Control Optimality for Ordinary Petri Nets

Vasiliu A.-I. † Alla H. ‡
Department of Automatic Control, GIPSA Lab, 961 Rue de la Houille Blanche, BP 46, F - 38402 St. Martin d’Heres Cedex, France
† andra-ionna.vasiliu@gipsa-lab.grenoble-inp.fr
‡ hassane.alla@gipsa-lab.grenoble-inp.fr

Abstract: This paper addresses the problem of optimal control in the context of ordinary Petri net (PN) models. The meaning and importance of both formal and structural PN control optimality is discussed. A novel approach, based on a spacial view of the system’s state space, is used to determine control constraints that insure optimality. A simplification technique producing the minimal optimal control solution is proposed, and outlined in an algorithm. Finally, the approach is illustrated via an example.

Keywords: Discrete-event systems; Controller synthesis; Petri-nets; Optimal control; Marking invariant

1. INTRODUCTION

The two fundamental objectives of supervisory control, given a manufacturing system and the specification describing its desired performance, are (1) to limit the behaviour of the resulting closed-loop system to the confines stated in the specification, and (2) to ensure the system is submitted to no unnecessary restrictions. Optimal supervisory control may only be achieved when the condition of maximally permissive behaviour is also observed. A third, more practical, focus for supervisory control is that of structural minimalism. It is highly desirable that the control solution be as simple as possible, while maintaining optimality. We shall, in the following of this paper, refer to this endeavor towards straightforwardness as structural optimality, and it is on this, as well as on traditional optimality, that this paper is centered.

When referring to discrete-event systems (DES), two very general techniques may be used to obtain optimality, depending on the type of model employed. The general DES synthesis theory developed by Ramadge and Wonham (Ramadge and Wonham (1987), Wonham and Ramadge (1987)) renders the maximally permissive controller for finite state machine (FSM) models. Unfortunately, the FSM’s high sensitivity to the problem of combinatory state explosion makes this approach little suitable for real-life-sized systems. For this reason alternative DES modeling frameworks have been developed, and of these one of the most popular is the Petri Net (PN). One of the greatest advantages offered by the PN is its scalability and versatility. The richness of the available modeling structures allows for a much more intuitive and elegant modeling alternative for large-scale, complex systems, with stuffy FSM models. The straight-forward representation of the various sub-processes usually involved in such structures and of their mutual correlations ensures a more efficient depiction of the distributed activities within the system, while the problem of combinatory state explosion is greatly diminished, and the implementation alternatives are more varied. The combination of all these factors has made PN models very attractive, when dealing with large-scale systems, and thus the need for a supervisory-controller synthesis method based on this modeling framework arose. However, the problem of finding an optimal controller synthesis method applicable to any type of PN model has proven to be a redoubtable one.

A hybrid controller-synthesis method, coupling automaton supervisors to PN-modeled processes via inhibitory arcs was proposed in Uzam and Wonham (2006). While the control solution is optimal and applicable to any type of bounded PN, the resulting hybrid model is often of considerable size, in spite of the reduction techniques applied to both the PN plant-model and the supervisor. Ghaffari et al. (2003a) advances a maximally permissive PN supervisor synthesis method combining the RW technique with the theory of regions. Unfortunately, there is no significant optimization at the structural level, and therefore the complexity of the final controller can be significant.

A feedback-control synthesis method is given in Holloway and Krogh (1990), and Holloway et al. (1996). This approach allows the computation of a maximally permissive closed-loop controller without having to generate or analyze the system’s state space. Unfortunately, the resulting control predicates are often very complex, despite the various simplification techniques advanced in Kumar and Holloway (1996), Ghaffari et al. (2003b), Dideban and Alla (2006), and Vasiliu et al. (2009).

The place-invariants controller synthesis method introduced in Yamalidou et al. (1996) focuses on computing a place-invariant-based PN controller that enforces a given set of linear constraints placed on the marking behaviour of a PN model. The method has become very popular, due to its clarity and computational efficiency. Problems arise, however, when the model-controller synchronization is made via uncontrollable transitions, in which case the...
observance of the constraints is no longer certain. A solution to this problem for the specific case of safe conservational PN models, as well as several constraint-simplification techniques, are provided in Dideban and Alla (2005) and Dideban and Alla (2007), and later extended to deal with the more general case of safe PNs in Dideban and Alla (2008) and Vasiliiu and Alla (2010b). By the same token, the internal report Vasiliiu and Alla (2010a) offers a general optimal control solution applicable to any ordinary (safe or non-safe) PN model. As this method might still produce quite complex controllers, our work hence has been focused on minimizing the number of constraints, in order to assure the highest possible level of structural optimality. The present paper reviews our results and complements them with an efficient simplification technique ensuring the minimal optimal controller is obtained.

The primary goal of this paper is that of providing the means to obtaining the minimal set of control conditions allowing the successful computation, via the place-invariants synthesis method, of a maximally permissive controller for any given ordinary PN model. Section 2 discusses the meaning of control optimality and its implications on the construction of suitable constraints, in the context of ordinary PN models. A constraint is perceived as optimal, where optimality has the sense of maximal permissiveness, if its effects are localized to and only to members of the set of forbidden states. A simple example, to be used throughout the paper, is introduced at this point to better illustrate this problem. The technique for obtaining the simplest maximally permissive constraint is briefly brought to mind. Section 3 introduces a solution to the problem of structural optimality — ensuring that a minimum of control places and arcs are to be added to the system. A simplification algorithm is proposed for the determination of the minimal final set of constraints. The complete implementation of the approach is illustrated in section 4, while conclusions are inferred in section 5.

2. CONSTRAINTS AND CONTROL OPTIMALITY FOR ORDINARY PETRI NETS

Given a closed-loop system, the following two basic conditions must be obeyed at all times in order to achieve control optimality: (1) all behaviour inconsistent with the specification is forbidden, and (2) any other system behaviour is allowed. In terms of PN models, this means that each control constraint must exclusively inhibit its associated forbidden state. While the forbidding implication is trivial, ensuring the reciprocal has however proved to be somewhat more challenging. We have therefore concentrated our efforts on finding a means of writing constraints that would ensure that no unrequired restrictions are placed upon the system’s behaviour, first dealing with the instance of safe (binary) PN models (results presented in Vasiliiu and Alla (2010b)), and later with the much larger case of ordinary PNs (in Vasiliiu and Alla (2010a)). Once the equivalence between each forbidden state and its associated constraint has been established, controller synthesis may be performed for the given model using the place-invariants method developed in Yamalidou et al. (1996).

Before proceeding to present our approach, let us first briefly remind in an intuitive way the fundamental notions essential to its understanding.

Definition 1. Given the PN model $\mathcal{N}$ of a closed loop system, its reachable state space consists of the following subsets:

1. The set $M_A$ of authorized states, comprising all the states possible in the process and permitted by the specification.
2. The set $M_F$ of forbidden states, comprising any state possible in the process but prohibited by the specification.

2.1. Of $M_F$, the subset $M_B$, of border-forbidden states, presents particular interest for controller synthesis, as it comprises any forbidden state reachable from an authorized state via a controllable transition: $M_B = \{ M_{B_i} | M_{B_i} \in M_F$ and $\exists M_{A_i} \in M_A$ and $\exists T_e \in T_C$ such that $M_{A_i} \xrightarrow{T_e} M_{B_i} \}$, where $T_C$ is the set of controllable transitions of $\mathcal{N}$. □

Let $\mathcal{N}$ be a PN with $n$ places, $M_A$ and $M_B$ be the sets of authorized- and respectively border-forbidden states for $\mathcal{N}$, and $M_{B_i} = p_1^{m_1}...p_n^{m_n}, \ t \leq n$, $M_{B_i} \in M_B$, be a forbidden state of $\mathcal{N}$. The intuitive control solution, presented in Dideban and Alla (2005) and Dideban and Alla (2007), would be to simply limit the sum of tokens in the places involved in $M_{B_i}$, so that the forbidden state in question is never reached:

$$c_i: \sum_{j=1}^{t} M(P_{ij}) \leq k_i$$

where $t$ is the number of marked places of state $M_{B_i}$, $M(P_{ij}) = m_{ij}$ is the marking of place $P_{ij}$, and $k_i = b_i - 1$ is the bound of the constraint, $b_i$ representing the sum of all tokens in the net in state $M_{B_i}$, $b_i = \sum_{j=1}^{t} M_{B_i}(P_{ij}) = \sum_{j=1}^{n} M_{B_i}(P_{ij})$. This type of constraint is however over-restrictive for optimality, as it provides no guarantee of the fact that it does not affect any authorized state. One such case is apparent in the following example:

**Example 2.** Let us consider a manufacturing system comprising two machines of type $M_1$, one machine of type $M_2$, and three robots. The processing capacity of each

![Fig. 1. PN models of the process and specification](image-url)
The machine is of one part at a time. Only the beginning of each processing task (event $b_i$) is controllable. The PN model of the machines is presented in Figure 1a. Once a machine has finished processing a part (uncontrollable event $c_i$), a robot begins its unloading. At the end of the discharge operation (event $d_i$), the robot transfers the processed part into a buffer, and then returns to its initial (waiting) state. (The end of the transfer is signaled by the apparition of the uncontrollable event $r$.) The specification allows for only three robots. Figure 1b presents the PN model of the specification. The complete system’s schema and the PN model of its desired closed-loop performance are presented in Figure 2; the reachability graph is given in Figure 3.

Using the formula (1) we obtain the following control condition for the forbidden state $M_{B_1} = P_1P_2P_4P_9$:

$$c_{B_1} = \sum_{j=1}^{4} M(P_{B_1j}) = M(P_1) + M(P_2) + M(P_4) + M(P_9) \leq 5$$

It may be easily observed, that this constraint also forbids access to the authorized state $P_1P_4P_9$.  

From the desire to ensure control optimality there arises, therefore, the need for an improved constraint determination technique, that would account for both the forbidding– and the permissive– aspect of optimality. To this end, we have proposed examining the problem from a spacial point of view. In the interest of clarity, we shall briefly recall the results here.

The $n$-dimensional state space of a PN model essentially comprises two regions, connected to the sets of authorized– and forbidden– states. An affine hyperplane constitutes the border between the two, and from its equation a valid control constraint may be derived:

$$c_i : a_{i_1}m_1 + a_{i_2}m_2 + \ldots + a_{i_n}m_n \leq a_{i_{n+1}}$$

where $c_i$ is the constraint forbidding state $M_{R_i}$, and $m_j = M(P_j)$. An intuitive 2-dimensional representation of this concept is given in Figure 4.

The set $A_i = [a_{i_j}], j = 1, n + 1$, of coefficients defining the constraint must be computed such that the two optimality conditions are observed: (1) the forbidden state $M_{B_j}$ is in violation of the constraint, and (2) any authorized state $M_{R_i} \in M_A$ verifies the constraint. Furthermore, these coefficients must be positive integers, as they will eventually be part of the controlled system’s place-invariants.

It follows that the set $A_i$ may be computed by solving the following system, under the initial conditions $a_{i_j} \geq 0$, $j = 1, n + 1$:

$$\begin{align*}
\sum_{j=1}^{n} a_{i_j} \cdot M_{A_i}(P_j) - a_{i_{n+1}} & \leq 0 \\
\sum_{j=1}^{n} a_{i_j} \cdot M_{A_i}(P_j) & \leq 0 \\
\sum_{j=1}^{n} a_{i_j} \cdot M_{A_i}(P_j) - a_{i_{n+1}} & > 0
\end{align*}$$

with $A_i = [a_{i_j}]$ satisfying system (3) under the initial conditions $a_{i_j} \geq 0, j = 1, n + 1$.

**Remark 3.** The set of constraints determined in this manner for a given PN model defines a state space that is equal to the model’s authorized state space and that is optimal, even in the event where the system-controller synchronization is done via uncontrollable transitions. 

### 3. Obtaining the Minimal Optimal Controller

A careful observation of the results obtained using the techniques presented in section 2 shows that in many cases some of the constraints are redundant, being covered by other, stronger constraints. Further simplification is therefore necessary, as, beyond the primary goal of optimal control, it is highly desirable to enforce a secondary level of optimality — the structural level — which stipulates that a minimum of control places and arcs are to be added to the model. While the number and complexity of the arcs is linked to the complexity of the constraints (aspect dealt with in section 2), the number of control places is directly determined by the final number of constraints used for the synthesis.

In order to have the simplest possible controller, we must therefore have the minimal number of constraints. Once having ascertained the fact that an optimal controller indeed exists (i.e. the system (3) is solvable for any $M_{B_i} \in M_B$), the forbidden states must also be analyzed by subsets in order to obtain the minimal solution. In the interest of efficiency, this analysis should be descending. Namely, it should firstly be investigated if
This technique is formally described in the following algorithm:

Algorithm 1. Given the sets \( M_A \) and \( M_B \) of authorized– and respectively, border–forbidden– states of a PN model, this algorithm finds the minimal set \( \mathcal{C}_F \) of control constraints needed to compute the optimal controller via the place-invariants method.

1. Verify that an optimal control solution exists (i.e. the system (3) is solvable \( \forall M_{B_j} \in M_B \)). If yes, go to step 2. Else go to step 9.
2. Initialize the subset to be investigated: \( S = M_B \).
3. If the current subset offers a solution \( A^S = [a^S_j] \), go to step 6.
4. For each subset $\mathcal{S} \subseteq \mathcal{S}$, with $\text{Card}[\mathcal{S}] \equiv \text{Card}[\mathcal{S}] - 1$

4.1. If a subset solution $A^{S_1} = [a_i^S]$ exists for $\mathcal{S}_i$, go to step 6.

5. For each subset $\mathcal{S} \subseteq \mathcal{S}$, with $\text{Card}[\mathcal{S}] \equiv \text{Card}[\mathcal{S}] - 1$

5.1. $\mathcal{S} = \mathcal{S}_i$ and go to step 4.

6. Add the current solution $c^\mathcal{S}$ to the final set of constraints $\mathcal{C}_F$.

7. Exclude from $M_B$ the subset for which a solution has already been found: $M_B = M_B - \mathcal{S}$.

8. Have all the forbidden states been investigated? If yes, go to step 9. Else go to step 2.

9. Stop.

The final set of constraints $\mathcal{C}_F$ computed with algorithm 1 ensures minimal optimal control in terms of number of control places and constraint complexity.

4. EXAMPLE

This section presents the implementation of the techniques presented in sections 2 and 3 on the manufacturing system presented in example 2.

The first step is to identify the sets of authorized- and border-forbidden states:

- $M_A = \{P_2P_1P_2P_3; P_1P_2P_1P_2P_3; P_1P_3P_4P_2P_3; P_2P_4P_2P_3; P_3P_1P_4P_2P_3; P_1P_5P_2P_2P_3; P_2P_1P_5P_2P_3; P_3P_1P_5P_2P_3; P_2P_1P_5P_2P_3; P_2P_1P_5P_2P_3; P_2P_1P_5P_2P_3; P_2P_1P_5P_2P_3; P_2P_1P_5P_2P_3; P_2P_1P_5P_2P_3; P_2P_1P_5P_2P_3; P_2P_1P_5P_2P_3; P_2P_6P_2P_3; P_3P_2P_2P_3; P_1P_2P_5P_2P_3; P_1P_2P_5P_2P_3; P_1P_2P_5P_2P_3; P_1P_2P_5P_2P_3; P_1P_2P_5P_2P_3; P_1P_2P_5P_2P_3; P_1P_2P_5P_2P_3; P_1P_2P_5P_2P_3; P_1P_2P_5P_2P_3; P_1P_2P_5P_2P_3; P_1P_2P_5P_2P_3; P_1P_2P_5P_2P_3; P_1P_2P_5P_2P_3; P_1P_2P_5P_2P_3; P_1P_2P_5P_2P_3; P_1P_2P_5P_2P_3; P_1P_2P_5P_2P_3; P_1P_2P_5P_2P_3; P_1P_2P_5P_2P_3; P_1P_2P_5P_2P_3; P_1P_2P_5P_2P_3; P_1P_2P_5P_2P_3; P_1P_2P_5P_2P_3; P_1P_2P_5P_2P_3; P_1P_2P_5P_2P_3; P_1P_2P_5P_2P_3; P_1P_2P_5P_2P_3; P_1P_2P_5P_2P_3; P_1P_2P_5P_2P_3; P_1P_2P_5P_2P_3; P_1P_2P_5P_2P_3; P_1P_2P_5P_2P_3; P_1P_2P_5P_2P_3; P_1P_2P_5P_2P_3; P_1P_2P_5P_2P_3; P_1P_2P_5P_2P_3; P_1P_2P_5P_2P_3; P_1P_2P_5P_2P_3; P_1P_2P_5P_2P_3; P_1P_2P_5P_2P_3; P_1P_2P_5P_2P_3; P_1P_2P_5P_2P_3; P_1P_2P_5P_2P_3; P_1P_2P_5P_2P_3; P_1P_2P_5P_2P_3; P_1P_2P_5P_2P_3; P_1P_2P_5P_2P_3; P_1P_2P_5P_2P_3; P_1P_2P_5P_2P_3; P_1P_2P_5P_2P_3; P_1P_2P_5P_2P_3; P_1P_2P_5P_2P_3; P_1P_2P_5P_2P_3; P_1P_2P_5P_2P_3; P_1P_2P_5P_2P_3; P_1P_2P_5P_2P_3; P_1P_2P_5P_2P_3; P_1P_2P_5P_2P_3; P_1P_2P_5P_2P_3; P_1P_2P_5P_2P_3; P_1P_2P_5P_2P_3; P_1P_2P_5P_2P_3; P_1P_2P_5P_2P_3; P_1P_2P_5P_2P_3; P_1P_2P_5P_2P_3; P_1P_2P_5P_2P_3; P_1P_2P_5P_2P_3; P_1P_2P_5P_2P_3; P_1P_2P_5P_2P_3; P_1P_2P_5P_2P_3; P_1P_2P_5P_2P_3; P_1P_2P_5P_2P_3; P_1P_2P_5P_2P_3; P_1P_2P_5P_2P_3; P_1P_2P_5P_2P_3; P_1P_2P_5P_2P_3; P_1P_2P_5P_2P_3; P_1P_2P_5P_2P_3; P_1P_2P_5P_2P_3; P_1P_2P_5P_2P_3; P_1P_2P_5P_2P_3; P_1P_2P_5P_2P_3;
minimal controller and illustrates the power of the technique we have presented. Other simplification techniques, such as the coverage table presented in Dideban and Alla (2007) or the invariant-based reduction techniques from Dideban and Alla (2005), may be adapted and used for the diminution of the initial set of constraints (given by the coefficient matrix $A$). All of these, however, work with an already determined set of constraints, rather than with the forbidden states themselves. None of them thus take into account the possibility of having a more general subset solution covering the individual constraints obtained for various disjunct subsets of $M_B$.

Once the final set of constraints determined, the controller is computed using the place-invariants method presented in Yamalidou et al. (1996):

$$W_C = -L^T \cdot W_P = \begin{bmatrix} -1 & 0 & 0 & -1 & 0 & 0 & 1 \end{bmatrix}$$

$$M_{C_{ini}} = k - L^T \cdot M_{P_{ini}} = [3]$$

The closed loop performance and reachability graph of the controlled system are given in figure 5.

5. CONCLUSION

This paper offers an original solution for the problem of control optimality in the context of an ordinary PN model. More specifically, a systematic and easily implementable method for writing optimal constraints allowing controller synthesis via the place-invariants method is proposed. The optimization of the shape of each constraint assures the attainment of the minimal complexity controller, while an effective simplification technique ensures the minimization of the number of control places.

REFERENCES


