Z} - B_{Z,:|k} \bar{f}}_{R_{Z,:|k}}. \end{aligned} \tag{71}$$

Here, $A_{Z,:|k}$, $B_{Z,:|k}$, and $R_{Z,:|k}$ are matrices of the configuration dynamics denoted by the index Z, determined at time k, and a prediction time denoted by $:|k$. Since the linearized dynamics (71) is defined in the configuration space, but the obstacle $\mathcal{P}_{x,k+j}$ in the Cartesian space, there are basically two options to bring the system and the obstacle into the same space:

1. using forward kinematics by means of the Denavit-Hartenberg convention to describe a certain point on the manipulator (e.g. the TCP) by its Cartesian states: $\Psi(t) = (x_1(t), \dot{x}_1(t), x_2(t), \dot{x}_2(t))^T$
2. mapping the obstacle into the configuration space by a dicretization-based mapping.

From [50], it is known that the nonlinear mapping of the obstacle into the configuration space is very time demanding, let alone for predicted obstacles. The non-convex shapes of the transformed obstacles do not allow for an intuitively planning procedure and, for the homotopy-based control method, not for an intuitively parameterization of so-called *base trajectories*. Therefore, rather than mapping the obstacle into the configuration space, the linearized dynamics (71) is transferred into the Cartesian space by forward kinematics of

the TCP. However, this procedure limits the obstacle-avoidance problem to the TCP (or respectively a single considered point on the manipulator).

To determine the TCP position $x(t)$ depending on $Z(t)$, the forward kinematics is given by the nonlinear function $r_2 : \mathbb{R}^4 \to \mathbb{R}^2$, and the TCP velocity $\dot{x}(t)$ by its time derivative:

$$x(t) = r_2(Z(t)),\ \dot{x}(t) = \frac{\partial r_2(Z(t))}{\partial t}. \tag{72}$$

Rearranging the two equations (72) into a combined vector $\Psi(t) = (x_1(t), \dot{x}_1(t), x_2(t), \dot{x}_2(t))^T$ leads to the nonlinear equation:

$$\Psi(t) = \Xi(Z(t)). \tag{73}$$

Evaluating (73) at discrete points $Z_{k+j|k}$ from (71), and by further linearizing the nonlinear function (73) around the current state vector $\bar{Z}_{k|k}$, leads to:

$$\begin{aligned}\Psi_{k+j|k} = &\underbrace{\left.\frac{\partial \Xi(Z_{k+j|k})}{\partial Z_{k+j|k}}\right|_{\bar{Z}_{k|k}}}_{\Phi_{:|k}} Z_{k+j|k} \\ &+ \underbrace{\Xi(\bar{Z}_{k|k}) - \left.\frac{\partial \Xi(Z_{k+j|k})}{\partial Z_{k+j|k}}\right|_{\bar{Z}_{k|k}} \bar{Z}_{k|k}}_{\Omega_{:|k}}.\end{aligned} \tag{74}$$

This equation means that the predicted Cartesian state vector $\Psi_{k+j|k}$ is determined from the configuration vector $Z_{k+j|k}$ with the matrices $\Phi_{:|k}$, and $\Omega_{:|k}$. By solving (74) for $Z_{k+j|k}$:

$$Z_{k+j|k} = (\Phi_{:|k})^{-1}\Psi_{k+j|k} - (\Phi_{:|k})^{-1}\Omega_{:|k}, \tag{75}$$

and insertion of (75) into (71), the linearized dynamics in the Cartesian space becomes finally:

$$\Psi_{k+j|k} = A_{\Psi,:|k}\Psi_{k+j|k} + B_{\Psi,:|k}f_{k+j|k} + R_{\Psi,:|k}, \tag{76}$$

with the similarity transformations:

$$A_{\Psi,:|k} = \Phi_{:|k}A_{Z,:|k}(\Phi_{:|k})^{-1}, \tag{77}$$

$$B_{\Psi,:|k} = \Phi_{:|k}B_{Z,:|k}, \tag{78}$$

$$R_{\Psi,:|k} = \Phi_{:|k}R_{Z,:|k} + \Omega_{:|k} - \Phi_{:|k}A_{Z,:|k}(\Phi_{:|k})^{-1}\Omega_{:|k}. \tag{79}$$

5.1.2 Conclusion

The derived Cartesian state space model describes the motion of the TCP, and can be used for the homotopy-based obstacle avoidance procedure. Since the model is linearized at each

time step, the properties of homotopies as described in Deliverable D2.2 apply. As typical in trajectory optimization, a high computational effort results from the number of inputs and/or state variables that have to be considered for reasons of obstacle avoidance over the prediction horizon. However, the homotopic control method reduces the number of variables by selecting a desired trajectory only from a set of homotopic trajectories. While these trajectories are parametrized by a low dimensional variable vector, the problem can be reduced to obtain real-time applicability.

One may argue that the use of linearized dynamics leads to only coarse predictions for states being far away from x_k (typically those states predicted to be attained for times $k + j|k$). However, since the linearization is adapted in any time-step k, the predictions for the considered example show only small deviations to the original nonlinear dynamics, when the online-control method is applied.

In addition, to guarantee that no collision between the links of the robot and the obstacle occur, the robotic manipulator can be approximated by a set of particles along each link of the robot, which likewise are considered in the homotopic control method, when selecting a collision-free homotopic trajectory. Papers on the homotopic control method with application to the robotic manipulator dynamics are in preparation, as well as papers on cooperative manufacturing for multiple robots.

5.2 Approximated Modeling of Automated Vehicles for Online Control

The case study of automated driving as in UnCoVerCPS constitutes an example of a CPS being composed of several subsystems (representing the automated vehicles) which evolve over time in interacting / cooperative manner. The number of vehicles involved in a scenario determines the information exchanged through the communication network and the number of restrictions to be considered for the driving plans of each vehicle. This requires suitable procedures for any vehicle to select the relevant information and to compute timely a suitable trajectory compliant to the current constraint. A basic cooperative objective is obviously that the local maneuvers of the single vehicles guarantee safety with respect to avoiding inter-vehicle collision. One possibility to achieve this is to over-approximate for any vehicle the region of possible positions over a future time span, and to exclude this region from the planning space available for the other vehicles. The following description presents a modeling concept as basis of vehicle planning techniques. First, an appropriate nonlinear model of the vehicle dynamics is provided, and then a collision avoidance scheme is presented to discuss different approximating models in a predictive planning algorithm.

In deliverable D5.1, a model of the vehicle dynamics for the case study of automated driving was already reported. The model structure was that of a bicycle model, which is quite frequently used in trajectory planning and vehicle control. The model in D5.1 was tailored to constant longitudinal velocity – here, we instead intend to use this velocity as degree of freedom and thus consider a slightly more general model as motivated by [58]. Consider the variables and notation as shown in Fig.23, where COG abbreviates 'center of gravity' (positioned in the vehicle center), a and b are the distances of the front and rear axes to the $COG[m]$, and m is the mass of the vehicle $[kg]$.

Assume that the state of the vehicle is modeled by the following six state variables:

- x_1: longitudinal velocity v_x $[m/s]$ in the local coordinate system,
- x_2: lateral velocity v_y $[m/s]$ in the local coordinate system,
- x_3: yaw rate $[rad/s]$,
- x_4, x_5: position in the world coordinate system,
- x_6: angle to the lateral axis α in the world coordinate system,

while the inputs are:

- u_1, u_2: front left/right tire torque $[ratio]$,
- u_3: steering angle $[rad]$ of the front tires.

Then, the vehicle dynamics in a world coordinate system can be described by the following state-space model:

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \\ \dot{x}_5 \\ \dot{x}_6 \end{pmatrix} = \begin{pmatrix} x_1 x_3 + \frac{1}{m}(C_x(u_1+u_2)cos(u_3) - 2C_y(u_2 - \frac{x_2+ax_3}{x_1})sin(u_3)) \\ -x_1 x_3 + \frac{1}{m}(C_x(u_1+u_2)sin(u_3) + 2C_y(u_3 - \frac{x_2+ax_3}{x_1})cos(u_3) + 2C_y\frac{bx_3-x_2}{x_1}) \\ J(a(C_x(u_1+u_1)sin(u_2)) + 2C_y(u_2 - \frac{x_2+ax_3}{x_1}cos(u_2)) - 2bC_y\frac{bx_3-x_2}{x_1}) \\ x_1 cos(x_6) + x_2 sin(x_6) \\ x_1 sin(x_6) - x_2 cos(x_6) \\ x_3 \end{pmatrix}. \tag{80}$$

Here, C_x and C_y represent the longitudinal and lateral tire stiffness, for the moment of inertia applies $J = \frac{1}{0.5(a+b)^2 m}$, and we assume the following parametrization: $m = 1700kg$, $a = b = 1.5m$, $C_x = 150000$ and $C_y = 40000$ in the rest of this chapter.

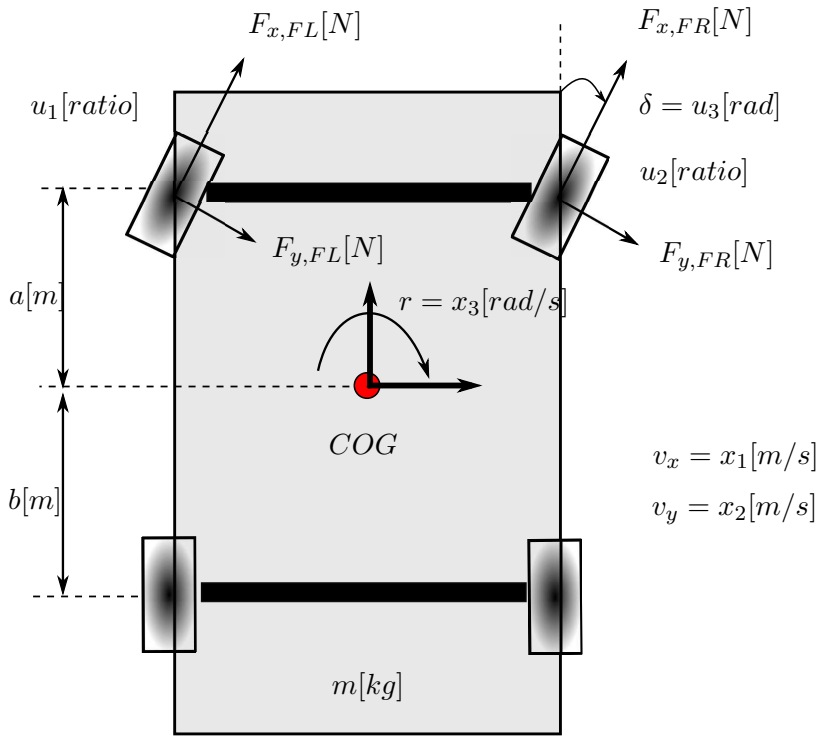

Figure 23: Scheme of a vehicle with relevant variables.

For control techniques involving online optimization (such as the schemes of distributed model predictive control investigated in WP2), the model is too complex for real-time solution. A natural work-around is to transfer the model to discrete time and to linearize it around the current state $x(t_k)$ at any discrete point of time t_k. Then, the question arises how well such a model can be used for online prediction and optimization – this point is studied in the following.

Let the model (80) first be discretized using a zero-order hold approximation of the input, and a sampling time $\delta(t) = t_{k+1} - t_k$ of 0.2 sec, which is suitable given the time constants of the model. If the resulting model of type:

$$x_{k+1} = A_{\bar{x}_k,\bar{u}_k} x_k + B_{\bar{x}_k,\bar{u}_k} u_k + R_{\bar{x}_k,\bar{u}_k} \tag{81}$$

with $k \in \mathbb{N}_0$ is subsequently linearized around the current state $\bar{x}_k$ and input $\bar{u}_k$, a model of the structure:

$$x_{k+1} = A_{\bar{x}_k,\bar{u}_k} x_k + B_{\bar{x}_k,\bar{u}_k} u_k + R_{\bar{x}_k,\bar{u}_k} \tag{82}$$

is obtained, where the matrices $A_{\bar{x}_k,\bar{u}_k}$, $B_{\bar{x}_k,\bar{u}_k}$, and $R_{\bar{x}_k,\bar{u}_k}$ are specific for the linearization point and the discretization time.

Let us now focus on a reach-avoid problem, in which the autonomously driving vehicle has to reach a goal position from the current position while circumventing an obstacle, as

sketched in Fig.24.

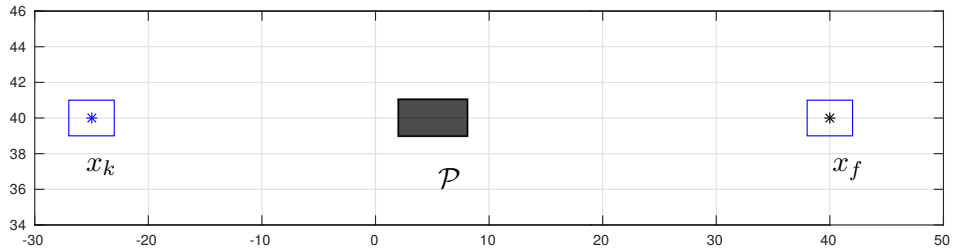

Figure 24: The dark rectangle represents a convex obstacle in between the current position x_k and a goal position x_f. The blue rectangle represents the vehicle.

For the numeric study, let the current position be given as $x_k = [3, 0, 0, 25, 40, 0]^T$, the goal position as $x_f = [0, 0, 0, 40, 40, 0]^T$, and the obstacle $\mathcal{P}$ be modeled by:

$$\mathcal{P} := \left\{ x \middle| \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 \end{pmatrix} x \leq \begin{pmatrix} 8 \\ -2 \\ 41 \\ -39 \end{pmatrix} \right\} \tag{83}$$

Furthermore, let the restriction of the vehicle to the region of the street be represented by limiting the vertical position to the range: $x_5 \in [34, 46]$. To consider restrictions of the longitudinal velocity, the lateral velocity, the yaw rate as well as the slip and steering angle, the following constraints are added:

$$x_1 \in [-5, 15],\ x_2 \in [-5, 5],\ x_3 \in [-0.5, 0.5],\ u_1 = u_2 \in [-0.002, 0.002],\ u_3 \in [-0.2, 0.2]. \tag{84}$$

Now, by adopting $\bar{x_k} = x_k$ and $\bar{u}_k = [0, 0, 0]^T$, exemplarily the following parametrization of (82) is obtained:

$$x_{k+1} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.0019 & -0.0079 & 0 & 0 & 0 \\ 0 & 0 & 0.0434 & 0 & 0 & 0 \\ 0.2 & 0 & 0 & 1 & 0 & 0 \\ 0 & -0.0318 & 0.0322 & 0 & 1 & 0.6 \\ 0 & 0 & 0.061 & 0 & 0 & 1 \end{pmatrix} x_k + \begin{pmatrix} 17.6470 & 17.6470 & 0 \\ 0 & 0 & 1.4097 \\ 0 & 0 & 0.9566 \\ 1.7647 & 1.7647 & 0 \\ 0 & 0 & -0.2084 \\ 0 & 0 & 0.1390 \end{pmatrix} u_k \tag{85}$$

With reference to the general model definition of CPS provided in Sec. 2, this setting refers to a subsystem with discrete-time time-varying linearized dynamics, and a state constraint

imposed by the condition $x_k \notin \mathcal{P}$ for any k. The complement of $\mathcal{P}$ is identical to the space encoded by an invariant function I_z in Def. 3 (Sec.2). While, for simplicity of notation, the obstacle was here introduced as being static, the extension to time-varying state constraints (stemming from the region occupied by a moving other vehicle) is straightforward.

To solve the reach-avoid problem online by a technique like model-predictive control (as investigated in Task 2.2), the following a cost functional Ω is defined:

$$\Omega = \sum_{j=0}^{H-1} (x_{k+j+1|k} - x_f)^T Q (x_{k+j+1|k} - x_f) + (u_{k+j|k})^T R (u_{k+j|k}), \tag{86}$$

with $Q = Q^{\mathrm{T}} \geq 0$ and $R = R^{\mathrm{T}} > 0$.

The minimization of Ω is complemented by the constraints listed above, as well as by collision avoidance constraints. These can be formulated by linear inequalities which are enforced by binary variables implying that the vehicle is on the safe side of hyperplanes determining the boundary of the space occupied by the obstacle. The result is an optimization problem of the type MIQP (Mixed Integer Quadratic Programming), for which the solution (if existing) leads to a feasible, collision-free vehicle trajectory, see Fig.25. The corresponding input signal is shown in Fig.26.

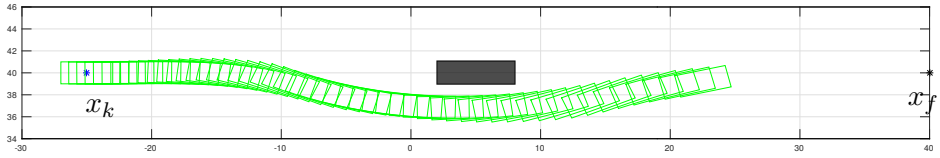

Figure 25: Trajectory planned at time t_k over a prediction horizon of $H = 50$ steps.

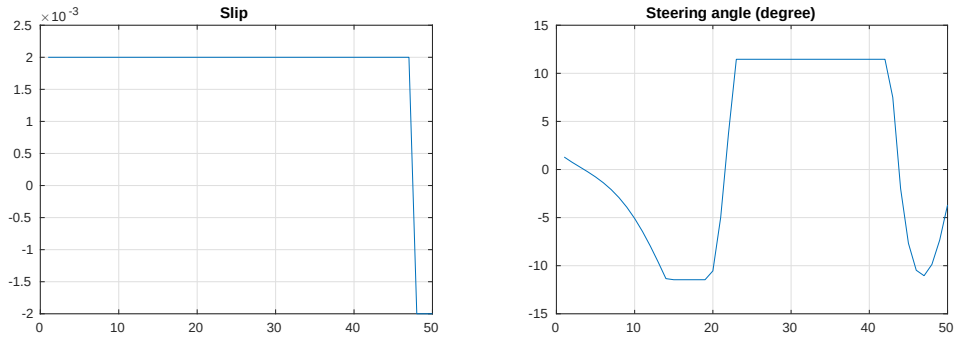

Figure 26: Input trajectory for the state trajectory shown in Fig. 25: the left figure shows the tire forces ($u_1 = u_2$), while the right figure illustrates the steering angle.

When applying the same input trajectory to the original nonlinear dynamics (80), it becomes obvious from Fig.s27 and 28 that the approximation error remains relatively small

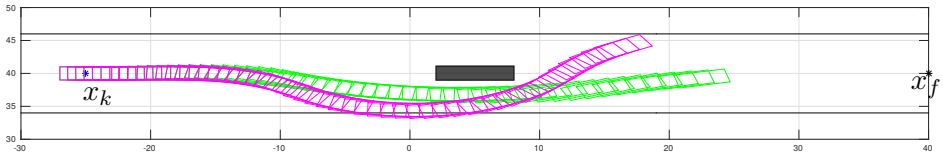

Figure 27: Comparison of vehicle trajectory planned at time t_k for the approximating linear (green) and the original nonlinear dynamics (magenta).

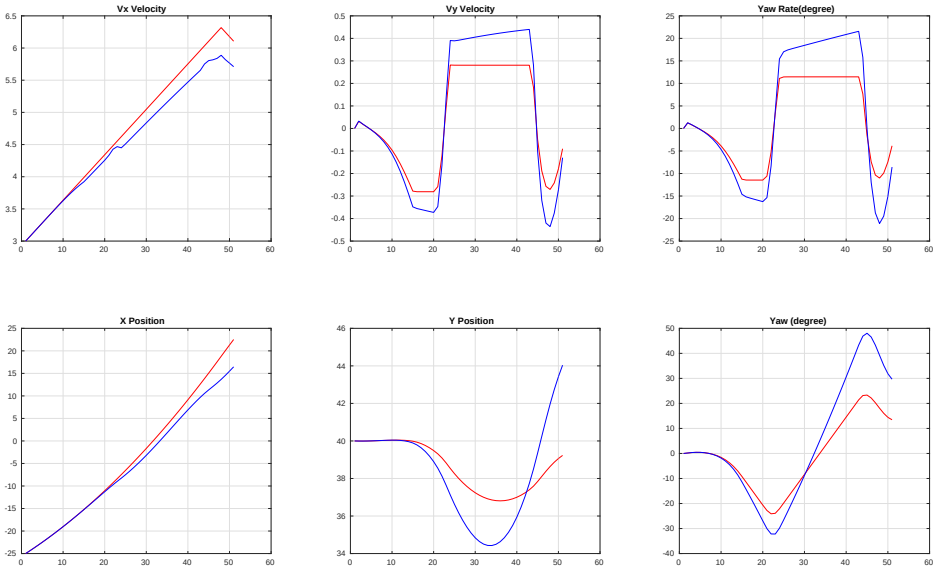

Figure 28: Comparison of all states for the cases referenced in Fig. 27: planned state trajectory for the linear dynamics (red), and the one obtained for the nonlinear dynamics (blue) with the same input trajectory.

for the first 20 steps, but increases afterwards. However, if a repeated optimization (planning) is applied in any time t_k (as in model-predictive control), the the deviations for later points of time within the prediction horizon are compensated, and the model accuracy can be rated as sufficiently good for development and application of the MPC techniques for CPS, as described in deliverable D2.2. This can be seen in Fig. 29, showing the solution for an MPC scheme using linearization in any t_k and a prediction horizon of $H = 15$, such that the goal state x_f is reached in 66 steps withou collision.

To further improve the model accuracy, in particular for larger horizons, the use of hybrid dynamics is a suitable extension. Especially, if the vehicle has to accomplish one or more swerves within one prediction, the approximation error may significantly increase with respect to changes of x_3 (yaw rate) and x_6 (angle to the lateral axis in the world coordinate

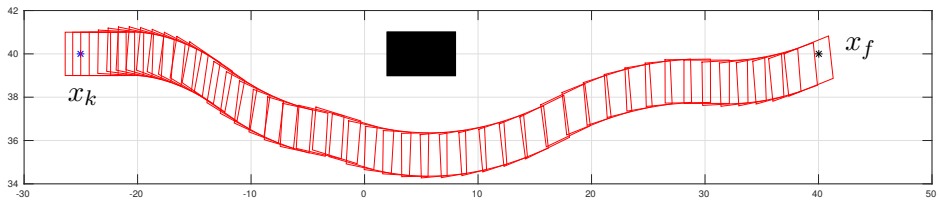

Figure 29: Vehicle trajectory obtained by using MPC with a prediction horizon of $H = 15$.

system). Fig.30 shows a case where the initial position of the vehicle is selected near to an obstacle (compare to the example in Fig.27, which means a swerve should be accomplished immediately): the optimization result obtained is unsuitable for the nonlinear model (as it leads to collision), while the computation time is relatively high.

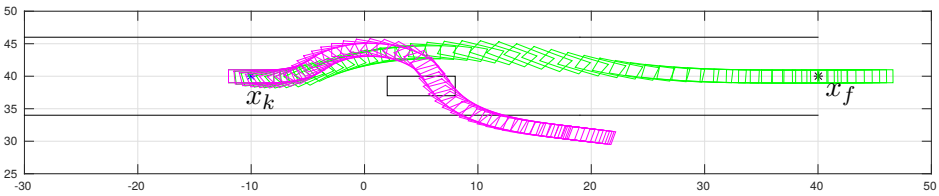

Figure 30: If the lateral initial position is selected to be -10 instead of -25 (the value in Fig.27), while the other settings are kept constant, a larger approximation error between the trajectories of the linear and nonlinear model are obtained.

Note that if the vehicle is driving straight along its longitudinal direction in the local coordinate system (i.e. $x_2 = 0$, $x_3 = 0$ and $u_3 = 0$), then the state-space model takes the following from:

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \\ \dot{x}_5 \\ \dot{x}_6 \end{pmatrix} = \begin{pmatrix} \frac{C_x}{m}u_1 + \frac{C_x}{m}u_2 \\ 0 \\ 0 \\ x_1 cos(x_6) + x_2 sin(x_6) \\ x_1 sin(x_6) - x_2 cos(x_6) \\ 0 \end{pmatrix}, \tag{87}$$

which can be further rewritten to:

$$\begin{pmatrix}\dot{x}_1\\ \dot{x}_2\\ \dot{x}_3\\ \dot{x}_4\\ \dot{x}_5\\ \dot{x}_6\end{pmatrix} = \begin{pmatrix}0 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 0\\ cos(x_6) & sin(x_6) & 0 & 0 & 0 & 0\\ sin(x_6) & -cos(x_6) & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 0\end{pmatrix}\begin{pmatrix}x_1\\ x_2\\ x_3\\ x_4\\ x_5\\ x_6\end{pmatrix} + \begin{pmatrix}\frac{C_x}{m} & \frac{C_x}{m} & 0\\ 0 & 0 & 0\\ 0 & 0 & 0\\ 0 & 0 & 0\\ 0 & 0 & 0\\ 0 & 0 & 0\end{pmatrix}\begin{pmatrix}u_1\\ u_2\\ u_3\end{pmatrix} \tag{88}$$

Now, if the angle to the lateral axis (x_6) selected to be inside a finite set of constant values $\mathcal{V} = \{v_1, v_2, \cdots, v_{n_v}\}$ instead of being a real variable (in $[0, 2\Pi]$), then, the state-space model (88) is linear and time-invariant for any $x_6 \in \mathcal{V}$. After eliminating the constant states and inputs in (88), the original state-space model (80) can be approximated by using the following hybrid dynamics:

$$\begin{pmatrix}\dot{x}_1\\ \dot{x}_2\\ \dot{x}_4\\ \dot{x}_5\end{pmatrix} = \begin{pmatrix}0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0\\ cos(x_6) & sin(x_6) & 0 & 0\\ sin(x_6) & -cos(x_6) & 0 & 0\end{pmatrix}\begin{pmatrix}x_1\\ x_2\\ x_4\\ x_5\end{pmatrix} + \begin{pmatrix}\frac{C_x}{m} & \frac{C_x}{m}\\ 0 & 0\\ 0 & 0\\ 0 & 0\end{pmatrix}\begin{pmatrix}u_1\\ u_2\end{pmatrix},\ x_6 \in \mathcal{V} \tag{89}$$

The model (89) is actually describing a straight-driving behavior of the vehicle with different driving directions. Clearly, this approach omits the state evolution during a change of the driving direction. Motivated by the work in [7]), one can determine a controller K_{v_i,v_j}, $\forall v_i, v_j \in \mathcal{V}$ and a set of states X_{v_i}, X_{v_j}, such that applies: if $x_{k+j|k} \in X_{v_i}$, then through applying the controller K_{v_i,v_j}, a change of the driving direction from v_i to v_j within $H_{v_i,v_j} > 0$ steps can be realized, and the constraint $x_{k+j+H_{v_i,v_j}|k} \in X_{v_j}$ is satisfied. With the determination of the controller K_{v_i,v_j}, the time step $H_{v_i,v_j} > 0$, and setting X_{v_i} and X_{v_j}, $\forall v_i, v_j \in \mathcal{V}$, we can obtain the following result by solving the same problem as was done in Fig. 30. (The set $\mathcal{V}$ is selected to be $\mathcal{V} = \{0, \frac{\Pi}{8}, -\frac{\Pi}{8}\}$).

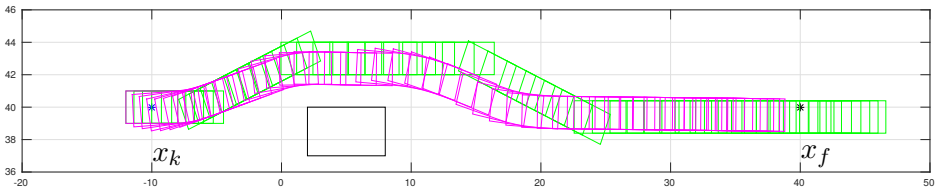

Figure 31: Comparison between the planned trajectory (in green) by using model (89), and the trajectory generated for the nonlinear model (magenta).

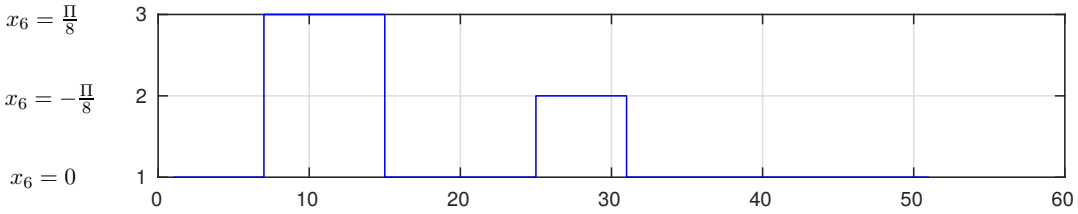

Figure 32: Sequence of the events in x_6 for all predicted steps, where '1' represents '$x_6 = 0$', '2' represents '$x_6 = -\frac{\Pi}{8}$' and '3' represents '$x_6 = \frac{\Pi}{8}$'.

Obviously, the new model results in a much smaller approximation error than the one in Fig.30. Therefore, with the help of (89), much better plans can be computed based on more accurate predictions. As the optimization of hybrid systems usually involves binary variables (what typically increases the computation time significantly), different methods have been developed for this case and are described in Deliverable 2.2.

5.2.1 Conclusions

Starting from a four-wheel vehicle dynamics in state-space realization, we have discussed different model approximations in form of discrete-time and time-varying linearizations with respect to the suitably for MPC schemes to solve reach-avoid problem for vehicles online. The numerical studies show that:

- the use of linearized discrete-time approximations in an optimization-based scheme is mandatory to obtain the optimization results timely,
- the deviations between the results for the original and the approximated model are largely compensated by the adapted linearization and the repetitive optimization as used in MPC,
- larger deviations for particular driving scenarios are avoided by using a hybrid dynamics which includes tailored linearizations for different driving modes (here modeled by selected discrete values of the state variable x_6).

Note that the modeling of one vehicle as described before extends straightforwardly to the case of groups of vehicles: Let the obstacles (as the one denoted by $\mathcal{P}$) refer to the predicted and communicated (or estimated) trajectories of other interacting vehicles. Then the complement of this occupied space defines the free space on which the vehicle can plan its own trajectory. To ensure convergence of the optimization, the free space should be defined by invariant functions $I^i_{z^i}(t_k)$ (see Def. 3 in Sec. 2) which determine convex sets.

Note that a multi-vehicle scenario, in which the dynamics of each vehicle is represented by hybrid dynamics, in which the vehicles communicate planned trajectories, and in which collision avoidance is cast into time-varying constraints leads exactly to the model class of CPS with networked, time-varying, and hybrid dynamics as specified in Def. 3 and 4 of Sec. 2.

In order to ensure recursive feasibility of the MPC scheme, the tasks 2.2 and 2.3 of WP2 investigate which conditions have to be formulated for admissible replanning of the state trajectories of the involved automated vehicles. These investigations also include the study to which degree the planning of the vehicles should be combined (in a cooperative / centralized scheme), or executed in decentralized fashion.

6 Summarizing Conclusions

This document has reported on the type of models and the set of techniques for modeling and model transformation as investigated in UnCoVerCPS. The contributions are as follows:

- A general model class for CPS has been proposed which combines properties of networked, hybrid, and time-varying models, and by this extends existing definitions of cyber-physical systems. These properties are (in combination or separately) relevant for the developments of methods for control and verification as envisaged in WP2 and WP3 of the project. It was shown for an example how subsystems of the general model (including the coupling to other subsystems) can be represented in the tool SpaceEx. Furthermore, the relation to the (more restricted) model classes underlying the techniques of model transformation has been pointed out.

- A method for conformance verification has been proposed which checks whether a concrete model conforms to an abstract one in the sense that a trace of the first is contained in the set of behaviors of the latter. For this purpose, a conformance monitor with hybrid dynamics was defined, for which verification with SpaceEx reveals, whether conformance applies. This technique is particularly useful if initial high-level specifications of controllers are refined to more detailed representations.

- In the context of verifcation, two approaches for model reduction have been described: The first one shows how balanced truncation can be used to approximate continuous-time switched affine systems by switched linear systems of reduced order, but with state resets. The second one simplifies the structure of discrete-time mixed-logical dynamical systems (an equivalent to piecewise-affine systems) by using a notion of obervability, while the input/ouput behavior is preserved. The ideas presented here include to remove the non-observable part of the dynamics, and to merge modes with the same dynamics. All options contributed to reduce the compuational burden of system verification.

- With specific relation to two of the project case studies, methods for providing appropriate models for online control were proposed: For reach-avoid problems in robot-human interaction, a modeling scheme has been presented which starts from the nonlinear continuous-time robot dynamics, and then step-by-step reduces the model (including the constraints arising from human motion) to a format which is amenable to an online optimizing control technique based on homotopies. Likewise, for a reach-avoid problem within an automated driving scenario, different model approximations were discussed

with respect to model accuracy and computational effort in solving online optimization problems for trajectory planning and control.

Overall, the proposed techniques provide a toolset for modeling and model transformation which is suitable as basis for the development of the methods for control synthesis and verification as carried out in WP2 and WP3 of the project. Note that the latter methods itself establish in certain steps additional means of model transformation (as, e.g., the mapping of the robot dynamics according to (76) into a space of homotopic trajectories, as described in D2.2).

References

[1] A. Abate, S. Amin, M. Prandini, J. Lygeros, and S. S. Sastry. Computational approaches to reachability analysis of stochastic hybrid systems. In *Hybrid Systems: Computation and Control*, volume 4416, pages 4–17. 2007.

[2] A. Abate, J. P. Katoen, J. Lygeros, and M. Prandini. Approximate model checking of stochastic hybrid systems. *European Journal of Control*, 16(6):624–641, 2010.

[3] A. Abate and M. Prandini. Approximate abstractions of stochastic systems: a randomized method. In *50th IEEE Conf. on Decision and Control and European Control Conf.*, pages 4861–4866, 2011.

[4] R. Alur, C. Courcoubetis, N. Halbwachs, T. A. Henzinger, P.-H. Ho, X. Nicollin, A. Olivero, J. Sifakis, and S. Yovine. The algorithmic analysis of hybrid systems. *Theoretical computer science*, 138(1):3–34, 1995.

[5] R. Alur and D. L. Dill. A theory of timed automata. *Theoretical computer science*, 126(2):183–235, 1994.

[6] A. C. Antoulas. *Approximation of large-scale dynamical systems*, volume 6. Society for Industrial Mathematics, 2005.

[7] L. Asselborn, D. Gross, and O. Stursberg. Control of uncertain nonlinear systems using ellipsoidal reachability calculus. *IFAC Proceedings Volumes*, 46(23):50–55, 2013.

[8] A. Bemporad. Efficient conversion of mixed logical dynamical systems into an equivalent piecewise affine form. *IEEE Transactions on Automatic Control*, 49(5):832–838, 2004.

[9] A. Bemporad, G. Ferrari-Trecate, and M. Morari. Observability and Controllability of Piecewise Affine and Hybrid Systems. *IEEE Transactions on Automatic Control*, 45(10):1864–1876, 2000.

[10] A. Bemporad and N. Giorgetti. Logic-based solution methods for optimal control of hybrid systems. *Automatic Control, IEEE Transactions on*, 51(6):963–976, 2006.

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

[12] A. Bemporad and M. Morari. Verification of hybrid systems via mathematical programming. In *Hybrid Systems: Computation and Control*, pages 31–45. Springer, 1999.

[13] A. Bemporad and M. Morari. Predictive Control of Constrained Hybrid Systems. In *Nonlinear Model Predictive Control*, pages 71–98, Ascona, Switzerland, 2000.

[14] S. P. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan. *Linear matrix inequalities in system and control theory*, volume 15. SIAM, 1994.

[15] M. C. 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.

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

[17] H. Ding, G. Reissig, D. Gross, and O. Stursberg. Mixed-Integer Programming for Optimal Path Planning of Robotic Manipulators. In 7^{th} *IEEE Conf. on Automation Science and Engineering*, pages 133–138, 2011.

[18] H. Ding, G. Reissig, and O. Stursberg. Increasing Efficiency of Optimization-based Path Planning for Robotic Manipulators. In 50^{th} *IEEE Conf. on Decision and Control*, pages 1399–1404, 2011.

[19] H. Ding, G. Reissig, K. Wijaya, D. Bortot, K. Bengler, and O. Stursberg. Human Arm Motion Modeling and Long-Term Prediction for Safe and Efficient Human-Robot-Interaction. In *2011 IEEE Conf. on Robotics and Automation*, pages 5875–5880, 2011.

[20] A. Donzé and G. Frehse. Modular, hierarchical models of control systems in spaceex. In *Control Conference (ECC), 2013 European*, pages 4244–4251. IEEE, 2013.

[21] A. Fehnker and F. Ivancic. Benchmarks for hybrid systems verification. In *Hybrid Systems: Computation and Control*, volume 2993, pages 326–341. 2004.

[22] G. Ferrari-Trecate and M. Gati. Computation observability regions for discrete-time hybrid systems. In *Proceedings of the 42nd IEEE Conference on Decision and Control*, volume 2, pages 1153–1158, Dec 2003.

[23] G. Frehse. PHAVer: Algorithmic verification of hybrid systems past hytech. In *Hybrid Systems: Computation and Control*, volume 3414, pages 258–273. 2005.

[24] G. Frehse, A. Hamann, S. Quinton, and M. Woehrle. Formal analysis of timing effects on closed-loop properties of control software. In *Proceedings of the IEEE 35th IEEE Real-Time Systems Symposium, RTSS 2014, Rome, Italy, December 2-5, 2014*, pages 53–62, 2014.

[25] G. Frehse, C. Le Guernic, A. Donzé, S. Cotton, R. Ray, O. Lebeltel, R. Ripado, A. Girard, T. Dang, and O. Maler. Spaceex: Scalable verification of hybrid systems. In *International Conference on Computer Aided Verification*, pages 379–395. Springer, 2011.

[26] S. Garatti and M. Prandini. A simulation-based approach to the approximation of stochastic hybrid systems. In *Analysis and Design of Hybrid Systems*, pages 406–411, 2012.

[27] T. Geyer, F. D. Torrisi, and M. Morari. Optimal complexity reduction of polyhedral piecewise affine systems. *Automatica*, 44(7):1728–1740, 2008.

[28] A. Girard and C. Guernic. Zonotope/hyperplane intersection for hybrid systems reachability analysis. In *11th international workshop on Hybrid Systems: Computation and Control*, pages 215–228, 2008.

[29] A. Girard, A. Julius, and G. J. Pappas. Approximate simulation relations for hybrid systems. *Discrete Event Dynamic Systems*, 18(2):163–179, 2008.

[30] A. Girard, A. A. Julius, and G. J. Pappas. Approximate simulation relations for hybrid systems. *Discrete Event Dynamic Systems*, 18(2):163–179, 2008.

[31] A. Girard and G. J. Pappas. Approximation metrics for discrete and continuous systems. *IEEE Trans. on Automatic Control*, 52(5):782–798, 2007.

[32] A. Girard, G. Pola, and P. Tabuada. Approximately bisimilar symbolic models for incrementally stable switched systems. *IEEE Trans. on Automatic Control*, 55(1):116–126, 2010.

[33] L. Grüne and J. Pannek. *Nonlinear Model Predictive Control*, pages 43–66. Springer London, London, 2011.

[34] W. Heemels, B. D. Schutter, and A. Bemporad. Equivalence of hybrid dynamical models. *Automatica*, 37(7):1085–1091, July 2001.

[35] T. A. Henzinger. The theory of hybrid automata. In *Verification of Digital and Hybrid Systems*, pages 265–292. Springer, 2000.

[36] J. P. Hespanha and A. S. Morse. Stability of switched systems with average dwell-time. In *38th IEEE Conf. on Decision and Control*, volume 3, pages 2655–2660, 1999.

[37] J. Hu, J. Lygeros, and S. S. Sastry. Towards a theory of stochastic hybrid systems. *Lecture Notes in Computer Science LNCS*, 1790:160–173, 2000.

[38] A. A. Julius and G. J. Pappas. Approximations of stochastic hybrid systems. *IEEE Trans. on Automatic Control*, 54(6):1193–1203, 2009.

[39] X. Koutsoukos. Optimal control of stochastic hybrid systems based on locally consistent Markov Decision Processes. In *Proceedings of the 2005 IEEE International Symposium on Intelligent Control*, pages 435–440, June 2005.

[40] A. B. Kurzhanski and P. Varaiya. Ellipsoidal techniques for hybrid dynamics: the reachability problem. In *New Directions and Applications in Control Theory*, volume 321, pages 193–205. 2005.

[41] E. A. Lee. Cyber physical systems: Design challenges. In *Object Oriented Real-Time Distributed Computing (ISORC), 2008 11th IEEE International Symposium on*, pages 363–369. IEEE, 2008.

[42] D. Liberzon. *Switching in systems and control*. Birkhäuser Boston, 2003.

[43] J. Lunze and F. Lamnabhi-Lagarrigue, editors. *Handbook of Hybrid Systems Control – Theory, Tools, Applications*. Cambridge University Press, 2009.

[44] J. Lunze and F. Lamnabhi-Lagarrigue. *Handbook of hybrid systems control: theory, tools, applications*. Cambridge University Press, 2009.

[45] J. Lygeros and M. Prandini. Stochastic hybrid systems: a powerful framework for complex, large scale applications. *European Journal of Control*, 16(6):583–594, 2010.

[46] N. Lynch, R. Segala, and F. Vaandrager. Hybrid i/o automata. *Information and computation*, 185(1):105–157, 2003.

[47] E. Mazzi, A. Sangiovanni Vincentelli, A. Balluchi, and A. Bicchi. Hybrid system reduction. In *47th IEEE Conf. on Decision and Control*, pages 227–232, 2008.

[48] A. Mehta and J. Luck. Novel temporal behavior of a nonlinear dynamical system: The completely inelastic bouncing ball. *Physical review letters*, 65(4):393, 1990.

[49] S. Mitsch and A. Platzer. Modelplex: verified runtime validation of verified cyber-physical system models. *Formal Methods in System Design*, 49(1-2):33–74, 2016.

[50] W. Newman and M. Branicky. Real-time configuration space transforms for obstacle avoidance. *J. of Robotics Research*, 6:650–667, 1991.

[51] A. V. Papadopoulos and M. Prandini. Model reduction of switched affine systems: A method based on balanced truncation and randomized optimization. In *17th Int. Conf. on Hybrid Systems: Computation and Control*, pages 113–122, 2014.

[52] A. V. Papadopoulos and M. Prandini. Model reduction of switched affine systems. *Automatica*, 70:57 – 65, 2016.

[53] M. Petreczky. Realization theory of linear hybrid systems. In *Hybrid Dynamical Systems*, volume 457, pages 59–101. 2015.

[54] M. Petreczky and J. H. van Schuppen. Observability reduction of piecewise-affine hybrid systems. In *19th Int. Symposium on Mathematical Theory of Networks and Systems*, pages 203–210, 2010.

[55] M. Petreczky, R. Wisniewski, and J. Leth. Balanced truncation for linear switched systems. *Nonlinear Analysis: Hybrid Systems*, 10:4–20, 2013.

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

[57] M. Prandini, S. Garatti, and R. Vignali. Performance assessment and design of abstracted models for stochastic hybrid systems through a randomized approach. *Automatica*, 50(11):2852–2860, 2014.

[58] R. Rajamani. *Vehicle dynamics and control*. Springer Science & Business Media, 2011.

[59] H. Roehm, J. Oehlerking, M. Woehrle, and M. Althoff. Reachset conformance testing of hybrid automata. In *Proceedings of the 19th International Conference on Hybrid Systems: Computation and Control, HSCC 2016, Vienna, Austria, April 12-14, 2016*, pages 277–286, 2016.

[60] H. R. Shaker and R. Wisniewski. Model reduction of switched systems based on switching generalized gramians. *Int. Journal of Innovative Computing, Information and Control*, 8(7(B)):5025–5044, 2012.

[61] E. Sontag. Nonlinear regulation: The piecewise linear approach. *IEEE Transactions on Automatic Control*, 26(2):346–358, Apr 1981.

[62] T. Strathmann and J. Oehlerking. Verifying properties of an electro-mechanical braking system. In *1st and 2nd International Workshop on Applied veRification for Continuous and Hybrid Systems, ARCH@CPSWeek 2014, Berlin, Germany, April 14, 2014 / ARCH@CPSWeek 2015, Seattle, WA, USA, April 13, 2015.*, pages 49–56, 2015.

[63] P. Tabuada. *Verification and Control of Hybrid Systems - A Symbolic Approach*. Springer, 2009.

[64] C. J. Tomlin, J. Lygeros, and S. S. Sastry. A game theoretic approach to controller design for hybrid systems. *Proc. of the IEEE*, 88(7):949–970, 2000.

[65] C. J. Tomlin, I. Mitchell, A. M. Bayen, and M. Oishi. Computational techniques for the verification of hybrid systems. *Proc. of the IEEE*, 91(7):986–1001, 2003.

[66] F. D. Torrisi. *Modeling and reach-set computation for analysis and optimal control of discrete hybrid automata*. PhD thesis, Diss., Technische Wissenschaften ETH Zürich, Nr. 15064, 2003, 2003.

[67] F. D. Torrisi and A. Bemporad. Hysdel-a tool for generating computational hybrid models for analysis and synthesis problems. *Control Systems Technology, IEEE Transactions on*, 12(2):235–249, 2004.

[68] R. Vignali, L. Deori, and M. Prandini. Control input design: detecting non influential inputs while satisfying a reachability specification. In *19th World Congress of the International Federation of Automatic Control*, Cape Town, South Africa, August 2014.

[69] R. Vignali and M. Prandini. Input design for a cascading system: An approach based on system decomposition and non-influential input detection. In *2014 IEEE Multi-Conference on Systems and Control*, Antibes, France, October 2014.

[70] R. Vignali and M. Prandini. Model reduction of discrete time hybrid systems: A structural approach based on observability. In *2016 International Workshop on Symbolic and Numerical Methods for Reachability Analysis (SNR)*, pages 1–6, April 2016.