Integrating Field Measurements into a Model-Based Simulator for Industrial Communication Networks

Jing Geng*, Honglei Li*, Mohamed Kashef†, Yongkang Liu†,
Richard Candell†, Shuvra S. Bhattacharyya*
* Department of Electrical and Computer Engineering, University of Maryland, College Park, USA
† Intelligent Systems Division, National Institute of Standards and Technology, USA
Email: {jgeng, honglei}@umd.edu, {yongkang.liu, mohamed.hany, richard.candell}@nist.gov, ssb@umd.edu

***Abstract*— Efficient and accurate simulation methods are of increasing importance in the design and evaluation of factory communication systems. Model-based simulation methods are based on formal models that govern the interactions between components and subsystems in the systems that are being simulated. The formal models facilitate systematic integration across the system, and enable powerful methods for analysis and optimization of system performance. However conventional simulation approaches utilize communication channel models that do not fully reflect the characteristics and diversity of industrial communication channels. To help bridge this gap, we develop in this paper new methods for channel model construction for link-layer simulation that systematically incorporate field measurements of wireless communication channels from industrial networks, and derive corresponding channel modeling library components. The generated library components capture channel characteristics in the form of lookup tables, which can be flexibly integrated into system-level simulators or co-simulation tools. We integrate our new table-generation methods into a model-based co-simulator that jointly simulates the interactions among process flows, physical layouts of workcells, and communication channels in factory systems that are integrated with wireless networks. Experimental results using our lookup-table-augmented co-simulator demonstrate the utility of the proposed methods for flexibly and accurately integrating realistic industrial network channel conditions into simulation processes.**

I. Introduction

The integration of wireless communication capabilities into factory systems is of increasing interest due to the important potential advantages brought about by wireless communications technology in factory environments [1]. This integration leads to highly complex design spaces, which involve interactions among process flow algorithms, factory workcell layouts, and wireless communication networks. We refer to these design spaces as *wireless-integrated factory system* (*WIFS*) design spaces. Due the high complexity of WIFS design spaces, simulation methods are of great importance for designing new systems, and for investigating modifications, such as upgrades, to existing systems. Effective simulation tools enable rapid evaluation and comparison of alternative system designs, thereby facilitating the process of iterative design and system performance optimization.

A limitation in conventional simulation approaches used in navigating WIFS design spaces is that they are based on communication channel models that do not fully reflect the characteristics and diversity of industrial communication channels. Many network simulation tools have been developed whose capabilities are useful to aid in exploring WIFS design spaces. Popular examples include NS-3, OMNeT++, and Cloonix, which provide sophisticated capabilities for network simulation. However, these simulators apply channel abstractions at the physical layer — e.g., by utilizing mathematical equations associated with different channel models or synthetic (simulated) data that is derived from delay profiles obtained from third-party sources, such as the IEEE 802.11 Wireless LAN Working Group (see Section II. WIFS simulation approaches that employ such methods may fail to provide accurate assessment of system-level performance because they do not precisely incorporate characteristics of actual industrial wireless communication channels — for example, harsh conditions that arise due to the vibration of machinery, and the presence of metal objects and obstacles (e.g., see [2]).

In this paper, we develop new methods for integrating realistic models of wireless factory communication network channels into simulation tools and associated WIFS design space exploration processes. More specifically, we introduce an approach to link layer simulation that systematically incorporates data from field measurements. Our link layer simulation approach produces signal-to-noise ratio (SNR) to packet-error-rate (PER) conversion tables, which can be integrated into co-simulation tools for WIFS design space exploration. The resulting integration allows the co-simulation tools to assess factory system performance characteristics (e.g., real-time communication performance, communication reliability, factory production throughput, and workcell energy consumption) more accurately in the context of the reference networks from which the field measurements were taken.

To demonstrate our proposed new approach to integrating field measurements into WIFS design space exploration, we employ a recently-introduced co-simulation tool called *Tau Lide Factory Sim* (*TLFS*), which enables model-based representation and simulation of factory process-flows and systematic integration of its process-flow simulation models with arbitrary discrete event tools for network simulation [3]. We extend TLFS with an extensible channel library that can be populated with different SNR-to-PER lookup tables, such as those generated by our proposed link layer simulation

978-1-7281-5297-4/20/$31.00 ©2020 IEEE

approach. Through experiments, we demonstrate the utility of plugging in SNR-to-PER conversion tables generated from our new link layer simulation tool into TLFS through the new channel library extension. This utility includes both simulation results that more accurately incorporate actual industrial network characteristics, and faster simulation speed, which is enabled by the use of lookup tables as opposed to computational methods.

Although we demonstrate the contributions in this paper using the TLFS co-simulation tools, the utility of the proposed methods is not specific to TLFS. The proposed link layer simulator and the SNR-to-PER tables that it produces can readily be integrated into other network simulators or process-flow / network co-simulation frameworks to extend those tools with more accurate and customizable models for industrial wireless networks.

The remainder of this paper is organized as follows. Section II discusses related work on modeling and simulation of industrial communication networks, and summarizes the contribution of this paper in the context of the related work. Section III presents our new methods for field measurement integration, link layer simulation, and channel characterization library construction. Section III also presents our approach for integrating channel characterization libraries into the TLFS simulation tool, which we use to validate and demonstrate the methods of this paper. Section IV presents experiments involving the results of applying our proposed link layer simulation methods, and results of applying TLFS extended with channel characterizations that are generated using results from these simulation methods. Finally, Section V summarizes the contributions of the paper.

II. Related Work

A large body of work in the literature covers modeling and simulation for cyber-physical systems that are integrated with wireless communication capabilities. These works include two major directions — one that focuses on communication channel modeling for different types of channels, and another that focuses on link layer or system-level simulation that utilizes new channel models or off-the-shelf models, such as those available in MATLAB.

Many approaches have been proposed to model wireless communication channels involving different path loss models, small scale fading models, noise figures, etc. For example, the IEEE 802.11 Wireless LAN Working Group proposes channel models based on a set of WLAN channel prototypes, which model the signal delay profiles for different environments under different indoor/open space line-of-sight (LOS) and non-LOS (NLOS) scenarios (e.g., see [4]). A related class of models is the set of SISO models developed by Medbo and Schramm [5]. However, channel models such as these do not precisely reflect signal transmission in industrial wireless environments. For example, they do not capture harsh characteristics of industrial environments, such as those due to complex surroundings involving vibrating machines and metal structures, which in turn lead to significant multi-path effects, electromagnetic resonance, and other complicating factors (e.g., see [2]).

A variety of works has also investigated channel modeling under more complex communication environments. For example, Abbas et al. present an evaluation of vehicle-to-vehicle communication channel parameters through a detailed comparison between simulations and measurements [6]. Peil et al. use measurements to develop wireless propagation characteristics in an industrial environment, and then use these characteristics to derive a channel model [7].

Other works on modeling and simulation emphasize system-level evaluation based on novel applications of existing simulation frameworks. For example, Liu et al. apply the OMNeT++ simulation library to develop an integrated framework for factory process control simulation and wireless network simulation [8]. Patidar et al. [9] apply link layer simulation to improve the physical layer abstraction. Their approach uses channel profiles proposed from the IEEE WLAN Working Group [4]. Li et al. present a WIFS-oriented design space exploration tool that utilizes TLFS together with channel models that are available in the NS-3 simulation platform [10].

The key distinguishing aspect of our contribution in this paper, compared to the body of literature summarized above, is its focus on coupling the derivation of accurate channel characterizations for industrial wireless environments with system-level co-simulation processes for WIFS design space exploration. This is achieved through a new link layer simulation approach that utilizes field measurements from actual industrial wireless channels, and encapsulation of the resulting channel characterizations through a channel library, which is designed to plug into higher-level co-simulation processes.

III. Approach

In this section, we present our approach for constructing channel models that incorporate the characteristics of industrial wireless networks through field measurements, and applying the derived channel models to enable more realistic co-simulation between factory process flows and wireless communication networks in networked, smart factory environments. We refer to this type of co-simulation as *networked process flow* simulation. The field measurements used in our approach capture channel impulse responses (CIRs) from actual industrial wireless communication environments.

We demonstrate our approach through extensive experiments in Section IV. The field data used in these experiments is publicly available, and was collected from a measurement campaign performed by the U.S. National Institute of Standards and Technology (NIST) [11], [12]. Specifically, we use field data collected in this campaign from an automotive factory site. The field data employed in our experiments is in the form of one large MAT file (approximately 5 GB), which contains all of the measured CIRs from the automotive factory. The MAT file includes the physical location associated with each measured CIR, as well as the IQ data and duration for each CIR signal. The MAT file also includes metadata

associated with the measurements, such as the file name, antenna information, and frequency.

In our approach, the measured CIRs are input to a link layer simulator, which in turn produces a PER/SNR table that can be used to efficiently and accurately characterize the communication environment of interest for a higher-level simulator, such as a networked process flow simulator. The approach is extensible so that different wireless environments can be characterized using different PER/SNR tables, which are collected in the form of a channel library, as mentioned in Section I. A channel library populated with tables derived from multiple wireless environments enables comparison of a given factory/network configuration across different environments, as well as comparison of alternative configurations for a given environment.

To model a communication channel, we need to consider factors that include the channel gain, fading, multipath characteristics, and noise, as illustrated in Fig. 1. In our approach, these factors are taken into account by deriving channel models based on the CIR measurement data that is collected from the given factory environment.

Fig. 2 illustrates our link layer simulator, which processes field measurements in the form of CIRs, and produces PER/SNR tables. Our simulator takes as input a set of M location-specific channel impulse responses (CIRs) $C_1, C_2, \ldots, C_M$. Each C_i is associated with a distinct physical location p_i in the factory environment from which measurements are taken. Each CIR C_i is input in the form of an array A_i such that for each sample index j, $A_i[j]$ gives the value of the jth sample in the corresponding CIR signal. The time associated with each of these signal samples can easily be derived by dividing the sample index by the sample rate of the measured signal.

The simulator involves two phases of operation: *model construction* and *simulation*. The horizontal dashed line in Fig. 2 shows the separation of these phases. In the remainder of this section, we describe the different blocks shown in Fig. 2, as well as the operation of the simulation phase.

A. Pre-processing

The Pre-processing block transforms the set of location-specific CIRs $S_c = \{C_1, C_2, \ldots, C_M\}$ into a more compact, *refined* set of location-specific CIRs $S_r = \{R_1, R_2, \ldots, R_N\}$ that are more useful for further analysis than the original set S_c. Here $N \ll M$ since in typical measurement scenarios, M may be very large, and a much smaller value of N is needed for efficient channel library generation as well as for efficient networked factory simulation by tools that use the channel library. To compress across the set of CIRs, we select 1 out every D successive locations that are visited in the measurement process, where D is a parameter of the Pre-processing block. In our experiments, we use $D = 60$.

We also apply pre-processing operations to each individual CIR. In particular, to each CIR, we apply operations for filtering out noise; compression (intra-CIR compression) to reduce the number of samples; and deriving parameters, such as the delay spread and Rician K factor, that compactly characterize each of the CIRs. We apply intra-CIR compression to reduce the computational and memory cost of applying the CIR data. For intra-CIR compression, we first determine the first signal sample whose value exceeds a predetermined threshold, and then extract this sample together with the following 127 samples. The thresholding operation here is performed to ensure that we discard any prefix in the signal that falls within the noise floor, which is assumed not to be part of the desired impulse response. We retain the extracted 128 samples to represent the CIR more compactly and discard all of the other samples. In this context, 128 can be viewed as a particular setting that we use for the *block size* parameter associated with intra-CIR compression in our link layer simulator.

In general, appropriate settings for pre-processing parameters, such as the inter-CIR compression factor and block size parameter, are heavily dependent on the field measurement process. Systematic methods for setting pre-processing parameters based on measurement-process parameters is an interesting direction for future work.

B. Clustering

The Clustering block in Fig. 2 applies the clustering algorithm developed by Kashef et al. [13] to partition the set

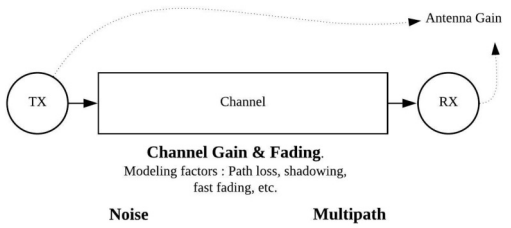

Fig. 1. An illustration of factors needed to model a communication channel.

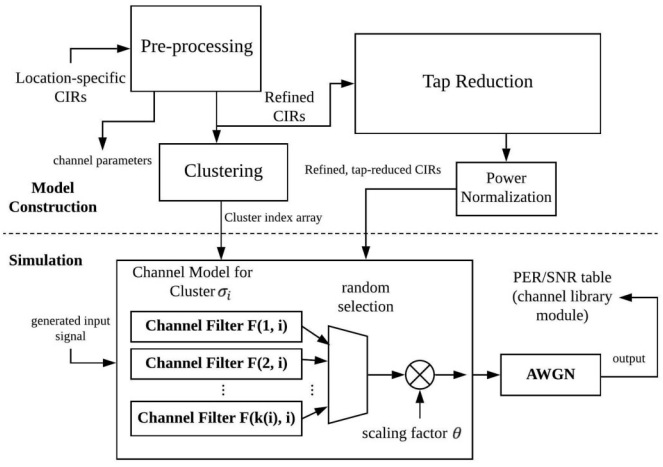

Fig. 2. An illustration of our proposed approach to link layer simulation and channel library generation.

of refined of CIRs into subsets $\sigma_1, \sigma_2, \ldots, \sigma_p$, where all of the CIRs within a given subset σ_i have similar characteristics. Each of the subsets σ_i is referred to as a *cluster* of refined CIRs. After the clustering process is applied, each refined CIR R_i has associated with it a unique cluster, which we denote as $\Sigma(R_i)$. Each cluster σ_i can be expressed as a set $\sigma_i = \{s_{i,1}, s_{i,2}, \ldots, s_{i,k(i)}\}$ of refined CIRs, where $k(i)$ represents the cardinality of (number of elements in) $\sigma(i)$, and $\Sigma(s_{i,j}) = \sigma_i$ for $j = 1, 2, \ldots, k(i)$.

The clustering block also creates a simple data structure, called the *cluster index array*, which maps indices of refined CIRs into indices of the clusters that contain them. Clustering helps to greatly reduce the complexity (number of generated PER/SNR tables) of the output of the link layer simulator, and the corresponding input that is operated on by the networked process flow simulators that utilize this data. This reduction in complexity translates into improved simulation speed during networked process flow simulation, as well as a more streamlined process for creating and storing the channel library. The reduction in complexity is achieved while preserving key characteristics of the original (unclustered) set of refined CIRs based on properties of Kashef's clustering algorithm.

C. Tap Reduction and Power Normalization

The tap reduction block converts each refined CIR R_i into a more compact tapped delay profile T_i using the algorithm of Mehlfuhrer and Rupp [14]. This is done, as with the inter-CIR and intra-CIR compression processes described above, to reduce model complexity in a way that does not substantially diminish the utility of the model. The number of taps in each of the T_is is a parameter of the tap reduction block that controls the complexity/accuracy trade-off of the tap-reduced CIR form. We refer to this as the *reduced tap count parameter* of our link layer simulator. In our experiments, we use 18 as the value of this parameter.

The output of the tap reduction process is post-processed by a simple power normalization block, as shown in Fig. 2. This block converts the taps to a form in which the sum of their squares equals unity. This normalization is performed to provide a channel profile without any path loss.

D. Construction of Cluster-Level Channel Models

Each cluster σ_i derived by the clustering block is converted during the model construction phase into a channel model, as illustrated by the block in Fig. 2 that is labeled Channel Model for Cluster σ_i. This block consists of a bank of filters $F(1,i), F(2,i), \ldots, F(k(i),i)$, where, as defined in Section III-B, $k(i)$ is the number of CIRs in cluster σ_i. Each $F(j,i)$ is the tap-reduced, power-normalized form of the refined CIR $s_{j,i}$.

Fig. 3 illustrates an example of a single filter $F(j,i)$ based on data from field measurements, and our reduced tap count parameter setting of 18. Section IV provides more details about the field measurement data that we have employed in our examples and experiments. The simulation model consists of $(k(1) + k(2) + \ldots k(p))$ such filters, where the filters are grouped into their corresponding clusters.

E. Simulation of the Constructed Channel Model

We use MATLAB to prototype our link layer simulator using the simulation model construction approach illustrated in the top part of Fig. 2. As a starting point for the prototype, we use the MATLAB WLAN toolbox, which contains channel modeling capabilities based on models proposed by Erceg et al. [4]. We then make various modifications and extensions to realize the proposed new simulation capabilities for bridging field measurements to networked process flow simulation.

First, we bypass the multipath features and fading profile from the MATLAB simulation. Instead, in our simulation approach, the multipath and fast fading effects are modeled through clusters and filters derived from the field measurements. For large scale fading, we apply the two-slope path-loss model proposed in by Damsaz et al. [15]. This is achieved in the simulation by scaling the filter response with a scaling factor θ whose value varies during simulation based on a randomly determined shadowing parameter. During simulation, applying this scaling factor is equivalent to multiplying the filter output by θ, as illustrated by the multiplication block shown in the lower part of Fig. 2.

In our approach, the link layer simulator is used to create a channel library module associated with a given factory environment. Each channel library module is populated with a set of PER/SNR tables, where one table is produced corresponding to each cluster. Different communication system parameters (e.g., the modulation and coding scheme or MCS) may be varied to produce a parameterized set of tables. In this case, for each set Γ of relevant parameter settings/combinations, link layer simulations are executed to derive the PER/SNR tables (table-subset) associated with Γ. Then the desired table-subset can be selected and used as input during networked process flow simulation.

To generate the table-subset for a given cluster σ_i and a given set of communication system parameters, the channel model for σ_i — based on the associated filters $F(1,i), F(2,i), \ldots, F(k(i),i)$ — is constructed as the core of the simulation model, as shown in the bottom part of Fig. 2. The simulation constructs physical layer service data unit (PSDU) signals from a stream of randomly-generated

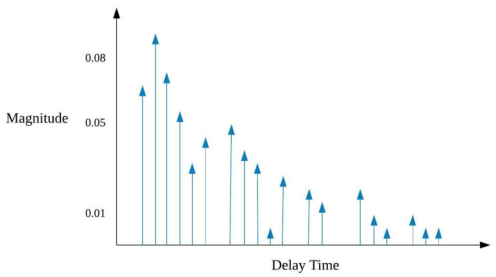

Fig. 3. An example of a single filter based on data from field measurements.

packets. The PSDU signals are passed through the channel model for σ_i to calculate the packet error rate.

For each packet, one of the $k(i)$ channel filters is selected randomly from $\phi_i = \{F(1,i), F(2,i), \ldots, F(k(i),i)\}$. This random selection allows us to incorporate the transient performance of the communication channel into the simulation in a realistic manner. The simulation to determine the packet error rate is run over a large number of packets and for a specific SNR value. This process is repeated over different SNR values to obtain the PER/SNR table for cluster σ_i.

The random selection from the set of filters ϕ_i is represented by the randomly-controlled switch that is connected to the output of the channel filters in Fig. 2. The output of the switch is connected to a multiplication block to simulate large scale fading, as described above.

Fig. 4 illustrates the use of the filter set ϕ_i to model a communication channel based on the corresponding cluster σ_i.

F. Integration with Networked Process Flow Simulation

As described in Section I, we use the TLFS tool for networked process flow simulation to demonstrate the utilization of a channel library produced from our link layer simulation approach. The model-based architecture of TLFS facilitates its extension with novel capabilities, such as those presented in this paper for channel modeling. Fig. 5 illustrates the high-level architecture of a TLFS-based networked process flow simulation together with the new TLFS extension to utilize channel libraries that are provided by external tools, such as our link layer simulator.

Here, the factory process flow is modeled (by the simulation tool user) as a dataflow graph, as illustrated in the bottom left part of Fig. 5. The dataflow graph illustration here incorporates various *actors* (dataflow-based functional components) that are useful for simulating factory process flows in TLFS. The actors labeled M and R represent a machine and a rail, respectively, while the actor labeled C is a controller actor that models the control of a set of machine and rail components. The dataflow graph also contains a number of send interface actors (SIAs) and receive interface actors (RIAs) which provide model-based interfaces between the process flow (dataflow) model and the communication network (discrete event) model.

The model-based architecture of TLFS makes it possible to plug in different network simulators to enable co-simulation between such simulators and the dataflow-based process flow simulator within TLFS. In our experiments, we utilize the popular NS-3 [16] simulator as the network simulator, as illustrated by the lower right part of Fig. 5. The blocks labeled NN and AP in this part of the figure respectively represent models for network nodes and an access point. In the extended version of TLFS illustrated in Fig. 5, we configure the network simulation to bypass the PER modeling features built-in to NS-3 and utilize instead the PER/SNR tables provided by the channel library.

For more details on the TLFS-based modeling and co-simulation methods illustrated in Fig. 5, see [3].

IV. Experiments

In this section, we present experiments that demonstrate the utility of our proposed linked layer simulation approach and its integration with networked process flow simulation.

A. Field Measurement Dataset

For the set of field-measurements that are input our link layer simulation experiments, we apply CIR measurements that have been obtained from a measurement campaign performed by NIST, as mentioned in Section III. Details of the measurement techniques involved in this campaign are presented by Candell et al. [11]. We specifically use the field data collected in this campaign from an automotive factory site. Among the different sites surveyed in the measurement campaign, the automotive factory most closely matches the factory production environment context to which this paper is most oriented. All of the field data is archived online [12].

Applying Kashef's clustering technique [13] on the CIR data from the automotive plant, after pre-processing, yields a total four clusters. Fig. 6 shows the *representative CIR* that is derived from each of the four clusters. The representative

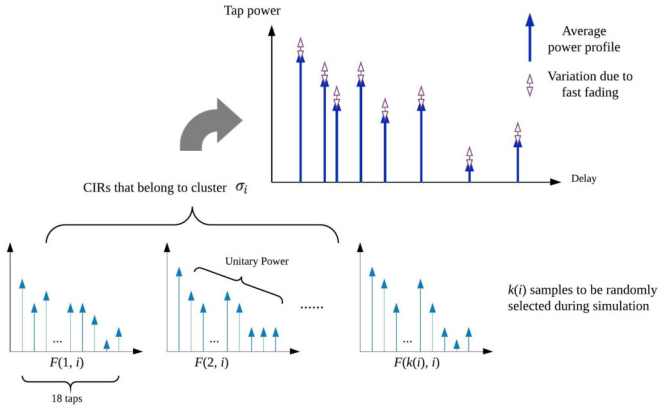

Fig. 4. An illustration of the use of the filter set ϕ_i to model a channel based on cluster σ_i.

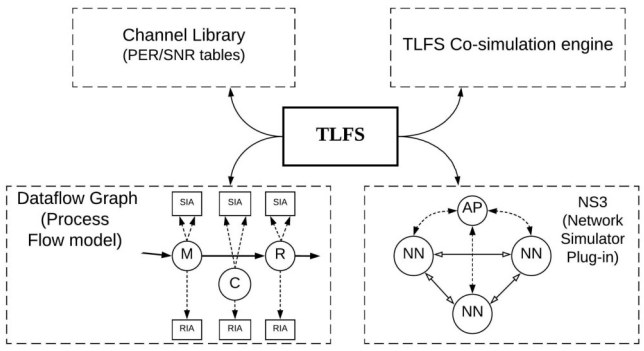

Fig. 5. An illustration of the architecture of a TLFS-based networked process flow simulation together with the new TLFS extension to utilize channel libraries.

CIR of a cluster is simply the CIR that results from averaging all of the CIRs in the cluster. We denote the clusters whose representatives are shown in Fig. 6a–d, respectively, by $\sigma_1, \sigma_2, \sigma_3, \sigma_4$. Cluster σ_1 corresponds to an NLOS channel in the factory, while clusters $\sigma_2, \sigma_3, \sigma_4$ correspond to LOS situations with different delay spreads. Differences in impact on channel performance due to CIRs from different clusters are demonstrated in Section IV-E.

B. Experimental Setup for Comparing PER/SNR Curves

In Section IV-C through Section IV-E, we present relevant experimental results and comparisons involving PER/SNR tables that are derived using our proposed linked layer simulation approach. These experiments are performed using the following parameters.

- Duration of simulation. For each SNR point in each generated table, we simulate until there are 10^5 packets transferred through the given channel or until 10^4 packet errors are detected. Whichever condition is detected first triggers the end of the simulation.
- Reduced tap count parameter. The value of this parameter is set to 18 taps, as stated in Section III-C.
- Protocol. All of the packets are generated randomly and transmitted using the 802.11n protocol with 2.4GHz frequency.
- Packet size. A packet size (PSDU length) of 100 bytes is used in all simulations.

C. Comparison with TGn Channel

In this subsection, we present a comparison between the channel model produced by our link layer simulator and the TGn model [4], which we choose here as a well-known, representative example of a model that is not specific to industrial environments. The differences between results using these two models help to concretely demonstrate the importance of customizing the channel model for an industrial environment when this type of environment is being studied.

Fig. 7 compares the PER/SNR curves generated from our link layer simulator using the measured automotive plant data with simulation results using the TGn model. To present the results without excessive clutter, we select one representative cluster (σ_3), and three TGn delay profiles (B, D, and E). In our settings for simulation, delay profiles B and D correspond to NLOS scenarios, while E is for LOS. The modulation and coding scheme (MCS) is set to MCS4 for all simulations. For a summary of all of the MCS indices for 802.11n, we refer the reader to [17]. The blue curve shows the PER/SNR characteristic derived by simulating cluster σ_3 in our link layer simulator, while the other three curves show PER/SNR results that we obtained by simulating the selected TGn delay profiles in MATLAB.

From Fig. 7, we can see that there can be significant difference between the performance of a real industrial channel and a TGn channel, and that this difference is captured by our link layer simulator. The disparity illustrated in Fig. 7 helps to quantitatively motivate the need for more accurate integration of channel characteristics into networked process flow simulations and related types of simulations for industrial wireless environments.

D. Comparison Involving Different MCS Settings

As mentioned in Section III-E, our link layer simulator can generate a parameterized family of PER/SNR tables based on designer-specified communication parameters. In this section, we demonstrate the utility of this capability by applying the MCS as a parameter for table generation. Fig. 8 shows the generated PER/SNR curves for cluster σ_3 (from the automotive factory) for MCS settings 0 through 6. Here, we use only these 7 MCS settings because the field measurements used in our experiments are based on a SISO antenna. However, our simulation framework of Section III is not restricted to the SISO case. Indeed, it can be applied to MIMO channels. Due to use of field measurements as opposed to mathematical models, the approach in our simulator for handling MIMO channels does not involve traditional MIMO modeling techniques based on correlation matrices (e.g., see [18]).

The results shown in Fig. 8 show that MCS selection can have significant impact on PER. Thus, in the context of WIFS design space exploration, it is useful to have available a parameterized collection of PER/SNR tables where one can easily vary the MCS scheme to assess the overall impact on

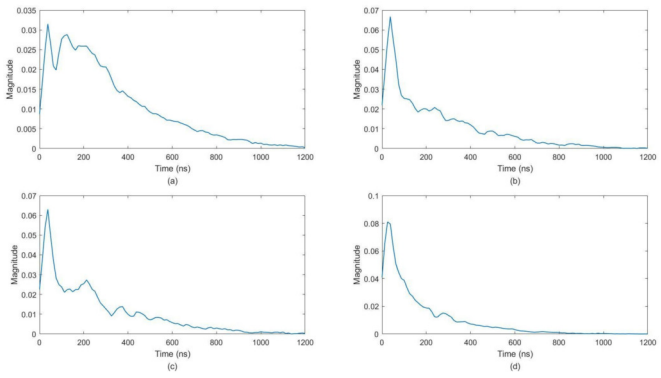

Fig. 6. Representative CIRs derived from the four clusters obtained from the automotive plant measurements. Parts (a)–(d) show the representative CIRs for clusters σ_1 through σ_4, respectively.

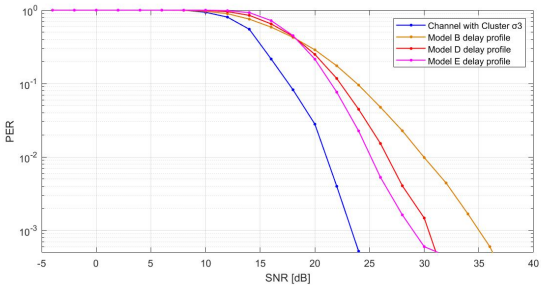

Fig. 7. A comparison between the PER/SNR curves generated from our link layer simulator using the measured automotive plant data with simulation results using the TGn model.

factory system performance in terms of PER and other relevant metrics.

E. Comparison Among Different Clusters

Fig. 9 illustrates the different PER/SNR curves generated by our link layer simulator for the four clusters in the automotive factory dataset. The MCS setting used in this experiment is 4. Cluster σ_1 exhibits poor PER due to its NLOS characteristic and large delay spread. Clusters $\sigma_2, \sigma_3, \sigma_4$, on the other hand, correspond to LOS scenarios. Among these, σ_4 has the lowest PER since signals transmitted in this type of channel have high K-factor, and low delay spread quality. Clusters σ_2 and σ_3 show similar performance characteristics in Fig. 9 with Cluster σ_3 exhibiting slightly better performance.

The results in Fig. 9 illustrate how networked factory simulation results may motivate changes to a factory layout. For example, if better communication reliability is desired than what an NLOS channel can support in a given deployment, then a rearrangement of the layout may be performed to provide a "clean" physical environment for the relevant communication path. This may lead to higher quality PER/SNR curves, which, when plugged into to the networked factory simulator, may provide the desired level of estimated reliability.

F. Integration into Networked Process Flow Simulation

In this section, we present experiments involving the integration into a networked process flow simulator of PER/SNR tables generated by our link layer simulator. As mentioned previously, the networked process flow simulator that we use to demonstrate this integration is TLFS. The experiments presented in this section help to validate the implementation of channel libraries for TLFS based on results of the new link layer simulator, and to demonstrate the kinds of networked process flow simulations and WIFS design space exploration that can be carried out based on field measurements, as enabled through the channel libraries.

The factory process flow model used in these experiments is a pipelined structure with one parts generator, one parts sink, three machines, four rails, and three machine controllers. The parts sink actor represents a subsystem that collects and stores parts after they are processed by the pipeline. For details on

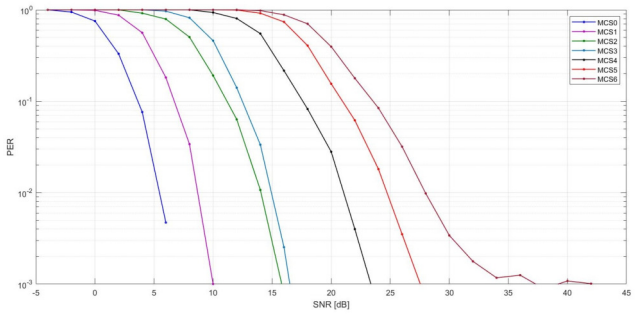

Fig. 8. A comparison across different MCS settings for cluster σ_3.

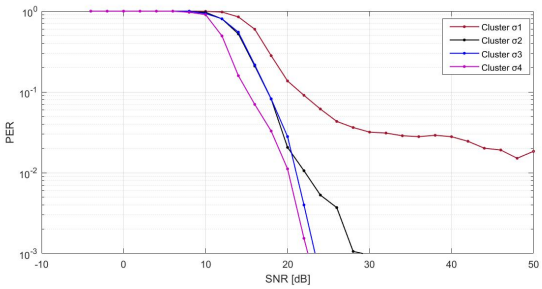

Fig. 9. PER/SNR curves generated by the link layer simulator for the four clusters in the automotive factory dataset.

the modeling of rails, machines, controllers and other types of factory subsystems in TLFS, see [3].

For simulation, the machine working time (the time for a machine to process a given part) is determined randomly from a uniform distribution within the range $[0.9\mu, 1.1\mu]$, were μ is a model parameter that is set to 40 sec. in our experiments. When initializing the process flow model for each simulation run, the physical spacing between the machines, rails, and machine controller was also determined randomly: the distance between adjacent subsystems was determined from a uniform distribution on $[0.9\delta, 1.1\delta]$, where $\delta = 2$ meters. In each simulation run, 10 products were generated by the parts generator actor and processed through the complete pipeline.

Fig. 10 shows networked process flow simulation results for the pipeline model described above using the generated PER/SNR tables associated with different MCS settings. Here we fixed the cluster to be σ_3, and varied MCS indices across the set $\{0, 1, 3, 6\}$. Here, to present the results without too much clutter, we have selected a subset of four representative MCS indices rather than evaluating all of them. For each MCS index, we ran 10 TLFS simulations independently and averaged the results over all the packets. The communication protocol employed in the experiment was the 802.11n protocol.

As shown in Fig. 10, the average communication delay decreases with increasing MCS indices, which is consistent with the increasing data rates associated with higher indices.

Fig. 11 shows networked process flow simulation results using the generated PER/SNR curves for the different clusters $\sigma_1, \sigma_2, \sigma_3, \sigma_4$ under a fixed MCS setting of 4. Here, we have randomly selected 4 as a representative MCS index to show results for a fixed index. In this experiment, we increase the mean spacing parameter to $\delta = 18$ meters. At this increased spacing, the SNR falls into a range where there is significant variation among the PER levels across the different clusters. All other parameters are kept the same as in the experiments associated with Fig. 10. The results in Fig. 11 are consistent with those in Fig. 9. We see that The PER/SNR curve for σ_4 leads to the lowest average communication delay and the one with σ_1 gives the highest delay.

V. Conclusions

In this paper, we have developed new methods for integrating realistic models of wireless factory communication

network channels into simulation tools and design space exploration processes for wireless-integrated factory systems. Our approach involves systematically incorporating data from field measurements into link layer layer simulation, and generating channel libraries that accurately incorporate channel characteristics into networked process flow simulation. We have demonstrated the proposed methods using a large dataset containing field data collected from an automotive factory. Interesting directions for future work include systematic methods for setting pre-processing parameters based on measurement-process parameters; integration of channel libraries into automated design optimization processes for factory process flows; and application of channel libraries to design space exploration for other types cyber-physical systems.

DISCLAIMER

Certain commercial equipment, instruments, materials, software or systems are identified in this paper in order to specify the experimental procedure adequately. Such identification is not intended to imply recommendation or endorsement by the National Institute of Standards and Technology, nor is it intended to imply that the materials or equipment identified are necessarily the best available for the purpose.

REFERENCES

[1] A. A. Kumar S., K. Ovsthus, and L. M. Kristensen, "An industrial perspective on wireless sensor networks — a survey of requirements, protocols, and challenges," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 3, pp. 1391–1412, 2014.

[2] K. Wiklundh, "Interference challenges for industry communication," 2019, pDF presentation slides from keynote at WFCS 2019, downloaded from https://www.miun.se/en/thank-you-for-participating on 01/22/2020.

[3] J. Geng *et al.*, "Model-based cosimulation for industrial wireless networks," in *Proceedings of the IEEE International Workshop on Factory Communication Systems*, 2018, pp. 1–10.

[4] V. Erceg, "TGn channel models," IEEE P802.11 Wireless LANs, Tech. Rep. IEEE 802.11-03/940r4, 2004.

[5] J. Medbo and P. Schramm, "Channel models for HIPERLAN/2 in different indoor scenarios," Ericsson Radio Systems AB, Tech. Rep. 3ERI085B, 1998.

[6] T. Abbas, J. Nuckelt, T. Kürner, T. Zemen, C. Mecklenbräuker, and F. Tufvesson, "Simulation and measurement based vehicle-to-vehicle channel characterization: Accuracy and constraint analysis," 2014, arXiv:1410.4187v1 [cs.NI].

[7] J. Peil, M. Damsaz, D. Guo, W. Stark, R. Candell, and N. Moayeri, "Channel modeling and performance of Zigbee radios in an industrial environment," National Institute of Standards and Technology, Tech. Rep., September 2016.

[8] Y. Liu, R. Candell, K. Lee, and N. Moayeri, "A simulation framework for industrial wireless networks and process control systems," in *Proceedings of the IEEE World Conference on Factory Communication Systems*, 2016, pp. 1–11.

[9] R. Patidar, S. Roy, T. R. Henderson, and A. Chandramohan, "Link-to-system mapping for ns-3 Wi-Fi OFDM error models," in *Proceedings of the Workshop on ns-3*, 2017, pp. 31–38.

[10] H. Li, J. Geng, Y. Liu, M. Kashef, R. Candell, and S. Bhattacharyya, "Design space exploration for wireless-integrated factory automation systems," in *Proceedings of the IEEE International Workshop on Factory Communication Systems*, May 2019, 8 pages in online proceedings.

[11] R. Candell *et al.*, "Industrial wireless systems radio propagation measurements," National Institute of Standards and Technology, Tech. Rep. 1951, 2017.

[12] "Networked control systems group — measurement data files," https://www.nist.gov/el/intelligent-systems-division-73500/networked-control-systems-group/measurement-data-files, 2020, visited in January 2020.

[13] M. Kashef, R. Candell, and Y. Liu, "Clustering and representation of time-varying industrial wireless channel measurements," in *Proceedings of the Annual Conference of the IEEE Industrial Electronics Society*, 2019, pp. 2823–2829.

[14] C. Mehlfuhrer and M. Rupp, "Approximation and resampling of tapped delay line channel models with guaranteed channel properties," in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, 2008, pp. 2869–2872.

[15] M. Damsaz, D. Guo, J. Peil, W. Stark, N. Moayeri, and R. Candell, "Channel modeling and performance of Zigbee radios in an industrial environment," in *Proceedings of the IEEE World Conference on Factory Communication Systems*, 2017, pp. 1–10.

[16] *ns–3 Tutorial, Release ns–3.25*, ns–3 Project, 2016.

[17] "MCS index for 802.11n and 802.11ac chart," https://www.wlanpros.com/resources/mcs-index-802-11ac-vht-chart/, 2020, visited in January 2020.

[18] L. Schumacher, K. I. Pedersen, and P. E. Mogensen, "From antenna spacings to theoretical capacities — guidelines for simulating MIMO systems," in *Proceedings of the International Symposium on Personal, Indoor and Mobile Radio Communications*, 2002, pp. 587–592.

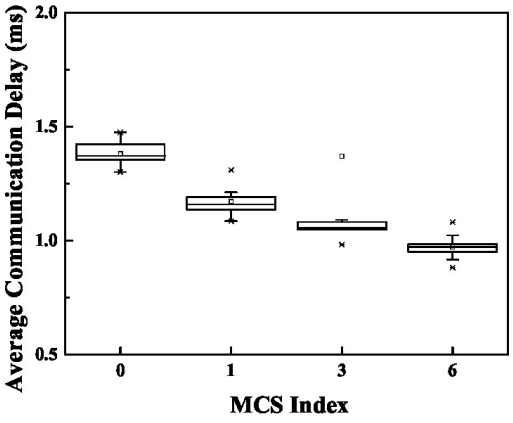

Fig. 10. An example of networked process flow simulation results using PER/SNR data from a channel library produced by our link layer simulator.

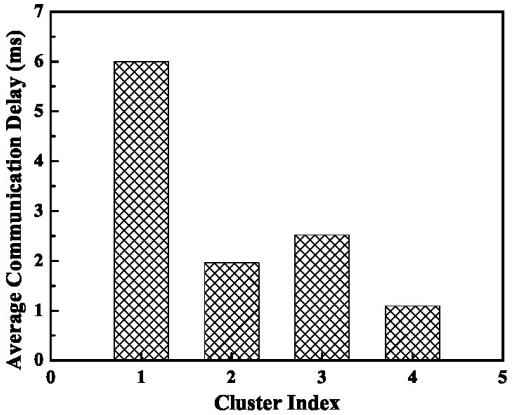

Fig. 11. Networked process flow simulation results for the four different clusters under $\mathrm{MCS} = 4$.