Available online at www.sciencedirect.com

ScienceDirect

Procedia CIRP 96 (2021) 295–300

www.elsevier.com/locate/procedia

CIRPe 2020 – 8th CIRP Global Web Conference – Flexible Mass Customisation

Implementation of a Vision-Based Worker Assistance System in Assembly: a Case Study

Carlos A. Paz Rocha[a,b], Erwin Rauch[a]*, Taavi Vaimel[b], Manuel A. Ruiz Garcia[a], Renato Vidoni[a]

[a] Free University of Bozen-Bolzano, Industrial Engineerin and Automation (IEA), Universitätsplatz 5, 39100 Bolzano, Italy
[b] Intercable srl, Rienzfeldstraße, 21, 39031 Bruneck, Italy

* Corresponding author. Tel.: +39 0471 017111. *E-mail address:* erwin.rauch@unibz.it

Abstract

The current introduction of Industry 4.0 is very challenging for industrial companies. On the one hand, there is an urge to implement concepts such as digital worker assistance systems or cyber-physical production systems, but besides theoretical work, there is very little research that shows examples of its practical implementation. Furthermore, there is currently a lack of a clear model of how sensor-based worker assistance systems for data acquisition and analytics can be designed and systematically implemented. In the present research, a model for a vision-based worker assistance system for assembly was developed based on an industrial case study regarding a manual assembly line. The proposed model consists of five integrated modules: data acquisition, data preprocessing, data storage, data analysis, and simulation. The data acquisition module was constructed in the assembly workstation of the production line by implementing a depth camera, which together with an algorithm developed in Python for preprocessing, tracks the activities of the operator and inserts the processing times into a SQL table of the data storage module. This module contains all the relevant information of the production system, from the shop floor to the Manufacturing Execution System, enabling vertical integration. The data analysis module, aimed at the streaming and predictive analytics, was deployed in the RStudio platform. Likewise, the simulation module was conceptualized to retrieve real-time data from the shop floor and to select the best strategy. To evaluate the model testing of the proposed system in real production was performed. The results of this use case provide useful information for academia as well as practitioners how to implement vision-based worker assistance systems.

Keywords: industry 4.0; data analytics; cyber-physical production system; computer vision; smart manufacturing; assembly

1. Introduction

The introduction of Industry 4.0 is firmly connected with the term cyber-physical systems. These are intended to connect the physical real world with the digital and virtual world and thus offer advantages in terms of networking and virtual commissioning and operation of production systems. Although the introduction of such cyber-physical systems promises great advantages, it is currently still very difficult for companies to change traditional production systems in this direction. Often there is simply a lack of models and application examples. This research aims to show exactly this focusing on data-driven worker assistance systems. A model is presented, which should support the introduction and implementation of vision-based worker assistance systems for industrial assembly. This work is based on a real case study of an Italian automotive supplier that has implemented and validated the model.

After the introduction, Section 2 examines the theoretical background in the form of a state of the art report. Section 3 then explains the model and describes the individual modules of the model. Finally, in Section 4 the application and testing of the model is presented using a real industrial case study. In Section 5 a critical discussion of the results follows and finally Section 6 summarizes the work again.

2212-8271 © 2021 The Authors. Published by Elsevier B.V.

10.1016/j.procir.2021.01.090

2. Theoretical Background

2.1. Smart Manufacturing

Manufacturing, boosted by recent technologies, is determined to evolve into new forms of production. One of these forms is smart manufacturing [1], which embraces the interaction and communication of heterogeneous components and services that are networked in the factory workspace [2], integrating physical elements of production with intelligent sensors, computing platforms, diverse communication protocols and technologies, simulation, automatized and autonomous control, data modelling, and data analytics [2, 3].

Smart Manufacturing is aimed at converting data acquired through the product and process lifecycle into relevant information that enables flexible, cost-efficient, and customized mass production [2, 3].

2.2. Cyber-Physical Systems

One of the modes to achieve smart manufacturing is through Cyber-Physical Systems (CPS). The CPS is a set of embedded physical and digital subsystems, sensors, actuators and mechanisms that allow the bidirectional communication in the network [4, 5].

The key of CPS is the seamless integration and interaction of the computational capabilities with its physical assets [3]. These physical resources that conform the physical space must be provided with the capability of computing, communicating, and controlling [6]; for its part, these computational capabilities in the cyber space are used to monitor, control, and coordinate the operations of systems in the physical space [7].

CPS involves smart devices, storage systems, and manufacturing facilities, implying a system that is able to collect data from the physical space through a wide variety of sensors, perform streaming analytics and simulation in the cyber space, and send the results back to the physical space [7,8].

In recent years, the term of Cyber-Physical Production System (CPPS) was introduced and spread as a CPS that essentially supports the production management field [9], CPPS go through all levels of manufacturing, from the smallest activities performed by machines and workers up to planning, logistics, and production networks. The importance of CPPS lays on their capacity of responding to unexpected conditions or states that have not been seen before, improving flexibility and the process of decision-making in real time [10, 11, 12].

Among the main benefits that can be expected from the generalization of CPPS, Rudtsch et al. [13] mention: optimization of production processes, optimized product customization, resource-efficient production, and human centered production processes.

2.3. Digital Worker Assistance Systems

Digital worker assistance systems increase the capabilities of operators in smart factories [14]. Mainly such worker assistance systems can be categorized into physical, sensorial and cognitive assistance systems [15]. Physical aid systems (e.g. exoskeletons) reduce the biomechanical workload of an operator. Sensorial aid systems are defined by the ability and capability of the worker to acquire information from the environment. This helps to create knowledge necessary for decision-making and for the orientation of the operator. Cognitive assistance systems can be defined by the ability and capacity of the worker to undertake mental tasks that are needed to properly perform the work task. In this work we put the focus on the latter two assistance systems for worker. Thanks to real-time sensed data from the physical space, such digital worker assistance systems process and analyze retrieved data and provide information and decisions to the worker in order to increase not only productivity but also responsiveness of the assembly system [16].

Such digital worker assistance systems require two streams of data, one from all the physical elements present in the shop floor environment that are sensed, and whose data are transmitted in real time towards the cyber/virtual space. The other stream represents the feedback that is sent to the shop floor [10, 17]. The feedback is the result of a series of algorithms that are executed to control the system and generate planning decisions. This information comprises actions to be performed by the physical world and includes other strategic and managerial decisions [18].

3. Model for Vision Based Worker Assistance Systems

3.1. General Model

The model was developed on the basis of an industrial problem and finally abstracted. The exemplary company, Intercable srl is an innovative, international and diversified company group, whose success formula was the identification and development of market niches with high quality level and innovation potential in the technology domain.

Some of the articles that Intercable produces are technical plastic parts, such as connectors, cable ducts, components for cable protection and connection systems, such as high-voltage components for the e-mobility, power distribution bars, connecting parts Cu/Al, battery clamps, jumpstart components and power distribution boxes.

The case study will be focused on the production of high-voltage power distribution bars, specifically on the assembly process, which consists in placing insulation caps, screws and other materials in the busbar.

Fig. 1. High-voltage power distribution bar (Source: Intercable srl).

The generalized model of a vision-based worker assistance system is composed of a set of different modules that are integrated and synchronized in real-time, and that enable the CPPS concept towards smart manufacturing. Figure 2 depicts the connection between the modules of the model and describes the bidirectional communication between the physical world, represented by the assembly line and sensing devices as well as the cyber world, which comprises different software, data models, computational resources, and historical/real-time data.

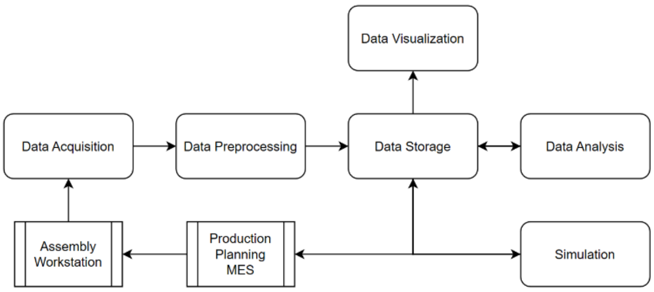

Fig. 2. Model for vision based worker assistance system in CPPS.

3.2. Module 1: Data Acquisition

The data acquisition module has as start point the assembly workstation, the physical space where the data of the operator's activities are captured by a sensing device. The Orbbec 3D Astra S camera was chosen for this task, and whose triangulation method is structured light [19]. The infrared projector provides a structured pattern that is previously known by the camera, what enables it to perform triangulation based on the unique correspondences of the codified rays of the pattern over the object [20]. This optical sensor should be placed on the top of the workstation to focus the depth-camera on the table of the workstation; and thus, sense the activities performed by the operator. The acquired data need to be sent to a computer for their preprocessing, for which, Orbbec Astra SDK and OpenNI2 has been installed in the computer. This SDK enables the communication between the sensor and the computer, whereas OpenNI2 lets these data be gathered by Python, software that will be used for their preprocessing.

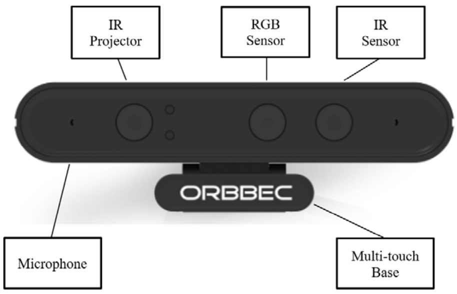

Fig. 3. Orbbec 3D Astra S.

3.3. Module 2: Data Preprocessing

The second module is aimed at the preprocessing and conversion of raw data from the workstation into meaningful data that will be used in the next modules (data storage, data analytics, and simulation). The raw data are read by Python by means of the Primesense and Openni2 imported libraries. These raw data consist of a set of points of the 3D operator's workspace in real time. To convert the raw data, an algorithm needs to be developed, which will have to detect a series of possible hand movements of the operator and interpret them as specific assembly activities (see Figure 4 and 5). Many algebraic operations and the use of matrices are expected; for which, it is indispensable to import the NumPy library. Likewise, it is necessary to import the Datetime library, to identify when the activities are being executed. This first part of the algorithm enables the system to convert points in the space into data of the executed activities, the time when they were performed, and their duration. The structuration of these meaningful data is the next step; the data should be written in tables of a SQL database. For this purpose, the pyodbc module must be imported; this library allows Python to write in SQL format. To enable the communication between Python and MS SQL, the ODBC Python-SQL must be installed in the system.

3.4. Module 3: Data Storage

The model raises the SQL Database as the center and core of the whole system. The data storage module is the only software that have plenty communication with the other software in the model. It behaves as server and client and integrates all the information of the system and basically fulfills the function of a Manufacturing Service Bus (MSB) as integration platform.

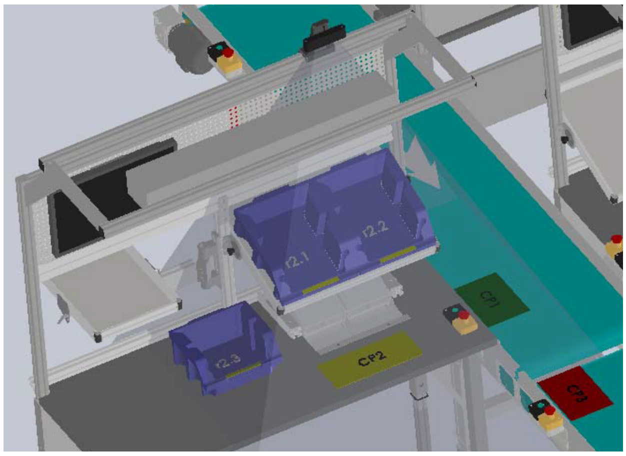

Fig. 4. Assembly table and control points.

Fig. 5. Detection of hand movements over control points.

The SQL server receives the structured data of the operator activities from Python through the ODBC driver. The data are stored row by row in real-time and are simultaneously retrieved by three different software: (1) Visual Shop Floor (data collection and visualization), (2) RStudio (data analytics) and (3) FlexSim (simulation). The first one is simply a client interface that takes the data from the database and shows them to the users in a visual manner. Here the main Key Performance Indicators (KPI) and component's status are depicted on a digital environment that could be accessed from any computer connected to the server. RStudio for its part, also through an ODBC driver, retrieves data from the database, and after carrying out data analytics, writes information into a new table in the SQL server that includes diagnosis, descriptions and predictions of the system. Likewise, FlexSim, again with another ODBC driver, takes information from the MS SQL database and processes it to make it suitable for simulation. After a simulation model was built, the software is able to evaluate different scenarios in real-time and provide the optimal solution under considerations previously given. The simulation results and prescriptions are sent back to the MS SQL and written in an additional table. In this way, the SQL server contains and integrates meaningful data from shop floor, information from analytics, simulation results, and even the production plan, becoming a powerful element for supporting decision-making process.

3.5. Module 4: Data Analysis

The module of data analysis is aimed at yielding streaming and predictive analytics, as well as data mining with historical data. As in the other cases, it is indispensable to install the specific ODBC to enable communication between MS SQL and RStudio; in the same way, it is needed to import the RODBC library in RStudio for coding SQL queries in R language.

RStudio extracts the table from database with the shop floor data and works with it in its own programming language. On a hand, streaming analytics is performed, taking and analyzing data in real-time; the results can be shown by means of plots on the RStudio interface, or by other interfaces like Visual Shop Floor. Histograms, control charts, pie charts and others can be built in this platform and displayed to the user for a better understanding of the production processes status.

RStudio, through the Pushover application and PushoverR library can also send alerts in real time to the user's smartphone or computer under certain conditions previously specified in the R script. Additionally, some valuable indicators, such as performance of the workstation, coefficient of variation, others can be calculated by streaming analytics and written in real time in a new SQL table; thus, the evolution of these indicators along time can be appreciated.

On the other hand, analytics based on the historical data can be executed. This process lets the user comprehend better the operation of the system and receive predictions and prescriptions that support decision-making processes.

3.6. Module 5: Simulation

The simulation module retrieves data from both the Manufacturing Execution System (MES) that includes the initial production plan, and the MS SQL database, that comprises data about the current status of the physical entities in the shop floor. The simulation software receives all the decision variables of the initial production plan and performs simulations based on a previously defined data model that includes all the rules, behaviors, constraints, and production factors of the assembly system.

Once the simulation of the initial production plan has been carried out, simulations results are sent back to both MES system and MS SQL database. As soon as the production line is running, real-time data from the shop floor is retrieved by the simulation software by means of the SQL Server. Then, real-time simulation is automatically performed. Different scenarios are run based on the target objective of the company, the decision variables, and the constraints of the system. The results are inserted into a SQL table in form of prescriptions, in other words, what input variables to modify and how to achieve the optimal solution.

4. Implementation and Testing

4.1. Description of the Setup

The first step in the implementation of the module 1 was to condition the workspace of the operator, especially the table where the worker performs the assembly tasks on. This conditioning responds to the need of standardizing the assembly activities and spaces in order to make the workstation suitable for the processing of different customable types of pieces without changing the configuration of the system. Moreover, the standardization of the workspace and assembly sequence would enhance the reliability of the data acquisition process by the minimization of possible interferences of objects over not corresponding regions that could cause sensing errors (see Figure 6).

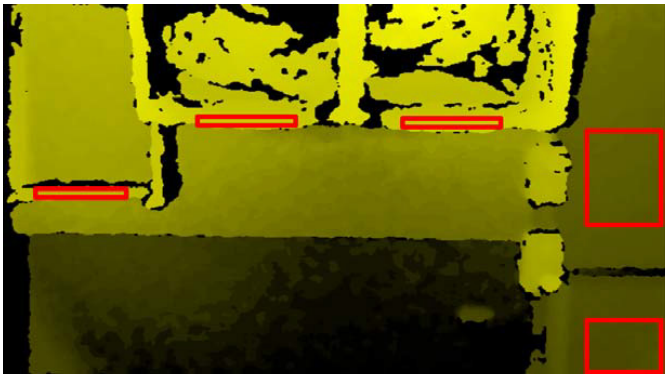

Fig. 6. Control point regions.

Module 2, basically consists in the development of the algorithm that allows to read raw data and convert them into meaningful data. Control points are the 3D regions on the workspace (Figure 6), where if the operator places the hand, a variable in the Python script is activated or deactivated, letting the workflow sequence be represented, These control points are placed in picking and releasing areas, as well as over the boxes that contain the different materials to be used during the assembly process. As mentioned, the match between the hands of the operator and the control points is identified and converted into Boolean values. The control point regions can be activated only if the previous activity was carried out and deactivated once the successor was started. In the first module, the OpenNI2 software was installed. To complete the communication between the sensor and Python, the OpenNI2 and PrimeSense libraries were imported in Python. NumPy and Datetime libraries were imported as well.

Every single control point is a matrix of x,y values with a certain depth. These values can be inserted into Python as shown in Figure 7. Sensitivity of the data acquisition system will depend on the ranges of x,y and depth given by the user. To minimize possible errors, the depth range detection is subjected to the average depth of the x,y matrix.

```
# Region 2 (pixels)
r2_x_min = 200; r2_x_max = 400; r2_y_min = 160; r2_y_max = 170

# Region 2 (depth)
d2_min = 9000; d2_max = 12000

# Sub regions (pixels)
r21_x_min = 390; r21_x_max = 475; r21_y_min = 270; r21_y_max = 275
r22_x_min = 215; r22_x_max = 290; r22_y_min = 270; r22_y_max = 275
r23_x_min = 140; r23_x_max = 330; r23_y_min = 450; r23_y_max = 460

# Sub regions (depth)
d21_min = 8800; d21_max = 10085
d22_min = 8800; d22_max = 10010
d23_min = 9500; d23_max = 11300
```

Fig. 7. Definition of regions and sub regions in the workspace.

Once one cycle is over, all the values (e.g. Activity Time, Process Time, Waiting Time, and Cycle Time) recorded previously in Python, are inserted into a specified table in MS SQL (see code in Figure 8). For this purpose, it is required to enable the communication between Python and MS SQL through the installation of the ODBC Python-SQL driver.

The MS SQL server is synchronized with Python, RStudio, FlexSim, Visual Shop Floor and MES by means of ODBC drivers. As described in the conceptual part, four different tables were created into the database to store the data of the whole system.

For its part, RStudio, as well as the latest version of R and the RODBC driver for MS SQL were installed. RODBC and PushoverR libraries were imported in RStudio.

Multiple Linear Regression model was deployed on R to perform predictive analytics in real time. Likewise, alert and status notifications were configured to be delivered to the user's computer and smartphone through Pushover.

Multiple Linear Regression is just one of many different tools and models that could be employed in streaming analytics to perform real-time predictions of the production line outcome. While in the case study just one process of the production line was virtually represented, once all the processes have a digital counterpart and are integrated with each other in a lean system, predictive models such as Multiple Linear Regression can perform a bigger role in the estimation of the final outcome of the line, what enhances the responsiveness of the production system, since it will retrieve the coefficient of variations, queues information and performances of the different processes, as well as it will identify in real time the input variables with the biggest impact.

Likewise, the visual analytics, through RStudio and Pushover let the user receive information from the production line in a clear and timely manner, what leads to the improvement of the decision-making process, for example, through corrective actions over critical input variables.

```
conn = pyodbc.connect('Driver={SQL Server};'
                'Server=DESKTOP-ND0K9I2\TNEWSQLSERVER;'
                'Database=Productivity5;'
                'Trusted_Connection=yes;')

cursor = conn.cursor()

cursor.execute(
        "INSERT INTO dbo.CycleTimeSLV (Date,Time,WorkerCode,"
        "InputBatchCode,ActivityTime1,ActivityTime2_0,"
        "ActivityTime2_1,ActivityTime2_2,ActivityTime2_3,ActivityTime2,"
        "ActivityTime3,ProcessTime,WaitingTime,CycleTime,Comment) "
        "VALUES (?,?,?,?,?,?,?,?,?,?,?,?,?,?,?)",
        (str_date, str_time, workercode, inputbatch, AT1, AT20,
        AT21, AT22, AT23, AT2, AT3, PT, WT, CT, ' '))
conn.commit()
```

Fig. 8. Python – SQL connection.

4.2. Results

The integration and operation of the data acquisition, data preprocessing, data storage, and data analysis modules were tested and analyzed in order to determine the reliability of the system in the manufacturing environment. Through the analysis of the collected data and the information generated during the testing activities, it was possible to define that the model works as expected.

Tests were carried out in the station where the sensor was implemented. Four well-trained operators were chosen to process five different articles and were given standard rules of operation: Other people cannot stand nearby the region covered by the sensor, an only standardized sequence had to be followed for all the different articles. Moreover, it was required to have a clean and tidy workspace.

Regarding the counting of good and bad parts, the measured number of pieces at the end of the tests coincided with the number of processed pieces in the assembly workstation. Moreover, a manual study was carried out to perform a comparison with the results obtained by the proposed system (activity times, process time, waiting time and cycle time). These results were similar to those attained by the manual time study and can be used for further improvements through method study.

The five articles that were studied had each one a specific distribution that better fits to its dataset, no matter which operator processed them. This means each article can be characterized by a specific model constructed in FlexSim with a very high relative score. These models were evaluated by the goodness-of-fit tests and obtained positive results, all the distribution models achieved from FlexSim should not be rejected and are able to be applied in simulation in real time, letting the user experiment with different scenarios, predict production outcome and receive prescriptions regarding the decision variables.

Python scripts without interruptions or errors during the testing process. The MS SQL database stored all the data that were retrieved from Python. RStudio was capable to extract the data from the real-time data streaming and insert new data from analytics processes into the historical database in MS SQL without errors.

Furthermore, the user received satisfactorily notifications to smartphone and computer through Pushover when specified conditions were met.

5. Discussion

Digital and vision based worker assistance systems are still in early stages. The literature related to CPPS usually provides general models or frameworks, but just a few practical case studies are presented. In addition, computer vision and data analytics will be enabling technologies for such digital worker assistance systems. Benefits and limitations of this recent technologies are not well estimated yet.

Regarding the implementation of such digital and vision based worker assistance systems, a proper system modelling with well-defined steps, rules, behaviors, and constraints is mandatory for a faithful interpretation of data from the physical shop floor, what could be very challenging. Furthermore, a module of simulation has been discussed in our work. It is important to highlight, that even when a simulation model is supposed to be as realistic as possible, it should be kept lightweight since the purpose of the proposed module is to perform simulation with real-time data, what could require huge computing resources.

Moreover, to fully exploit all the capabilities and functionalities of the proposed model, the lean manufacturing concept is required to be implemented in the production line.

From a company point of view the planning, design and implementation of the case study took several months in form of a student project. Based on the scientific supervision of the university tutors and the practical supervision of the company tutor the student was able to perform all required tasks. As lessons learned the most challenging part was to define how data can be retrieved and processed. In this regard the model developed in Fig. 2 helped a lot during the implementation of the case study itself.

6. Conclusions and Outlook

While the present study carries out the implementation of a vision sensor provided with smart capabilities in one workstation only, the analytics module is able and intended to retrieve and integrate the cycle times of all the sensors that could compose the production line, what would grant a better understanding of the whole manufacturing process. Therefore, future works should emphasize to test the model in a larger production line combining and comparing different computer vision systems.

Likewise, the integration of simulation and MES would enable the system to perform prescriptive analytics for the manufacturing process, providing optimal input variables in real time that can be deployed by decision takers. In this case, further research should focus on the application of the developed model on a whole production line or factory in order to test real-time data-based simulation also in more complex situations.

References

[1] A. Kusiak, „Smart manufacturing," International Journal of Production Research, pp. 508-517, 2018.

[2] A. Zeid, S. Sundaram, M. Moghaddam, S. Kamarthi und T. Marion, „Interoperability in Smart Manufacturing: Research Challenges," Machines, 2019.

[3] H. S. Kang, J. Y. Lee, S. Choi, H. Kim, J. H. Park, J. Y. Son, B. H. Kim und S. D. Noh, „Smart manufacturing: Past research, present findings, and future directions," International Journal of Precision Engineering and Manufacturing-Green Technology, pp. 111-128, 2016.

[4] E. Lee, „Cyber Physical Systems: Design Challenges," International Symposium on Object/Component/Service-Oriented Real-Time Distributed Computing (ISORC), pp. 364-369, 2008.

[5] N. Nikolakis, K. Alexopoulos, E. Xanthakis und G. Chryssolouris, „The digital twin implementation for linking the virtual representation of human-based production tasks to their physical counterpart in the factory-floor," International Journal of Computer Integrated, 2019.

[6] F. Tao und M. Zhang, „Digital Twin Shop-Floor: A New Shop-Floor Paradigm Towards Smart Manufacturing," IEEE Access, pp. 20418-20427, 2017.

[7] Q. Qi, D. Zhao, W. Llao und F. Tao, „Modeling of Cyber-Physical Systems and Digital Twin Based on Edge Computing, Fog Computing and Cloud Computing Towards Smart Manufacturing," Proceedings of the ASME 2018 13th International Manufacturing Science and Engineering Conference MSEC2018, 2018.

[8] Y. Tan, W. Yang, K. Yoshida und S. Takakuwa, „Application of IoT-Aided Simulation to Manufacturing Systems in Cyber-Physical System," Machines, 2019.

[9] L. Monostori, „Cyber-physical Production Systems: Roots, Expectations and R&D Challenges," Procedia CIRP 17, pp. 9-13, 2014.

[10] G. Shao, S. Shin und S. Jain, „Data analytics using simulation for smart manufacturing.," Proceeding of the 2014 Winter Simulation Conference, pp. 192-203, February 2014.

[11] O. Cardin, „Classification of cyber-physical production systems applications: proposition of an analysis framework," 2018.

[12] Dallasega, P., Rauch, E., & Frosolini, M. (2018). A lean approach for real-time planning and monitoring in engineer-to-order construction projects. Buildings, 8(3), 38.

[13] V. Rudtsch, J. Gausemeier, J. Gesing, T. Mittag und S. Peter, „Pattern-based Business Model Development for Cyber-Physical Production Systems," Procedia CIRP. 25, pp. 313-319, 2014.

[14] Mark, B. G., Hofmayer, S., Rauch, E., & Matt, D. T. (2019). Inclusion of Workers with Disabilities in Production 4.0: Legal Foundations in Europe and Potentials Through Worker Assistance Systems. Sustainability, 11(21), 5978.

[15] Romero, D., Stahre, J., Wuest, T., Noran, O., Bernus, P., Fast-Berglund, Å., & Gorecky, D. (2016, October). Towards an operator 4.0 typology: a human-centric perspective on the fourth industrial revolution technologies. In proceedings of the international conference on computers and industrial engineering (CIE46), Tianjin, China (pp. 29-31)..

[16] E. Negri, L. Fumagalli, C. Cimino und M. Macchi, „FMU-supported simulation for CPSDigital Twin," Procedia Manufacturing 28, pp. 201-206, 2019.

[17] B. Schleich, N. Anwer, L. Mathieu und S. Wartzack, „Shaping the digital twin for design and production engineering," CIRP Annals - Manufacturing Technology, pp. 141-144, 2017.

[18] G. E. Modoni, E. G. Caldarola, M. Sacco und W. Terkaj, „Synchronizing physical and digital factory: benefits and technical challenges," 12th CIRP Conference on Intelligent Computation in Manufacturing Engineering, Gult of Naples, Italy, pp. 472-477, 2018.

[19] L. Granero, F. Díaz, R. Domínguez, J. Hervás, S. Simón und E. Pérez, „Application of optical techniques in documentation and identification of archaeological rests: The case study of the Roman bronze rest found in Lucentum," in Proceedings of SPIE - The International Society for Optical Engineering, Paterna, 2009.

[20] S. Giancola, M. Valenti und R. Sala, A Survey on 3D Cameras: Metrological Comparison of Time-of-Flight, Structured-Light and Active Stereoscopy Technologies, Milan: Springer International, 2018.