Chapter 17
The On-Line Diagnosis of Time Petri Nets

René K. Boel and George Jiroveanu

17.1 Introduction

The Petri Net models that we analyse in the following consider the time as a quantifiable and continuous parameter whereas in an untimed Petri Net model the time is taken into account only via the partial order relation between the transitions that are executed in the plant.

As presented in Chapter 16 there are two main time extensions of Petri Nets (PNs) namely Time Petri Nets [21] and Timed Petri Nets [22]. The difference between the two is that in Time Petri Nets (TPNs) a transition can be fired after its enabling with a delay that belongs to a given time-interval and the execution takes no time to complete, while in Timed Petri Nets a transition fires as soon as possible (without delay) but its execution requires a certain amount of time to complete. Among the two we have chosen Time Petri Nets for modelling our plant since this formalism is convenient for expressing most of the temporal constraints regarding the execution and the duration of the events.

For the diagnosis problem we adopt the setting proposed in [13] and [23]. Thus, we consider the plant observation given via a subset of transitions (observable transitions) and the faults that should be detected are modelled by a subset of the unobservable (silent) transitions. We assume that the model is correct and there are no delays in receiving the plant observation. Moreover, the execution time of an observed transition is measured with perfect accuracy w.r.t. a global clock. In this

René K. Boel
EESA - SYSTeMS Research Group, Ghent University, Technologiepark 914,
Zwijnaarde 9052, Belgium
e-mail: rene.boel@ugent.be

George Jiroveanu
Transelectrica SA - Romanian National Power Grid Company, Brestei 5, 200581, Craiova, Romania
e-mail: george.jiroveanu@transelectrica.ro
setting, the goal of this chapter is to design an on-line algorithm that derives the plant diagnosis at time $\xi$ based on the known plant model and on the observation received up to time $\xi$.

As for untimed PNs all the interesting problems for the analysis of TPN models can be reduced to reachability analysis (e.g. for the diagnosis problem one must calculate all the feasible time-traces that obey the received observation to check whether or not they contain fault transitions). In a TPN, a transition $t$ can be executed in a given marking at a certain time if $t$ is enabled as in an untimed PN and additionally some timing constrains are satisfied. There are timing constrains that relate the delay between the time when $t$ has become enabled and the time when $t$ is allowed to be executed, as well as timing constrains regarding the execution time of $t$ before some other transitions that are also enabled in that marking are forced to be executed by reaching the maximum delay allowed by the model for their execution.

In general to decide if a time-trace $\tau^0$ is legal (valid) or not it is necessary to check if its untimed support trace $\tau$ is legal in the untimed PN support of the TPN model and additionally to check if the times when the transitions in $\tau$ are assumed to be executed provide a solution for a system of linear inequalities (called the characteristic system of $\tau$). To calculate all the legal time-traces in the TPN model requires to derive for each legal trace $\tau$ of the untimed PN model the entire set of solutions of its characteristic system. Since a transition in a TPN can fire at any time in its firing domain, TPN models have in general infinite state spaces because a state may have an infinite number of successor states. Methods based on grouping states that are equivalent under a certain equivalence relation into so-called state classes were proposed in [4, 5, 12, 29, 30] where it was shown in [4] that for bounded TPNs the state class graph is finite and it compactly represents the set of all sequences of transitions that can be executed. Thus, the potentially infinite state spaces of TPNs can be finitely represented and the analysis can be reduced to a decidable problem.

The on-line algorithms that we design can be briefly described as follows. First, we construct the state class graph up to the time $\xi$. Then for each observable transition considered in the state class graph we derive the earlier time and the latest time when it can be executed. If an observable transition is either executed sooner than allowed or it is not observed prior to the latest time it could have been executed, then the arc labelled by that transition is deleted. The arc is also deleted if some other observable transition has been executed in the plant. Otherwise an equality relation is added to the characteristic system to express the fact that the observable transition occurred at the time given by the received observation. Hence, there could be cases when one can infer that a fault has happened for sure in the plant even though no observation is received from the plant, since traces can be deleted from the state class graph when the latest execution time of an observable transition has elapsed.

It is known that the analysis of PNs is a computationally hard problem because of the state space explosion due to the interleaving of concurrent transitions (e.g. the time computational complexity to construct the reachability graph is exponential in the number of places of the PN model [9]). The same problem remains also for TPN models where additionally the computation requires to solve for each untimed trace a system of linear inequalities (that can be solved by a standard algorithm in