State Estimation and Fault Detection Using Petri Nets

Alessandro Giua

DIEE, University of Cagliari, Piazza d'Armi, 09123 Cagliari, Italy

email: giua@diee.unica.it

Abstract

This extended abstract serves as a support for the plenary address given by the author at the 32nd International Conference on Application and Theory of Petri Nets and Concurrency.

Published as:

A. Giua, "State Estimation and Fault Detection Using Petri Nets", *Applications and Theory of Petri Nets*: Proc. 32nd Int. Conf. on Applications and Theory of Petri nets 2008 (Newcastle, UK), June 20-24, 2011. Lecture Notes in Computer Science Vol. 6709, pp. 419-428, L.M. Kristensen and L. Petrucci (Eds.) Springer-Verlag, 2011. The original publication is available at www.springerlink.com.

I. State estimation and observability

The *state estimation problem* for a dynamical system consists in reconstructing the current and past values of its internal states from the knowledge of the current and past values of its external measurable outputs. If such a problem admits a solution, the system is said to be *observable*.

Observability is a fundamental property that has received a lot of attention in the framework of time driven systems (TDS), given the importance of reconstructing plant states that cannot be measured for monitoring and for implementing state feedback control laws. Although less popular in the case of discrete event systems (DES), the problem of state estimation has been discussed in the literature and has generated several original issues.

A first issue in this framework consists in the definition of a suitable output. In fact, TDS are naturally described using the paradigm of input–state–output, the outputs being algebraic functions of the states. On the contrary, DES are typically described in terms of states and events and it may possible to associate labels to states or to events (e.g., as in a Moore or Mealy machine). When a system is in a given state, or when an event occurs, the corresponding label is observed. Hence this suggests that there are at least two possible ways of defining the outputs of a DES. The most popular choice is that of considering event occurrences as outputs. This talk will mostly focus on approaches of this type.

As a second issue, let us observe that there have been two main approaches to state estimation for DES.

- The first approach, that we call *control theory approach*, assumes that systems are deterministic, i.e., the labeling function that assigns to each event a label is injective. This means that any event occurrence can be precisely detected: the only uncertainty in this case is the lack of information on the initial state of the system. This setting is close to the state estimation problem in TDS.
- The second approach, that we call *computer science approach*, assumes that systems are non deterministic, i.e., the labeling function that assigns to each event a label is non-injective and possibly erasive. This means that an event occurrence may not be distinguishable from the occurrence of another different event or may even remain undetected. This adds an additional degree of complexity to the problem of state estimation. Furthermore, it also poses a new problem of *sequence estimation* (as opposed to state estimation), i.e., the

problem of reconstructing the sequence of events that has occurred.

As a third issue, we mention that while the state estimation problem in TDS consists in producing an estimate $\hat{x}(t)$ of the actual state $x(t)$ at time t, in the DES framework what is actually computed is an enumerable set of *consistent states*, i.e., states in which the system may be given the observed output.

A fourth issue consists in the fact that in a DES setting the state estimation problem is often solved with a bounded delay, i.e., due to nondeterminism an observer may not be able to correctly estimate the current system state but may be able to reconstruct it only after a given number of observations have elapsed.

Traditionally the problem of state estimation for DES has been addressed using automata models, or related models of computation. For a DES modeled as nondeterministic finite automata, the most common way of solving the problem of partial observation is that of converting, using a standard *determinization* procedure, the nondeterministic finite automaton (NFA) into an equivalent deterministic finite automaton (DFA) where: (i) each state of the DFA corresponds to a set of states of the NFA; (ii) the state reached on the DFA after the word w is observed, gives the set of *states consistent with the observed word* w.

We first observe that an analogous determinization procedure as that used in the case of automata, cannot be used in the Petri net (PN) framework. In fact, a nondeterministic PN cannot be converted into an equivalent deterministic PN, because of the strict inclusions:

$$\mathcal{L}_{det} \subsetneq \mathcal{L} \subsetneq \mathcal{L}_{\lambda}$$

where

- $\mathcal{L}_{det}$ is the set of deterministic PN languages;
- $\mathcal{L}$ is the set of λ-free PN languages, namely, languages accepted by nets where no transition is labeled with the empty string;
- $\mathcal{L}_{\lambda}$ is the set of arbitrary PN languages where a transition may also be labeled with the empty string.

Furthermore, the main drawback of the automata based approaches is the requirement that the set of consistent states must explicitly be enumerated.

In this talk several approaches for the state estimation of PN models will be discussed and it will be shown that they offer several computational advantages with respect to the automata approaches. In particular the fact that the state of a net, i.e., its marking, is a vector and that the state space does not require to be enumerated is a major asset and paves the way for the development of efficient linear algebraic tools for state estimation.

II. Fault detection and diagnosability

A *fault* is defined to be any deviation of a system from its normal or intended behavior. *Diagnosis* is the process of detecting an abnormality in the system's behavior and isolating the cause or the source of this abnormality.

Failures are inevitable in today's complex industrial environment and they could arise from several sources such as design errors, equipment malfunctions, operator mistakes, and so on. The need of automated mechanisms for the timely and accurate diagnosis of failures is well understood and appreciated both in industry and in academia.

In this framework two different problems can be solved: the problem of *diagnosis* and the problem of *diagnosability*. Solving a problem of diagnosis means that we associate to each observed evolution of the outputs a diagnosis state, such as "normal" or "faulty" or "uncertain". Solving a problem of diagnosability is equivalent to determine if the system is diagnosable, i.e., to determine if, once a fault has occurred, the system can detect its occurrence with an observation of bounded length.

Failure detection problems have been addressed by several communities. Within systems theory, both in the framework of TDS and DES the approaches for solving these problems have often been based on methodologies developed to cope with state estimation and sequence estimation. Recent approaches to diagnosis and diagnosability of Petri net models will be discussed in the second part of this talk.

III. Relevant literature for state estimation and fault detection of discrete event systems

A. State estimation using automata

For systems represented as finite automata, Ramadge [32] was the first to show how an observer could be designed for a partially observed system.

Caines *et al.* [9] showed how it is possible to use the information contained in the past sequence of observations (given as a sequence of observation states and control inputs) to compute the set of states consistent with observation. In [10] the observer output is used to steer the state of the plant to a desired terminal state. The approach of Caines is based on the construction of an observer tree to determine the set of markings consistent with the observed behavior. A similar approach was also used by Kumar *et al.* [26] when defining observer based dynamic controllers in the framework of supervisory predicate control problems.

Özveren and Willsky [30] proposed an approach for building observers that allows one to reconstruct the state of finite automata after a word of bounded length has been observed, showing that an observer may have an exponential number of states.

Several works dealt with state and sequence estimation in a decentralized setting with communications.

In [43] Tripakis defined a property that he calls *local observability*. The idea is the following: a set of local sites observe, through their own projection masks, a word w of symbols that is known to belong to a language L. A language $K \subset L$ is locally observable if, assuming all local sites send to a coordinator all observed strings, the coordinator can decide for any w if the word belongs to K or to $L \setminus K$. This property was shown to be undecidable even when languages L and K are regular: this is due to the fact that the length of a word w can be arbitrarily long.

A very general approach for observability with communication has been presented by Barret and Lafortune [1] in the context of supervisory control, and several techniques for designing a possibly optimal communication policy have also been discussed therein. By optimal we mean that the local sites communicate as late as possible, only when strictly necessary to prevent the undesirable behavior.

In the previous approach communications are decided by the local observers and are triggered by the observation of an event. Ricker and Caillaud [34] have considered a setting where communications may also be triggered by the receiver, that requests information from a sender. Furthermore, they also discuss policies where communication occurs after prefixes of any of the behaviors involved in a violation of co-observability, not just those that may result in undesired behavior.

A dual problem, strictly related to observability, is *opacity*. A system is (current-state) opaque if an observer cannot determine if the (current) state of the system certainly belongs to a given set of secret states. See the work of Saboori and Hadjicostis [37], [38], [39] and of Dubreil *et al.* [15].

Finally, we also mention that several other works have addressed the related issue of control under partial observations. However, they are not mentioned here because only partially relevant to the problem of state estimation.

B. State estimation using Petri nets

Our group was one of the first to address the problem of state estimation for Petri nets and has explored different methodologies.

In [18] some preliminary concepts dealing with the observability properties of nets where all transitions are observable have been introduced. As a extension of this work, in [20] a procedure that produces an estimate of the state was proposed, while the special structure of Petri nets allows one to determine, using linear algebraic tools, if a given marking is consistent with the observed behavior without the explicit enumeration of the (possibly infinite) consistent set. This approach was also successfully applied to timed nets: a procedure to exploit the timing information to improve the marking estimate was given in [21].

In [19] λ-free labeled PNs, i.e., PNs where no transition is labeled with the empty string, were studied. The restrictive assumption was that nondeterministic transitions are *contact-free*, i.e., for any two nondeterministic transitions t and t' the set of input and output places of t cannot intersect the set of input and output places of t'. In this case it is possible to give a linear algebraic characterization of the set of consistent markings that depends on some parameters that can be recursively computed.

Nets with unobservable transitions, i.e., transitions labeled with the empty string, were studied in [12]. Here we introduced the notion of *basis marking*. The idea is that under very general conditions, namely the acyclicity of the unobservable subnet, it is possible to characterize the set of markings consistent with an observation in terms of sequences of minimal length. The markings reached by these sequences are called basis markings and all other markings consistent with the observation can be obtained from the knowledge of this smaller set.

A similar approach that also deals with distributed implementation was presented by Jiroveanu *et al.* in [25]. The problem of finding upper bounds on the number of markings that are consistent with an observed sequence of labels was addressed by Ru and Hadjicostis in [35]. Here they show that the number of consistent markings, i.e., the complexity of solving a state estimation problem, in a Petri net with nondeterministic transitions is at most polynomial in the length of the observation sequence, though it remains exponential in certain parameters of the net.

Finally a different approach was proposed by Meda–Campaña *et al.* [29], using interpreted Petri nets to model the system and the observer. The main idea is to start the observer with a marking bigger than the real one and then eliminate some tokens until the observer and system markings are equal. Interpreted Petri nets using a mix of transition firings and place observations have also been used by Ramírez-Treviño *et al.* in [33] where it was shown that observability defined as in [29] is equivalent to observability in [18] and it was shown how to construct an observer for binary interpreted Petri nets when the observability property is verified.

C. Diagnosis and diagnosability using automata

In the contest of DES several original theoretical approaches have been proposed using automata.

Lin [28] proposed a state-based DES approach to failure diagnosis. The problems of off-line and on-line diagnosis are addressed separately and notions of diagnosability in both of these cases are presented. The author gives an algorithm for computing a diagnostic control, i.e., a sequence of test commands for diagnosing system failures. This algorithm is guaranteed to converge if the system satisfies the conditions for on-line diagnosability.

The group of Lafortune [40], [41], [13] in a series of papers proposed a seminal approach to failure diagnosis. The system is modeled as an automaton in which failures are modeled by a (possibly strict) subset of the set of unobservable events. The level of detail in a discrete event model appears to be quite adequate for a large class of systems and for a wide variety of failures to be diagnosed. The approach is applicable whenever failures cause a distinct change in the system status but do not necessarily bring the system to a halt. In [40] and [41] Sampath *et al.* provided a systematic approach to solve the problem of diagnosis using *diagnosers*. They also gave a formal definition of diagnosability in the framework of formal languages and established necessary and sufficient conditions for diagnosability of systems.

Hashtrudi Zad *et al.* [23] presented a state-based approach for on-line passive fault diagnosis. In this framework, the system and the fault detection system do not have to be initialized at the same time. Furthermore, no information about the state or even the condition (failure status) of the system before the initiation of diagnosis is required. The design of the fault detection system, in the worst case, has exponential complexity. A model reduction scheme with polynomial time complexity is introduced to reduce the computational complexity of the design. Diagnosability of failures is studied, and necessary and sufficient conditions for failure diagnosability are derived.

We conclude mentioning several works dealing with diagnosis in a decentralized settings, possibly with communications or with a coordinator.

Debouk *et al.* [13] addressed the problem of failure diagnosis in DES with decentralized information. A coordinated decentralized architecture was proposed that consists of two local sites communicating with a coordinator that is responsible for diagnosing the failures occurring in the system. They extend the notion of diagnosability, originally introduced in [40] for centralized systems, to the proposed coordinated decentralized architecture. In particular, they specify three protocols that realize the proposed architecture and analyze the diagnostic properties of these protocols.

Boel and van Schuppen [4] addressed the problem of synthesizing communication protocols and failure diagnosis algorithms for decentralized failure diagnosis of DES with costly communication between diagnosers. The costs on the communication channels may be described in terms of bits and complexity. The costs of communication and computation force the trade-off between the control objective of failure diagnosis and that of minimization of the costs of communication and computation. The main result of this paper is an algorithm for decentralized failure diagnosis of DES for the special case of only two diagnosers.

D. Diagnosis using Petri nets

Among the first pioneer works dealing with PNs, we recall the approach of Prock that in [31] proposed an on-line technique for fault detection that is based on monitoring the number of tokens residing into P-invariants: when the number of tokens inside P-invariants changes, then the error is detected.

Srinivasan and Jafari [42] employed time PNs to model a DES controller and backfiring transitions to determine whether a given state is invalid. Time PNs have also been considered by Ghazel *et al.* [17] to propose a monitoring approach for DES with unobservable events and to represent the "a priori" known behavior of the system, and track on-line its state to identify the events that occur.

Hadjicostis and Verghese [22] considered PN models introducing redundancy into the system and using P-invariants to allow the detection and isolation of faulty markings. Redundancy into a given PN was also exploited by Wu and Hadjicostis [47] to enable fault detection and identification using algebraic decoding techniques. Two types of faults are considered: place faults that corrupt the net marking, and transition faults that cause an incorrect update of the marking after event occurrence. Although this approach is general, the net marking has to be periodically observable even if unobservable events occur. Analogously, Lefebvre and Delherm [27] investigated a procedure to determine the set of places that must be observed for the exact and immediate estimation of faults occurrence.

Note that all above mentioned papers assume that the marking of certain places may be observed. On the contrary other approaches, that we review in the rest of this subsection, are based on the assumption that no place is observable.

The three different approaches mentioned in the following [3], [2], [14] are characterized by the fact that they use on-line approaches to diagnosis. This may offer some advantages in terms of generality and adaptability, but may require heavy computations in real time.

Benveniste *et al.* [3] derived a net unfolding approach for designing an on-line asynchronous diagnoser. The state explosion is avoided but the computational complexity can be high due to the requirement of unfolding on-line PN structures.

Basile *et al.* [2] considered an on-line diagnoser that requires solving Integer Linear Programming (ILP) problems. Assuming that the fault transitions are not observable, the net marking is computed by the state equation and, if the marking has negative components, an unobservable sequence has occurred. The linear programming solution provides the sequence and detects the fault occurrences. Moreover, an off-line analysis of the PN structure may reduce the computational complexity of solving the ILP problem.

Dotoli *et al.* [14] proposed a diagnoser that works on-line as a means to avoid its redesign when the structure of the system changes. In particular, the diagnoser waits for an observable event and an algorithm decides whether the system behavior is normal or may exhibit some possible faults. To this aim, some ILP problems are defined and provide eventually the minimal sequences of unobservable transitions containing the faults that may have occurred. The proposed approach is a general technique since no assumption is imposed on the reachable state set that can be unlimited, and only few properties must be fulfilled by the structure of the net modeling the system fault behavior.

The state estimation approach founded on basis markings introduced by our group in [12] has also been extended to address diagnosis of Petri nets. In [8] the fundamental diagnosis approach was presented assuming that all observable transitions have a distinct label: the two notions of basis marking and justification allow one to characterize the set of markings that are consistent with the actual observation, and the set of unobservable transitions whose firing enables it. This approach applies to all net systems whose unobservable subnet is acyclic. If the net system is also bounded the proposed approach may be significantly simplified by moving the most burdensome part of the procedure off-line, thanks to the construction of a graph, called the *basis reachability graph*. In [6] the approach has been extended to the case of nets with observable transitions sharing the same label.

Ru and Hadjicostis [36] studied fault diagnosis in discrete event systems modeled by partially observed Petri nets, i.e., Petri nets equipped with sensors that allow observation of the number of tokens in some of the places and/or partial observation of the firing of some of the transitions. Given an ordered sequence of observations from place and transition sensors, the goal is to calculate the belief (namely, the degree of confidence) regarding the occurrence of faults belonging to each type. This is done transforming a given partially observed PN into an equivalent labeled PN, i.e., a net with only transition sensors.

Finally, we mention the distributed approach by Genc and Lafortune [16] where the net is modular and local diagnosers performs the diagnosis of faults in each module. Subsequently, the local diagnosers recover the monolithic diagnosis information obtained when all the modules are combined into a single module that preserves the behavior of the underlying modular system. A communication system connects the different modules and updates the diagnosis information. Even if the approach does not avoid the state explosion problem, an improvement is obtained when the system can be modeled as a collection of PN modules coupled through common places.

E. Diagnosability using Petri nets

It should be noted that none of the above mentioned approaches for the diagnosis of PNs deal with *diagnosability*, namely none of them provide a procedure to determine a priori if a system is *diagnosable*, i.e., if it is possible to reconstruct the occurrence of fault events observing words of finite length.

The first contribution on diagnosability of PNs is due to Ushio *et al.* [44]. The main idea is to extend the diagnosability approaches for automata presented in [40], [41] to deal with unbounded PNs, assuming that the set of places is partitioned into observable and unobservable places, while all transitions are unobservable. Starting from the net they build a diagnoser called *simple ω diagnoser* that gives sufficient conditions for diagnosability of unbounded PNs.

Chung [11] generalized the previous setting assuming that some of the transitions of the net are observable and showing that the additional information from observed transitions in general adds diagnosability to the analyzed system. Moreover starting from the diagnoser he proposed an automaton called *verifier* that allows a polynomial check of diagnosability for finite state models.

Wen and Jeng [45] proposed an approach to test diagnosability by checking the T-invariants of the nets. They used Ushio's diagnoser to prove that their method is correct, however they do not construct a diagnoser for the system to do diagnosis. Wen *et al.* [46] also presented an algorithm, based on a linear programming problem, of polynomial complexity in the number of nodes, to compute a sufficient diagnosability condition for DES modeled by PNs.

Jiroveanu and Boel [24] used a reduced automaton for testing diagnosability of labeled nets with unobservable transitions. The automaton generates the same language as the net reachability graph after projecting out all non-faulty unobservable transitions, and contains significantly less states than the reachability graph. The reduced automaton is efficiently constructed computing the minimal explanations of the fault and of the observable transitions.

Our group described a diagnosability test for the diagnosis approach founded on basis markings in [7]. The proposed approach applies to bounded nets with unobservable transitions and is based on the analysis of two graphs that depend on the structure of the net, including the faults model, and on the initial marking. The first graph is called *basis reachability diagnoser*, the second one is called *modified basis reachability graph*.

In [5] necessary and sufficient conditions for diagnosability of unbounded nets were given. The constructive procedure to test diagnosability is based on the analysis of the coverability graph of a particular net, called *verifier net*. To the best of our knowledge, this is the first available test that provides necessary and sufficient conditions for diagnosability of labeled unbounded Petri nets.

REFERENCES

[1] G. Barrett and S. Lafortune. Decentralized supervisory control with communicating controllers. *IEEE Trans. on Automatic Control*, 45(9):1620 –1638, September 2000.

[2] F. Basile, P. Chiacchio, and G. De Tommasi. An efficient approach for online diagnosis of discrete event systems. *IEEE Trans. on Automatic Control*, 54(4):748–759, April 2008.

[3] A. Benveniste, E. Fabre, S. Haar, and C. Jard. Diagnosis of asynchronous discrete event systems: A net unfolding approach. *IEEE Trans. on Automatic Control*, 48(5):714–727, May 2003.

[4] R.K. Boel and J.H. van Schuppen. Decentralized failure diagnosis for discrete-event systems with costly communication between diagnosers. In *Proc. 6th Work. on Discrete Event Systems*, pages 175–181, Zaragoza, Spain, October 2002.

[5] M.P. Cabasino, A. Giua, S. Lafortune, and C. Seatzu. Diagnosability analysis of unbounded Petri nets. In *Proc. 48th IEEE Conf. on Decision and Control*, pages 1267–1272, Shanghai, China, December 2009.

[6] M.P. Cabasino, A. Giua, M. Pocci, and C. Seatzu. Discrete event diagnosis using labeled Petri nets. An application to manufacturing systems. *Control Engineering Practice*, doi:10.1016/j.conengprac.2010.12.010, 2010.

[7] M.P. Cabasino, A. Giua, and C. Seatzu. Diagnosability of bounded Petri nets. In *Proc. 48th IEEE Conf. on Decision and Control*, pages 1254–1260, Shanghai, China, December 2009.

[8] M.P. Cabasino, A. Giua, and C. Seatzu. Fault detection for discrete event systems using Petri nets with unobservable transitions. *Automatica*, 46(9):1531–1539, September 2010.

[9] P.E. Caines, R. Greiner, and S. Wang. Dynamical logic observers for finite automata. In *Proc. 27th IEEE Conf. on Decision and Control*, pages 226–233, Austin, TX, USA, December 1988.

[10] P.E. Caines and S. Wang. Classical and logic based regulator design and its complexity for partially observed automata. In *Proc. 28th IEEE Conf. on Decision and Control*, pages 132–137, Tampa, FL, USA, December 1989.

[11] S.L. Chung. Diagnosing PN-based models with partial observable transitions. *International Journal of Computer Integrated Manufacturing*, 12 (2):158–169, 2005.

[12] D. Corona, A. Giua, and C. Seatzu. Marking estimation of Petri nets with silent transitions. *IEEE Trans. on Automatic Control*, 52(9):1695–1699, September 2007.

[13] R. Debouk, S. Lafortune, and D. Teneketzis. Coordinated decentralized protocols for failure diagnosis of discrete-event systems. *Discrete Events Dynamical Systems*, 10(1):33–86, January 2000.

[14] M. Dotoli, M.P. Fanti, and A.M. Mangini. Fault detection of discrete event systems using Petri nets and integer linear programming. In *Proc. of 17th IFAC World Congress*, Seoul, Korea, July 2008.

[15] J. Dubreil, Ph. Darondeau, and H. Marchand. Supervisory control for opacity. *IEEE Trans. on Automatic Control*, 55(5):1089–1100, May 2010.

[16] S. Genc and S. Lafortune. Distributed diagnosis of place-bordered Petri nets. *IEEE Trans. on Automation Science and Engineering*, 4(2):206–219, April 2007.

[17] M. Ghazel, A. Togueni, and M. Bigang. A monitoring approach for discrete events systems based on a time Petri net model. In *Proc. of 16th IFAC World Congress*, Prague, Czech Republic, July 2005.

[18] A. Giua. Petri net state estimators based on event observation. In *Proc. 36th IEEE Conf. on Decision and Control*, pages 4086–4091, San Diego, CA, USA, December 1997.

[19] A. Giua, D. Corona, and C. Seatzu. State estimation of λ-free labeled Petri nets with contact-free nondeterministic transitions. *Discrete Events Dynamical Systems*, 15(1):85–108, January 2005.

[20] A. Giua and C. Seatzu. Observability of place/transition nets. *IEEE Trans. on Automatic Control*, 49(9):1424–437, September 2002.

[21] A. Giua, C. Seatzu, and F. Basile. Observer based state-feedback control of timed Petri nets with deadlock recovery. *IEEE Trans. on Automatic Control*, 49(1):17–29, January 2004.

[22] C.N. Hadjicostis and G.C. Verghese. Monitoring discrete event systems using Petri net embeddings. *Lecture Notes in Computer Science*, 1639:188–207, 1999.

[23] S. Hashtrudi Zad, R.H. Kwong, and W.M. Wonham. Fault diagnosis in discrete-event systems: framework and model reduction. *IEEE Trans. on Automatic Control*, 48(7):1199–1212, July 2003.

[24] G. Jiroveanu and R.K. Boel. The diagnosability of Petri net models using minimal explanations. *IEEE Trans. on Automatic Control*, 55(7):1663–1668, July 2010.

[25] G. Jiroveanu, R.K. Boel, and B. Bordbar. On-line monitoring of large Petri net models under partial observation. *Discrete Events Dynamical Systems*, 18(3):323–354, September 2008.

[26] R. Kumar, V. Garg, and S.I. Markus. Predicates and predicate transformers for supervisory control of discrete event dynamical systems. *IEEE Trans. on Automatic Control*, 38(2):232–247, February 1993.

[27] D. Lefebvre and C. Delherm. Diagnosis of DES with Petri net models. *IEEE Trans. on Automation Science and Engineering*, 4(1):114–118, January 2007.

[28] F. Lin. Diagnosability of discrete event systems and its applications. *Discrete Event Dynamic Systems*, 4(2):197–212, May 1994.

[29] M.E. Meda-Campaña, A. Ramírez-Treviño, and A. Malo. Identification in discrete event systems. In *Proc. IEEE Int. Conf. on Systems, Man and Cybernetics*, pages 740–745, San Diego, CA, USA, October 1998.

[30] C.M. Özveren and A.S. Willsky. Observability of discrete event dynamic systems. *IEEE Trans. on Automatic Control*, 35(7):797–806, July 1990.

[31] J. Prock. A new tecnique for fault detection using Petri nets. *Automatica*, 27(2):239–245, February 1991.

[32] P.J. Ramadge. Observability of discrete-event systems. In *Proc. 25th IEEE Conf. on Decision and Control*, pages 1108–1112, Athens, Greece, December 1986.

[33] A. Ramírez-Treviño, I. Rivera-Rangel, and E. López-Mellado. Observer design for discrete event systems modeled by interpreted Petri nets. In *Proc. 2000 IEEE Int. Conf. on Robotics and Automation*, pages 2871–2876, San Francisco, CA, USA, April 2000.

[34] S.L. Ricker and B. Caillaud. Mind the gap: Expanding communication options in decentralized discrete-event control. In *Proc. 46th IEEE Conf. on Decision and Control*, pages 5924 –5929, New Orleans, LA, USA, December 2007.

[35] Y. Ru and C.N. Hadjicostis. Bounds on the number of markings consistent with label observations in Petri nets. *IEEE Trans. on Automation Science and Engineering*, 6(2):334–344, April 2009.

[36] Y. Ru and C.N. Hadjicostis. Fault diagnosis in discrete event systems modeled by partially observed Petri nets. *Discrete Events Dynamical Systems*, 19(4):551–575, December 2009.

[37] A. Saboori and C.N. Hadjicostis. Opacity-enforcing supervisory strategies for secure discrete event systems. In *Proc. 47th IEEE Conf. on Decision and Control*, pages 889–894, Cancun, Mexico, December 2008.

[38] A. Saboori and C.N. Hadjicostis. Opacity verification in stochastic discrete event systems. In *Proc. 49th IEEE Conf. on Decision and Control*, pages 6759–6764, Atlanta, GA, USA, December 2010.

[39] A. Saboori and C.N. Hadjicostis. Reduced-complexity verification for initial-state opacity in modular discrete event systems. In *Proc. 10th Work. on Discrete Event Systems*, Berlin, Germany, August/September 2010.

[40] M. Sampath, R. Sengupta, S. Lafortune, K. Sinnamohideen, and D. Teneketzis. Diagnosability of discrete-event systems. *IEEE Trans. on Automatic Control*, 40(9):1555–1575, September 1995.

[41] M. Sampath, R. Sengupta, S. Lafortune, K. Sinnamohideen, and D. Teneketzis. Failure diagnosis using discrete-event models. *IEEE Trans. on Control Systems Technology*, 4(2):105–124, March 1996.

[42] V.S. Srinivasan and M.A. Jafari. Fault detection and monitoring using time Petri nets. *IEEE Trans. on Systems, Man and Cybernetics*, 23(4):1155–1162, July/August 1993.

[43] S. Tripakis. Undecidable problems of decentralized observation and control on regular languages. *Information Processing Letters*, 90(1):21–28, 2004.

[44] T. Ushio, L. Onishi, and K. Okuda. Fault detection based on Petri net models with faulty behaviors. In *Proc. 1998 IEEE Int. Conf. on Systems, Man, and Cybernetics*, pages 113–118, San Diego, CA, USA, October 1998.

[45] Y. Wen and M. Jeng. Diagnosability analysis based on T-invariants of Petri nets. In *Proc. 2005 IEEE Int. Conf. on Networking, Sensing and Control*, pages 371– 376, Tucson, AZ, USA, March 2005.

[46] Y. Wen, C. Li, and M. Jeng. A polynomial algorithm for checking diagnosability of Petri nets. In *Proc. 2005 IEEE Int. Conf. on Systems, Man, and Cybernetics*, pages 2542– 2547, October 2005.

[47] Y. Wu and C.N. Hadjicostis. Algebraic approaches for fault identification in discrete-event systems. *IEEE Trans. Robotics and Automation*, 50(12):2048–2053, December 2005.