An OPC UA Based Architecture for Testing Tracking Simulation Methods

Gerardo Santillán Martínez[1], Tommi Karhela[1, 2], Valeriy Vyatkin[1, 3], Tuomas Miettinen[2], Cheng Pang[1]
[1]Aalto University, Helsinki, Finland
[2]VTT Technical Research Centre of Finland Ltd, Espoo, Finland
[3]Luleå University of Technology, Luleå, Sweden
gerardo.santillan@aalto.fi, tommi.karhela@vtt.fi, vyatkin@ieee.org, tuomas.miettinen@vtt.fi, cheng.pang.phd@ieee.org

***Abstract*— A tracking simulator is a simulation system that runs in parallel with the real process. They integrate model and process using a persistent and direct communication between the physical system and the simulation. In a tracking simulator, the simulation system receives data measured from the process instrumentation in order to adjust the model parameters so that the simulated state matches the real process state. Tracking simulation systems have a number of applications and the benefits in comparison to conventional simulation systems are numerous. This paper presents an architecture for testing different methods that can be used in tracking simulation. The proposed approach employs the OPC Unified Architecture as the protocol for the communication between the simulation model and the physical system. This work first introduces the structure of the testbed. Then, in order to test the proposed system, a previously published tracking method based on the tuning of the model parameters using PI controllers is implemented and the test results are shown. Finally, the conclusions and future work are discussed.**

Keywords*— *tracking simulation; online simulation; OPC Unified Architecture; industrial internet; dynamic process simulation; automation systems;

I. Introduction

Simulation models are often used at the design stages of production systems. Using them also during later phases of production systems' lifecycle could be beneficial for many reasons but is hindered by their high maintenance costs and the time-consuming work needed for the integration between simulation and plant [1, 2]. Online simulation systems integrate model and physical system using a persistent and direct communication between the simulation and the real process [3]. In order to estimate the state variables needed, for example, to initialize the model, the simulation is run in parallel with the process, connecting the control system into the model [4]. Fig. 1 (a) shows the structure of an online simulation system. In an online simulator, there are no feedback connections from the process to the model, causing the outputs of the simulation to eventually diverge from the ones of the process due to non-ideality of the model. In order to address these problems, a tracking simulator is needed.

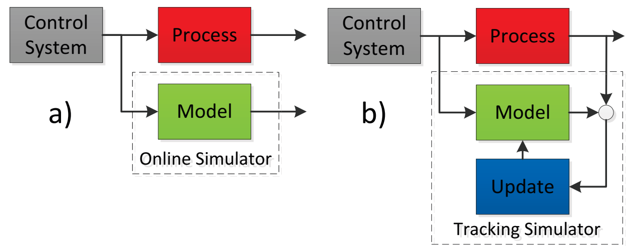

Fig. 1. Online (a) and tracking (b) simulation systems.

A tracking simulator, showed in Fig. 1 (b), is an online simulator that runs in parallel with the real process. By comparing the simulation and process outputs, the tracking simulator updates its parameters to converge the simulated state to the real one. Tracking simulation systems are similar to online simulation systems but the results of a tracking simulator are more accurate thanks to the parameters' adjustment performed by the update mechanism. Tracking simulation systems have a number of applications and the benefits in comparison to conventional simulation systems are numerous [4-6].

This paper presents an architecture for testing different methods for tracking simulation that uses OPC Unified Architecture (OPC UA) as the protocol for the communication between its components. The main elements of this architecture are the physical system and the simulation environment, however, a historical data repository is added to be used during different phases of the tracking process. An important characteristic of the suggested design is the inclusion of a model of the control system in the same simulation tool where the process is modelled. The model of the control system can be used, for example, to speed up the initialization of the simulation model or to control different instances of the simulation independently. The research presented in this work, addresses the need for simulation systems architectures that can be easily integrated with processes at the production phase of the plant lifecycle.

The paper is structured as follows. Section II describes the proposed architecture and how different components are used during the tracking simulation. In section III, an implementation of the proposed structure that uses a simple water heating process as the physical system is described. Then, in Section IV, the tracking method used to evaluate the developed testbed is explained and its results are showed. Finally, the conclusions and future work are presented in Section V.

978-1-4673-7952-6/15 $31.00 © 2015 IEEE
DOI 10.1109/Trustcom.2015.644

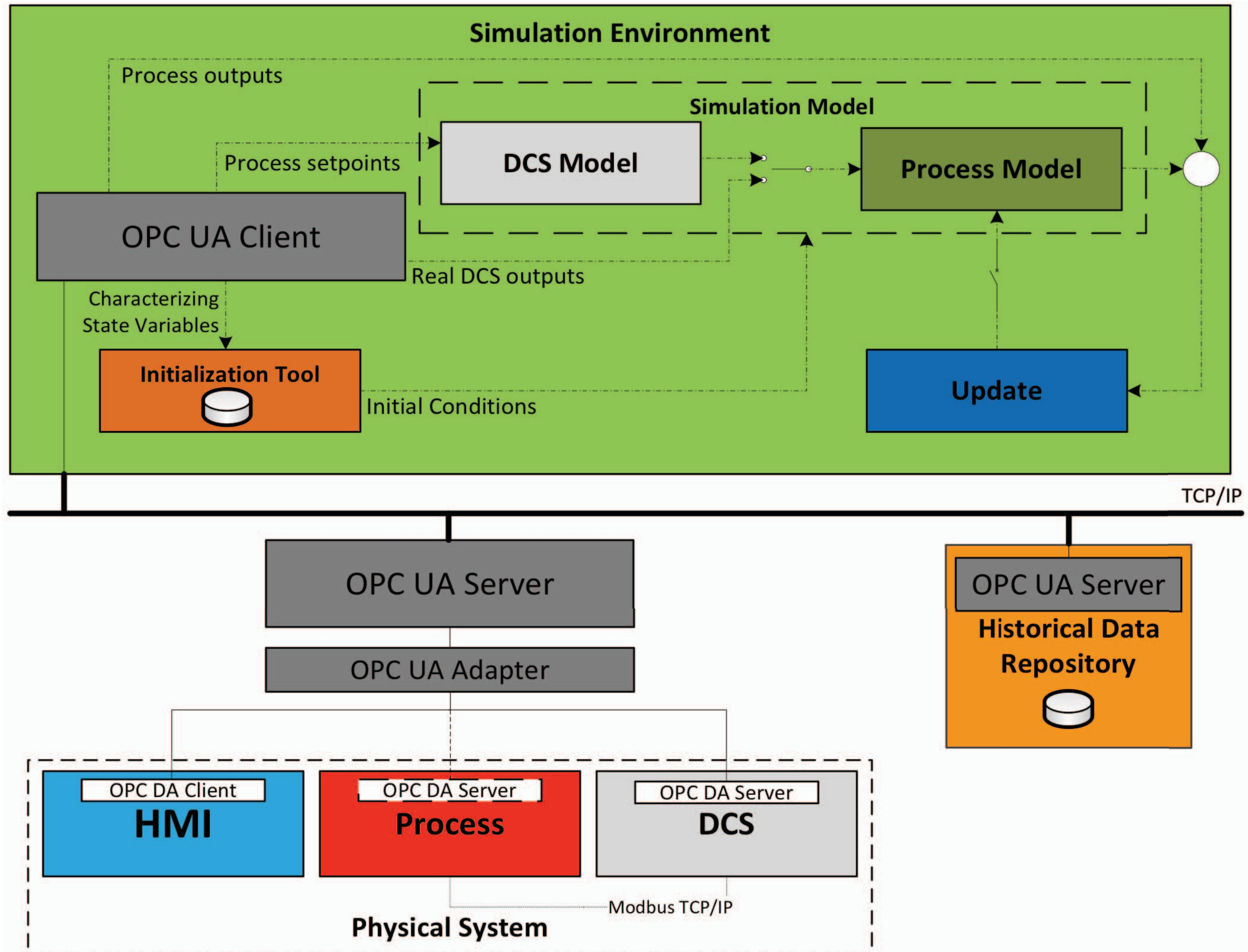

Fig. 2. Architecture of the testbed for tracking simulation systems.

II. Proposed Architecture

The architecture of the testbed for tacking simulation methods proposed in this work is presented in Fig. 2. The main components of tracking simulation systems are the physical system and the simulation model. In the suggested design, a historical data repository is added. This repository is used to save historical data coming from the process, the control system and the model.

OPC UA is used for the communication between all the components of the tracking simulation system. OPC UA is a platform independent service-oriented architecture that integrates all the functionality of the individual OPC Classic specifications into one extensible framework [7]. It allows software to connect devices, machines and systems from different manufacturers using a common interface, reducing integration time and development costs. OPC UA guarantees interoperability between multi-vendor systems and devices such as sensors, control systems, PLCs and HMIs [8]. It is possible to use other field communication standards for the communication in the control and instrumentation layers of the physical system. OPC UA adapters are required to interface classic OPC servers and clients with others that use the Unified Architecture version of the protocol.

A. Simulation Environment

While the physical system comprises the process, the distributed control system (DCS) and a HMI used to monitor the plant; the simulation environment includes the simulation model of the process, a model of the control system (DCS model) and the update mechanism to be tested for the tracking simulation. Because it is not possible to access all the information from the real plant, the state of the process cannot be completely copied into the model. For this reason, the simulation environment also includes a component defined as the Initialization Tool required to provide the model with initial conditions (IC) that are close

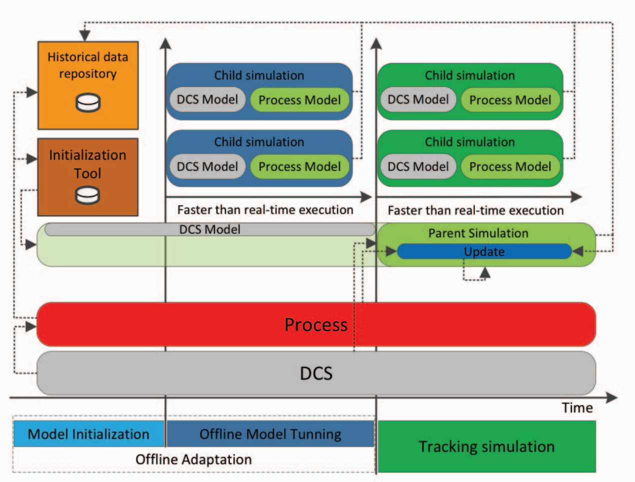

Fig. 3. Tracking simulation phases [9].

to the state of the physical system in order to start the simulation [9].

The DCS model, developed in the same simulation environment using the control components of the simulation tool and modelled taking the real DCS as a reference, is used during most of the tracking simulation phases to control the process model. Fig. 3 shows the tracking simulation phases: in the Model Initialization an Initialization Tool provides the model with initial conditions that are close to the state of the process. Later, in the Offline Model Tuning, multiple child simulations are created to adjust the parameters of the model using offline tuning techniques and historical data from the process. During the tracking simulation phase the adjustment of the parameters of the model is done using online tracking methods by the update mechanism. At this point, the tracking simulation model becomes the parent simulation from which child simulations can be instantiated. It is necessary to be able to select which control system (the real or the model) is used to run the simulation according to the application. Further details of the three main ways in which the DCS model is used in the tracking simulation phases are described as follows:

- **Model Initialization**: This is part of a phase defined as the Offline Adaptation where, before the tracking simulation can be started, the simulation model goes through an adaptation process in order to reach the same state as the real plant. During the first stage of the initialization sequence, the Initialization Tool loads IC that are close to the state of the real process into the simulation model. Then, the model of the DCS drives the simulation to the same state as the real system using the process setpoints. Finally, the model of the DCS is used to correct any difference in mass or energy between the simulation model and the real process. A more detailed description of the initialization process is available in [9].
- **Offline Model Tuning**: In the Offline Model Tuning, various instances of the simulation called child simulations are created to evaluate the parameters adjustment done by offline tuning techniques, such as dynamic data reconciliation. These offline methods use the historical data from the process stored in the repository to adjust different parameters of the simulation model. This is done so that the behavior of the model becomes closer to the one of the real process. In this stage, the DCS model is used to control child simulations. Each child simulation is run faster than real time controlled by its own DCS model. Because each child simulation is run faster than real-time, having the model of the control system as part of the same simulation environment is convenient to avoid any problem related to the synchronization of the communication between the DCS and the model during these runs. The outputs of the child simulations during the offline tuning are stored in the historical data repository for further analysis.
- **Tracking simulation**: In the tracking simulation phase, the tracking is performed using online tracking methods. In this phase, the simulation model running in parallel with the real plant becomes the parent simulation from which child simulations can be instantiated and then run offline. A single child simulation can be created for predictive simulation, model predictive control, etc.; or multiple child simulations can be run to compare the plant response to different transients or to tune multiple controller parameters. In the same way as in the Offline Model Tuning, a model of the DCS is used to control each child simulation. The outputs of the parent and child simulations are stored in the historical data repository to be analyzed.

The update mechanism of the tracking simulator is in charge of the parameters adjustment required for the outputs of the simulation to converge with the ones of the real process. This is the component to be tested using the suggested architecture. Different approaches used for tracking simulation can be implemented and easily integrated with the system for testing purposes using the designed testbed. Various methods used by the update mechanism for tracking simulation can be found in [4-6].

III. IMPLEMENTATION OF THE PROPOSED ARCHITECTURE

The implementation of the suggested approach follows the structure showed in Fig. 2. The process used to demonstrate the architecture of the testbed is the water heating system shown in Fig. 4. It is a small scale and simplified water heating plant where the water is first heated, circulated and pressurized to be later consumed. The piping and instrumentation diagram (P&ID) of the water heating system is shown in Fig. 5 [10]. Before the system reaches the production stage, the process goes through three preparation steps:

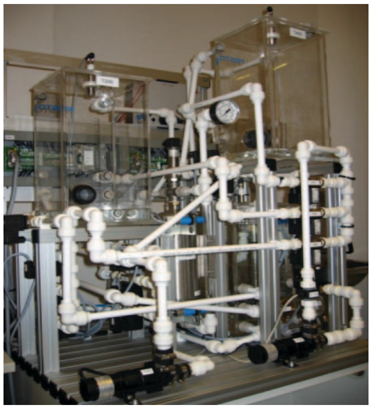

Fig. 4. Water heating process.

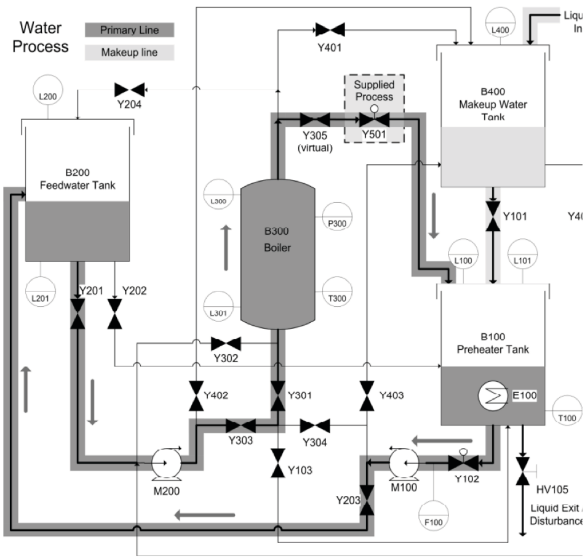

Fig. 5. P&ID diagram of the water heating system used as the physical system to demonstrate the proposed architecture. [10]

1. The water in the Preheater tank is heated to the setpoint temperature.
2. The water level of the Feedwater tank reaches its setpoint.
3. The boiler is filled and pressurized.

During the production phase, the position of the Supplied Process valve (Y501) can be adjusted to simulate different levels of hot water consumption. The consumed hot water flows back to the Preheater tank to be re-heated. The water level of the Feedwater tank is regulated by two PID controllers connected in cascade that control the flow F100 by adjusting the position of the proportional valve Y102. Another PID controller regulates the pressure in the Boiler using the rotation speed of the M200 pump. The water level of the Preheater tank is controlled using the position of the shut off valve Y101.

The DCS of the physical system runs on a soft PLC. It was developed following the IEC 61131-3 standard using the automation software CODESYS [11]. Modbus TCP/IP protocol is used for the communication between the process and the DCS. The CODESYS OPC Server is used for the communication with the HMI and to connect the physical system with the rest of the tracking simulation environment. The HMI is needed to monitor and control the process setpoints. As the physical system uses the classic OPC standard, the Unified Automation's UaGateway [12] interfaces the process, DCS and HMI with the rest of the tracking simulation environment.

The models of the process and of the DCS application were developed using the simulation environment Apros [13]. Apros is a multifunctional software tool for dynamic simulation of processes that also supports the simulation of automation system to control the models. It is possible to easily connect it to other simulation tools or DCS using its built-in OPC UA functionalities. The Initialization Tool, as it was previously explained, is a set of functions used to initialize the model. They are bundled into the simulation environment. In the implementation of the testbed, the Initialization Tool is able to store in a database snapshots of the model during simulation runs that can be used as IC. Then, using Characterizing State Variables published by the OPC UA client of Apros, it searchers, selects (from the previously saved snapshots) and provides the simulation model with IC that are the closest possible to the current state of the plant [9]. The Characterizing State Variables are a set of system-specific values that are available from the instrumentation of the real plant and that define the state of the physical system. The model of the DCS is a close version of the real one, however, it was not developed following the 61131-3 standard as it was modelled using the control components available in the simulation tool Apros.

IV. Experiment Results

A simplified version of the tracking method presented in [4] was implemented to demonstrate the tracking simulation testbed environment. In this approach, the update mechanism is performed by PI controllers that adjust the parameters of the model while the simulator is running in parallel with the process. In the method suggested, the outputs of the process are connected to the setpoints of the PI controller whereas the outputs of the simulation are connected to its measurements inputs. The control value of the controller is the parameter to be adjusted in the simulation for the outputs of the process and model to converge. The selection of this technique as the approach to be tested was motivated by its simplicity by the fact that the simulation environment Apros, where the process model is developed, supports the modeling of the control components required for the implementation of this tracking mechanism. This means that it is possible to have the update mechanism in the same simulation system without any extra communication interface required. In order to test the implementation, the tracking method previously described was simplified: the autotuning method that the original approach suggests for the controller tuning is not being used. Instead, a manual tuning was carried out. It is

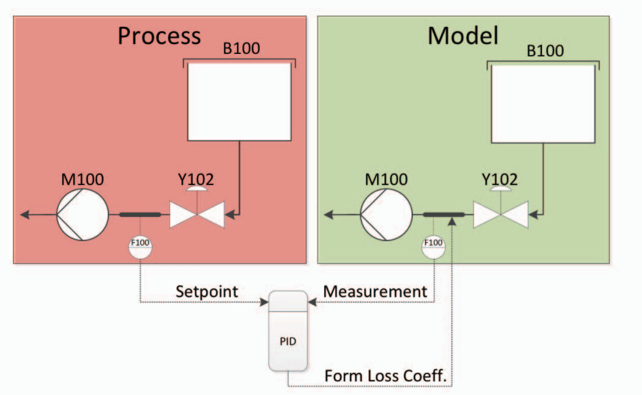

Fig. 10. Form loss coefficient adjustment using a PID controller for the tracking process.

important to point out that the Offline Model Adaptation phase, showed in Fig. 3, is not tackled by this work. After the initialization phase is completed, the tracking simulation is started.

For the tracking simulation experiment, the selected parameter to be adjusted is the water flow F100 measured between the valve Y102 and the pump M100, as shown in Fig. 10. It is the water flow between the Preheater and Feedwater tanks. This parameter was selected because adjusting this flow will also impact on two of the most important variables of the process: the water levels of the earlier mentioned tanks. The flow F100 is tracked adjusting the form loss coefficient of the piping section between the valve and the pump. In order to have a faster parameter adjustment, a derivative part is added to the suggested PI controller. The tracking PID controller uses the flow measurement from the process as the setpoint and the flow calculated by the simulation model as the measurement input. The controller output (control value) is the form loss coefficient of the piping section. A higher loss coefficient will reduce the flow and vice versa, thus, it is a direct-gain system where the tracking controller has a negative gain.

When the process is running in a steady production state and before starting the tracking simulation, the model is initialized using the modelled version of the DCS following the method described in [9]. Once the process and model are in the exact same state, the tracking procedure is started connecting the outputs of the real DCS into the simulation model and starting the PID controller to adjust the flow F100. For comparison purposes, Fig. 6 shows the flow F100 of the model and process running in parallel without the tracking mechanism modifying the form loss coefficient of the piping section. It can be seen that the simulation flow requires an adjustment in order to converge with the one measured in the process. Fig. 7 shows the result of the tracking of the flow F100. Fig. 8 and Fig. 9 show the comparison between the real water levels and the ones calculated by the simulation in the Preheater (L100) and Feedwater tank (L200) during the tracking simulation. The

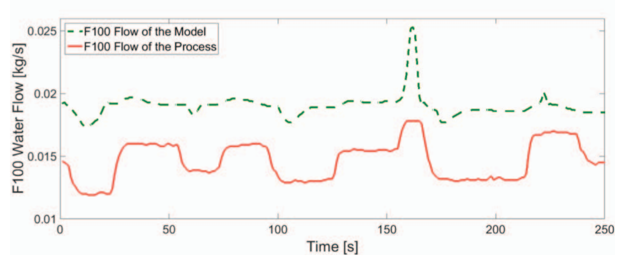

Fig. 6. Flow F100 during an online simulation where the form loss coefficient is not being adjusted. In an online simulation, the model runs in parallel with the process controlled by the real DCS.

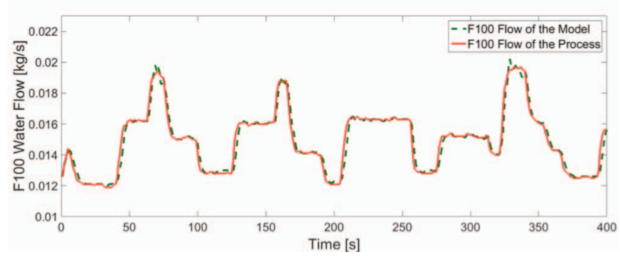

Fig. 7. Flow F100 during the tracking simulation. In a tracking simulation, the model runs in parallel with the process controlled by the real DCS and a PID controller tracks the flow F100 adjusting the loss coefficient of the piping section between the valve Y102 and the pump M100.

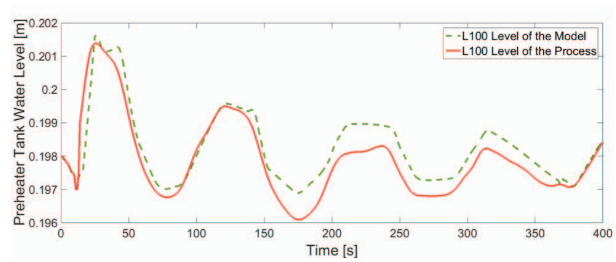

Fig. 8. Preheater tank water level L100 during the tracking simulation. The difference between the setpoint and measured value of the water level L100 is caused by the pre-set hysteresis value of the ultrasonic level sensor.

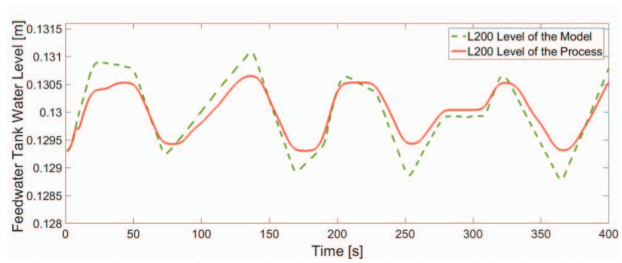

Fig. 9. Feedwater tank water level L200 during the tracking simulation.

water level setpoints in the Preheater and Feedwater tanks are 200 mm and 130 mm respectively.

The results show that, thanks to the tracking mechanism, the flow F100 of the model and the process converge. This causes the levels L100 and L200 calculated by the simulation to be very close to the real ones. The difference in the levels is due to the non-ideality of the model. Both levels could be also tracked using the same method as the one used for the flow tracking. However, the adjustment of the parameters needed to track extra variables could impact or interfere with the ongoing tracking process. For this reason, an offline multi-parameter adjustment method could improve the quality of the results during the tracking simulation.

V. Conclusions and Future Work

This paper presented an architecture for testing tracking simulation methods that uses the OPC UA standard for the communication between system components. The structure of the testbed is elaborated and its implementation is showed. Then, in order to assess the proposed architecture, a previously published tracking approach based on the tuning of model parameters using PI controllers was developed and tested. Finally, the results of the experiments performed were showed.

In a tracking simulation, the precise integration of the components that comprise the system is very important for the quality of its results. With OPC UA as the communication standard, a tight integration between different parts of the tracking simulation system is guaranteed. In addition, OPC UA reduces the development work needed to interface different components of the system and it becomes particularly useful for future development where extra components could be added to the testbed implementation. For the tracking simulation process it is very important to have a model of the control system (DCS) application as part of the simulation tool where the model of the process is developed. A DCS application model can be used for the initialization of the simulation and to control the model of the process during the tracking phases where the simulation is run faster than real-time. For this reason, a process simulation environment that supports for modelling and simulation of control systems functionality becomes the ideal choice to be used for this kind of application.

Future work will be focused on the addition of the historical data repository where data from all the components will be stored for further analysis. This repository will be the first step towards the study of offline multi-parameter tuning methods that can use historical data time series from the process to adjust different parameters of the model at the same time. Moreover, other online tracking techniques will be implemented and analyzed using the developed tracking simulation testbed environment.

Acknowledgments

This work was supported by the Smart Technologies for Lifecycle Performance (S-STEP) program [14], funded by the Finnish Funding Agency for Innovation (TEKES) [15] and coordinated by the Finnish Metals and Engineering Competence Cluster (FIMECC) [16]. Authors would also like to thank the participating companies Outotec Oy and Fortum Oyj; and the research organizations VTT Technical Research Center of Finland LTD and Aalto University, involved in the Project 3 of the S-STEP program.

References

[1] S. Kain, F. Schiller, and S. Dominka, "Reuse of models in the lifecycle of production plants using HiL simulation models for diagnosis," in *IEEE International Symposium on Industrial Electronics, 2008. ISIE 2008.* , 2008, pp. 1802-1807.

[2] S. Leong, Y. T. Lee, and F. Riddick, "A Core Manufacturing Simulation Data Information Model for Manufacturing Applications," presented at the Proceedings of the 2006 Fall Simulation Interoperability Workshop, Orlando, Florida, 2006.

[3] M. N. Yuniarto and A. W. Labib, "Designing an online and real time simulation, control and monitoring of disturbances in an intelligent manufacturing system," in *Industrial Informatics, 2003. INDIN 2003. Proceedings. IEEE International Conference on*, 2003, pp. 273-278.

[4] M. Friman and P. Airikka, "Tracking Simulation Based on PI Controllers and Autotuning," in *IFAC Conference on Advances in PID Control*, Brescia, Italy, 2012.

[5] M. Nakaya and X. Li, "On-line tracking simulator with a hybrid of physical and Just-In-Time models," *Journal of Process Control,* vol. 23, pp. 171-178, 2013.

[6] M. Nakaya, G. Fukano, Y. Onoe, and T. Ohtani, "On-line Simulator for Plant Operation," in *Intelligent Control and Automation, 2006. WCICA 2006. The Sixth World Congress on*, 2006, pp. 7882-7885.

[7] OPC Foundation. (2015). *OPC Unified Architecture*. Available: https://opcfoundation.org/about/opc-technologies/opc-ua/

[8] Prosys OPC. (2015). *OPC UA*. Available: https://www.prosysopc.com/opc/

[9] G. Santillán Martínez, T. Karhela, H. Niemistö, A. Rossi, C. Pang, and V. Vyatkin, "A Hybrid Approach for the Initialization of Tracking Simulation Systems," presented at the Ememrging Technologies and Factory Automation (ETFA), Luxembourg, 2015.

[10] T. Vepsalainen, S. Sierla, J. Peltola, and S. Kuikka, "Assessing the industrial applicability and adoption potential of the AUKOTON model driven control application engineering approach," in *Industrial Informatics (INDIN), 2010 8th IEEE International Conference on*, 2010, pp. 883-889.

[11] 3S-Smart Software Solutions GmbH. (2015). *CODESYS*. Available: http://www.codesys.com/

[12] Unified Automation. (2015). *UaGateway*. Available: https://www.unified-automation.com/products/wrapper-and-proxy/uagateway.html

[13] VTT and Fortum. (2015). *Advanced Process Simulation Software (Apros)*. Available: http://www.apros.fi/en/

[14] FIMECC. (2015). *Smart Technologies for Lifecycle performance (S-STEP) program*. Available: http://www.fimecc.com/content/s-step-smart-technologies-lifecycle-performance-0

[15] TEKES, "Finnish Funding Agency for Innovation," 2015.

[16] FIMECC. (2015). *Finnish Metals and Engineering Competence Cluster*. Available: http://www.fimecc.com/