12th IFAC/IEEE Workshop on Discrete Event Systems
Cachan, France. May 14-16, 2014

Natural Projections for the Synthesis of Non-Conflicting Supervisory Controllers

Thomas Moor *

* *Lehrstuhl für Regelungstechnik*
Friedrich-Alexander Universität Erlangen-Nürnberg, Germany

Abstract: A common strategy in the design of discrete-event systems is to apply synthesis algorithms not to the actual plant model but to an abstraction that is realised on a significantly smaller state set. Depending on the control objectives, certain conditions are imposed on the plant and on the abstraction, in order to end up with an appropriate controller. A well known result from the literature is that abstractions obtained by a so called *natural observer* can be used for the purpose of non-blocking supervisory control. Despite additional favourable properties of natural observers regarding state count and composed plants, as a condition for non-blocking supervisory control it is restrictive, i.e., sufficient but not necessary. This contrasts the sufficient and necessary condition developed in this paper.

Keywords: Discrete-event systems, supervisory control, abstraction-based synthesis, liveness properties.

1. INTRODUCTION

When the plant model provides more detail than required for the controller design problem at hand, one may resort to an appropriate plant abstraction instead. A crucial question in such an *abstraction-based controller design* is whether the resulting controller enforces relevant control objectives not only for the abstraction but also for the original plant model.

More specifically, we consider the situation where the plant model is given as a formal language and the *natural projection* to strings of *high-level events* is considered as a candidate for an abstraction; see also (Feng and Wonham, 2008, 2010). This setting applies to the design of hierarchical control architectures when a group of plant components, each subject to low-level control (Ramadge and Wonham, 1987, 1989), are composed and the subsequent task is the synthesis of a supervisor that addresses cooperative behaviour, specified w.r.t. high-level events. For computational procedures, including the choice of a suitable high-level alphabet, see e.g. (Schmidt et al., 2008; Feng and Wonham, 2010). In the present paper, we rephrase the question, whether the abstraction-based design solves the original problem as a requirement imposed on the high-level alphabet and we develop an implementable test to verify this requirement.

Our study relates to (Wong and Wonham, 1996), where within a general framework the notion of an *observer* is defined and proven to be a sufficient condition for the purpose of non-blocking hierarchical controller synthesis. Variations of the *natural observer* property that explicitly take into account controllability are presented in (Feng and Wonham, 2008) and address minimal restrictive hierarchical supervision (Schmidt and Breindl, 2011). In (Malik et al., 2007), it is shown that the observer property is not only sufficient but also necessary to obtain a conflict equivalent abstraction used for *compositional non-blocking verification*. The present paper is a further development of the reachability analysis presented in (Moor et al., 2013). In contrast to the earlier results, the novel condition obtained in the present paper is not only sufficient but also necessary for *non-conflicting controller synthesis*, i.e., we characterise precisely those projections, for which an abstraction-based controller design is guaranteed to exhibit a non-conflicting closed-loop behaviour.

The paper is organised as follows. Preliminaries and notational conventions are given in Section 2 and prepare for the technical problem statement in Section 3. To obtain a characterisation of a non-conflicting closed-loop configuration, Section 4 relates individual conflicts to the minimal restrictive solution of a particular controller synthesis problem. Consequences for the situation of regular languages are drawn in Section 5 to provide the basis for a software implementation. Finally, Section 6 interprets the results in the context of the reachability analysis proposed in (Moor et al., 2013).

2. PRELIMINARIES AND NOTATION

Let Σ be a *finite alphabet*, i.e., a finite set of symbols $\sigma \in \Sigma$. The *Kleene-closure* Σ^* is the set of finite strings $s = \sigma_1\sigma_2\cdots\sigma_n$, $n \in \mathbb{N}$, $\sigma_i \in \Sigma$, and the *empty string* $\epsilon \in \Sigma^*$, $\epsilon \notin \Sigma$. If, for two strings $s, r \in \Sigma^*$, there exists $t \in \Sigma^*$ such that $s = rt$, we say r is a *prefix* of s, and write $r \leq s$.

A *formal language* (or short a *language*) over Σ is a subset $L \subseteq \Sigma^*$. Given a language $L \subseteq \Sigma^*$, the equivalence relation $[\equiv_L]$ on Σ^* is defined by $s' \,[\equiv_L]\, s''$ if and only if $(\,\forall\, t \in \Sigma^*\,)[\, s't \in L \leftrightarrow s''t \in L\,]$. The language L is *regular* if $[\equiv_L]$ has only finitely many equivalence classes.

The *prefix* of a language $L \subseteq \Sigma^*$ is defined by $\mathrm{pre}\, L := \{r \in \Sigma^* \,|\, \exists\, s \in L : r \leq s\}$. A language L is *prefix-closed* (or short *closed*) if $L = \mathrm{pre}\, L$. A language K is *relatively closed w.r.t. L* if $K = (\mathrm{pre}\, K) \cap L$. The languages L and K are *non-conflicting* if $\mathrm{pre}\,(L \cap K) = (\mathrm{pre}\, L) \cap (\mathrm{pre}\, K)$. The prefix operator distributes over arbitrary unions of languages.

For the *observable events* $\Sigma_\mathrm{o} \subseteq \Sigma$, the *natural projection* $\mathrm{p_o} : \Sigma^* \to \Sigma_\mathrm{o}^*$ is defined iteratively: (1) let $\mathrm{p_o}\epsilon := \epsilon$; (2) for $s \in \Sigma^*$, $\sigma \in \Sigma$, let $\mathrm{p_o}(s\sigma) := (\mathrm{p_o}s)\sigma$ if $\sigma \in \Sigma_\mathrm{o}$, or, if $\sigma \notin \Sigma_\mathrm{o}$, let $\mathrm{p_o}(s\sigma) := \mathrm{p_o}s$. The set-valued inverse $\mathrm{p_o^{-1}}$ of $\mathrm{p_o}$ is defined by $p_o^{-1}(r) := \{s \in \Sigma^* \mid p_o(s) = r\}$ for $r \in \Sigma_o^*$. When applied to languages, the projection distributes over unions, and the inverse projection distributes over unions and intersections. The prefix operator commutes with projection and inverse projection.

The projection $p_o : \Sigma^* \to \Sigma_o^*$ is a *natural observer* for a language $L \subseteq \Sigma^*$, if for all $s \in \mathrm{pre}\, L$ and all $u \in \Sigma_o^*$ with $(p_o s)u \in p_o L$ there exists $t \in \Sigma^*$ such that $st \in L$ and $p_o t = u$; see e.g. Feng and Wonham (2010).

Given two languages $L,\ K \subseteq \Sigma^*$, and a set of *uncontrollable events* $\Sigma_{uc} \subseteq \Sigma$, we say K is *controllable w.r.t.* L, if $(\mathrm{pre}\, K)\Sigma_{uc} \cap (\mathrm{pre}\, L) \subseteq \mathrm{pre}\, K$. Note that, in contrast to e.g. (Ramadge and Wonham, 1987) but in compliance with e.g. (Cassandras and Lafortune, 2008), this variant of controllability does not insist in $K \subseteq L$. Controllability, closedness and relative closedness are each retained under arbitrary union.

Unless otherwise noted, the alphabets Σ, Σ_c, Σ_{uc}, Σ_o and Σ_{uo} refer to the *common partitioning* $\Sigma = \Sigma_c \dot\cup \Sigma_{uc} = \Sigma_o \dot\cup \Sigma_{uo}$ in controllable, uncontrollable, observable and unobservable events, respectively.

3. PROBLEM STATEMENT

For the purpose of this paper, let the *plant* and the *controller* be represented by formal languages $L \subseteq \Sigma^*$ and $H \subseteq \Sigma^*$, respectively, to obtain the *closed-loop behaviour* $K \subseteq \Sigma^*$ by intersection, i.e., $K = L \cap H$. The following definition imposes conditions on the controller for a well-posed closed-loop configuration.

Definition 1. Given a *plant* $L \subseteq \Sigma^*$, $\Sigma = \Sigma_c \dot\cup \Sigma_{uc}$, a *controller* $H \subseteq \Sigma^*$ is *admissible w.r.t.* L, if

(i) H is prefix-closed;
(ii) H is controllable w.r.t. L; and,
(iii) L and H are non-conflicting. □

It is readily verified that a closed-loop behaviour $K \subseteq L$ can be achieved by an admissible controller H if and only if K is controllable w.r.t. L and relatively closed w.r.t. L. This corresponds to *non-blocking supervision* as originally proposed by Ramadge and Wonham (1987). There, control is exercised by a causal feedback map $V\colon \mathrm{pre}\, L \to \Gamma$, which maps the respective past string $s \in \mathrm{pre}\, L$ to a control pattern $\gamma = V(s)$, $\Sigma_{uc} \subseteq \gamma \subseteq \Sigma$, to indicate the set of enabled successor events after the occurrence of s. In this paper, the controller H is interpreted as a representation of the feedback map V, and we omit explicit references to V in the subsequent development.

When a *language inclusion specification* $E \subseteq L$ is given, controller design amounts to the computation of the supremal achievable closed-loop behaviour $K^{\uparrow} \subseteq E$ in order to extract a corresponding controller $H := \mathrm{pre}\, K^{\uparrow}$; see e.g. (Wonham and Ramadge, 1987) for a computational procedure. Now consider the case, where the controller can only observe events from a restricted alphabet $\Sigma_o \subseteq \Sigma$. This paper takes the perspective of *hierarchical control*, see e.g. (Wong and Wonham, 1996), where one motivation in the deliberate restriction of observable events is to gain computational benefits. In this setting, one may assume that any aspects of the specification that relates to unobservable events has been dealt with by a low-level controller and that the specification at hand exclusively refers to Σ_o, i.e., $E = p_o^{-1} p_o E$. It is then proposed to synthesise an admissible controller $H_o \subseteq \Sigma_o^*$ for the projected plant $L_o := p_o L \subseteq \Sigma_o^*$ to satisfy the projected specification $E_o := p_o E$. In this approach, L_o is interpreted as an *abstraction* of the plant L, and, in turn, $H := p_o^{-1} H_o$ as an *implementation* of the *high-level controller* H_o to operate on the actual plant L. By construction, we obtain

$$L \cap H \subseteq p_o^{-1}(L_o \cap H_o)\,, \qquad L_o \cap H_o = p_o(L \cap H)\,,$$

where the latter equality is referred to as *hierarchical consistency*; see also (Zhong and Wonham, 1990). In particular, the actual closed-loop behaviour $K = L \cap H$ satisfies the language inclusion specification:

$$K = L \cap H \subseteq p_o^{-1}(L_o \cap H_o) \subseteq p_o^{-1} E_o = E\,.$$

It must be noted, that in the worst case the number of states required to realise L_o is even larger when compared to L; see (Wong, 1998). However, for relevant applications a substantial reduction of the required state set can be observed. In such a prospective situation, there remains the question whether admissibility of the high-level controller H_o implies admissibility of the implementation $H := p_o^{-1} H_o$. This question is readily rephrased as a formal requirement imposed on the abstraction.

Definition 2. Given a plant $L \subseteq \Sigma^*$ with the common alphabet partitioning, the plant abstraction $L_o := p_o L$ is *consistent for the purpose of controller design* (or short *consistent*), if admissibility is retained under implementation; i.e., if for all $H_o \subseteq \Sigma_o^*$, $H := p_o^{-1} H_o$, the following implication holds:

$$H_o \text{ is admissible w.r.t. } L_o \implies H \text{ is admissible w.r.t. } L\,.$$ □

Regarding the individual properties closedness, controllability and non-conflictingness, we recall well-known facts from the literature.

Proposition 3. Given a plant $L \subseteq \Sigma^*$ with the common alphabet partitioning, consider the abstraction $L_o := p_o L$, a controller candidate $H_o \subseteq \Sigma_o^*$ and its implementation $H := p_o^{-1} H_o$. Then each of the following three implications holds true individually:

$$H_o \text{ is prefix-closed} \implies H \text{ is prefix-closed}\,;$$

$$H_o \text{ is controllable w.r.t. } L_o \implies H \text{ is controllable w.r.t. } L\,;$$

and, provided that p_o is a natural observer for L,

$$L_o \text{ and } H_o \text{ are non-conflicting} \implies L \text{ and } H \text{ are non-conflicting}\,.$$

Proof. For the first implication, recall that the prefix operator $\mathrm{pre}\,(\cdot)$ and the projection $p_o(\cdot)$ commute. The second and the third implication are consequences of the more general results given in (Zhong and Wonham, 1990), Theorem 4.1, and in (Wong and Wonham, 1996), Theorem 6, respectively. For a direct proof addressing the specific situation at hand, see also (Moor et al., 2013). □

In particular, the above proposition identifies the natural observer property as a *sufficient condition* for the consistency of an abstraction.

Theorem 4. If for a plant $L \subseteq \Sigma^*$ with the common alphabet partitioning the projection $p_o : \Sigma^* \to \Sigma_o^*$ is a natural observer, then the abstraction $L_o := p_o L$ is consistent for the purpose of controller design. □