Construction of Model-based Diagnosis of Cyber-Attack in Cyber-Physical Systems Using Labeled Petri Nets

Romain Thibert, Gregory Faraut, Saïd Amari

Construction of Model-based Diagnosis of Cyber-Attack in Cyber-Physical Systems Using Labeled Petri Nets

Romain Thibert, Gregory Faraut, Saïd Amari
LURPA
ENS Paris-Saclay
Gif-sur-Yvette, France
name.surname@ens-paris-saclay.fr

***Abstract*—Cyber-Physical Systems (CPS) brought connectivity to factories, and with connectivity comes a risk of cyber attack. CPSs are vulnerable to malicious attacks in which an attacker is inserted between a process and its control unit. Some papers have proposed to model attacked CPSs with discrete event systems (DES), have successfully modeled attacked systems and characterised different types of attacks. Nevertheless, these papers tend to treat attacks after they have become problematic. The objective of this paper is to extend previous works on cyber attack in DES to build models for diagnosis of attacks. We will design a model using Labeled Petri Nets and construct a reachability graph of an attacked net in order to enrich a diagnosis model which will allow the detection of attacks before they completely destabilise the system. This construction is illustrated by an industrial example.**

***Index Terms*—Discrete event systems, Cyber attacks, Petri nets, Cyber physical systems, Diagnosis**

I. Introduction

Cyber-Physical Systems (CPSs) are intelligent systems, integrating control, communication and computing and they have a wide range of applications, including smart power grids, transportation systems, smart manufacturing and medical monitoring [1]. These CPSs are highly networked systems and are dependant on a heavy flow of information in wired and wireless networks. CPSs are one of the cornerstones of industry 4.0 thanks to their heavily networked nature. Having models deeply connected with systems allows continuous feedback between control and simulation within the CPS and prompt adaptability to new production conditions.

Due to the increasing communication flow, CPSs are vulnerable to malicious attacks which can gather, modify or nullify data exchanges [2]. Those intended modifications can lead the system into critical states in which the controller issues inappropriate orders resulting in the blockage of the process [3] or in physical damages to the system. The development of control strategies robust to these attacks is a crucial point. Cyber attacks are defined by [2] as *actions that exploit the system's vulnerabilities and result in some kind of damage*.

Attacks based on sensors modification to destabilise the system modeled by Discrete Event Systems (DES) have been studied in [4] and [5], these stealthy attacks, where the attacker's goal is to remain undetectable until the system is in a critical state, were also developed in [6]. These approaches design observers which assess the process' status and the possible occurrence of attacks in order to prevent them by pointing out vulnerabilities.

Other works model attacks and define defense strategies. In [7], a security module is designed to prevent the system from reaching a critical state by disabling events, ensuring the safety of the process. In [8], DES under supervisory control are modeled. An observer of the attacker and an observer of the attacked plant are synchronized to obtain an automaton called attack structure which allows to estimate attack effectiveness and detectability. Works have been done to model attacks and evaluate attack effectiveness on systems but few works have been done on diagnosis of attacks and one of the main concerns in this field is the distinction between faults and attacks that has still to be made. That last point is the major inconvenience from all cited works, a fault is only different from an attack by hypothesis.

When dealing with modelisation of cyber attacks in DES, authors have mainly focused on finite automata [7], [9]–[15] but few of them have focused on Petri nets modeling CPSs under attack [16]–[18]. In [16], labeled Petri nets are used to integrate attacks in models of nominal behaviors to build observers in order to compute effective attacks. In [17], a new formalism called output synchronized Petri nets is used to perform cost analysis of attacks and vulnerability detection in CPSs. In those works, attacks are still indiscernible from faults and attacks are detected too late, when the system has reached a destabilised state. A distinction can be made between finite automata and Petri nets on the size of the system during conception, with Petri nets it is possible to avoid a combinatory explosion at the conception step.

In this paper we propose to build a model for attacked systems based on the work of [16] to model the attacks and to perform online diagnosis of attacks as they could be considered as a new class of defects for CPSs. Using a reachability graph and a deadlock analysis to determine attack trajectories a model for diagnosis will be built. This aims to develop techniques to diagnose attacks differently from faults,

as faults can be treated when production stops but attacks need to be circumvented as soon as they are detected to prevent the attacker from taking full control of the process.

This paper will be structured as follows. In section II some theoretical basis will be introduced before exposing the hypothesis of the problem. The proposed modelisation will be detailed and discussed in section III and the paper will be concluded in section IV.

II. Preliminary and Hypothesis

In this section, Petri nets models are described. If need be, further information can be found in [19], [20].

A. Petri Nets

A Petri net (PN) is defined as a structure $N = \langle P, T, Pre, Post \rangle$, where:

- P is a set of m places
- T is a set of n transitions
- $Pre : P \times T \to \mathbb{N}$ is the pre-incidence matrix
- $Post : P \times T \to \mathbb{N}$ is the post-incidence matrix

$Pre(p,t) = w$ (resp. $Post(p,t) = w$) means that an arc goes from place p to transition t (resp. from t to p) with a weight $w \in \mathbb{N}^*$. $C = Post - Pre$ is the incidence matrix. ${}^\bullet p$ (${}^\bullet t$) and $p^\bullet$ ($t^\bullet$) are respectively referring to the pre-set and post-set of a place p (transition t), e.g. ${}^\bullet t = \{p \in P | Pre(p,t) \neq 0\}$. $M \in \mathbb{N}^m$ represents the marking vector of the net.

A transition t is enabled at M iff $M \geq Pre(.,t)$. After a transition has fired the new marking vector is given by $M' = M + C(.,t)$, with $M[t\rangle M'$. A firing sequence is a sequence of transitions $\sigma = t_1 t_2 \dots t_k$ such that $M[t_1\rangle M_1[t_2\rangle \dots [t_k\rangle M_k$ and this is written $M[\sigma\rangle M_k$. If the initial marking of the net is M_0, the relation

$$M = M_0 + C.\sigma$$

is called the state equation of the net.

A net is said to be k-bounded if each component of M is lower than k i.e. $\forall p \in P,\ M(p) \leq k$. If $k = 1$ the net is *one-bounded*.

B. Labeled Petri Nets

In addition to PN, a language is needed to synchronize a model with the actual process. With the addition of this alphabet, a new Petri net class is built. A Labeled Petri Net (LPN) is defined as a structure $L = \langle N, M_0, E, l \rangle$

- N is a Petri Net
- M_0 is the initial marking of the net
- E is an alphabet
- $\ell : T \to E \cup \{\varepsilon\}$ is the labelling function

To analyse the behavior of systems represented by PN, reachability graphs are commonly used. It represents all the marking accessible from M_0 in nodes and arcs labeled with transitions. In the case of LPN, elements from E will be on arcs. This graph has the same structure as an automaton which is defined as a finite automaton is defined as a 5-tuple $G = \langle Q, E, \sigma, x_i, X_m \rangle$ where Q is a finite set of states, E is an alphabet, $\sigma : Q \times E \to Q$ is the transition function, x_i is the initial state and X_m are the marked states.

C. Attack Language

Different alphabets are manipulated in this paper, their definitions are listed in this subsection.
Let E be the set of events of the process. The set of compromised events will be denoted $E_{com} \subseteq E$ and will contain all the events that the attacker can manipulate. Furthermore, events that can be erased by the attacker will be part of a set E_{era}, and events that can be inserted will be part of a set E_{ins}. With this partitioning we have $E_{com} = E_{ins} \cup E_{era}$, E_{ins} and E_{era} are not necessarily disjoint as illustrated in Fig.1.

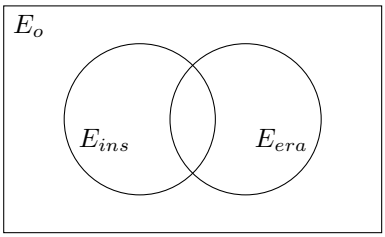

Fig. 1. Attack language definition

Then we define two sets of events, copies of E_{era} (resp. E_{ins}) handled only by the attacker, in order to distinguish those events a character '+' (resp '−') is added, then $E^+ = \{e^+ \mid e \in E_{ins}\}$ and $E^- = \{e^- \mid e \in E_{era}\}$. Because E^+ and E^- are copies of the two previous sets but differentiated by an added character, they are disjoint by construct. An occurrence of an event e^+ means that the attacker adds an observation e for the operator, an occurrence of e^- means that the attacker has erased the observation of e for the operator. Hence we define the set of events possible during an attack $E_a = E \cup E^+ \cup E^-$. When talking about languages, $|u|$ will denote the length of word u.

D. Hypothesis

In this work, we will consider the following hypothesis.

- The formalism of Discrete Event Systems (DES) is used to model the system and the events. More specifically, bounded LPN will be our chosen formalism.
- As stated in [2], the attacker wants to harm the system by destabilising it.
- We assume that faults are different from attacks as an attack detected as a fault can be reproduced by the attacker later and cause harm that could have been avoided.
- In this paper we will assume that the attacker's goal is to block the system by leading it into a deadlock. In other words, an attack will be successful only if the system is in a deadlock state. A deadlock state is defined as a state from which there is no evolution possible in the language of the process.
- We consider an ongoing attack on systems which is described in Fig.3, the attacker is inserted in the sensor communication channel and can alter them. In this example, we only consider erasure attacks.

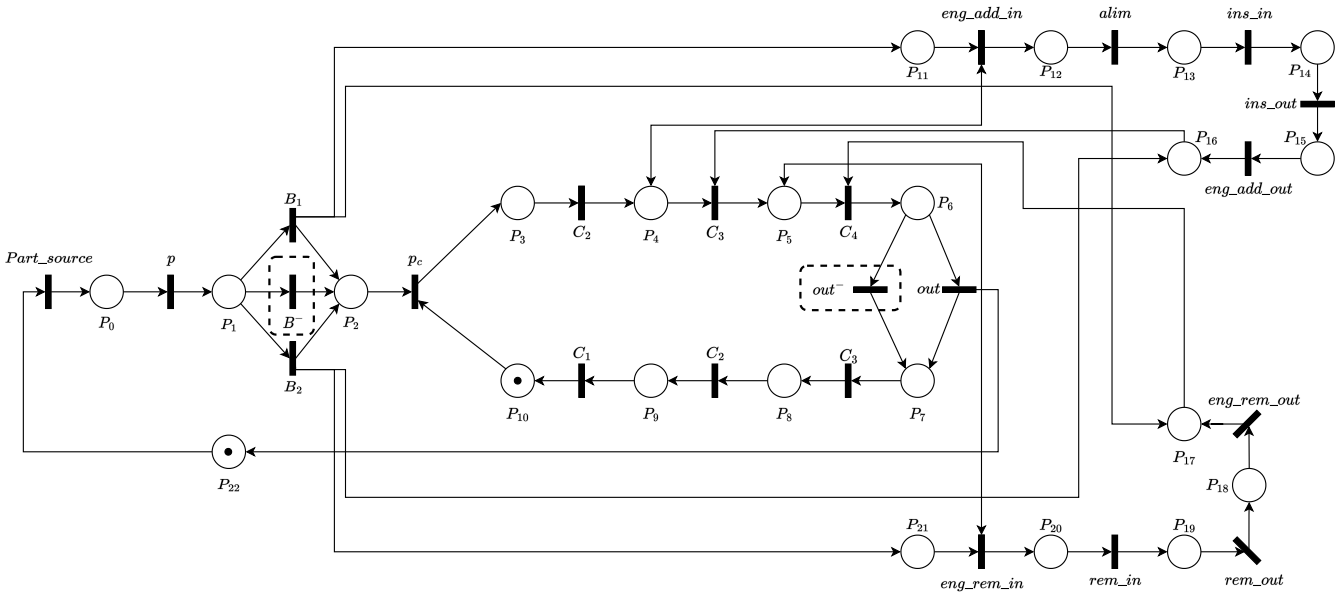

Fig. 2. Simplified behavior of Station 2 and 3 from [21]. Transitions in dashed areas are added for the attacks (see III-B) and are not part of nominal behavior. Nominal net is denoted L_N and the attacked one L_{att}

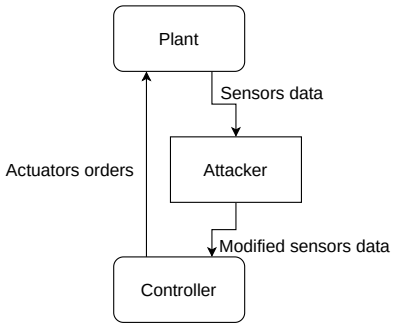

Fig. 3. Architecture of the attack

TABLE I
EVENTS USED IN THE SYSTEM

$Part\_source$	Part arrives in the line
p	Part is detected in the line
B_1	Part has no bearing
B_2	Part has bearing
pc	Pressure sensor on the cart
C_i	Detectors along the cart line
eng_add	Piston feeding the bearing insertion machine
$alim$	Bearing drop
ins	Piston inserting the bearing in the part
eng_rem	Piston feeding the bearing removal machine
rem	Piston removing the bearing from the part
out	Part is detected out of the line

III. Model construction method for diagnosis

In this section, a model suitable for attack prevention is built. To do so our methodology is summed up in Fig.4. From a model of the nominal behavior of a system (**A**), possible attacks are taken into account and added to the model to simulate the attacked behavior (**B**). From this new model, we obtain its reachability graph (**C**) in order to detect deadlocks and obtain attack words that lead to them (**D**). Finally, the reachability graph representing the nominal behavior is enriched with that knowledge to provide a base model for diagnosis (**E**).

A. Modeled system

To facilitate comprehension, the general workflow description from Fig.4 will be supported by a case study defined in this subsection.

The base model presented in Fig.2 represents a simplified behavior of a part of a production line presented in the work of [21]. Our example is composed of Station 2 and Station 3 in a case where all workpieces are made of the same material. In Fig.2, all transitions are labeled by events from sensors in the system, they are presented in Table I. The net in Fig.2 is one-bounded, that is enforced by place P_{22} which is a necessary resource to introduce a workpiece in the system and which is given back at the end of the process when the workpiece is discharged. The language recognised in this net will be called $\mathcal{L}$, its reachability graph is presented in Fig.5.
Two transitions of this net are attacked transitions and will be described and used later in Section III-B. Those attacked parts are denoted by dotted areas in Fig.2.

B. Attack implementation

Starting from the model of the system it is now necessary to implement some new behavior to take into account the effect of attacks on the system. The only attack considered here is an erasure attack on sensors, the attacker can absorb an event coming from the sensors and hide it from the controller. As stated in part II-C, the attacker can erase an attack by replacing an event $e \in E_{era}$ with an event $e^- \in E^-$ that is invisible to

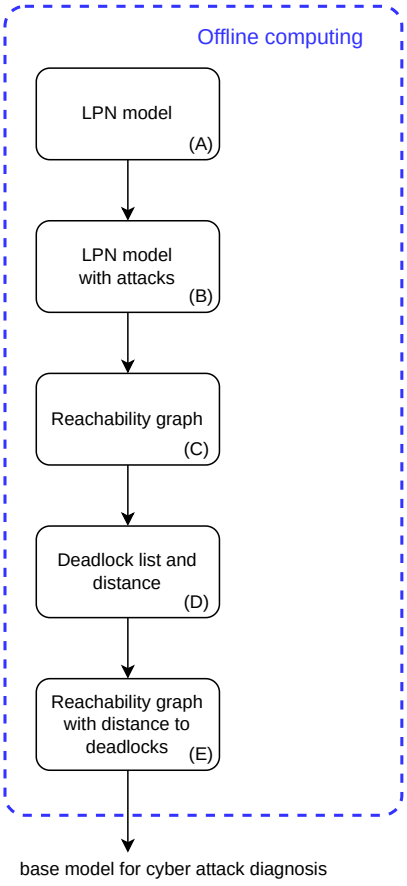

Fig. 4. Proposed workflow to provide diagnosis model

the controller.
That capability is illustrated in the example in Fig.2 by the addition of two *attacked transitions* in the net, $B-$ that erases the signal from sensor B and $out-$ that erases the output signal from the line. Those two alternative transitions allow the attacker to hide an evolution of the system to the controller and thus potentially lead the system to a deadlock.

With these new additions to the net, the marking graph in Fig.5 is bound to change due to the attacks.

C. Reachability Graph

The new attacked language is denoted $\mathcal{L}_a$. The attacked marking graph of L_{att} is presented in Fig.6, it is denoted R_{att}. It now contains two deadlock states induced by the attacks, one for lack of information in the system for the treatment of parts, and another one due to the erasure of the event out, which prevents the system from resetting.

To detect transition sequences that lead to deadlocks the reachability graph of the attack model is built. The language produced by the reachability graph is in E_a^*. This new graph takes into account the attacks added earlier to the model and shows labels from the attack alphabet.

The Reachability graph of the attacked net L_{att} is presented in Fig.6. Two paths in the graph are clearly distinguishable,

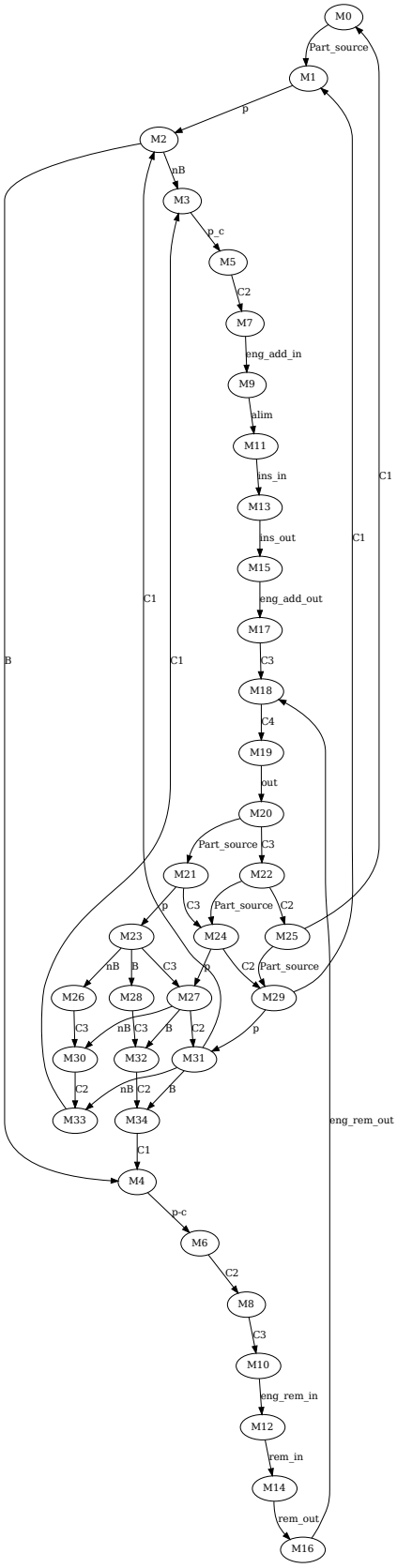

Fig. 5. Reachability graph R_N of the example net L_N

ending with two deadlocks that we consider here as successful attacks. New states created by the attacked transitions are printed in red in the graph.

D. Model transformation

After the construction of R_{att} it is now necessary to detect and isolate deadlocks to identify traces that lead to these deadlocks.

Based on the reachability graph produced it is possible to identify sequences that lead to deadlocks by identifying finite words $w \in E_a^*$ which lead to a deadlock state. As shown in Fig.6, attacked trajectories leading to deadlocks can be isolated.

From this graph, after identifying all deadlocks, *i.e.* all states deprived of outgoing arc, it is possible to determine for each nominal state of the reachability graph of the identified model a set of possible attack trajectories Ω. This step isolates finite words containing elements from $E^+ \cup E^-$ and leading to deadlocks, *i.e.* $\omega \in \Omega \Leftrightarrow \{\omega \in \mathcal{L}_a \setminus \mathcal{L} \mid \sigma(q,\omega) = q_f \text{ and } \forall u \in \mathcal{L}_a \nexists \sigma(q_f,u) \text{ does not exist}\}$. By extension from the definition in the previous section, $\sigma(q,\omega)$ returns the state reached by the successive occurrence of all the labels in ω. Function f associates a subset of $2^{\mathcal{L}_a}$ to all states based on the previous definition

$$f : Q \rightarrow 2^{\mathcal{L}_a}$$
$$q \mapsto \Omega$$

The objective is, from each marking state, to compute the shortest attack sequence to deadlocks. Let π_a be the projection of E_a on $E^+ \cup E^-$, $\pi_a : E_a \rightarrow E^+ \cup E^-$. The shortest path to a deadlock is defined as follows

$$S(q) = \underset{\omega \in f(q)}{argmin}(|\omega|)$$

It is then possible to inform the user, from each state of the marking graph, of the closest deadlock, the distance from it and of the number of attacks comprised in the sequence leading to it.

$$(S(q); |S(q)|; |\pi_a(S(q))|)$$

Starting from the initial reachability graph from Fig.5, each state is enriched with trajectories leading to deadlocks and a distance indicator to the closest one. We now have a graph representing the nominal behavior of the system but for each state of the system, information is added about attacks that can lead to deadlocks.

E. Discussions

Performing diagnosis with this model can provide, at each time, attack trajectories possibly ongoing with the length of their associated sequence to assess their imminence. With this enriched graph it is possible to say whether a detected deviation from nominal behavior could be part of an attack or not.

For example in Fig.6, from state $M19$ it is possible to reach the deadlock state $M44$ with the attack sequence $\omega_1 = out^- C_3 C_2 C_1$ which corresponds to an erasure of the output

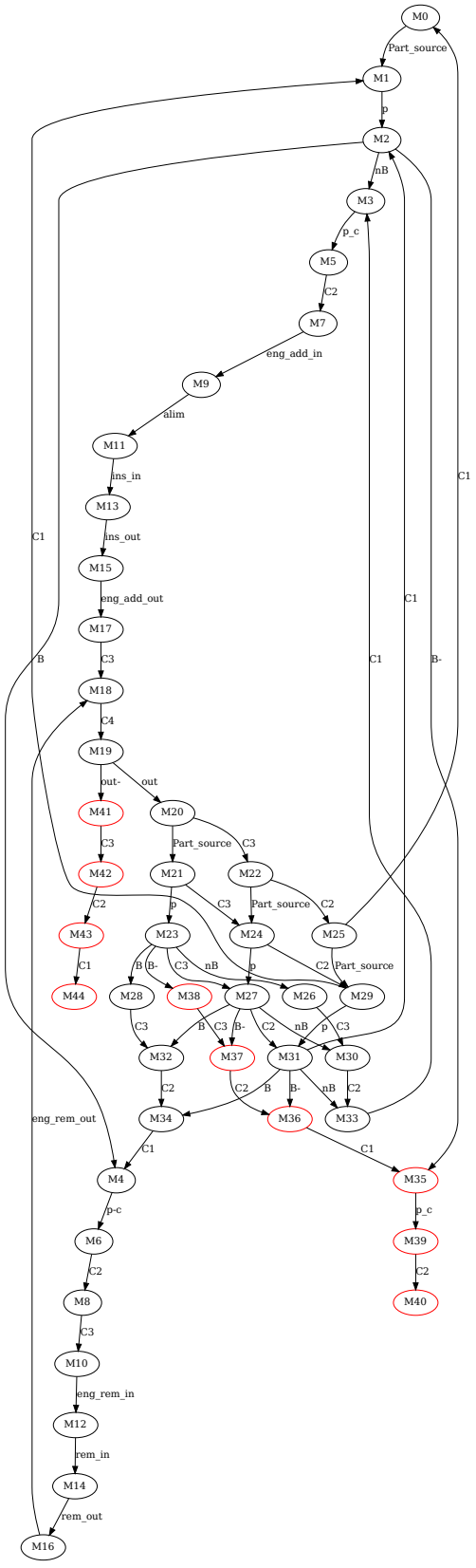

Fig. 6. Reachability graph R_{att} of the attacked example net L_{att}, new attacked trajectories in red color

signal out and lead to an impossibility to reset the system as the shared resource controlling the number of parts in the system is not returned and thus lead to blockage. One indicator that demonstrates the dangerousness of an attack is the number of attacks present in it, as trajectories with more modifications require deeper access to the system and induce a greater risk for the attacker to be detected. Conversely, an attack word containing only one attack is highly dangerous as its occurrence may lead directly to a deadlock state. The criterion of warning based on word length and the number of attacks in it is to be discussed in further works.

Another point of concern is the scalability of the model. In the proposed workflow every trajectory has to be computed in advance in order to enrich the model. This means predefined attacks that are not adaptable on the flight. A possible solution would be to adopt a distributed approach that would allow to modify or add parts of the model after the first conception.

IV. Conclusion

In this paper, we proposed a method to build a model to perform model-based diagnosis on systems subject to cyber-attacks in order to detect them. To do that, a model composed of the reachability graph of the identified nominal system enriched with knowledge on attacks which lead to deadlocks is built. This model provides additional information when a deviation from nominal behavior is detected, but in the case of a nominal run, this information is also accessible. It provides an additional source of information for decision-making when a non-nominal behavior is detected during the execution of the system. This model is limited by the type of attacks it takes into account, only simple insertion or erasure, and by the fact that all those attacks are predefined during model building. The scalability of the model is also a concern, as considering every label as compromised labels create a great many transitions in the attacked marking graph. A more distributed approach is to be considered. Our future work will mainly be focused on relaxing three of the hypothesis, firstly by considering attacks that do not lead to a deadlock, as livelocks can also be a mean of paralysing a production line, secondly by modeling more complex attacks such as entire sequences of events repeated or delayed by an attacker and lastly by considering a non-bounded system.

References

[1] G. Putnik, L. Ferreira, N. Lopes, and Z. Putnik, "What is a Cyber-Physical System: Definitions and models spectrum," *FME Transactions*, vol. 47, no. 4, pp. 663–674, 2019. [Online]. Available: https://scindeks.ceon.rs/Article.aspx?artid=1451-20921904663P

[2] H. S. Sánchez, D. Rotondo, T. Escobet, V. Puig, and J. Quevedo, "Bibliographical review on cyber attacks from a control oriented perspective," *Annual Reviews in Control*, vol. 48, pp. 103–128, Jan. 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1367578819300288

[3] A. Beaudet, F. Sicard, C. Escudero, and E. Zamaï, "Process-Aware Model-based Intrusion Detection System on Filtering Approach: Further Investigations," in *2020 IEEE International Conference on Industrial Technology (ICIT)*, Feb. 2020, pp. 310–315, iSSN: 2643-2978.

[4] R. M. Góes, E. Kang, R. Kwong, and S. Lafortune, "Stealthy deception attacks for cyber-physical systems," in *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, Dec. 2017, pp. 4224–4230.

[5] R. Su, "Supervisor synthesis to thwart cyber attack with bounded sensor reading alterations," *Automatica*, vol. 94, pp. 35–44, Aug. 2018. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0005109818301912

[6] Q. Zhang, Z. Li, C. Seatzu, and A. Giua, "Stealthy Attacks for Partially-Observed Discrete Event Systems," in *2018 IEEE 23rd International Conference on Emerging Technologies and Factory Automation (ETFA)*, vol. 1, Sep. 2018, pp. 1161–1164, iSSN: 1946-0759.

[7] P. M. Lima, M. V. S. Alves, L. K. Carvalho, and M. V. Moreira, "Security Against Communication Network Attacks of Cyber-Physical Systems," *Journal of Control, Automation and Electrical Systems*, vol. 30, no. 1, pp. 125–135, Sep. 2018. [Online]. Available: http://link.springer.com/10.1007/s40313-018-0420-9

[8] Q. Zhang, C. Seatzu, Z. Li, and A. Giua, "A framework for the analysis of supervised discrete event systems under attack," *arXiv:2005.00212 [cs, eess]*, 2019, arXiv: 2005.00212. [Online]. Available: http://arxiv.org/abs/2005.00212

[9] L. K. Carvalho, Y.-C. Wu, R. Kwong, and S. Lafortune, "Detection and mitigation of classes of attacks in supervisory control systems," *Automatica*, vol. 97, pp. 121–133, Nov. 2018. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0005109818303741

[10] R. Fritz and P. Zhang, "Modeling and detection of cyber attacks on discrete event systems," *IFAC-PapersOnLine*, vol. 51, no. 7, pp. 285–290, 2018. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S240589631830644X

[11] C. Gao, C. Seatzu, Z. Li, and A. Giua, "Multiple Attacks Detection on Discrete Event Systems," in *2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)*. Bari, Italy: IEEE, Oct. 2019, pp. 2352–2357. [Online]. Available: https://ieeexplore.ieee.org/document/8914035/

[12] R. Meira-Góes, E. Kang, R. H. Kwong, and S. Lafortune, "Synthesis of sensor deception attacks at the supervisory layer of Cyber–Physical Systems," *Automatica*, vol. 121, p. 109172, Nov. 2020. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0005109820303708

[13] S. Mohajerani, R. Meira-Góes, and S. Lafortune, "Efficient Synthesis of Sensor Deception Attacks Using Observation Equivalence-Based Abstraction," in *IFAC-PapersOnLine*, vol. 53, 2020, pp. 28–34. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S2405896321001440

[14] Q. Zhang, C. Seatzu, Z. Li, and A. Giua, "Joint State Estimation Under Attack of Discrete Event Systems," *IEEE Access*, vol. 9, pp. 168 068–168 079, 2021, arXiv: 1906.10207. [Online]. Available: http://arxiv.org/abs/1906.10207

[15] P. M. Lima, M. V. S. Alves, L. K. Carvalho, and M. V. Moreira, "Security of Cyber-Physical Systems: Design of a Security Supervisor to Thwart Attacks," *IEEE Transactions on Automation Science and Engineering*, pp. 1–12, 2021. [Online]. Available: https://ieeexplore.ieee.org/document/9429937/

[16] Q. Zhang, C. Seatzu, Z. Li, and A. Giua, "Stealthy Sensor Attacks for Plants Modeled by Labeled Petri Nets," in *IFAC-PapersOnLine*, vol. 53, 2020, pp. 14–20. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S2405896321000987

[17] R. Ammour, L. Brenner, I. Demongodin, S. Amari, and D. Lefebvre, "Costs analysis of stealthy attacks with bounded output synchronized Petri nets," in *2021 IEEE 17th International Conference on Automation Science and Engineering (CASE)*. Lyon, France: IEEE, Aug. 2021, pp. 799–804. [Online]. Available: https://ieeexplore.ieee.org/document/9551583/

[18] R. Ammour, S. Amari, L. Brenner, I. Demongodin, and D. Lefebvre, "Robust Stealthy Attacks Based on Uncertain Costs and Labeled Finite Automata With Inputs," *IEEE Robotics and Automation Letters*, vol. 8, no. 5, pp. 2732–2739, May 2023, conference Name: IEEE Robotics and Automation Letters.

[19] R. David and H. Alla, *Discrete, Continuous, and Hybrid Petri Nets*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010. [Online]. Available: http://link.springer.com/10.1007/978-3-642-10669-9

[20] C. G. Cassandras and S. Lafortune, *Introduction to Discrete Event Systems*. Cham: Springer International Publishing, 2021. [Online]. Available: https://link.springer.com/10.1007/978-3-030-72274-6

[21] J. Saives, G. Faraut, and J.-J. Lesage, "Automated Partitioning of Concurrent Discrete-Event Systems for Distributed Behavioral Identification," in *IEEE Transactions on Automation Science and Engineering*, vol. 15, Apr. 2018, pp. 832–841. [Online]. Available: https://ieeexplore.ieee.org/document/7976383/