A Collaborative Work Cell Testbed for Industrial Wireless Communications — The Baseline Design

Yongkang Liu*, Richard Candell†, Mohamed Kashef*, Karl Montgomery†
*Advanced Network Technologies Division †Intelligent Systems Division
National Institute of Standards and Technology, Gaithersburg, Maryland, USA
Email: {yongkang.liu, richard.candell, mohamed.hany, karl.montgomery}@nist.gov

***Abstract*—A work cell is an essential industrial environment for testing wireless communication techniques in factory automation processes. A new testbed was recently designed and developed to facilitate such studies in work cells by replicating various data flows in an emulated production environment. In this paper, the testbed's baseline design is presented which characterizes deterministic and reliable communication needs between work cell components in a typical machine tending application. Special design issues are discussed regarding safety measures in collaborative robotic operations and network synchronization among distributed machines. Measurement plans in the hardwired baseline are also introduced along with further wireless extensions. The testbed can serve as a representative cyber-physical system (CPS) model to verify industrial wireless techniques in support of mission-critical data transmissions.**

***Index Terms*—industrial wireless communications, industrial wireless networks, industrial wireless testbed, factory automation processes, testbed design.**

I. INTRODUCTION

Industrial communication networks leverage operational technology (OT) insights/decisions in recent Industry 4.0 and Smart Manufacturing initiatives through mission-critical data sharing between field instruments and factory automation controllers [3], [4]. Compared with hardwired connections, wireless links have unique advantages in connecting field sensors and actuators with reduced cabling cost and natural support of mobility [1]. A number of industrial wireless solutions have been proposed for improved production efficiency, asset health, and workplace safety [2]. However, they need to prove the full support on agile plant operations with trusted transmission timeliness and reliability before being adopted on the factory floor. Evaluating the capabilities of various and diverse wireless technologies has turned out to be a challenging but essential task to promote industrial wireless applications [5], [6].

An evaluation platform provides necessary details of performance requirements and operation specifications in typical industrial wireless use cases, e.g., the plant layout, process workflow, wireless channel model, and data traffic pattern. It plays an important role in verifying wireless network design and comparing the performance of different wireless technologies. Such modeling efforts have been taken both at the macroscopic level, e.g., on spatial statistics of the node density and traffic load on the factory floor, and at the microscopic level, e.g., on the latency and interference in individual transceiver pairs. Based on these models, system verification methods using co-simulation platforms [7], [8], [9], [10], hardware-in-the-loop (HIL) experiments [11], and testbeds [12] become popular in studying the unique industrial environments and service characteristics.

A few new challenges have emerged which need a further investigation when modeling various plant factors. First, most evaluation workflows are one-way, i.e., describing the impact from industrial environments and operations onto wireless transmissions. Since industrial systems, as complex cyber-physical systems (CPS), are featured with the interplay between industrial processes and data networks, the model is expected to represent interactive connections between OT and information technology (IT) systems. Second, machine-to-machine (M2M) communications carry and distribute data in a vastly different way compared to the conventional Internet data. The model also needs to characterize and verify various traffic patterns, both empirical and statistical. Last but not least, current models are usually built upon snapshots of existing industrial practices which only capture environments and activities of the status quo for the network design and optimization. As CPS innovations have been evolving in emerging industrial use cases, the new model has to be more flexible in compliance with both short- and long-term network implementations.

Through measurements of process and network activities to finely tune performance requirements on industrial wireless networks, a testbed is being developed at the National Institute of Standards and Technology (NIST). This paper introduces the baseline design which identifies a variety of data needs in the emulated industrial operations and calibrates the performance under hardwired connections before extending to wireless alternatives. Generally, the testbed is featured with three aspects of innovations.

First, the testbed picks a work cell as the target model which is at the "right" size to capture essential data traffic patterns between industrial devices in a manufacturing cycle. By inspecting both internal module coordination and external interactions with upper-level management systems, the proposed work cell testbed serves as a good reference to verify industrial wireless networks in supporting efficient manufacturing operations.

Second, the testbed is specialized in emulating collaborative operation scenarios various machining tools working with industrial robots. Cooperations between machines and their

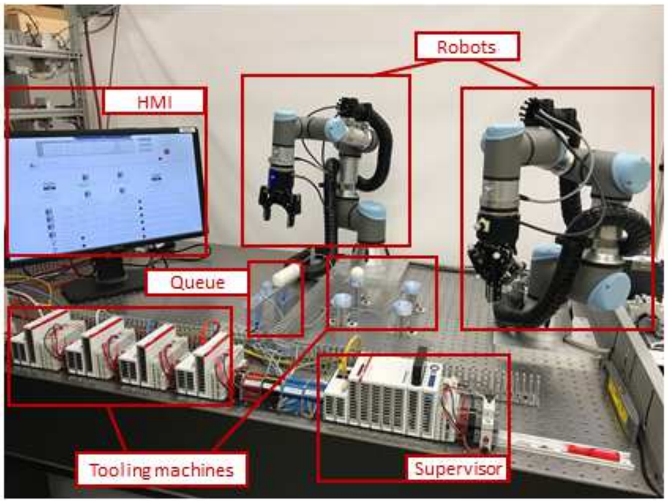

Fig. 1: Collaborative work cell testbed

robotic partners are managed by the work cell supervisor through customized data flows, e.g., the context information, size, and frequency, following industrial specifications.

Third, the testbed provides rich work cell footprints in production operations which facilitates network measurements and evaluation. Compared with previous modeling efforts which simply treated individual work cells as buffers of working parts/orders [10], the testbed further characterizes data flows within and beyond work cells to fully represent data features in complex industrial scenarios.

The remainder of the paper is organized as follows. The system architecture is introduced along with brief discussions on the emulated production processes in Section II. Details about the design of machine emulators are presented in Section V. The network synchronization issues and safety-related operations are discussed in Section VI and Section IV, respectively. The ongoing measurement and wireless extensions are introduced in Section VII. Concluding remarks are given in Section VIII.

II. OVERVIEW OF TESTBED DESIGN

A. Work Cell Components

As shown in Fig. 1, the testbed emulates a generic work cell in the manufacturing factory which consists of multiple components including a supervisory control unit, machines, interstage buffers, robots, and human workers.

1) Supervisor: The supervisory control unit, or supervisor, manages its work cell by monitoring the whole production process, scheduling production based on incoming orders, and coordinating inter-node actions. Meanwhile, it also serves as the agent on behalf of the entire work cell to communicate with the upper-level managing systems in the factory, such as supervisory control and data acquisition systems (SCADA) and manufacturing execution systems (MES). A programmable logic controller (PLC) usually plays the supervisor's role in the work cell. In the testbed, we use a Beckhoff CX2020 PLC as the work cell supervisor which is equipped with various communication interfaces for internal and external information exchanges [13].

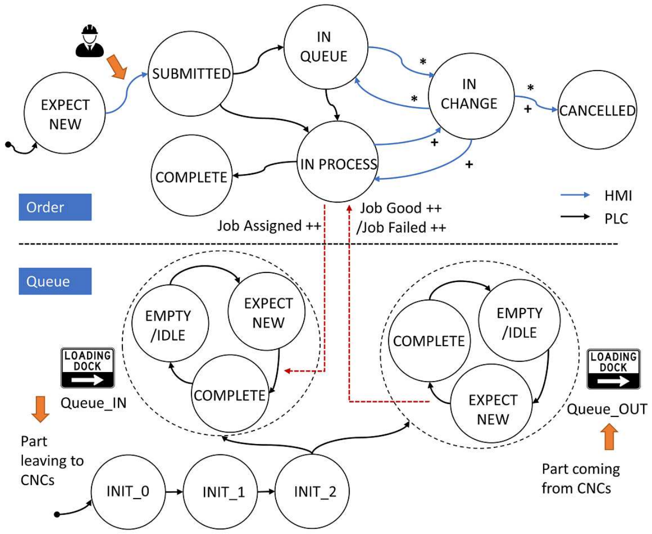

Fig. 2: State machines of the order and queue modules

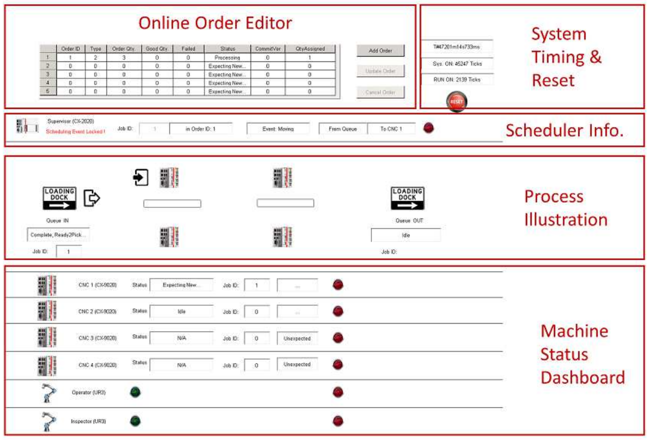

Fig. 3: Snapshot of the testbed human-machine interface

2) Interstage Queue: The interstage queue is comprised of two loading zones in the work cell, i.e., the input (Queue_IN) and output (Queue_OUT) buffers, which serve as the start and end points for a single job, respectively. As shown in Fig. 2, the input buffer accommodates the incoming raw parts into the work cell and the output buffer collects the finished parts, either good or failed. The supervisor detects the arrival/departure events in the buffers with proximity sensors, one for each, and updates the order status accordingly.

3) Machines: Four computer numerical control (CNC) machines are considered in the testbed, whose behaviors in the machine tooling and communications are characterized by emulation models. Each CNC machine consists of a PLC, a part holder, and a proximity sensor. The PLC mimics state transitions of the CNC machine in its tooling cycle and exchanges the machine status and job information with the supervisor. The part holder represents the machine's working zone where the proximity sensor is used to monitor the part arrival/departure. The PLC connects the sensor to its digital input/output (I/O) module and samples the input signal. Four

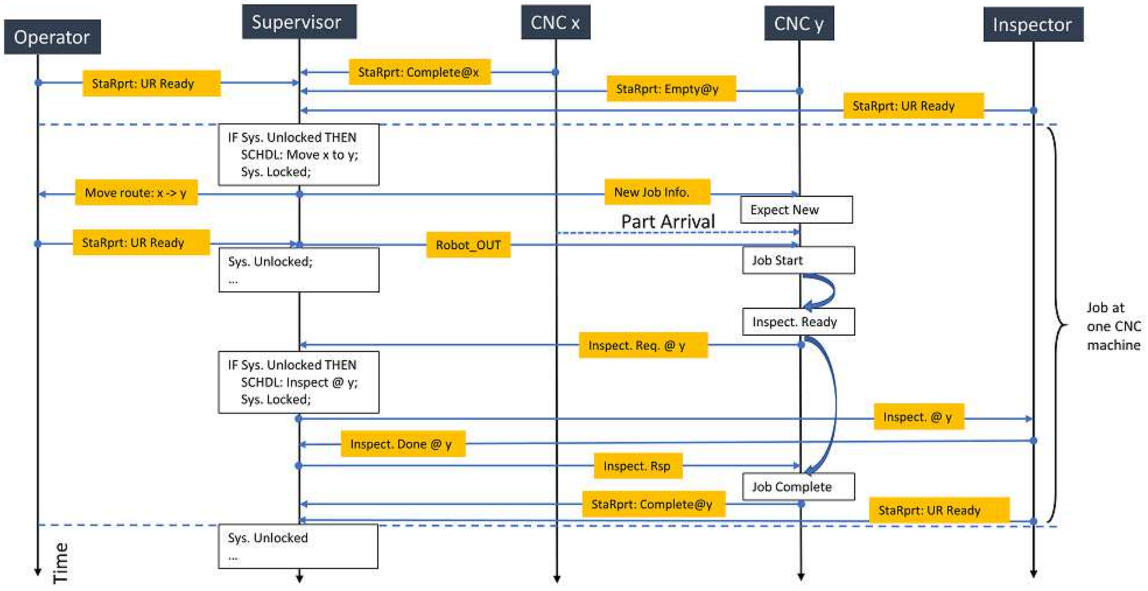

Fig. 4: Timeline illustration of communication messages in an intermediate tooling procedure

Beckhoff CX9020 PLC are used as the emulators along with the propriety I/O modules [14]. Details of the emulator design are discussed in Section V.

4) Robotic Laborers: Two UR3 robots are used in the testbed [15]. Each robot has six degrees of freedom (6 DoF) and is equipped with a gripper and a 6 DoF force torque sensor [16], [17]. Robots mainly communicate with the supervisor to receive actuation commands and report their status. Based on each robot's role in work cell operations, UR3 programs perform motion commands such as waypoint selection and trajectory planning.

5) Human Workers: A collaborative work cell may be operated by human workers or not. In the testbed, human workers can remotely monitor and interact with the automated production process, such as placing orders and stopping/resetting the production, through a human-machine interface (HMI) as shown in Fig. 3. The real-time status information as displayed by HMI is updated through the supervisor and collected from distributed components.

B. Baseline Use Case: Machine Tending

The baseline design studies a machine tending use case. Jobs are assigned to the work cell in batches through the HMI as shown in Fig. 2. Each batch, namely an order, contains a number of jobs/parts of the same type with a specific tool path, i.e., a sequence of moves operated at one or more machines. The two robots play different roles in the production: one as the operator (OPT) and another as the inspector (INS). OPT is in charge of transporting parts between job stops. A job stop refers to the working zone of a machine or the input/output loading zone. The INS robot checks the part quality after each tooling step and reports the inspection result to the supervisor. Based on the result, the supervisor then orders the operator to either move the part to the next stop along the path (if it passed the check) or drop it to Queue_OUT with a defect mark (if it failed). The inspection result is simulated at the inspector by a random variable associated with the emulated tooling operation.

TABLE I: Exemplary specifications of data flows between work cell components

Link	Data	Update Rate	Size (Bytes)	Protocol
Supervisor -CNC	Status report	1 Hz - 100 Hz	10s	ADS
	Safety	100+ Hz	10s	ADS
	Inspection request/response	On-demand	10s	ADS
CNC-CNC	Motion control	1000 Hz	A few	ADS
Supervisor -Robot	Actuation	1 Hz - 50 Hz	A few	Modbus
	Safety	125 Hz	A few	Modbus
Robot -Peripheral	6 axis force and torque sensor	100 Hz - 500 Hz	100s	TCP/IP
Supervisor -External	HMI	10 Hz - 50 Hz	100s	ADS
	IoT	$>$1 Hz	10s - 100s	MQTT

C. Work Cell Communications

The topology of the work cell network is centered around the supervisor which acts as the information hub and gateway for both internal and external data flows. Fig. 4 illustrates messages that are transported between work cell components in a job move. Connections within and beyond the work cell are managed by different communication protocols. Among them, the inter-PLC links are carrying transmission control protocol/Internet protocol (TCP/IP) based TwinCAT Automation Device Specification (ADS) messages [18]. ADS is a medium-independent protocol for the communication between Beckhoff's TwinCAT devices. The supervisor communicates with robots through Modbus which allows the data exchange between heterogeneous industrial appliances in the shared registers at the supervisor.

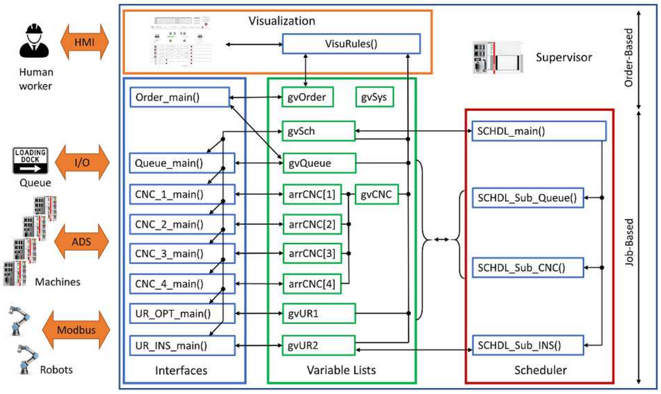

Fig. 5: Architecture of work cell supervisor functions implemented in the PLC

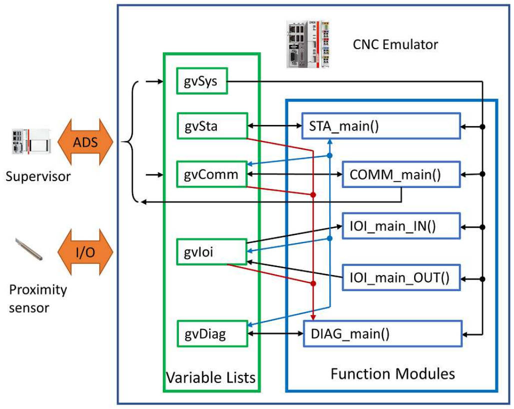

Fig. 6: Architecture of the CNC emulator

Generally, the data exchange in a work cell is determined by the associated production operations. For process variables (PV) regarding the production efficiency, the supervisor needs to collect the updates from remote machines to estimate the loads of individual stations and ensure the quality. For the ones with the asset health, the supervisor uses them to schedule the maintenance downtime and estimate the cost. To coordinate the collaborative operations in the work cell, the real-time status of a machine should be made known to its partners so that the synchronous operation can mitigate errors and improve the quality. Besides routine exchanges, part of the CNC machine data is state-related, i.e., data are transmitted according to the current state in which the machine stays. Table I summarizes the emulated data flows in the testbed.

III. Design of Work Cell Supervisor

As shown in Fig. 5, the supervisor of the testbed consists of four main function blocks in its architecture: the scheduler (SCHDL), interfaces, visualization, and global variable lists (GVL). Specifically, the scheduler is in charge of assigning production jobs to machines and robots. The interface block handles the communications with various CNC machines and UR3 robots in the work cell. Meanwhile, it also updates the order and queue information in emulation experiments. Using the PLC's visualization library, the visualization block controls the testbed's HMI. The supervisor's first three blocks are implemented as the PLC function modules which maintain their own status locally. The system-wide data sharing between function modules takes place in GVL. System variables associated with a specific function or work cell component are managed in the GVL named after it, e.g., "gvCNC" contains four array objects each of which stores the relevant information of a CNC machine in the work cell.

From the perspective of information processing, functions and data memories in the supervisor can be divided into two planes: order-based and job-based. Order-based functions deal with incoming orders, update the order status based on real-time production results, and maintain the inventory. On the other hand, job-based functions mainly work on the associated work cell components and coordinate their production activities following the schedule. Such a modular design allows the supervisor to easily adapt to the composition of a specific work cell and utilize the state-of-the-art techniques to leverage individual functions.

IV. Safety-aware Scheduling and Operations

In the baseline, the testbed considers production activities without physical human contact where human workers stay in the remote safety zone and interact with the process through HMI. Major safety concerns include collision risks between robots and the interruption of machining when a robot hits the running machine. Therefore, the testbed is designed with multiple safety approaches to eliminate possible risks to protect the asset.

First, the supervisor sets a safety flag in its scheduler to indicate if there is an active robot moving in the work cell. The scheduler only assigns at most one robot to be actively operating. Once the flag is set, the locked scheduler would not assign a new job to another robot so that collisions are avoided.

Second, the active robot will keep notifying the contacted machine(s) in the current job so that the machine would not start to process the part until the robot returns to the safety zone. As shown in Fig. 4, the "Robot_OUT" message indicates the clearance of the contact.

Besides, an additional logic check on the waypoint information is performed at the robot to verify the fetched instruction through Modbus. Meanwhile, the supervisor will clear the waypoint information set in the registers right after the robot confirms the reception. In this way, it prevents the robot from repeating out-of-date operations in case that the new waypoint information is lost in the transmission. Initial experiments confirm that introducing such an approach allows error-free operations through very light supervisor-robot Modbus communications as low as 1 Hz.

V. Machine Tool Emulation

The testbed is aimed to evaluate the mutual impact between data transmissions and the work cell performance. Therefore, the CNC emulator is mainly focused on mimicking the machine's behaviors with time dependent and statistical performance features, such as the production efficiency, error

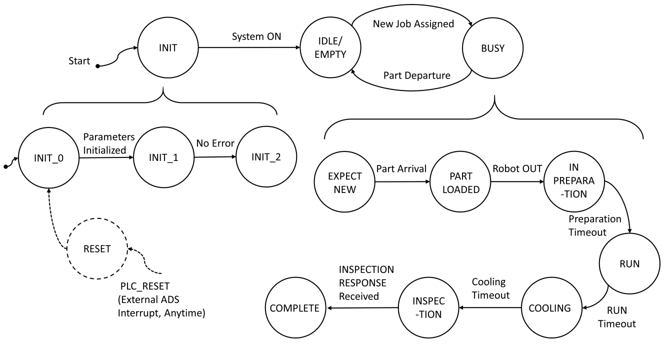

Fig. 7: CNC state machine

and downtime distributions, and part defects. Meanwhile, the emulators also help shape the work cell data traffic with their periodic status updates and on-demand messages during the production. Following the similar modular design as the supervisor, the CNC emulator also defines its function modules and GVL in the implemented PLC. Specifically, function modules include the state machine (STA), communications (COMM), I/O module interfaces (IOI)[1], and diagnostics (DIAG). The associated system variables shared in between are organized in GVL, e.g., “gvSta” maintains variables related with the state machine and “gvSys” contains the system-wide information such as the machine’s identification (ID) and network address.

To fully capture the operational and communication activities of a machine tool, the CNC emulator conducts state-dependent operations and communications characterized by the state machine as shown in Fig. 7.

The state machine is defined in the STA function module which contains three main states: initialization (INIT), idle, and busy. Each main state can contain a few substates which characterize further details of operations. INIT along with its substates facilitate the synchronization among distributed nodes whose design will be discussed with more detail in Section VI. The substates of the busy state represent a series of machine operations regarding a single job. The dwelling time in each (sub-)state can be either timed according to the machine’s specification, e.g., the approximate G-code execution time and material removal rate, or determined by external events that trigger state transitions, e.g., a notification message. The randomness can also be introduced based on statistical machine/production models. Examples of randomness components in the models include: 1) the time of a tooling procedure; 2) time varying energy consumption in different states, e.g., power variations in material-drilling processes; 3) tool life estimation; 4) part defect rate; 5) measurement drift between calibrations; and 6) safety related events, e.g., unexpected interrupts due to object intrusion. Using empirical models and measurement data, we can model the above performance metrics statistically and regenerate the state-related traffic for the studied machine.

Therefore, the machines emulated in the testbed can be programmed to highlight the details of real practices to study the network impact on the work cell performance. PV can be modeled in the testbed focusing on different topics such as 1) the production (task) efficiency, e.g., the execution time, material removal rate, energy consumption, and part defect rate; 2) asset health, e.g., the tool life time, failure probability, and downtime schedules for calibration and maintenance; and 3) work cell collaboration, e.g., the clock drift, coordination precision, and safety. Besides checking the network support on routine data transmissions as scheduled, the testbed is particularly useful for testing the network performance in extreme cases with rare occurrences. The machine emulator can produce the traffic in the special use cases, such as the recovery from unexpected overload events or in emergency cases, and repeat it for comparative studies.

The quality of the “product” is also virtually rendered in the testbed. The result of each single part after a machining process is randomly generated following the statistical model to mimic the defect rate in a real machine. The inspector is in charge of generating the result and returning it to the supervisor for scheduling the next move. According to the study of the quality and quantity relationship in production systems [19], [20], part failures have both independent and dependent causes. The independent failure follows a Bernoulli distribution with the uncertainty of temporal independence. On the other hand, the dependent types of failures, which are often referred to as “persistent” or “systematic” ones, are those caused by tool failures, such as the broken drill or clog in the painting tube. In such cases, the failure of product is highly related with the asset failure rate. Since both types of failures are decoupled by their nature, the testbed carries the failures of the product as well as the ones related with assets to emulate the occurrences of various failures across time. The delivery delay or loss in communication links also affect the performance of operations and safety measures.

VI. Synchronization of Networked Components

Since work cell components are collaboratively working in the production, the testbed implements multiple approaches to coordinate these distributed nodes.

First, we develop a phased initialization process at the beginning of each experiment. The testbed initialization includes three steps:

INIT_0: Parameter initialization/reset;
INIT_1: Logic error check and confirmation; and
INIT_2: Loading ready-to-go state.

The supervisor keeps the pace by triggering the state transition only after all components have met the state-specific conditions. Meanwhile, the testbed also supports the online reset through the HMI as shown in Fig. 3. Once the reset button is clicked, the supervisor will send the reset commands to individual nodes and direct them to restart from INIT_0.

Besides signaling procedures, the testbed also introduces global clock synchronization throughout the work cell. In the work cell, a Meinberg Lantime M900 time server provides the IEEE 1588 precision time protocol (PTP) synchronization service as the grandmaster [21]. The supervisor PLC is

---

[1] The IOI functions are further grouped into IOI IN and IOI OUT, respectively.

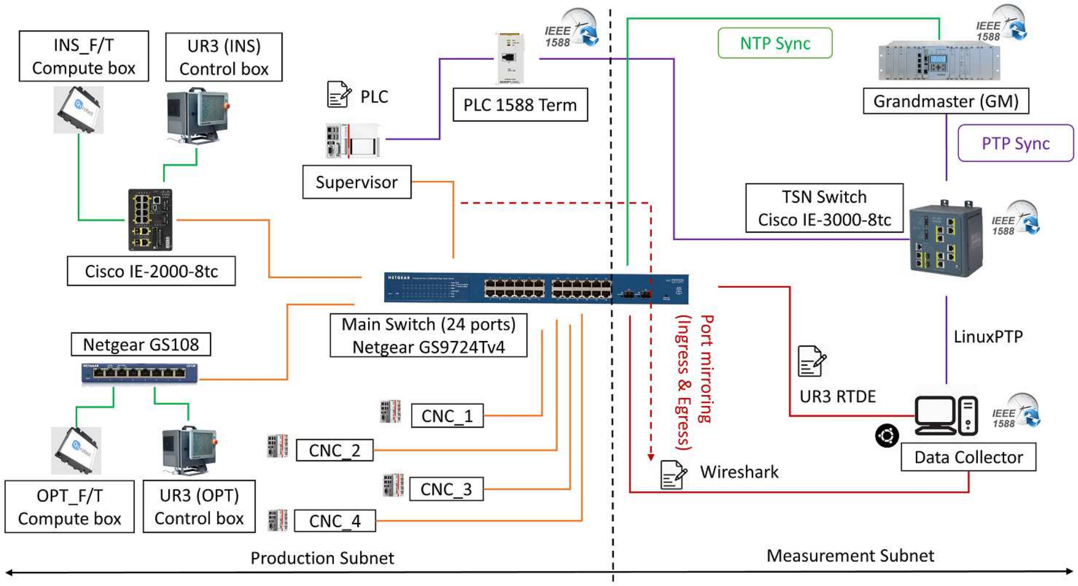

Fig. 8: Illustration of network topology in baseline measurements. Green and orange lines indicate connections for production data and process-related traffics where the orange ones are candidates to be replaced by wireless alternatives in the next phase. Purple lines are used for the PTP synchronization purpose; red lines carry measurement data.

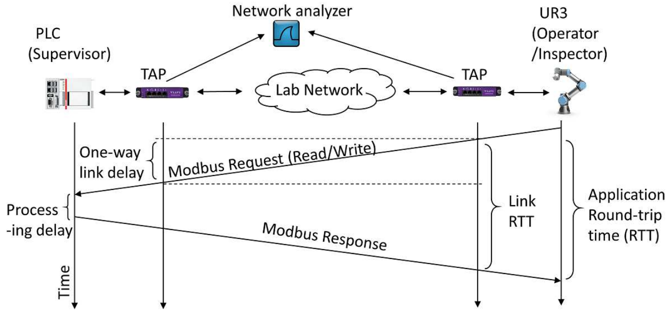

Fig. 9: Illustration of using network TAP devices in the link level delay measurement

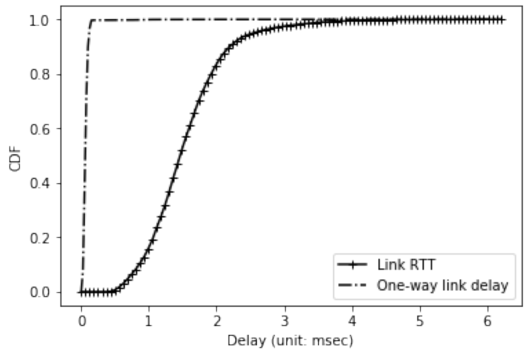

Fig. 10: Result of the link level delay measurement of Modbus transactions in the baseline network

equipped with a Beckhoff IEEE 1588 terminal to synchronize its local clock with the time server [22]. Measurement devices also run the LinuxPTP software to render time stamps in collecting the real time status of UR3 robots and network traffic captures [23].

VII. MEASUREMENTS AND WIRELESS EXTENSIONS

A. *Testbed Measurements*

System and network measurements are performed in the testbed which employs various performance metrics regarding the production efficiency, product quality, and network utility in highly discrete manufacturing processes [24]. An illustration of the network topology used in baseline measurements is shown in Fig. 8. The main observation point for network traffic is set at the supervisor as the testbed takes a centralized topology. Table I also indicates that the majority of data flows assumed in the work cell operations are associated with the supervisor. Therefore, data that are routed from/to the supervisor are collected. Specifically, all work cell components are connected to an industrial-grade switch whose ports are further separated into production operation and measurement uses. Utilizing the switch's "port mirroring" function, we copy and forward the data from the supervisor's operation port to the measurement port where a computer collects the data with network packet analyzers, e.g., WireShark. Data collected in individual experiments will be stored for future analysis and modeling.

As part of the proof of concept, we introduce the network test access point (TAP) devices in the link-level measurements to study the impact of link-level transmissions on the work cell performance, e.g., packet losses of mission-critical PV updates. As shown in Fig. 9, we use two TAP devices to collect data copies at both ends of a Modbus link between the supervisor and the UR3 robot and employ Using Python's Scapy library to obtain link delay statistics [26]. Fig. 10 presents the cumulative distribution functions (CDFs): one for the round-trip time (RTT) in Modbus transactions between Supervisor and UR3 operator, and another for the one-way link delay. The average values of the link RTT and one-way link delay are 1.528 msec and 0.0627 msec, respectively, in a 3-hop Ethernet path as shown in Fig. 8. A longer delay would be expected along with link failures in lossy wireless channels.

B. Wireless Extensions

Based on the baseline design, wireless extensions are also underway. As each network node is equipped with Ethernet adapter(s), hardwired connections between work cell components can be replaced by wireless links if the Ethernet-wireless adapters are used. Currently, we are working with industrial partners to verify the wireless solution using wireless local area network (WLAN) radios. To reduce the conversion delay between Ethernet packets and WLAN packets, the Ethernet-WLAN conversion takes place in the link layer (Layer 2 forwarding) where both Ethernet packets and wireless packets share the same network address of the node. Channel emulation is also considered in the testbed evaluation to mimic the channel response in real factory radio environments [25]. The time sensitive networking study over wireless links is another target in this project. The synchronized clocks of industrial equipment facilitate collaborative operations, e.g., in coordinated robot movements, and leverage the management of orthogonal time-frequency radio resources.

VIII. CONCLUSIONS

In this paper, we have presented a work cell testbed and explained design details for both hardware and software implementations. In addition, measurement techniques and the applicability of wireless links to the design have also been discussed. The testbed is aimed to serve as an evaluation platform for verifying the performance of different wireless technologies in support of deterministic and reliable industrial communications. As an ongoing effort, the current version is built as a baseline with hardwired Ethernet connections between individual components. In future work, we will introduce wireless links and evaluate their performance in harsh industrial radio environments. The future progress and measurement data will be released in the NIST public domain repository as a reference for modeling efforts and comparative studies on industrial wireless technologies [27].

DISCLAIMER

Certain commercial equipment, instruments, or materials 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] L. L. Bello, J. Åkerberg, M. Gidlund and E. Uhlemann, "Guest Editorial Special Section on New Perspectives on Wireless Communications in Automation: From Industrial Monitoring and Control to Cyber-Physical Systems," *IEEE Trans. Ind. Informat.*, vol. 13, no. 3, pp. 1393-1397, June 2017.

[2] V. K. L. Huang, Z. Pang, C. J. A. Chen, and K. F. Tsang, "New Trends in the Practical Deployment of Industrial Wireless: From Noncritical to Critical Use Cases," *IEEE Ind. Electron. Mag.*, vol. 12, no. 2, pp. 50–58, 2018.

[3] H. Kagermann, W. Wahlster, and J. Helbig, "Recommendations for Implementing the Strategic Initiative Industrie 4.0", Industrie 4.0 Working Group, 2013.

[4] A. B. Feeney, S. Frechette and V. Srinivasan, "Cyber-Physical Systems Engineering for Manufacturing", in *Industrial Internet of Things - Cybermanufacturing Systems*, S. Jeschke et al., Eds. Springer Nature, 2017, pp. 81–110.

[5] J. Ansari et al., "Demo: a realistic use-case for wireless industrial automation and control," in *Proc. NetSys 2017*, Gottingen, 2017, pp. 1-2.

[6] Y. Liu, R. Candell, M. Kashef, and L. Benmohamed, "Dimensioning Wireless Use Cases in Industrial Internet of Things," in *Proc. IEEE WFCS'18*, Imperia, Italy, Jun. 2018.

[7] E. Galli, G. Cavarretta and S. Tucci, "HLA-OMNET++: an HLA compliant network simulation," in *Proc. DS-RT'08*, pp. 319-321, 2008.

[8] H. Neema, et. al. "Model-Based Integration Platform for FMI Co-Simulation and Heterogeneous Simulations of Cyber-Physical Systems," in *Proc. 10th International Modelica Conference*, 2014.

[9] Y. Liu, R. Candell, K. Lee, and N. Moayeri, "A Simulation Framework for Industrial Wireless Networks and Process Control Systems," in *Proc. IEEE WFCS'16*, Aveiro, Portugal, May 2016.

[10] J. Geng et al., "Model-based cosimulation for industrial wireless networks," in *Proc. WFCS 2018*, Imperia, 2018, pp. 1-10.

[11] J. Kölsch, C. Heinz, S. Schumb and C. Grimm, "Hardware-in-the-loop simulation for Internet of Things scenarios," in *Proc. Workshop MSCPES 2018*, Porto, 2018, pp. 1-6.

[12] R. Candell, T.A. Zimmerman, and K.A. Stouffer, "An Industrial Control System Cybersecurity Performance Testbed", NISTIR 8089, Dec. 2015.

[13] Beckhoff, "CX2020 Basic CPU Module", 2018. [online]. Available: https://www.beckhoff.com/english.asp?embedded_pc/cx2020.htm. [Accessed January 6, 2019].

[14] Beckhoff, "CX9020 Basic CPU Module", 2018. [online]. Available: https://www.beckhoff.com/english.asp?embedded_pc/cx9020.htm. [Accessed January 6, 2019].

[15] Universal Robots, "Universal Robot UR3", 2018. [online]. Available: https://www.universal-robots.com/products/ur3-robot/. [Accessed January 6, 2019].

[16] Robotiq, "2F-85 and 2F-140 Grippers", 2019. [online]. Available: https://robotiq.com/products/2f85-140-adaptive-robot-gripper. [Accessed on January 10, 2019].

[17] OnRobot, "6 axis Force Torque Sensor", 2018. [online]. Available: https://onrobot.com/products/hex-force-torque-sensors/. [Accessed January 6, 2019].

[18] Beckhoff, "ADS Introduction", Beckhoff Information System, 2018. [online]. Available: https://infosys.beckhoff.com/. [Accessed January 6, 2019].

[19] I. C. Schick, S. B. Gershwin, and J. Kim, "Quality/Quantity Modeling and Analysis of Production Lines Subject to Uncertainty, Phase I, Final Report", May 2005. [online]. Available: http://cell1.mit.edu/papers/GM_PhaseI_FinalReport-2005.pdf

[20] J. Kim and S. B. Gershwin, "Integrated quality and quantity modeling of a production line", OR Spectrum, Vol 27, No. 2-3, pp. 287–314, June 2005.

[21] Meinberg, "LANTIME M900/PTP", 2018. [online]. Available: https://www.meinbergglobal.com/english/products/modular-3u-ieee-1588-grandmaster-clock.htm. [Accessed January 6, 2019].

[22] Beckhoff, "IEEE 1588 external synchrnozation interface (EL6688)", Beckhoff Information System, 2018. [online]. Available: https://www.beckhoff.com/english.asp?ethercat/el6688.htm. [Accessed April 25, 2019].

[23] Red Hat, "Chapter 23. Configuration PTP using PTP4L", in *Deployment, Configuration and Administration of Red Hat Enterprise Linux 6*. [online]. Available: https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/6/html/deployment_guide/ch-configuring_ptp_using_ptp4l

[24] R. Candell, K. A. Stouffer, and D. Anand, "A Cybersecurity Testbed for Industrial Control Systems", in *Proc. PCS 2014*, Houston, TX, Oct. 2014.

[25] R. Candell et al., "Industrial Wireless Systems Radio Propagation Measurements", NIST Technical Note 1951, 2017.

[26] Scapy, "Packet crafting for Python2 and Python3", 2019. [online]. Available: https://scapy.net/ [Accessed on April 25, 2019].

[27] NIST, "Wireless Systems for Industrial Environments", 2019. [online]. Available: https://www.nist.gov/programs-projects/wireless-systems-industrial-environments [Accessed on January 7, 2019].