Studies in Systems, Decision and Control 89

Calin Belta
Boyan Yordanov
Ebru Aydin Gol

Formal Methods for Discrete-Time Dynamical Systems

Springer

Chapter 8
Parameter Synthesis

To study the satisfaction of LTL formulas by trajectories of a PWA system $\mathscr{W}$, in Chap. 6 we defined the embedding transition system $T_{\mathscr{W}}$, which was infinite. We showed that finite quotients of $T_{\mathscr{W}}$ can be computed through polyhedral operations and in Chap. 7 we used such quotients to develop an analysis procedure for PWA systems. We discussed how this procedure can be used to find a region of initial conditions of $\mathscr{W}$, from which all trajectories are guaranteed to satisfy the specification. Our procedure was also capable of handling PWA systems with uncertain parameters restricted to polytopic ranges. We assumed that parameter uncertainty was inherent in the system and, in order to guarantee satisfaction, trajectories must satisfy the specification regardless of the (nondeterministic) choice of parameters from the allowed range.

In this chapter, we take a different approach to study autonomous PWA systems with uncertain parameters. The parameters of the system are allowed to vary in predefined polytopic ranges as before, but in this chapter we assume that those ranges can be restricted further. In other words, we treat the parameter ranges not as an uncertainty inherent in the system, but rather as allowed ranges in which the system parameters must be tuned. Our goal is to find subsets of the allowed parameters for each region, such that the satisfaction of a specification can be guaranteed. The problem that we consider in this chapter can be formally stated as follows:

Problem 8.1 (*Parameter synthesis*) Given an autonomous, uncertain parameter, discrete-time piecewise affine system $\mathscr{W}$ (Definition 6.3) and an LTL formula ϕ over $L \cup \{\text{Out}\}$, find subsets of system parameters $\mathbf{P}^A_{l,\phi} \subseteq \mathbf{P}^A_l$ and $\mathbf{P}^c_{l,\phi} \subseteq \mathbf{P}^c_l$ for each region $l \in L$ and a non-empty set of initial states $\mathbf{X}_{0,\phi}$, such that all trajectories of the system originating there satisfy the formula under all identified parameters.

In other words, we are interested in excluding parameters from the allowed sets $\mathbf{P}^A_l$ and $\mathbf{P}^c_l$ for each region $l \in L$, for which the formula is not satisfied. As it will become clear later, for each region $l \in L$, our solution will be in the form of a union of disjoint open polytopes, which are subsets of the allowed polytopes $\mathbf{P}^A_l$ and $\mathbf{P}^c_l$. In general, it is possible that for some states, no allowed parameters can be found such that the satisfaction of the specification can be guaranteed. Such states are excluded from the

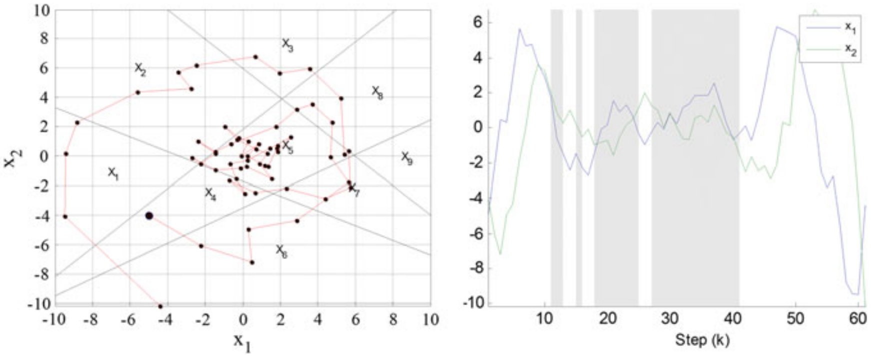
C. Belta et al., *Formal Methods for Discrete-Time Dynamical Systems*,
Studies in Systems, Decision and Control 89, DOI 10.1007/978-3-319-50763-7_8

allowed initial states of the system $\mathbf{X}_{0,\phi}$ and, therefore, the overall problem involves searching for both parameter ranges and initial states from which the satisfaction of the specification can be guaranteed, leading to the formulation of Problem 8.1.

Our approach to Problem 8.1 involves the construction of discrete abstractions in the form of finite transition systems as described in Chap. 7 and a counterexample-guided strategy allowing the identification and elimination of parameters leading to violating trajectories in the system. We first embed the autonomous PWA system $\mathscr{W}$ into $T_{\mathscr{W}}$ (see Definition 6.6 in Chap. 6), which has infinitely many states and is, in general, non-deterministic: given a state of $T_{\mathscr{W}}$ several states might be reachable under different parameters from the allowed sets. Thus, there is a correspondence between the transitions of $T_{\mathscr{W}}$ and the parameters of $\mathscr{W}$ and a possible approach to Problem 8.1 involves iteratively model checking $T_{\mathscr{W}}$ with LTL formula ϕ in order to generate counterexamples as described in Chap. 3 and eliminating such violating behavior by restricting the allowed parameters of $\mathscr{W}$ to appropriate subsets. When no additional counterexamples can be generated, the satisfaction of the specification by the system is guaranteed. Such an approach resembles iterative, counterexample-guided "debugging" of system $T_{\mathscr{W}}$, corresponding to a parameter synthesis procedure for $\mathscr{W}$. However, since $T_{\mathscr{W}}$ is infinite and model checking cannot be applied directly, our parameter synthesis strategy relies on the construction of the finite over-approximation quotient $\overline{T_{\mathscr{W}}/_{\sim}}$ (or the quotient $T_{\mathscr{W}}/_{\sim}$ for autonomous, additive uncertainty PWA systems), whose language includes the language of $T_{\mathscr{W}}$ as described in Chap. 7. Model checking can then be used to cut transitions from $\overline{T_{\mathscr{W}}/_{\sim}}$ and, correspondingly, cut sets of parameters from $\mathscr{W}$ until all system trajectories satisfy the formula.

To provide a solution to Problem 8.1, in Sect. 8.1 we discuss the problem of identifying a subset of the transitions for a finite transition system (such as $\overline{T_{\mathscr{W}}/_{\sim}}$) in order to satisfy an LTL formula. In Sects. 8.2 and 8.3, we characterize sets of parameters associated to transitions in quotients of uncertain parameter PWA systems.Then, in Sect. 8.4, we present a solution to Problem 8.1. Alternatively, in Sect. 8.5, we propose a method for the direct construction of a bisimulation quotient.

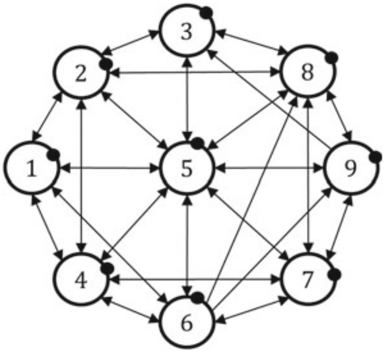
Example 8.1 A 2-dimensional ($N = 2$) PWA system is defined on the set of polytopes $\mathbf{X}_1 \ldots \mathbf{X}_9$ (L = {1…9}) shown in Fig. 8.1a. This system is similar to the one defined in Example 1.8 but is autonomous and has uncertain parameters. For each mode $l \in L$ the parameters are restricted to the ranges $\mathbf{P}_l^A$ and $\mathbf{P}_l^c$. Initially, all modes have identical allowed parameter ranges (i.e., $\forall l \in L$, $\mathbf{P}_l^A = \mathbf{P}^A$, $\mathbf{P}_l^c = \mathbf{P}^c$), which are defined by restricting the individual parameter components of $A = \begin{bmatrix} a_1 & a_2 \\ a_3 & a_4 \end{bmatrix}$ and $c = [c_1, c_2]^T$ to the ranges $0.8 \le a_1 \le 1$, $-0.55 \le a_2 \le -0.05$, $0.05 \le a_3 \le 0.55$, $0.8 \le a_4 \le 1$, $-1 \le c_1 \le 1$ and $-1 \le c_2 \le 1$.



(a) Simulated trajectory in state space (b) Simulated trajectory over time

Fig. 8.1 A simulation of an uncertain parameter PWA system in state space (**a**) and over time (**b**). The trajectory's initial state is marked by a disk in **a**, while visits to the "unsafe" region $\mathbf{X}_5$ are highlighted in *gray* in **b**. See Example 8.1 for details

Fig. 8.2 Quotient of the example PWA system from Fig. 8.1 described in Example 8.1. A dot next to a state indicates a self-loop transition, while the Out state as well as transitions to it from all other states are omitted



A simulated trajectory of the system visits region $\mathbf{X}_5$ (see Fig. 8.1a, b), which is considered "unsafe" in this example, and eventually leaves the defined state space $\mathbf{X}$. Through the rest of this chapter, we will develop a framework that allows us to formulate requirements such as "stay inside $\mathbf{X}$ and never visit $\mathbf{X}_5$", or richer behaviors such as permanent oscillations, and to enforce such specifications by restricting the allowed parameters in each mode of the system to different subsets of the allowed parameters. The method is based on the construction of finite quotients as in Fig. 8.2.

8.1 Counterexample-Guided Pruning of Finite Systems

In this section, we focus on finite transition systems with no inputs and consider the following synthesis problem:

Problem 8.2 (*Transition system synthesis*) Given a finite transition system $T = (X, \delta, O, o)$ and an LTL formula ϕ over its set of observations O, find a subset of transitions $\delta_\phi \subseteq \delta$ and a region $X_{0,\phi} \subseteq X$ such that transition system $T_\phi = (X, \delta_\phi, O, o)$ satisfies ϕ from $X_{0,\phi}$ (i.e., $T_\phi(X_{0,\phi}) \vDash \phi$).

Our goal in Problem 8.2 is to obtain a transition system T_ϕ that satisfies specification ϕ but preserves the states of T. One possible approach to Problem 8.2 is based on the idea of counterexample guided "debugging". Using the LTL model checking procedures described in Chap. 3 we generate a counterexample—a run of T satisfying the negation of the formula $\neg\phi$. If such a run exists, then we eliminate it by removing one of the traversed transitions. Then, we reiterate the process until no additional counterexamples are found, in which case we obtain the transition system T_ϕ satisfying ϕ.

In general, several different transitions are taken during the generation of a counterexample and removing any one of them will remove the counterexample from the language of the quotient. Selecting the best transition to remove at each step is nontrivial and, in general, it is not clear if removing a particular transition will lead to a solution or to the "best" solution when several solutions exist. In order to obtain more general results, we exhaustively generate all solutions by testing all transitions taken by a counterexample. This process can be seen as generating a tree (see Fig. 8.3), having as its root the initial system T, together with a set of states to be considered as initial (e.g. $X_0 = X$). Each child node in the tree represents a transition system that has the same set of states as the parent, but only a subset of its transitions. To construct the children of a node, model checking is applied to the system represented by it and a counterexample is generated. Each child represents the system obtained by removing one of the transitions traversed by the counterexample from the parent system. The exact order in which counterexamples are generated is not important because the entire tree is explored by this procedure.

When transitions are removed, a state might become blocking, resulting in the appearance of finite words in the system's language. Since the semantics of LTL are defined only over infinite words, we recursively make all blocking states unreachable by removing all their incoming transitions. This allows us to guarantee that blocking states are never reached. We must also guarantee that the system is not initialized in a blocking state and, therefore, we remove any blocking states from the initial set X_0. If the initial set X_0 becomes empty at a node of the tree, further removal of transitions will not lead to a solution and such systems are ignored.

A leaf node in the tree constructed as described above represents a transition system for which computation has stopped and no additional counterexamples can be generated. Such systems either include empty initial sets (since blocking states are removed) and are ignored, or otherwise, do not include any blocking states and therefore their languages are non-empty and contain infinite words only. Furthermore, since no additional counterexamples can be generated, all words in the languages of such transition systems are guaranteed to satisfy the LTL formula.

Some additional optimizations to the procedure described above can be performed. It is possible that the same transition system is obtained through different branches of the tree (i.e., through a different sequence of counterexamples). This might lead to unnecessary computation and therefore, the tree is always pruned so that repeated nodes are never added. Furthermore, once a satisfying transition system is found, the tree is pruned to remove all its subsets, thus preserving the richest system with most transitions. Finally, if the negation of the LTL formula is satisfied at any step of the procedure, then no satisfying runs exist in the system and further computation will not lead to solutions, so the tree is pruned accordingly.

The procedure described so far is summarized as Algorithm 15. Since T is finite and therefore contains a finite number of transitions, the termination of Algorithm 15 is guaranteed—in the worst case, all transitions of T will be eliminated. Since the entire computation tree of transition systems described above is explored, the solution to Problem 8.2 generated by this procedure is complete. When $X_0 = \emptyset$ for all leaf nodes of T_{tr} then no T_ϕ and $X_{0,\phi}$ that solve Problem 8.2 exist. In general, several satisfying transition systems (represented by leaf nodes in the computation tree) might be obtained. In this case, selecting the "best" solution (line 15) might involve additional metrics, for example by ranking all transition systems providing a solution to Problem 8.2 based on the number of transitions they include or the expressivity of their languages.

Algorithm 15 Given a finite $T = (X, \delta, O, o)$ and an LTL formula ϕ over O, find $T_\phi = (X, \delta_\phi, O, o)$ and $X_{0,\phi} \subseteq X$, such that $\delta_\phi \subseteq \delta$ and $T_\phi(X_{0,\phi}) \vDash \phi$

```
1: $T_{tr} := \{(T, X)\}$
2: **for all** leaf nodes $(T', X'_0)$, $T' = (X, \delta', O, o)$ of $T_{tr}$ where $X'_0 \neq \emptyset$ **do**
3:   Generate a counterexample $w \in \mathscr{L}_{T'} \cap \mathscr{L}_{\neg\phi}$
4:   **for all** transitions $x' \in \delta'(x)$ from $w$ **do**
5:     Construct $\delta''$ by removing $x' \in \delta'(x)$ from $\delta'$
6:     Construct $T'' = (X, \delta'', O, o)$
7:     Construct $X''_0 \subseteq X'_0$ and adjust $\delta''$ to ensure that $T''$ is non-blocking
8:     **if** $T'' \notin T_{tr}$ **and** $T'' \nvDash \neg\phi$ **then**
9:       Add $(T'', X''_0)$ as a child of $(T', X'_0)$ in $T_{tr}$
10:    **end if**
11:  **end for**
12: **end for**
13: Select a leaf node $(T_\phi, X_{0,\phi})$ of $T_{tr}$ where $X_{0,\phi} \neq \emptyset$
14: **return** $(T_\phi, X_{0,\phi})$
```

Example 8.2 To illustrate the counterexample-guided pruning of finite systems, we apply Algorithm 15 to the simple system shown as the root of the tree in Fig. 8.3. The observations of the system, where each state $X = \{X_1 \ldots X_5\}$ has an unique observations $o(X_i) = o_i$, are omitted in the illustration. We are interested in pruning the transitions of the system such that all trajectories keep visiting states with observation o_2 (i.e., state X_2). Therefore, we apply Algorithm 15 using the specification $\Box\Diamond o_2$. To simplify the illustration, we

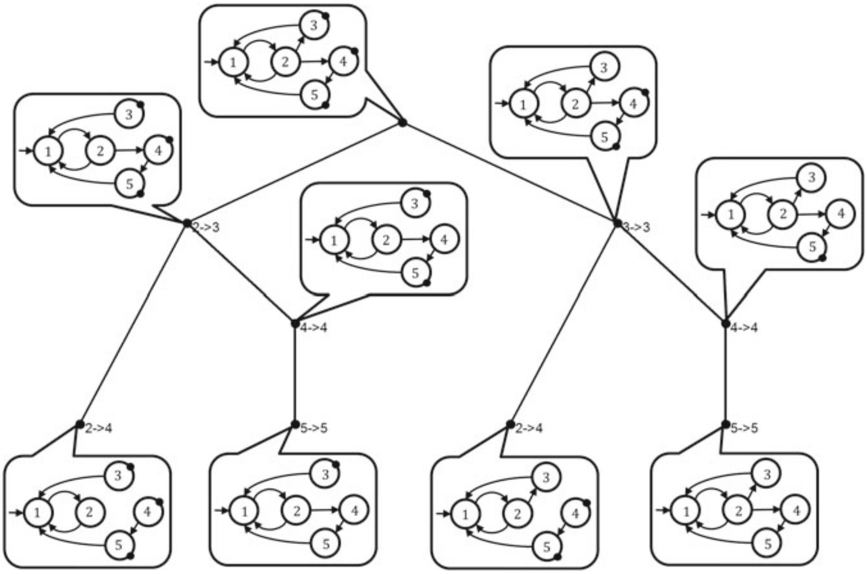


Fig. 8.3 The computation tree generated by Algorithm 15 when applied to a simple, finite transition system (illustrated at the root of the tree) using the specification $\Box\Diamond o_2$. In other words, we require that trajectories of the system (originating in state X_1) always keep visiting states with observation o_2, which in this case is only state X_2 (the unique observations $o(X_i) = o_i$ are omitted from the illustrations). At each node, the indicated transition is removed from the system and the resulting transition systems are illustrated next to each node. A dot next to a state indicates a self-transition. See Example 8.2 for additional details

consider only state X_1 as initial, instead of identifying the set of satisfying initial states while pruning the transitions.

The first counter-example generated by the algorithm is $o_1 o_2 (o_3)^\omega$, which can be removed by pruning the transition from state X_1 to X_2, the one from X_2 to X_3 or the self-transition at X_3. Pruning the transition between states X_1 and X_2 leads to a blocking system that does not satisfy the specification. The pruning of each one of the other two transitions is explored separately and additional counterexamples are generated.

When Algorithm 15 terminates, the entire computation tree illustrated in Fig. 8.3 is explored. The leaf nodes represent transition systems that have been pruned in such a way that the specification is satisfied.

8.2 Parameter Sets and Transitions

For simplicity of presentation, in the rest of this chapter we use a slightly different notation from the one introduced in Sect. 6.1. There, the parameter ranges for an autonomous, uncertain-parameter PWA system were defined as $\mathbf{P}_l^A$ and $\mathbf{P}_l^c$ for each region $l \in L$. For the following discussion, it is convenient to consider a notation with only a single parameter set for each region. We define the set of parameters $\mathbf{P}_l$ for each region $l \in L$ as a polytope in $\mathbb{R}^{(N^2+N)}$ that combines $\mathbf{P}_l^A$ and $\mathbf{P}_l^c$. The linear functions $A : \mathbb{R}^{(N^2+N)} \to \mathbb{R}^{N^2}$ and $c : \mathbb{R}^{(N^2+N)} \to \mathbb{R}^N$ take the first N^2 and the last N components of $p \in \mathbb{R}^{(N^2+N)}$ and form a $N \times N$ matrix and $N \times 1$ vector, respectively. The dynamics of the autonomous, uncertain parameter PWA system from Definition 6.3 are then described by

$$\mathcal{W} \;:\; x(k+1) = A(p)x(k) + c(p),\;\; x(k) \in \mathbf{X}_l,\;\; p \in \mathbf{P}_l,\;\; l \in L,\;\; k = 0, 1, \ldots \tag{8.1}$$

Remark 8.1 For the case when the parameter set $\mathbf{P}_l$ is given as an union of polytopes $\mathbf{P}_l = \bigcup_{i=1}^{n_l} \mathbf{P}_l^i$, the formulas for the calculation of $Post$ and $\overline{Post}$ of $\mathbf{X}_l$ from Eqs. (1.4) and (7.6) can be extended to $\bigcup_{i=1}^{n_l} \text{hull}(\{A(p)v + c(p), v \in V(\mathbf{X}_l), p \in V(\mathbf{P}_l^i)\}$.

In Sect. 7.2, we described the construction of the over-approximation quotient $\overline{T_{\mathcal{W}}/_\sim}$ of an uncertain parameter PWA system $\mathcal{W}$. In this section, we use LTL model checking to "cut" transitions from $\overline{T_{\mathcal{W}}/_\sim}$ until we obtain a transition system $\overline{T_{\mathcal{W}}/_\sim}_\phi$ satisfying the formula ϕ. Once a satisfying transition system is obtained, we modify the original PWA system $\mathcal{W}$ by removing parameter values in such a way that the language of the embedding transition $T_{\mathcal{W}\phi}$ system of the modified $\mathcal{W}_\phi$ is included in the language of $\overline{T_{\mathcal{W}}/_\sim}_\phi$. In other words, $\overline{T_{\mathcal{W}}/_\sim}_\phi$ becomes a quotient of the modified embedding $T_{\mathcal{W}\phi}$, which guarantees the satisfaction of the formula by the system.

The finite quotient $T_{\mathcal{W}}/_\sim = (X_{\mathcal{W}}/_\sim, \delta_{\mathcal{W},\sim}, O_{\mathcal{W}}, o_{\mathcal{W},\sim})$ is constructed so that it captures all possible transitions of the embedding $T_{\mathcal{W}} = (X_{\mathcal{W}}, \delta_{\mathcal{W}}, O_{\mathcal{W}}, o_{\mathcal{W}})$. By the definition of the embedding, transitions are included in the embedding if and only if appropriate parameters for such a transition are allowed. Therefore, we can relate the transitions present in the finite quotient to sets of allowed parameters for the PWA system. These relationships are formalized in the following and are used as part of the parameter synthesis procedures.

Definition 8.1 Given states $l, l' \in X_{\mathcal{W}}/_\sim$, let

$$\mathbf{P}^{l \to l'} = \{p \in \mathbb{R}^{N^2+N} \mid Post(\mathbf{X}_l, p) \cap \mathbf{X}_{l'} \neq \emptyset\} \tag{8.2}$$

denote the set of parameters for which a state from region $\mathbf{X}_l$ makes a transition into region $\mathbf{X}_{l'}$ in $\mathcal{W}$.

In other words

$$p \in \mathbf{P}^{l \to l'} \Leftrightarrow \exists x \in \mathbf{X}_l \text{ such that } A(p)x + c(p) \in \mathbf{X}_{l'} \tag{8.3}$$

and the transition $l \to l'$ is included in the quotient $T_{\mathscr{W}}/_{\sim}$ (or its over-approximation $\overline{T_{\mathscr{W}}/_{\sim}}$) if and only if some parameters from the set $\mathbf{P}^{l \to l'}$ are allowed in mode $l \in L$ of PWA system $\mathscr{W}$. This relates the transitions of the quotient with the set of parameters allowed at a given region as

$$l' \in \delta_{\mathscr{W},\sim}(l) \Leftrightarrow \mathbf{P}_l \cap \mathbf{P}^{l \to l'} \neq \emptyset. \tag{8.4}$$

Equivalently, if the set of parameters allowed for $\mathscr{W}$ in mode $l \in L$ is restricted to any subset of set

$$\mathbf{P}^{l \not\to l'} = \mathbf{P}_c^{l \to l'} = \{p \in \mathbb{R}^{N^2+N} \mid Post(\mathbf{X}_l) \cap \mathbf{X}_{l'} = \emptyset\}, \tag{8.5}$$

where $\mathbf{P}_c^{l \to l'}$ denotes the complement of $\mathbf{P}^{l \to l'}$, then a transition between states l and l' is impossible in $T_{\mathscr{W}}/_{\sim}$. In other words

$$l' \notin \delta_{\mathscr{W},\sim}(l) \Leftrightarrow \mathbf{P}_l \subseteq \mathbf{P}^{l \not\to l'}. \tag{8.6}$$

The equivalence from Eq. (8.6) provides a strategy for eliminating transitions of $T_{\mathscr{W}}/_{\sim}$ (or $\overline{T_{\mathscr{W}}/_{\sim}}$) by restricting the parameters of $\mathscr{W}$ to appropriate sets. This allows the implementation of the algorithm developed in Sect. 8.1 to the quotient of $\mathscr{W}$, provided that the set $\mathbf{P}^{l \not\to l'}$ can be computed.

We first consider the computation of the sets from Definition 8.1 and Eq. (8.5) for autonomous, additive-uncertainty PWA systems, where the following computation is possible:

Proposition 8.1 *Given states $l, l' \in X_{\mathscr{W}}/_{\sim}$, the V-representation of set $\boldsymbol{P}^{l \to l'}$ from Definition 8.1 can be computed from the V-representations of $\boldsymbol{X}_l$ and $\boldsymbol{X}_{l'}$ as*

$$\boldsymbol{P}^{l \to l'} = \{p \in \mathbb{R}^{N^2+N} \mid A(p) = A_l, c(p) \in B\} \textit{ where} \tag{8.7}$$

$$B = hull(\{v' - A_l v, v \in V(\boldsymbol{X}_l), v' \in V(\boldsymbol{X}_{l'})\})\}. \tag{8.8}$$

Proof We need to show that

$$p \in \mathbf{P}^{l \to l'} \Leftrightarrow A(p) = A_l, c(p) \in B \overset{?}{\Leftrightarrow} \exists x \in \mathbf{X}_l \text{ such that } A_l x + c(p) \in \mathbf{X}_{l'}$$

($\Leftarrow$) Let $\exists x \in \mathbf{X}_l$ such that $A_l x + c(p) \in \mathbf{X}_{l'}$. Let $m = |V(\mathbf{X}_l)|$ and $x = \Sigma_{i=1}^m \lambda_i v_i$, where $0 < \lambda_i < 1$ for all $i = 1, \ldots, m$ and $\Sigma_{i=1}^m \lambda_i = 1$. Let $n = |V(\mathbf{X}_{l'})|$ and $x' = \Sigma_{j=1}^n \mu_j v'_j$, where $0 < \mu_j < 1$ for all $j = 1, \ldots, n$ and $\Sigma_{j=1}^n \mu_j = 1$. Then,

$$A_l \Sigma_{i=1}^m \lambda_i v_i + c(p) = \Sigma_{j=1}^n \mu_j v'_j \Rightarrow c(p) = \Sigma_{j=1}^n \mu_j v'_j - A_l \Sigma_{i=1}^m \lambda_i v_i =$$
$$= \Sigma_{i=1}^m \Sigma_{j=1}^n \lambda_i \mu_j (v'_j - A_l v_i) \Rightarrow c(p) \in hull(\{v' - A_l v, v \in V(\mathbf{X}_l),\ v' \in V(\mathbf{X}_{l'})\})$$

($\Rightarrow$) Let $c(p) \in hull(\{v' - A_l v, v \in V(\mathbf{X}_l),\ v' \in V(\mathbf{X}_{l'})\})$, $c(p) = \Sigma_{i=1}^m \Sigma_{j=1}^n \nu_{ij} (v'_j - A_l v_i)$, where $0 < \nu_{ij} < 1$, $i = 1, \ldots, m$, $j = 1, \ldots, n$ and $\Sigma_{i=1}^m \Sigma_{j=1}^n \nu_{ij} = 1$.

Let $\lambda_i = \Sigma_{j=1}^{m} \nu_{ij}$ and $\mu_j = \Sigma_{i=1}^{n} \nu_{ij}$. Of course, $0 < \lambda_i < 1$ for all $i = 1, \ldots, m$, $0 < \mu_j < 1$ for all $j = 1, \ldots, n$ and $\Sigma_{i=1}^{m} \lambda_i = \Sigma_{j=1}^{n} \mu_j = \Sigma_{i=1}^{m} \Sigma_{j=1}^{n} \nu_{ij} = 1$. Then, for $x = \Sigma_{i=1}^{m} \lambda_i v_i$ and $x' = \Sigma_{j=1}^{n} \mu_j v'_j$ we have $A_l x + c(p) = x'$ and therefore $\exists x \in \mathbf{X}_l$ such that $A_l x + c(p) \in \mathbf{X}_{l'}$. ■

In the equations above, A_l represents the (singleton) parameter value of the matrix corresponding to mode $l \in L$. Following from Proposition 8.1, the removal of a transition between states l and l' in $T_{\mathscr{W}}/_{\sim}$ amounts to the restriction of the allowed parameters in mode l of $\mathscr{W}$ to any set

$$\mathbf{P}'_l \subseteq \mathbf{P}_l \setminus \mathbf{P}^{l \rightarrow l'}, \tag{8.9}$$

and all computation can be performed using polyhedral operations.

For autonomous PWA systems where the matrix component of the parameters is allowed to vary, the set $\mathbf{P}^{l \rightarrow l'}$ cannot be computed easily. Instead, we focus directly on the computation of a subset

$$\underline{\mathbf{P}}^{l \nrightarrow l'} \subseteq \mathbf{P}^{l \nrightarrow l'}, \tag{8.10}$$

guaranteeing that a transition from all states in region $\mathbf{X}_l$ to any state in region $\mathbf{X}_{l'}$ is impossible in $T_{\mathscr{W}}$. This allows the removal of transition $l \rightarrow_{e\sim} l'$ in $T_{\mathscr{W}}/_{\sim}$ by restricting the parameters of $\mathscr{W}$ to a set $\mathbf{P}'_l \subseteq (\mathbf{P}_l \cap \underline{\mathbf{P}}^{l \nrightarrow l'})$. In the following, we develop two different strategies leading to the the construction of $\underline{\mathbf{P}}^{l \nrightarrow l'}$, which offer a tradeoff between computational complexity and the accuracy of the obtained approximation.

Our first computational strategy is based on the following observation: a transition from l to l' is impossible in $T_{\mathscr{W}}/_{\sim}$ under parameters $p \in \mathbf{P}_l$ (i.e., $Post(\mathbf{X}_l, p) \cap \mathbf{X}_{l'} = \emptyset$) when one of the inequalities defining the H-representation of region $\mathbf{X}_{l'}$ is violated by all successor states $A(p)x + c(p) \in Post(\mathbf{X}_l)$, $x \in \mathbf{X}_l$.

Proposition 8.2 *Given states $l, l' \in X_{\mathscr{W}}/_{\sim}$, the H-representation of a conservative under-approximation of set $\boldsymbol{P}^{l \nrightarrow l'}$ from Eq. (8.5) can be computed from the V-representation of $\boldsymbol{X}_l$ and the H-representation of $\boldsymbol{X}_{l'} = \{x \in \mathbb{R}^N \mid h_i^T x \leq k_i, i = 1, \ldots, n\}$ as*

$$\underline{\boldsymbol{P}}^{l \nrightarrow l'} = \bigcup_{i=1}^{n} \{p \in \mathbb{R}^{(N^2+N)} \mid h_i^T \hat{v} p > k_i, \forall v \in V(\boldsymbol{X}_l), \forall i = 1, \ldots, n\} \subseteq \boldsymbol{P}^{l \nrightarrow l'}, \tag{8.11}$$

where, for any state $x \in \mathbb{R}^N$ and parameters $p \in \mathbb{R}^{N^2+N}$, the $\hat{x}$ operator reshapes vector x so that $\hat{x} p = A(p)x + c(p)$.

Proof

$$\text{Let } p \in \underline{\mathbf{P}}^{l \nrightarrow l'} = \bigcup_{i=1}^{n} \{p \in \mathbb{R}^{(N^2+N)} \mid h_i^T \hat{v} p > k_i, \forall v \in V(\mathbf{X}_l), \forall i = 1, \ldots, n\} =$$

$$
\begin{aligned}
&= \bigcup_{i=1}^{n}\{p \in \mathbb{R}^{(N^2+N)} \mid h_i^T(A(p)v + c(p)) > k_i, \forall v \in V(\mathbf{X}_l)\} \Rightarrow \\
&\Rightarrow \exists i, 1 \leq i \leq n, \forall v \in V(\mathbf{X}), h_i^T(A(p)v + c(p)) > k_i \Leftrightarrow \\
&\Leftrightarrow \forall x \in \mathbf{X}, h_i^T(A(p)v + c(p)) > k_i \Leftrightarrow \\
&\Leftrightarrow \nexists x \in \mathbf{X}, A(p)x + c(p) \in \mathbf{X}_{l'} \Leftrightarrow \\
&\Leftrightarrow Post(\mathbf{X}_l, p) \cap \mathbf{X}_{l'} = \emptyset
\end{aligned}
$$

■

Using the computation from Proposition 8.2, the under-approximation $\underline{\mathbf{P}}^{l \nrightarrow l'}$ is obtained as a union of polyhedral sets and allows the computation of a set of allowed parameters $\mathbf{P}'_l = (\mathbf{P}_l \cap \underline{\mathbf{P}}^{l \nrightarrow l'})$ for mode l of $\mathscr{W}$, guaranteeing the removal of the transition between states l and l' in $T_{\mathscr{W}}/_{\sim}$.

A second strategy for the computation of an under-approximation to $\mathbf{P}^{l \nrightarrow l'}$ for some $l, l' \in X_{\mathscr{W}}/_{\sim}$ is based on the set difference operator $\ominus$ introduced in Definition A.8 and the observation that, given parameters $p \in \mathbf{P}_l$,

$$Post(\mathbf{X}_l, p) \cap \mathbf{X}_{l'} \neq \emptyset \Leftrightarrow \mathbf{0} \in Post(\mathbf{X}_l, p) \ominus \mathbf{X}_{l'} \tag{8.12}$$

Proposition 8.3 *Given states $l, l' \in X_{\mathscr{W}}/_{\sim}$ and the V-representations of $\boldsymbol{X}_l$ and $\boldsymbol{X}_{l'}$, the V-representation of a conservative under-approximation of the set from Eq. (8.5) can be computed as*

$$\underline{\boldsymbol{P}}^{l \nrightarrow l'} = \{p \in \mathbb{R}^{N^2+N} \mid h^T \hat{v} p < h^T v', \forall v \in V(\boldsymbol{X}_l), \forall v' \in V(\boldsymbol{X}_{l'})\} \subseteq \boldsymbol{P}^{l \nrightarrow l'} \tag{8.13}$$

for any $h \in \mathbb{R}^N$. Furthermore, a less conservative under-approximation can be computed as

$$\underline{\boldsymbol{P}}^{l \nrightarrow l'} = \bigcup_{h \in H}\{p \in \mathbb{R}^{N^2+N} \mid h^T \hat{v} p < h^T v', \forall v \in V(\boldsymbol{X}_l), \forall v' \in V(\boldsymbol{X}_{l'})\} \subseteq \boldsymbol{P}^{l \nrightarrow l'} \tag{8.14}$$

where $H \subseteq \mathbb{R}^N$.

Proof To guarantee that all points from a polytope $x \in \mathbf{X}_l$ satisfy a linear inequality $h^T x \leq k$ it is necessary and sufficient to guarantee that the inequality is satisfied at all vertices $v_x \in V(\mathbf{X}_l)$

$$\forall v \in V(\mathbf{X}_l), h^T v \leq k \Leftrightarrow \forall x \in \mathbf{X}_l, h^T x \leq k \tag{8.15}$$

Let $p \in \underline{\mathbf{P}}^{l \nrightarrow' l}$. Then, for some $h \in \mathbb{R}^N$ we have

$$
\begin{aligned}
&\forall v \in V(\mathbf{X}_l), \forall v \in V(\mathbf{X}_l), h^T \hat{v} p < h^T v' \Rightarrow \\
&\Rightarrow \forall v \in V(\mathbf{X}_l), \forall v \in V(\mathbf{X}_l), h^T(A(p)v + c(p) - v') < 0,
\end{aligned}
$$

which, from Eq. (8.15), implies that

$$\forall x \in Post(\mathbf{X}_l, p) \ominus \mathbf{X}_{l'}, h^T x < 0 \Rightarrow \mathbf{0} \notin Post(\mathbf{X}_l, p) \ominus \mathbf{X}_{l'}.$$

From Eq. (8.12) this guarantees that $Post(\mathbf{X}_l, p) \cap \mathbf{X}_{l'} = \emptyset$. Extending this computation to a union of sets as in Eq. (8.14) is straightforward, since the result holds for any $h \in H$, where $H \subset \mathbb{R}^N$ is a set of samples. ■

The quality of the under-approximation $\underline{\mathbf{P}}^{l \not\to l'}$ computed through Proposition 8.3 improves as the number of samples in set H from Eq. (8.14) is increased. To obtain better coverage, the set H can be obtained through uniform sampling of rotation groups. While, in terms of computational complexity, this approach could be more costly than obtaining the under-approximation $\underline{\mathbf{P}}^{l \not\to l'}$ through Proposition 8.2, it allows control over the quality of the approximation by selecting the number of samples in H.

8.3 Transient Parameters

In this section, we consider the particular case when a self loop (i.e., a transition $l \to l$ for some $l \in X_{\mathscr{W}}/_{\sim}$) must be removed during the application of the procedure described in Sect. 8.1 to the quotient $T_{\mathscr{W}}/_{\sim}$. Although such transitions can be removed by restricting the parameters of PWA system $\mathscr{W}$ to appropriate subsets using the computation from Proposition 8.2 or Proposition 8.3, this might lead to very conservative results. In fact, such an approach would require that whenever a self loop at a state $l \in X_{\mathscr{W}}/_{\sim}$ is removed, all trajectories of $\mathscr{W}$ leave region $\mathbf{X}_l$ in a single step, which is hard to enforce. Instead, in the following we derive conditions guaranteeing that if the parameters are restricted appropriately, trajectories of the system leave the region eventually, but not necessarily in a single step, which leads to a less conservative parameter synthesis procedure.

Definition 8.2 A subset of parameters $\mathbf{P}'_l \subseteq \mathbf{P}_l$ is *transient* at mode $l \in L$ of PWA system $\mathscr{W}$ if and only if, for all trajectories $x(0)x(1)x(2)\ldots$ such that $x(0) \in \mathbf{X}_l$, there exists a finite $k > 1$ such that $x(0), \ldots, x(k) \in \mathbf{X}_l$ and $x(k+1) \notin \mathbf{X}_l$.

Remark 8.2 The above definition of transient parameters is related to the definition of stuttering inputs from Definition 9.2 in Sect. 9.3 from Chap. 9. Propositions 8.4 and 8.5 are stated in more general forms as Propositions 9.5 and 9.6, respectively, and their proofs are therefore postponed to Chap. 9.

Restricting the parameters of a PWA system $\mathscr{W}$ to a transient subset $\mathbf{P}'_l \subseteq \mathbf{P}_l$ guarantees that region $\mathbf{X}_l$ becomes a transient region for all trajectories of the system. While this does not eliminate the self loop at state $l \in X_{\mathscr{W}}/_{\sim}$ in the quotient $T_{\mathscr{W}}/_{\sim}$, it guarantees that this loop cannot be followed infinitely often along any trajectory. In the particular case when a specification expressed as a formula ϕ from the LTL$\setminus\bigcirc$ fragment is considered, this allows us to safely ignore transition $l \to l$ without violating the inclusion of the system language within the quotient language. Formally, this result follows from the stutter equivalence (i.e., an equivalence with respect to the order of observations visited along all trajectories but not the exact number of repetitions of each observation) of the quotients constructed with and without stutter transitions such as the self loop at state l. In the following, we derive a necessary and sufficient condition characterizing a subset of parameters as transient and use it to develop a computational procedure based on polyhedral operations for restricting the parameters of a PWA system to transient subsets.

Proposition 8.4 *A subset of parameters $\boldsymbol{P}_l' \subseteq \boldsymbol{P}_l$ is transient at mode $l \in L$ of PWA system $\mathcal{W}$ if and only if*

$$\mathbf{0} \notin hull(\{(A(p) - I)v + c(p), \forall v \in V(\boldsymbol{X}_l), \forall p \in V(\boldsymbol{P}_l')\}) \qquad (8.16)$$

where $I \in \mathbb{R}^{N \times N}$ denotes the identity matrix.

The characterization from Proposition 8.4 provides a computational procedure that allows us to check if a set of parameters is transient. Since this condition is necessary and sufficient, in the following we use it to restrict a set of parameters to a transient subset.

Proposition 8.5 *Given state $l \in X_{\mathcal{W}}/_{\sim}$ and the V-representation $V(\boldsymbol{X}_l)$ of $\boldsymbol{X}_l$, the subset of parameters with the following H-representation*

$$\boldsymbol{P}_l' = \{p \in \mathbb{R}^{N^2+N} \mid h^T \hat{v} p < h^T v, \forall v \in V(\boldsymbol{X}_l)\} \qquad (8.17)$$

is transient at mode $l \in L$ for any $h \in \mathbb{R}^N$.

Note that the computation from Proposition 8.4 is similar to the one from Proposition 8.3 but does not allow a larger transient set of parameters to be computed as a union of smaller ones as in Eq. (8.14). However, sampling might still be beneficial in order to find larger subsets of parameters that are transient. In other words, given the set of samples H, generated by uniformly sampling rotation groups as in Proposition 8.3, we seek to find an $h \in H$ that maximizes the volume of the intersection

$$\mathbf{P}_l' = \mathbf{P}_l \cap \{p \in \mathbb{R}^{N^2+N} \mid h^T \hat{v} p < h^T v, \forall v \in V(\mathbf{X}_l)\}). \qquad (8.18)$$

8.4 Parameter Synthesis for PWA Systems

Using the results described in Sects. 8.3 and 8.2 and the method discussed in Sect. 8.1, a solution to Problem 8.1 can be obtained as follows. Given PWA system $\mathcal{W}$ with embedding $T_{\mathcal{W}}$ and LTL formula ϕ, the quotient $T_{\mathcal{W}}/_{\sim}$ (or its over-approximation $\overline{T_{\mathcal{W}}/_{\sim}}$) is constructed as described in Chap. 7 and Algorithm 15 is applied to it. Whenever a transition between states l and l' is removed as part of the computation, the set of system parameters of $\mathscr{W}$ is restricted using Propositions 8.2 or 8.3. In the particular case when $l = l'$ and ϕ is from the LTL$\backslash\bigcirc$ fragment, the self loop $l \rightarrow l$ is removed through Proposition 8.5 in order to obtain less conservative results.

The computation from Algorithm 15 is guaranteed to terminate returning a tree, where the root node represents the quotient $T_{\mathscr{W}}/_\sim$ (or $\overline{T_{\mathscr{W}}/_\sim}$) and every other node represents a transition system that has the same states as $T_{\mathscr{W}}/_\sim$ (and $\overline{T_{\mathscr{W}}/_\sim}$) but only a subset of its transitions (see Sect. 8.1). Each transition system from this tree is, in fact, a quotient of a PWA system where parameters have been restricted to appropriate subsets using the computation described so far. In other words, each node in the tree represents a triple $(T'_{\mathscr{W}}/_\sim, \mathscr{W}', \mathbf{X}'_0)$ where

i. $X'_0 \subseteq X'_{\mathscr{W}}/_\sim$ is a set of initial states from the quotient $T'_{\mathscr{W}}/_\sim$,
ii. the difference between $\mathscr{W}$ and $\mathscr{W}'$ is in the parameter sets where, for each mode $l \in L$, the parameter set of $\mathscr{W}'$ is a subset of the allowed parameter set for $\mathscr{W}$ (i.e., $\mathbf{P}'_l \subseteq \mathbf{P}_l$ for some $l \in L$),
iii. the difference between $T'_{\mathscr{W}}/_\sim$ and $T_{\mathscr{W}}/_\sim$ is the set of transitions where, for each state $l \in X_{\mathscr{W}}/_\sim$, the set of reachable states from l in $T'_{\mathscr{W}}/_\sim$ is a subset of the set of reachable states in $T_{\mathscr{W}}/_\sim$, and
iv. T'_W is the embedding of $\mathscr{W}'$, while $T'_{\mathscr{W}}/_\sim$ is the quotient of T'_e induced by the observational equivalence relation $\sim$.

Therefore, $T'_{\mathscr{W}}/_\sim$ simulates $T'_{\mathscr{W}}$ and the language inclusion $\mathscr{L}_{T'_{\mathscr{W}}} \subseteq \mathscr{L}_{T'_{\mathscr{W}}/_\sim}$ is guaranteed. When a leaf node in the tree returned by Algorithm 15 represents the triple $(T'_{\mathscr{W}}/_\sim, \mathscr{W}', X'_0)$, we know that $T'_{\mathscr{W}}/_\sim$ satisfies ϕ from region X_0 (see Sect. 8.1) and, therefore, $\mathscr{W}'$ satisfies ϕ from region $con(X_0)$, which provides a solution to Problem 8.1.

The PWA system structure allows different regions to share the same set of parameters, for example, whenever the initial partitioning of X is refined to accommodate a specification. Therefore, it is possible that additional transitions besides the target one are removed at each step of the computation described above and, to account for this, we reconstruct the quotient every time parameters are cut. If, during the computation, the set of allowed parameters for some mode $l \in L$ of $\mathscr{W}$ becomes empty, then we consider state l and all states from region $con(l)$ as blocking in $T_{\mathscr{W}}/_\sim$ and $T_{\mathscr{W}}$, respectively, and we make them unreachable by restricting the parameters of all other regions accordingly. Whenever an over-approximation quotient is constructed, a spurious transition might appear in place of one that was already eliminated but we prevent this by enforcing that once a transition is removed it never reappears in the quotient.

The computation described so far in this section is summarized as Algorithm 16, which covers the most general case of parameter uncertainty in $\mathscr{W}$. This procedure is guaranteed to terminate but, while our solution to the purely discrete problem discussed in Sect. 8.1 was complete, Algorithm 16 might not find a solution to Problem 8.1 even when one exists. Due to the construction and use of (over-approximation) quotients to guide the removal of parameters and the computation of under-approximate parameter sets for the removal of transitions, the overall method becomes conservative, but the correctness of the solution (when one is found) is guaranteed. As for Algorithm 16, the procedure summarized in Algorithm 16 returns a tree, where several solutions to Problem 8.1 might be possible and are represented by its leaf nodes. Selecting the “best” solution is a non-trivial problem, and might depend on the application. It is possible to introduce additional constraints (such as requiring that a particular transition is present) or compare total number of transitions of the solutions, since more reachable states from the initial one with more transitions result in a richer language.

Both the number of states and transitions in the embedding $\overline{T_{\mathscr{W}}/_\sim}$ contribute to the complexity of Algorithm 16. A high dimensional system with many regions of different dynamics would be embedded in a transition system with a large number of states. This, together with the size of the LTL formula affects the time required to perform model checking on the system. The number of transitions in the original embedding, on the other hand, depends on the dynamics of the system and determines how many iterative model checking steps must be performed during the generation of the computation tree. As a result, Algorithm 16 can perform well even on high dimensional systems, as long as the total number of transitions is low or only few transitions must be removed to reach a solution.

Algorithm 16 Given a PWA system $\mathscr{W}$ (embedded in $T_{\mathscr{W}}$) and an LTL formula ϕ, identify a region $\mathbf{X}_{0,\phi}$ and construct a system $\mathscr{W}^{\phi}$ that satisfies ϕ from $\mathbf{X}_{0,\phi}$ and, for each mode $l \in L$, its parameters are a subset of the allowed parameters for $\mathscr{W}$.

```
Construct quotient $\overline{T_{\mathscr{W}}/_\sim}$ from $T_{\mathscr{W}}$
Let $X_0 := X_{\mathscr{W}}/_\sim \setminus \{\text{Out}\}$
Initialize $T_{tr} := \{(\overline{T_{\mathscr{W}}/_\sim}, \mathscr{W}, X_0)\}$
for all leaf nodes $(\bar{T}'_{\mathscr{W}}/_\sim, \mathscr{W}', X'_0)$ of $T_{tr}$ where $X'_0 \neq \emptyset$ do
  Generate a counterexample $w \in \mathscr{L}_{\bar{T}'_{\mathscr{W}}/_\sim} \cap \mathscr{L}_{\neg\phi}$
  for all transitions $l$ to $l'$ from $w$ do
    if $l = l'$ and $\phi \in \text{LTL}\backslash\bigcirc$ then
      Construct $\mathbf{P}''_l$ as a transient subset of $\mathbf{P}'_l$
    else
      $\mathbf{P}''_l := (\mathbf{P}'_l \cap \underline{\mathbf{P}}^{l \nrightarrow l'})$
    end if
    Construct PWA system $\mathscr{W}''$ (the parameter set for mode $l \in L$ is $\mathbf{P}''_l$)
    Construct quotient $\bar{T}''_{\mathscr{W}}/_\sim$ of $\mathscr{W}''$
    Enforce no transition between $l$ and $l'$ in $\bar{T}''_{\mathscr{W}}/_\sim$
    Recursively make all blocking states of $\bar{T}''_{\mathscr{W}}/_\sim$ unreachable and restrict the parameters of
   $\mathscr{W}''$ appropriately
    Construct initial set $X''_0 \subseteq X'_0$ by excluding blocking initial states
    if no node from $T_{tr}$ has $\bar{T}''_{\mathscr{W}}/_\sim$ as its quotient then
      add $(\bar{T}''_{\mathscr{W}}/_\sim, \mathscr{W}'', X''_0)$ as a child of $(\bar{T}'_{\mathscr{W}}/_\sim, \mathscr{W}', X_0)$ in $T_{tr}$
    end if
  end for
end for
select a leaf node $(\bar{T}_{\mathscr{W}^{\phi}}/_\sim, \mathscr{W}^{\phi}, X_{0,\phi})$ of $T_{tr}$ where $X_{0,\phi} \neq \emptyset$
return $\mathscr{W}^{\phi}$ and $\mathbf{X}_{0,\phi} = con(X_{0,\phi})$
```

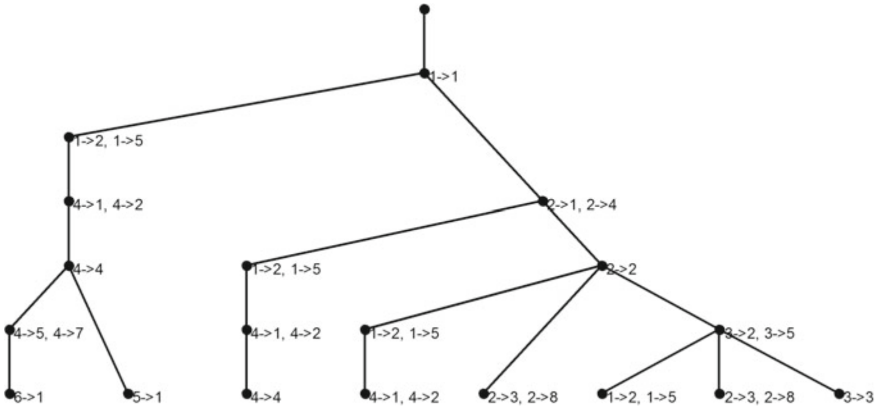


Fig. 8.4 First 20 nodes of the computation tree generated by Algorithm 16 when applied to the PWA system from Example 8.1 (simulated trajectory in Fig. 8.1 and quotient in Fig. 8.2). In contrast to the transition system synthesis algorithm application illustrated in Fig. 8.3, here multiple transitions are often removed at each node as a result of restricting the parameters of the system as explained in Sect. 8.2

Remark 8.3 In order to prevent unnecessary computation, we can first model-check the quotient $T_{\mathscr{W}}/_{\sim}$ from each initial state against $\neg\phi$. The satisfaction of this negation from an initial state implies that there are no satisfying trajectories originating there under any of the allowed parameters and a solution to Problem 8.1 will not be found by our parameter synthesis procedure.

Example 8.3 To illustrate the parameter synthesis for PWA systems approach presented in this chapter, we apply Algorithm 16 to the PWA system defined in Example 8.1. The initial system has uncertain parameters and, for some parameter values, trajectories of the system visit region $\mathbf{X}_5$, which we consider unsafe. In addition, it is possible that trajectories of the initial system leave $\mathbf{X}$, which is required to be invariant. Finally, we require that trajectories of the system oscillate, visiting regions $\mathbf{X}_1$ and $\mathbf{X}_9$, while avoiding region $\mathbf{X}_5$. To capture all these properties, we use the specification $\Box(\neg 5 \wedge \neg \text{Out} \wedge \Diamond 1 \wedge \Diamond 9)$. The invariant $\mathbf{X}$ is enforced as a pre-processing step by restricting the parameters of the system to suitable sub-sets and therefore the specification reduces to $\Box(\neg 5 \wedge \Diamond 1 \wedge \Diamond 9)$.

The first 20 nodes (corresponding to different quotients) of the computation tree generated by Algorithm 16 are shown in Fig. 8.4. Compared to the counterexample guided pruning of transitions in a finite system (illustrated in Fig. 8.3), multiple transitions are removed in this case since the parameters of the PWA system are restricted at each step and the quotient is recomputed.

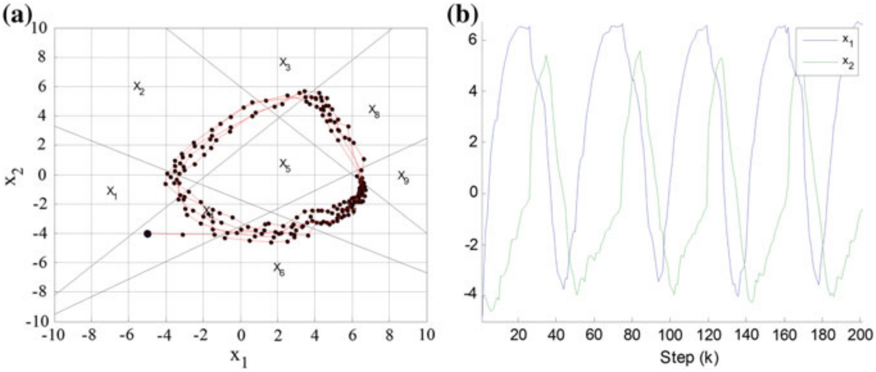
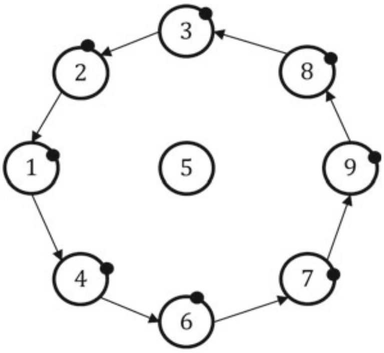


Fig. 8.5 A satisfying trajectory in state space (**a**) and time (**b**) for the system synthesized in Example 8.3. The initial state is shown as a disk in **a**

Fig. 8.6 Resulting quotient after parameter synthesis for the system in Example 8.3. All transitions to state Out are also removed to enforce **X** as an invariant, while self-transitions are preserved but the parameters are restricted as stuttering to enforce the progress of trajectories of the system



Only a single satisfying PWA system is identified in this case after the algorithm terminates and a simulated trajectory of the resulting system is shown in Fig. 8.5. This illustrates that the system satisfies the specification and the trajectory oscillates without visiting region $\mathbf{X}_5$ or leaving the invariant **X**. The quotient of the resulting system is shown in Fig. 8.6, which illustrates that a number of transitions were removed in order to satisfy the specification compared to the quotient of the initial system shown in Fig. 8.2.

Interestingly, for this example no solutions are identified unless the computation of transient parameters described in Sect. 8.3 is utilized. Indeed, the quotient shown in Fig. 8.6 retains self-transitions at a number of the states and the trajectory illustrated in Fig. 8.5 makes several steps within the same region before making a transition to an adjacent region along the oscillations.

8.5 Parameter Synthesis Using Bisimulations

The construction of finite bisimulation quotients for PWA systems is attractive due to their equivalence with respect to model checking. One possible approach to the computation of bisimulation quotients involves the construction and subsequent refinement of simulation quotients as discussed in Chap. 7. However, this procedure is not guaranteed to terminate since, in general, a PWA system might not admit a finite bisimulation quotient. Here, we consider an orthogonal approach where, instead of attempting to construct a bisimulation quotient for a given PWA system directly (which might be infeasible), we first restrict the system's behavior by restricting its sets of parameters appropriately. While, in doing so, we sacrifice possible behavior and restrict the richness of the system's language, we obtain a system for which a bisimulation quotient can be computed trivially.

A bisimulation quotient constructed using the approach outlined above is not directly guaranteed to satisfy a given specification (Problem 8.1). However, as it will become clear in the following, each transition of such a bisimulation quotient corresponds to a specific set of parameters. Then, a particular property can be easily enforced (for example by pruning transitions and the corresponding parameter sets using the approach described in Sect. 8.1). In addition, the bisimulation quotient can be used interchangeably with the restricted-parameter PWA system for model checking or analysis.

As before, we begin by studying the relation between the set of parameters of $\mathscr{W}$ and transitions in the quotient $T_{\mathscr{W}}/_{\sim}$.

Definition 8.3 Given states $l, l' \in X_{\mathscr{W}}/_{\sim}$, let

$$\mathbf{P}^{l \Rightarrow l'} = \{p \in \mathbb{R}^{(N^2+N)} \mid Post(\mathbf{X}_l, p) \subseteq \mathbf{X}_{l'}\} \tag{8.19}$$

denote the set of parameters for which every state $x \in \mathbf{X}_l$ makes transitions to states in $\mathbf{X}_{l'}$ only.

In other words,

$$p \in \mathbf{P}^{l \Rightarrow l'} \Leftrightarrow \forall x \in \mathbf{X}_l,\ A(p)x + c(p) \in \mathbf{X}_{l'}. \tag{8.20}$$

From Eq. (8.19) it follows that restricting the allowed parameters in mode $l \in L$ of PWA system $\mathscr{W}$ to any subset $\mathbf{P}'_l \subseteq (\mathbf{P}_l \cap \mathbf{P}^{l \Rightarrow l'})$ guarantees that only the deterministic transition $l \to l'$ is possible in $T_{\mathscr{W}}/_{\sim}$.

Proposition 8.6 *If, for each mode $l \in L$, the parameters of PWA system $\mathscr{W}$ are restricted to the subset $\boldsymbol{P}'_l = \boldsymbol{P}_l \cap (\bigcup_{l' \in L} \boldsymbol{P}^{X_l \Rightarrow X_{l'}})$, then the quotient $T_{\mathscr{W}}/_{\sim}$ is a bisimulation quotient.*

Proof The proof follows immediately from Definitions 8.3, 1.4 (bisimulation) and 6.6 (embedding of PWA systems). ■

After the parameters of $\mathscr{W}$ are restricted as described in Proposition 8.6, the quotient $T_{\mathscr{W}}/_{\sim}$ is still computable as before but its transitions are implicitly induced by

$l \rightarrow l' \Leftrightarrow \mathbf{P}'_l \cap \mathbf{P}^{l \Rightarrow l'} \neq \emptyset$. Note that $T_{\mathscr{W}}/_{\sim}$ is in general nondeterministic since the parameter set $\mathbf{P}_l \cap \mathbf{P}^{\mathbf{X}_l \Rightarrow \mathbf{X}_{l'}}$ might be nonempty for several different $l' \in L$.

Proposition 8.7 *Given states $l, l' \in X_{\mathscr{W}}/_{\sim}$, the H-representation of parameter set $\boldsymbol{P}^{l \Rightarrow l'}$ from Definition 8.3 can be computed from the V-representation $con(l) = \boldsymbol{X}_l = hull(V(\boldsymbol{X}_l))$ and the H-representation $con(l') = \boldsymbol{X}_{l'} = \{x \in \mathbb{R}^N \mid h_i^T x \leq k_i, i = 1, \ldots, n\}$ as*

$$\boldsymbol{P}^{l \Rightarrow l'} = \{p \in \mathbb{R}^{(N^2+N)} \mid h_i^T(A(p)v + c(p)) < k_i, \forall v \in V(\boldsymbol{X}_l), \forall i = 1, \ldots, n\}$$

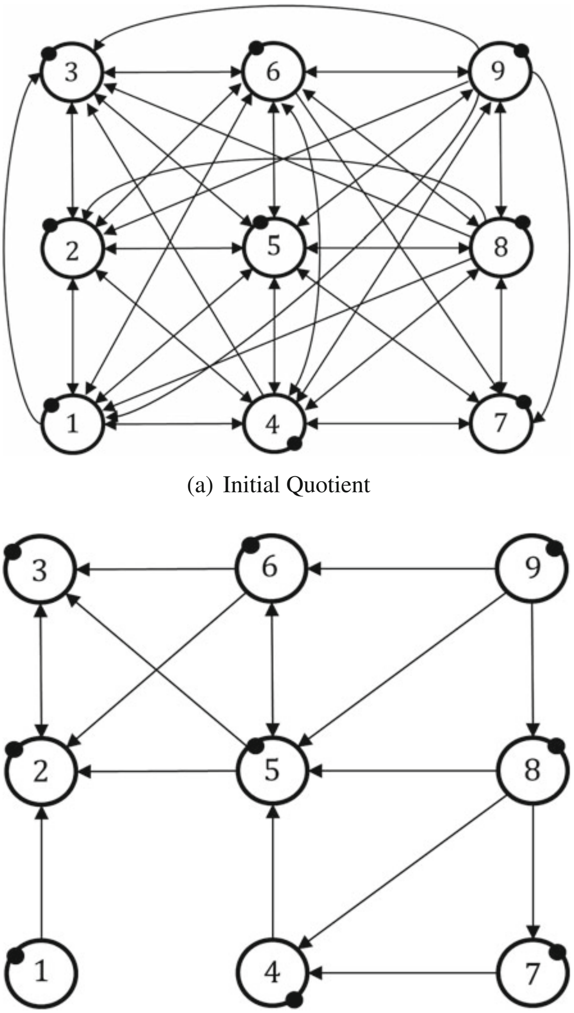
Proof ($\Rightarrow$) Let $p \in \mathbb{R}^{N^2+N}$ such that $\forall v \in V(\mathbf{X}_l), \forall i = 1, \ldots, n, h_i^T(A(p)v + c(p)) < k_i$. From Eq. (8.15), this implies that $\forall i = 1, \ldots, n,\ \forall x \in \mathbf{X}_l, h_i^T(A(p)x + c(p)) < k_i$ or, equivalently, $\forall x \in \mathbf{X}_l, A(p)x + c(p) \in \mathbf{X}_{l'}$ and therefore $Post(\mathbf{X}_l, p) \subseteq \mathbf{X}_{l'}$.

($\Leftarrow$) Let $p \in \mathbb{R}^{N^2+N}$ such that $Post(\mathbf{X}_l, p) \subseteq \mathbf{X}_{l'}$. Then, $hull(\{A(p)v + c(p), v \in V(\mathbf{X}_l)\}) \subseteq \mathbf{X}_{l'}$, which implies that $\forall i = 1, \ldots, n,\ \forall v \in V(\mathbf{X}_l), h_i^T(A(p)v + c(p)) < k_i$. ■

The computation from Proposition 8.7 allows us to restrict the parameters of $\mathscr{W}$ as described in Proposition 8.6, which guarantees that the quotient $T_{\mathscr{W}}/_{\sim}$ is, in fact, a bisimulation quotient of $T_{\mathscr{W}}$.

Example 8.4 We illustrate the parameter synthesis using bisimulations method presented in this chapter on the two-dimensional ($N = 2$) PWA system defined in Example 7.4. As before, we assume hyper-rectangular parameter sets, where each parameter is restricted to a range defined by ±50% of the fixed parameter values from Example 7.4 (parameters equal to 0 are restricted to the range $[-1.0E^{-2}, 1.0E^{-2}]$). Furthermore, the parameters for each region $l \in L$ are restricted to ensure that $\mathbf{X}$ is an invariant of all trajectories of the system. This can be accomplished using a computation similar to the one described in Proposition 8.7, where the entire $\mathbf{X}$ is used as the target (instead of a specific $\mathbf{X}_{l'}$) to restrict the parameters for each region $\mathbf{X}_l$. The quotient of this system under the uncertain parameters is illustrated in Fig. 8.7a.

We apply the method described in Sect. 8.5 in order to modify the parameters of the system and obtain a bisimulation quotient directly. A graphical representation of the resulting bisimulation quotient is shown in Fig. 8.7. As expected, a number of transitions of the system are lost when parameters are restricted to the smaller sets guaranteeing bisimulation. For example, the parameter set for region $\mathbf{X}_7$ retains around 50% of its original volume, while the one for region $\mathbf{X}_5$ retains only 0.017% when only parameters guaranteeing bisimulation are preserved. However, due to the language equivalence between the bisimulation quotient and the initial PWA system, the two systems can be used equivalently for model checking.



(a) Initial Quotient

(b) Bisimulation Quotient (after parameter synthesis)

Fig. 8.7 **a** Quotient of the PWA system from Example 7.4 with uncertain parameters restricted to ±50% of the fixed-parameter values. **b** Bisimulation quotient after the application of the parameter synthesis approach described in Sect. 8.5. Self-loop transitions at a state are indicated by a dot next to it. See Example 8.4 for details

8.6 Notes

In this chapter, which is based on [178], we showed that an iterative procedure can be used to obtain subsets of parameters for a PWA system, such that an LTL formula is satisfied (Problem 8.1).

The counterexample-guided pruning method presented in Sect. 8.1 focuses on the problem of enforcing that a finite transition system satisfies an LTL specification. The approach involves iteratively generating counterexamples through model-checking (Chap. 3) and removing these violating behaviours by pruning the transitions of the system. This approach is similar to counterexample-guided abstraction refinement (CEGAR) [44] in the sense that counterexamples are generated iteratively. However, while with CEGAR counterexamples are removed from the language of the system by refining the abstraction (unless a counterexamples is not spurious, in which case the analysis terminates), here we directly remove each counterexample by pruning transitions. More generally, enforcing a particular property for a finite system could be considered as an instance of the supervisory control problem (e.g., where a separate controller is synthesized to disable unwanted transitions and behaviors of the system) but here we consider autonomous systems, while the removal of transitions corresponds to a memoryless control strategy. The counterexample-guided pruning approach from this chapter is also different from other synthesis methods, since the states of the system remain unchanged and only its transitions are modified. This restriction is required in order to apply the approach for synthesis of PWA systems, where only the system's parameters (and the transitions they induced) can be restricted.

To apply this approach to PWA systems, in Sect. 8.2 we derived conditions guaranteeing that a given transition is present or not present in the quotient. For systems with additive parameter uncertainty only these conditions are exact, while otherwise the results are conservative (e.g., the sets of parameters preventing a transition between two states of the quotient are under-approximated). Two strategies were proposed for computing an under-approximation of parameter subsets guaranteeing that a transition between two regions (states in the quotient) is not possible. The computation from Proposition 8.3 could be more costly than the one from Proposition 8.2 but it allows some control over the quality of the approximation (e.g., by sampling of rotation groups, for example using the methods described in [133]). Using the properties from Sect. 8.2, the removal of a transition in the finite quotient can then be enforced in the PWA system by restricting the parameter sets to a suitable subset. In general, several transitions are removed at each step as a result of restricting the system's parameters, which was illustrated in Fig. 8.4.

In addition, weaker conditions were derived for restricting the parameters of a system to subsets that enforce transitions eventually rather than in a single step or, equivalently, enforcing that unwanted self-transitions at specific states of the quotient cannot be taken infinitely. While this computation can be applied only for a subset of specifications (LTL formulas without the next operator), this enables the synthesis of satisfying PWA systems in the cases where no subset of parameters can be found otherwise as in Example 8.4. The computation of such parameter sets is related to approaches for dealing with liveness properties (e.g., [24]) and stuttering behaviour in the quotient, which is discussed in more detail in Sect. 9.3.

In Sect. 8.5, we also derived conditions guaranteeing that only deterministic transitions exist in the quotient for certain subsets of parameters. This allowed us to restrict the parameters of a PWA system to subsets guaranteeing that a finite bisimulation quotient is constructed. While this restricts the possible behaviour of the system as illustrated in Example 8.4, it provides a strategy for obtaining finite bisimulation quotients, which can be used equivalently with the modified, concrete PWA systems (with restricted parameters) for analysis or model checking.

In the controls and formal methods communities, the parameter synthesis problem has been considered for a variety of systems. For example, in [86] parametric constraints were derived for guaranteeing the correctness of hybrid automaton models with respect to safety properties and a counterexample-guided parameter synthesis approach was developed for linear hybrid automata in [62]. In [53] a parameter synthesis approach was developed for dynamical systems modeled as nonlinear differential equations based on sensitivity analysis and search over initial conditions. Parameter synthesis has also been explored for other continuous time hybrid models such as piecewise multi-affine systems [22–25]. These approaches share many similarities with the methods for discrete time systems presented in this chapter, including the construction of finite abstractions as well as the identification of parameter sets inducing particular transitions or transient behaviors in such quotients.

The counterexample-guided pruning approach employed in this chapter for finite transition systems to drive the parameter synthesis in Sect. 8.4 is complete (even though our solution to the overall PWA parameter synthesis method is conservative) but computationally intensive as it explores the entire computation tree (where each node corresponded to a PWA system with different parameter sets and quotient) exhaustively. As formulated, the method performs best when the system has "little" violating behavior consisting of short executions, which can be removed after the generation of a small number of counterexamples. However, the conditions relating the parameter ranges of a PWA systems to transitions in the resulting quotient can also serve as a foundation for more efficient approaches, based on improved strategies for enforcing the satisfaction of an LTL property in a finite transition system or various heuristics. For example, in [24] a parameter synthesis approach similar to a binary search was developed. However, this approach cannot be applied directly here, since the set of parameters cannot always be partitioned into a subset inducing a given transition and one that does not due to the use of under-approximations. Also, the parameter sets for each state region were independent rather than linked through a common parameter. Alternatively, parallel model checking methods could improve the scalability of parameter synthesis approaches as demonstrated in [17].

The algorithms presented in this chapter were implemented as a software tool for Parameter Synthesis for Piecewise Affine Systems ParSyPAS, which is freely downloadable at http://hyness.bu.edu/software. The tool is built under MATLAB, and uses our in-house LTL model checker described in [103], LTL2BA [65] for the conversion of an LTL formula to a Büchi automaton, and the MPT toolbox [113] for polyhedral operations. The evaluation presented in Example 8.4 was performed on a 3.6 GHz machine with 32 GB of memory and required around 70 min for the synthesis of a satisfying PWA system.