Towards Resilient and Reliable Distributed Automation for Smart Manufacturing Systems

Vuk Lesi
Duke University
Durham, North Carolina
vuk.lesi@duke.edu

Zivana Jakovljevic
University of Belgrade
Belgrade, Serbia
zjakovljevic@mas.bg.ac.rs

Miroslav Pajic
Duke University
Durham, North Carolina
miroslav.pajic@duke.edu

ABSTRACT

Industrial Internet-of-Things (IIoT) -enabled equipment supports highly dynamic production environments by offering easily configurable building blocks for distributed manufacturing. These building blocks encapsulate sensing, actuation and local control in smart manufacturing modules that expose network connectivity. However, most existing production systems are based on sequential automation that is conventionally centralized, i.e., not tailored for the distributed setting. In our work, we propose a framework for resilient and reliable automation distribution that starts from a centralized controller design, and distributes the control functionality over IoT-enabled controllers in a manner that preserves functional equivalence. Due to possible communication faults and potentially congested environments, we develop techniques to include channel fault and adversarial models in verification of correctness of the newly obtained distributed control. Additionally, we support high-assurance code generation from the verified models, by introducing suitable patches for control code for IoT-enabled controllers. Furthermore, due to dynamic operating environments of IoT-based deployments, we introduce edge-based performance/reliability runtime monitoring that is used to promptly alert at operational trends leading to failures based on runtime process and channel measurements. We successfully apply our framework on real-world industrial systems, and show how an existing centralized control design can be used to automatically obtain an IoT-enabled distributed manufacturing deployment.

1 INTRODUCTION

In alignment with the Industry 4.0 vision [7], manufacturers are moving towards mass customization, as opposed to mass production [3]. However, to support such a vision, highly flexible production equipment that is composed of smart building blocks is needed, in order to facilitate efficient structural and functional adaptability [8]. One of the main enabling technology for development of the next generation of manufacturing equipment and systems is Industrial Internet of Things (IIoT) [6]. These smart manufacturing resources directly support reconfigurability of smart manufacturing systems, by physically integrating communication capabilities with sensing, local control/decision making (i.e., computation), and actuation close to the physical process under control.

Key shortcomings of centralized automation reflect in the need to perform extensive hardware rewiring when reconfiguring the system, together with complex control software configurations/-modifications. On the other hand, IIoT-enabled devices that integrate sensing, actuation, and (local) control components with the mechanical system represent smart manufacturing building blocks that enable fast reconfiguration in dynamic production environments, as they perform a set of tasks locally and expose control over their capabilities through standard network interfaces. Therefore, there exists a need to break up the conventional centralized control paradigm and shift towards the networked, distributed setting in which a network of local controllers (LCs) collectively performs the tasks of the legacy global (centralized) controller correctly and efficiently, and in addition enables adaptability. On the other hand, robust operation of such distributed control requires highly reliable and secure connectivity of deployed smart devices.

On the other hand, most legacy automation controllers in manufacturing systems suffer from lack of connectivity, originating from the use of conventional centralized control paradigms. Design of such sequential automation controllers is commonly based on Petri nets formalism, and their semantically equivalent industrial programming languages (GRAFCET/SFC). Consequently, we focus on challenges that arise with distributed control of sequential automation, and provide a general framework for resilient and reliable distribution of such control functionalities, while supporting the use of specific industry-adopted modeling formalisms. Our framework supports automated (i) mapping of the centralized control functionalities to distributed controllers, (ii) resilience analysis of the obtained distributed system and patches for code generation to improve security guarantees, (iii) performance/reliability analysis of the obtained distributed system and patches for code generation to improve fault-tolerance, and finally (iv) edge-based performance/reliability monitoring.

This paper is organized as follows. Section 2 summarizes the reliable and resilient distribution framework. Section 2.2 introduces reliability- and resiliency-aware modeling for distributed sequential control, with emphasis on their differences. System analysis is presented in Section 3, while Section 4 discusses real-world evaluation of our framework. Finally, Section 5 concludes the paper.

2 FRAMEWORK FOR DISTRIBUTED INDUSTRIAL AUTOMATION

In this section, we provide the outline of our framework for reliable and resilient distribution of sequential automation presented in Fig. 1). Specifically, the framework consists of the following stages:

- Distribution of formally-specified centralized control functionalities over to IIoT-enabled LCs, including mapping of sensing/actuation signals to smart devices and code generation for LCs,
- System-level resilience analysis and inclusion of security measures during code generation for LCs, to provide strong performance guarantees even in the presence of attacks,

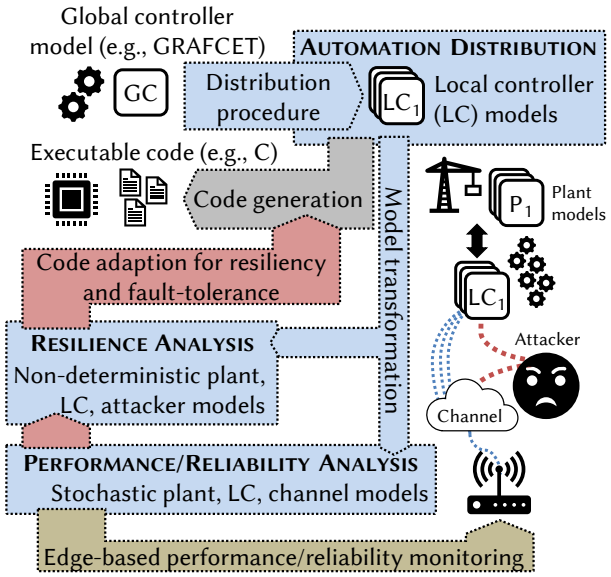


Figure 1: Framework for resilient and reliable distributed automation in smart reconfigurable manufacturing systems.

- Performance and reliability analysis and inclusion of fault-tolerant communication primitives during code generation for LCs, and
- Edge-based performance and reliability monitoring.

It is important to highlight that each stage of the framework provides feedback to the code generation stage (Fig. 1)), in order to improve reliability and resilience of the overall distributed automation. In the following sections, we summarize our existing contributions and ongoing work towards improving efficiency of the framework, and show how it can be used to successfully distribute control in a number of industrial applications.

2.1 Distribution of Control Functionalities

Existing sequential control designs are typically given in the form of discrete event systems. In [5], we consider the industry-adopted formalism of Control Interpreted Petri Nets (CIPN); the reason for widespread adoption of CIPN-based models originates from the semantical equivalence with GRAFCET/SFC languages for programming event-based automation. Petri nets (PN) are conveniently comprised of *places* that represent system's states, and *transitions* that represent the state changes. The current state of the system is represented with a set of *tokens* that dwell in places, while the token flow through the net corresponds to the system state evolution (a comprehensive review of this formalism is available in Ch. 12 of [11]). CIPNs extend PNs by allowing synchronization of places with actuation commands and conditioning transitions with sensor measurements — the key modeling features that allow specification of cyber-physical controllers.

We have recently derived a method to automatically transform a centralized automation controller (i.e., control model) into a set of distributed controllers (number depending on the number of available resources), while preserving functional correctness [5]. This process includes supervised mapping of sensing and actuation components to LCs, after which places and transitions from the global (i.e., centralized) CPIN-based control specification are automatically mapped into LC models in a manner ensuring that parallel composition of the LCs provides the same control functionality as its centralized counterpart. To achieve this, communication API calls provided by the target LC runtime environment are inserted at suitable places within the LCs' CIPN specifications, where synchronization and information exchange between LCs is needed. For instance, if a sensor value exceeds a threshold, or a limit switch is triggered, a multicast message transmission is inserted in the code of the LC that has local access to this sensory information (i.e., physical access to the sensor), while a corresponding message reception is inserted in the code of the LCs that require this update.

Note that in manufacturing systems, as in all cyber-physical systems, the concept of correctness spans beyond traditional safety and liveness requirements. Hence, our design framework also maintains safety properties specified relative to the physical process under control, for instance, *handled object will not be dropped between picking and placing positions in a distributed pick & place manipulator*. In addition, the distributed control models can be used in conjunction with existing code generation techniques to automatically obtain executable control code, suitable for immediate deployment on target platforms [5], as illustrated in Fig. 1.

2.2 Safety- and Security- Aware Distributed Automation Modeling

While functionally equivalent to its centralized counterpart when reliable communication links are employed, the obtained distributed control implementations are susceptible to communication failures and attacks, as the basic communication API does not ensure fault-tolerant and secure information exchange. However, without thorough analysis of the distributed system implementation, it is challenging to determine operational importance and criticality of specific messages, which is necessary when system-resources may be limited. Therefore, our design framework provides support for analysis of such distributed controllers, by composing LCs' models with suitable plant and communication channel/attack models.

2.2.1 Fault VS Adversary Modeling. Fundamental modeling decisions are conditioned by the intended model use-case. Communication faults introduced by control distribution over smart devices can be modeled by experimentally measuring network patterns and delays, or by utilizing existing knowledge on the deployed communication system and protocols. While communication faults can be captured using stochastic models, such models are medium-, protocol-, and implementation- dependent. On the other hand, using a stochastic model to capture generally unpredictable adversarial behavior severely limits attack modeling. Therefore, the use of non-deterministic modeling formalisms is required for effective adversarial modeling. In what follows, we present our approach to obtain suitable communication channel model (based on [9]), and the corresponding adversarial model.

2.2.2 Half-duplex, acknowledgment-required unicast CSMA-CA-based communication channel model. Half-duplex, acknowledge-required unicast CSMA-CA-based communication channels are widely used in wireless IoT deployments. While other channel models can be as easily adopted, we show the Stochastic Reward Net (SRN) model of this channel in Fig. 2(a). SRNs extend PNs in

that they allow transitions in the model to have a stochastic firing time (e.g., network propagation time), and are therefore suitable for performance/reliability modeling. As shown in Fig. 2(a), the channel can be idle, busy transmitting a message, just completed transmitting a message (i.e., ready to transmit the acknowledgement — ACK), or busy with the ACK. The corresponding transitions model the change of the channel's state triggered by initiation of message/ACK transmissions in the transmitter/receiver LC models, or transmission completion. This coupling between models is achieved through *guard functions*, denoted by `g_x()` next to the guarded transition (more details on interfacing models is given in Sec. 2.2.4). Notice that transitions depicted as rectangles (rather than bars) take a stochastic time to fire (i.e., remove token from incoming place, and deposit it into the outgoing). Recall that the current token position indicates the current channel state (e.g., initially idle in Fig. 2(a)).

2.2.3 Attacker model. In our resilience analysis, we assume a powerful attacker that has full knowledge of nominal event propagation and states of all LCs, and has network access and full communication protocol compliance. Therefore, the attacker is capable of

- Intercepting or delaying communication messages between LCs (e.g., as in Denial-of-Service (DOS) attacks),
- Intercepting the ACK packet, or
- Impersonation, i.e., sending events on behalf of LCs (e.g., as in *masquerading* attacks).

The choice of the attack start time and type should be modeled as *non-deterministic*, as it cannot be anticipated or fitted to a model in reality. Additionally, while the nature of communication faults/delays is known in the adversary-free environment, no realistic assumptions can be imposed on the attacker-induced packet delays or interception/injection rates. This type of non-deterministic semantics can be encoded with Time Petri Nets (TPN). Fig. 2(b) shows a TPN for the described attacker model for the previously considered wireless channel; possible attacker's choices are modeled with specific portions of the net (e.g., *normal transmission*, *injection*). As their firing time distributions are not known, timed transitions in TPN are specified via time intervals (denoted next to them as in Fig. 2(b)).

2.2.4 Plant model and components' interactions. To perform system-wide analysis, besides the controller and channel/attacker models, a corresponding plant model is necessary. Expected plant behavior is usually known at legacy controller design time. Plant models can conveniently be captured within stochastic or non-deterministic modeling frameworks as plant dynamics and response times (e.g., actuator travel times, sensor measurement ranges) can be measured or are known. Semantic compatibility between CIPNs and SRNs/TPNs originating from the mother formalism of PNs allows straightforward transformation of the CIPN controller models obtained from the distribution stage, into compatible models that are used in reliability and resilience analysis as we have done in [9].

To integrate plant, controller, and channel/attacker models into the overall system model, flexible interfaces between models must exist to support execution semantics conditioned by the employed runtime environment (e.g., blocking VS non-blocking communication API, polling VS event-based sensing). The expressiveness of

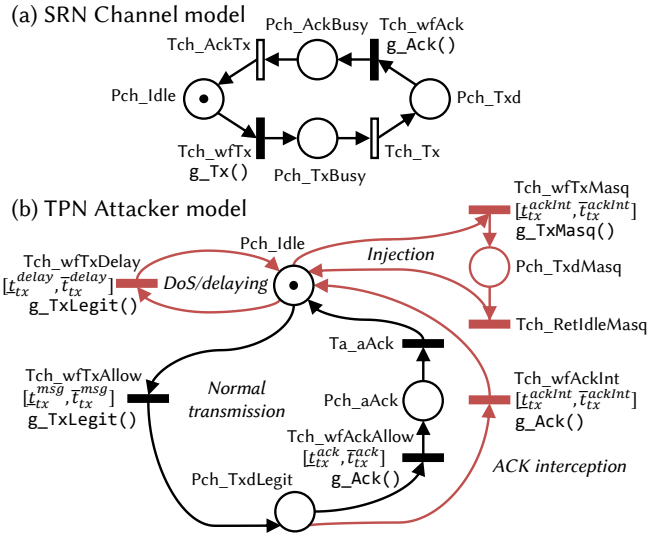


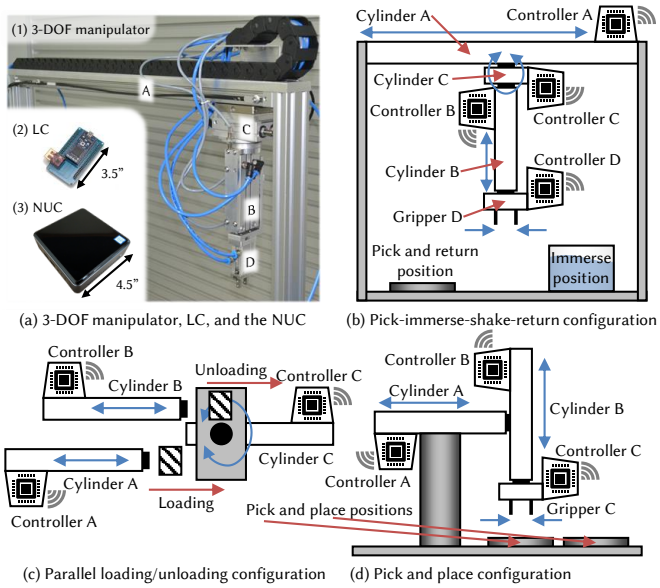
Figure 2: (a) Stochastic Reward Net channel model from [9], and (b) Time Petri Net non-deterministic attacker model.

Petri nets in the form of token multiplication over net branches and joining of parallel branches of the net [10] can be used in conjunction with rich support in existing tools for encoding model interdependencies in a semantically sound manner. For performance/reliability modeling, there exists support for state-dependent transition firing rates and arbitrary state-dependent transition guard functions [4]. Moreover, state-of-the-art tools support formal race condition resolution, i.e., conditions that occur when one transition becomes enabled to fire while stochastic firing time of another is elapsing, and firing of the former disables the latter. For non-deterministic analysis, tools provide transition guard functions [1]. Additionally, basic synchronization constructs such as global variables are generally available [1, 4].

3 RESILIENCE AND PERFORMANCE/RELIABILITY ANALYSIS

Once the distributed system model is formed and components' interactions encoded, corresponding application-specific performance/reliability measures of interest are translated into formal safety and liveness properties. Essential safety properties formally defined for Petri nets such as *1-boundedness* and *absence of deadlock* [2] do not capture safe operation from the physical process standpoint. Therefore, we consider a larger class of safety and liveness properties that capture operational aspects of the system [5, 9].

For stochastic models, choice of distribution firing times limits its applicable solution methods and the nature of obtained results. Namely, if transition firing times are constrained to the exponential distribution, analytic/numeric solution methods can be applied and performance/reliability measures obtained with strong guarantees. However, for generally-distributed firing times, analysis resorts to simulative methods, and measures are obtained with probabilistic guarantees [11]. Yet, some non-exponential distributions (e.g., Erlang-k) can be modeled with multiple exponential transitions, making it possible for a fully analytical solution to capture non-exponentially distributed firing times, which is often practically



(a) 3-DOF manipulator, LC, and the NUC (b) Pick-immerse-shake-return configuration

(c) Parallel loading/unloading configuration (d) Pick and place configuration

Figure 3: IoT-enabled pneumatic manipulator [9]; (a) physical components including the LC and the edge-based monitor; (b), (c), and (d) different manipulator configurations.

needed. Pertaining to non-deterministic analysis, we make no assumptions on the firing time distributions (only lower, and upper firing times are encoded in the model), and the non-deterministic adversary model is included. This analysis allows for verification of properties of interest with strong guarantees.

3.0.1 Edge-Based Performance and Reliability Monitoring. Due to highly-dynamic IIoT operating environments, verification of performance and reliability metrics offline may be insufficient. For example, introducing additional IIoT-enabled equipment may affect channel properties. Thus, in [9] we propose edge-based performance and reliability monitoring driven by process and channel measurements obtained at system runtime and demonstrate computational tractability of near-gateway monitor deployment. We exploit rich functionality of SRN analysis tools [4] to directly feed measurements to the simulation engine as transition times in the model, bypassing the stage of fitting a probability distribution to real-world data. This significantly improves accuracy of obtained measures, as no assumptions (e.g., independent, identically-distributed samples) need to be introduced for transition firing times.

4 CASE STUDIES

We evaluated our framework on an IoT-enabled industrial pneumatic manipulator (Fig. 3) that is easily reconfigurable. Fig. 3(a1) shows the pneumatic cylinders in a *pick-immerse-shake-return* configuration (also shown in Fig. 3(b)). Fig. 3(c) and (d) show other possible configuration with the IoT-enabled cylinders. We start from a legacy centralized control specification and apply our framework to obtain executable LC instances. Our target LC platform is ARM Cortex-M3-based and features IEEE 802.15.4 wireless connectivity, and is shown in Fig. 3(a2). We perform analysis assuming the channel and attacker models presented in Sec. 2.2. We fit stochastic channel models based on experimentally-obtained measurements. For instance, our analysis finds that protocol-level retransmissions are not sufficient to provide safe operation of the distributed system (e.g., the object may be dropped between picking and placing positions in the *pick and place* configuration). We thus introduce application-level message retransmissions that ensure satisfaction of safety properties. Furthermore, as described in [9], we developed edge-based monitoring using the power- and size- aware Intel Next Unite of Computing (NUC) platform (Fig. 3(a3)). We showed that the edge-based monitor can warn at operational trends leading to unsafe behaviors based on channel and process measurements collected at system runtime, before failures occur. A more comprehensive reliability and performance analysis is given in [9]. Resilience analysis of this industrial setup is a part of our ongoing work.

5 CONCLUSIONS

In this paper, we reviewed our framework for security- and safety-aware distribution of legacy centralized controllers. While our framework is generally applicable to a wide range of automation systems, we instantiated it and demonstrated its utility on an industry-adopted formalism of CIPNs. We showed how derived stochastic channel and non-deterministic adversarial models can be composed with distributed control models in order to perform system-wide performance/reliability evaluation. Additionally, we showed how analysis results can be used to patch control code generated from the distributed model, and thus improve system reliability and resiliency. Furthermore, we demonstrated tractability of our approach for edge-based performance/reliability monitoring that supports highly dynamical environments of modern production systems.

ACKNOWLEDGMENTS

This work is sponsored in part by the ONR under agreements N00014-17-1-2012 and N00014-17-1-2504, as well as the NSF CNS-1652544 grant. It was also partially supported by the Serbian Ministry of Education, Science and Technology research grant TR35004.

REFERENCES

[1] Bernard Berthomieu*, P-O Ribet, and François Vernadat. 2004. The tool TINA–construction of abstract state spaces for Petri nets and time Petri nets. *International journal of production research* 42, 14 (2004), 2741–2756.
[2] R. David and H. Alla. 2010. *Discrete, continuous, and hybrid petri nets (2nd edition).*
[3] H. ElMaraghy, G. Schuh, W. ElMaraghy, F. Piller, P. Schönsleben, M. Tseng, and A. Bernard. 2013. Product variety management. *[CIRP] Annals - Manufacturing Technology* 62, 2 (2013), 629 – 652. https://doi.org/10.1016/j.cirp.2013.05.007
[4] Christophe Hirel, Bruno Tuffin, and Kishor S. Trivedi. 2000. SPNP: Stochastic Petri Nets. Version 6.0. In *Computer Performance Evaluation.Modelling Techniques and Tools*. Springer Berlin Heidelberg, 354–357.
[5] Z. Jakovljevic, V. Lesi, S. Mitrovic, and M. Pajic. 2018. Distributing Sequential Control for Manufacturing Automation Systems. *IEEE Transactions on Control Systems and Technology* (2018). submitted.
[6] Zivana Jakovljevic, Vidosav Majstorovic, Slavenko Stojadinovic, Srdjan Zivkovic, Nemanja Gligorijevic, and Miroslav Pajic. 2017. Cyber-Physical Manufacturing Systems (CPMS). In *Proceedings of 5th International Conference on Advanced Manufacturing Engineering and Technologies*. Springer International, 199–214.
[7] Henning Kagermann, Johannes Helbig, Ariane Hellinger, and Wolfgang Wahlster. 2013. *Recommendations for implementing the strategic initiative INDUSTRIE 4.0.* Forschungsunion.
[8] Y. Koren, X. Gu, and W. Guo. 2018. Reconfigurable manufacturing systems: Principles, design, and future trends. *Frontiers of Mechanical Engineering* 13, 2 (2018), 121–136. https://doi.org/10.1007/s11465-018-0483-0
[9] V. Lesi, Z. Jakovljevic, and M. Pajic. 2019. Reliable Industrial IoT-Based Distributed Automation. *ACM/IEEE Conference on Internet of Things Design and Implementation* (2019).
[10] James L Peterson. 1977. Petri nets. *ACM Computing Surveys (CSUR)* 9, 3 (1977), 223–252.
[11] Kishor S Trivedi and Andrea Bobbio. 2017. *Reliability and Availability Engineering: Modeling, Analysis, and Applications.* Cambridge University Press.