



Universal identification and control of industrial manufacturing equipment as a service

Verena Tessaro^a, Axel Vick^{b*} and Jörg Krüger^b

^a Chair of Industrial Automation Technology, Technische Universität Berlin, Pascalstr. 8–9, 10587 Berlin, Germany

^b Fraunhofer Institute for Production Systems and Design Technology IPK, Pascalstr. 8–9, 10587 Berlin, Germany

Received 23 September 2021, accepted 4 October 2021, available online 4 November 2021

© 2021 Authors. This is an Open Access article distributed under the terms and conditions of the Creative Commons Attribution-NonCommercial 4.0 International License (<http://creativecommons.org/licenses/by-nc/4.0/>).

Abstract. This paper presents a universal approach of identification and closed-loop control of manufacturing equipment, delivered through web services using Open Platform Communications United Architecture (OPC UA). Rapid prototyping as well as retrofitting and digitization of legacy systems often need design and application of closed-loop controllers. The analysis and modelling for systems such as energy-conversion or material transport devices is labour-intensive and needs process understanding. Current identification and control toolboxes require systematic preparation of input/output data, modification and tuning of the derived models, also proper design of classic PID controllers. An on-demand service paradigm is applied to allow identification and control with direct access to the controlled system over a network connection. The identified parameters are used to adapt a model predictive controller (MPC), which stabilizes the system and drives trajectories to different operating points. To evaluate the performance of the controllers in terms of stability, accuracy, and time response, several target trajectories and disturbances (signal noise, external physical disturbances, latency in communication) were investigated. The identification service was used to model the linear dynamics of a 6-DOF industrial robot and a laboratory-scale waterworks containing two separately controllable pumps. The robot's axes and the waterworks' pumps were successfully controlled with current set-points by using their respective identified state-space models. Simulation and laboratory experiments show promising results for the control of diverse systems with varying time-constants, and imply broad applicability. As a major achievement, this approach enables to efficiently implement system identification and model predictive control in manufacturing.

Key words: system identification, model predictive control, industrial robots, process technology, manufacturing as a service.

1. INTRODUCTION

Usually, the control and regulation technology is located close to the associated industrial plant or industrial robot. Local controllers are limited in their computing and storage capacity, which imposes restrictions on complex, computationally intensive functionalities. Therefore, outsourcing them, e.g. as on-demand services in a cloud, can prove attractive and is currently intensively investigated in research and industry.

In the course of transformation to Industry 4.0, automation and digitization of factories and production systems is advancing rapidly. The trend develops towards high connectivity of distributed system components. This opens new opportunities for the use of cloud services. In addition to computing and storage

* Corresponding author, axel.vick@ipk.fraunhofer.de

capacities, cloud services can also provide complex functionalities (value added services) such as the control of industrial robots and plants [1]. Due to their accessibility via network interfaces, they eliminate the need for local installation and execution of software [2]. Moreover, cloud services can cost-effectively add new features to existing production systems [3]. They increase the mobility and flexibility of these production systems, and thus their efficiency. The subdivision and software separation of communication, hardware, and value-added services also creates scalable, distributed systems that can be adapted to dynamic factory environments.

One functionality with high demands on computing power is, for example, image processing. Rudorfer and Krüger [4] designed a cloud-service-based system architecture in which image processing is offered as a web service. Based on their experiments, they conclude the possibility of rapid prototyping of new image processing applications. By using OPC UA, this can involve the use of various modular services from different sources, although the authors noted the lack of real-time capability of the service offered.

Lambrecht et al. [5] implemented a cloud-based system architecture with path planning for mobile robots as a value-added service. Using a 4G campus network as the communication medium, they detected a negligible impact of communication latencies on navigation success. However, the authors restricted this finding to reduced speeds of their mobile robot considering a maximum latency for their use case at 150 ms.

Wassermann et al. [6] also designed a cloud-service-based system architecture, whereas path planning, robot control, visualization and local hardware services were offered as modular, independent, relocatable web services. They successfully demonstrated the approach using a pick-and-place task in an environment with obstacles. Object detection and path planning were performed correctly and the processing time of the planning tasks was considered acceptable. The authors suggested the use of OPC UA as a communication framework.

A comparison of the performance of a model predictive control (MPC) and a proportional-proportional integral (P-PI) control as cloud services in network environments with fluctuating latency and bandwidth was performed by Vick et al. [7] on a six-axis industrial robot. The state-space model on which the MPC controller is based was derived analytically and control parameters were tuned manually. In experiments, the MPC controller was found to be more stable and faster for increasing latencies, and exhibited less overshoot and lower steady state error than the P-PI control. Latencies up to 160 ms were tested.

Instead of deriving a state-space model of the robot to be controlled, Briese et al. [8] investigated the possibility of using a non-model-based control approach (modified Switching Active Disturbance Rejection Control) for cloud-service-based distributed robot control in a network with jitter. In experiments, they achieved stabilization of all six axes of their robot for cycle times up to 100 ms. The small number of tuning parameters and thus a simpler model adaptation than with the model synthesis using first-principle methods are advantages of the authors' approach.

A different grey-box approach was investigated by Gillespie et al. [9]. The authors performed a system identification of a soft-robotic axis by first training a neural network with measurement data of the system to be identified and then deriving a linearized state space model from the neural network. They compared the MPC based on this model with a conventionally synthesized model. For the control based on the system model learned by the neural network, the authors detected a steady state error of two degrees, as well as a higher overshoot and a larger rise time than for the conventionally synthesized model.

Cloud-based control structures in conjunction with system identification services can further facilitate the integration of both conventional industrial plants and novel prototypes into a factory cloud [10]. This is because the universal applicability of this combination of services can eliminate the need for labour-intensive modelling and the adaptation of control algorithms on the engineering side. Research potential arises in investigating the feasibility of the universal approach and its applicability to unknown, dynamic systems. In a first approximation, only linear and linearizable systems are to be considered.

2. METHODS

2.1. Service-based automation

In terms of service-based automation systems, the architecture is split into hardware services and value-added services. The hardware services are responsible for running the driver to control a particular device and provide a common interface for the orchestration client or other services. The value-added services are responsible for high-level algorithms and data processing providing an interface to execute their functions. In contrast to choreography, the concept of orchestration is chosen because it facilitates the flexible addition and disconnection of services, and contributes to the reusability of individual services in the future. In addition, orchestration is more scalable with respect to a possible extension of the system to multi-client capability. The greater communication overhead in the orchestration client due to the redirection of all exchanged information must be accepted as a drawback. The OPC UA standard is chosen for communication between the services and clients. This is an M2M communication protocol for industrial applications and it is often used to enable the use of data from different sources.

2.2. Subspace system identification

The system descriptive matrices of a state space representation can be obtained using subspace system identification methods based on input and measured data. The most common methods in this regard are Canonical Variate Analysis (CVA), Multivariable Output Error State Space (MOESP), and Numerical Subspace State Space System Identification (N4SID) [11]. In general, these methods are based on finding the subspace spanned by the columns of an extended observability matrix.

Starting from a given set of system input U and response Y , using several decompositions and transformations, $\{A^* = TAT^{-1}, B^* = TB, C^* = CT^{-1}, D^* = D\}$ are found. By setting $C^* = I$, and $D^* = 0$ for causal systems, only two matrices A^* and B^* (using another T) have to be transported later. Adding OPC UA communication interfaces for input, response, parameters and system matrices provides a value-added service for identification of arbitrary linear systems.

2.3. Model predictive control

With the help of the system model, e.g. a discrete-time state space model, the future measured variables $\hat{y}(t+k|t)$ are estimated up to the prediction horizon at time step $t+N_p$ on the basis of the past input variables $u(t)$ and measured variables $y(t)$ as well as the future input variables $u(t+k|t)$ up to the control horizon at time step $t+N_c$. Initial values are first used for the future input variables.

A cost function J is calculated from the deviation of the estimated measured variable trajectory $\hat{y}(t+k|t)$ from a reference trajectory $w(t+k|t)$ up to the prediction horizon and from the required control effort $u(t+k|t)$ up to the control horizon. Both terms are weighted with corresponding weighting matrices Q and R , respectively. Manipulated variable constraints u_{min} and u_{max} or other constraints can be incorporated. Combining both horizons N_p and N_c to a common horizon N allows using the cost function (1):

$$J(y, u) = \sum_{k=0}^N [(y(k) - w(k))^T Q (y(k) - w(k)) + u^T(k) R u(k)] \quad (1)$$

Adding OPC UA communication interfaces for system model, state, target and manipulated variables provides a value-added service for model predictive control of arbitrary systems.

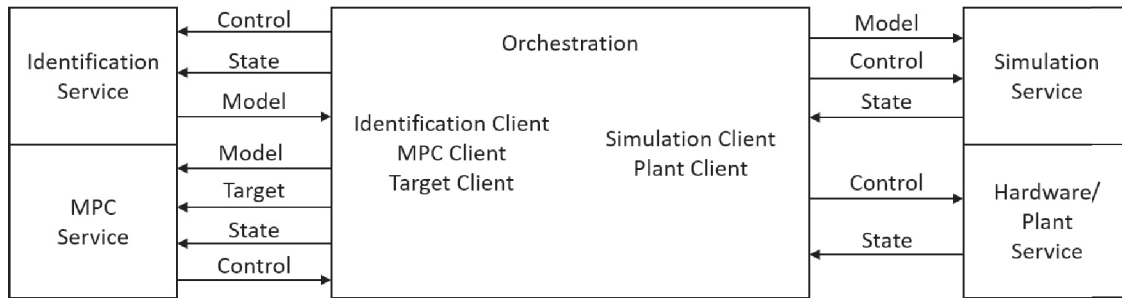


Fig. 1. System architecture for service-based identification and control.

3. CONCEPT AND IMPLEMENTATION OF UNIVERSAL IDENTIFICATION AND CONTROL

3.1. System architecture

Services depend on the results of other services to progress in the program flow. The exchange of information and the temporal coordination take place via the orchestration client. The services connected to the orchestration client, their inputs, tasks and outputs are shown in Fig. 1.

The hardware service has access to the physical system hardware. By controlling actuators it applies the control signals, and by polling sensors it returns the current system state. The hardware service can be replaced by a simulation service containing the plant simulation. It must be initialized with the system model matrices. Like the real plant, it then accepts control signals and returns the system state.

The identification service identifies and returns the system matrices based on the logged input and output variables. In addition to the simulation service, these matrices are also required by the MPC service. The latter is initialized with these matrices. It calculates a control trajectory to reach the target position using the target state and the current state, and then returns the entire manipulated variable sequence.

The orchestration client initializes the services and forwards the results to other endpoints. In addition, the orchestration client communicates with another client, the target client, with the help of an integrated server. It is able to accept target positions entered by the user, produce random manipulated variable sequences, play back recorded manipulated variable sequences, generate target positions in the form of a step/ramp function and forward all this data to the orchestration client.

The principle of communication between the services and clients is shown in Fig. 1. In addition, this sketch also illustrates the possibility of outsourcing simulation, identification and control services to a cloud.

3.2. System Components

Robot Server. The Robot Server is the OPC UA capable server of the six-axis Amtec robot. The Amtec robot consists of six PowerCube modules, which are arranged in a kinematic chain and can be controlled individually via a CAN interface. The robot control hardware service inherits from an OPC UA Adapter and provides variables for accessing the PowerCube driver. The PowerCube driver implements the following operations:

- Motion with target positions;
- Motion with target speeds;
- Motion with target motor currents.

Depending on the operation mode, outer closed-loop controllers can be defined, e.g. axes position control by motor currents, which will be used by the MPC service.

Waterworks Server. The waterworks is a basin with two inlets and one outlet. The flows of the inlets are controlled by one pump each. The water supply to the pumps is provided by the outlet of the basin. Thus, the waterworks is a closed-loop system. The input variables of the waterworks system are the set-points of the pump capacities. The corresponding output variable is the pressure change caused by the pumps. This is specified in Pascal and measured starting from the system normal pressure, which is approx. 0.21 Pascal. The Waterworks Server implemented in Codesys accesses a PLC that controls the pumps and reads the pressure sensor. The Waterworks Server is designed to run close to the system to be controlled, since a direct bus connection to the PLC of the waterworks is necessary to control it in real time. The Waterworks Server inherits from the OPC UA Adapter, providing the input and output variables respectively.

Simulation Server. The Simulation Server is a server instance for simulating linear systems of arbitrary order using their state space representation. The Simulation Server program consists of two threads that are executed in parallel.

One thread runs the OPC UA Server, which provides access to various variables of the simulation. Since the sizes of these variables and the associated elements of the simulation class object depend on the system to be simulated, they are generated dynamically. OPC UA variables are created, transmitting the matrices of the state space representation and a constant disturbance of the system to be simulated as well as the input and output variables. Furthermore, when simulating systems, it is possible to add a Gaussian noise to the state variables. The standard deviation of this noise can also be specified by the OPC UA client.

In another thread, the actual simulation is executed, where the evolution of the simulated system is discretized. The time step length depends on the discretization time constant when discretizing the model and can also be specified by a client. Unlike the Hardware Server, the Simulation Server is not limited to local execution. The only limiting criterion is the cycle time, which increases with distance from the Control Client. Therefore, the Simulation Server is suitable for outsourcing to a cloud.

Identification Server. The Identification Server is an OPC UA server implemented in Python that provides a service for identifying state space models based on input and output data. Subspace identification is performed using the Python module *SIPPY* (Systems Identification Package for Python). This module provides methods for identification and simulation of linear, dynamic SISO and MIMO systems. In the case of noisy data, it is first smoothed before the actual identification takes place.

The identified matrices are transformed so that the initial output matrix C^* is a unit matrix. A simulation of the state trajectory can be initiated server-side using the identified model to compare with the output data from the submitted dataset and calculate an error measure. This functionality will be used to validate the identification results.

MPC Server. The MPC Server calculates manipulated variable sequences according to the principle of model predictive control. For this purpose, the matrices of an identified state space representation required for the MPC, the weighting matrices R and Q from the cost function (1) and the manipulated variable constraints of the system must be sent via OPC UA. Since the size of the matrices depends on the system to be controlled, the OPC UA variables and the MPC automation class object are dynamically generated on client request.

During cyclic control, the current system state and the target are each sent from the client to the MPC server. Once the client requests the OPC UA variable of the input variable sequence, the MPC calculations are executed and the result is returned to the client. The actual MPC was implemented using algorithms and data structures from the control and matrix sections of the C++ toolbox *dlib*.

4. EXPERIMENTS AND RESULTS

The aspects under which the identification quality is to be considered are the correctness of the identified models and their statistical certainty, i.e. trustworthiness. To evaluate the control quality, the performance of the MPC shall be checked with respect to the steady state error, the overshoot and the rise time. The model correctness shall be quantified using the normalized root mean square error (NRMSE). The value range of the NRMSE is from $-\infty$ to one. A smaller value means a worse fit and a value of one stands for a perfect fit. If the NRMSE is zero, then the fit is as good as using the mean as a model of the data.

The generality and universal applicability of the combination of services for identification and model predictive control will be tested by three-stage experiments. The identification performance and the control performance will first be verified using the simulation of the Amtec robot. This is followed by experiments on the real Amtec robot system, in which both disturbances can occur and constraints must be considered. However, the real robot system has the same structure as the robot in the simulation. Finally, investigations are carried out on a structurally different system, the waterworks. For both systems, the average steady state error, the average overshoot and the average rise time are calculated.

4.1. Model accuracy and control performance in simulation

In the simulation experiments, model accuracy was higher for axis velocities than for their positions, decreasing with increasing latency from a mean NRMSE of 0.73 at 7 ms latency to 0.025 at 140 ms latency and remaining constant as noise increases.

The NRMSE is comparable for all axes and it decreases with increasing cycle time (see Fig. 2a). A curve fitting shows that the modelling quality of the velocity courses of all axes, except axis 4, decreases exponentially and with similar strength. At a cycle time of 7 ms, the NRMSE of the velocities is 0.725 on average for all axes and 0.025 at a cycle time of 140 ms.

The modelling quality of the position trajectories decreases less with increasing cycle time (see Fig. 2b). For all axes, there is a similar negative development from an average NRMSE of 0.315 to 0.035.

The steady-state error from the feedback controlling the identified model in simulation ranges from -1% to 1% for all noise levels and latencies tested. Overshoot increases linearly from 0% to an average of 2% while rise times have a minimum at 100 ms. Rise times and overshoot depend on MPC weights, with rise times being significantly more dependent on MPC weights than on noise and latencies.

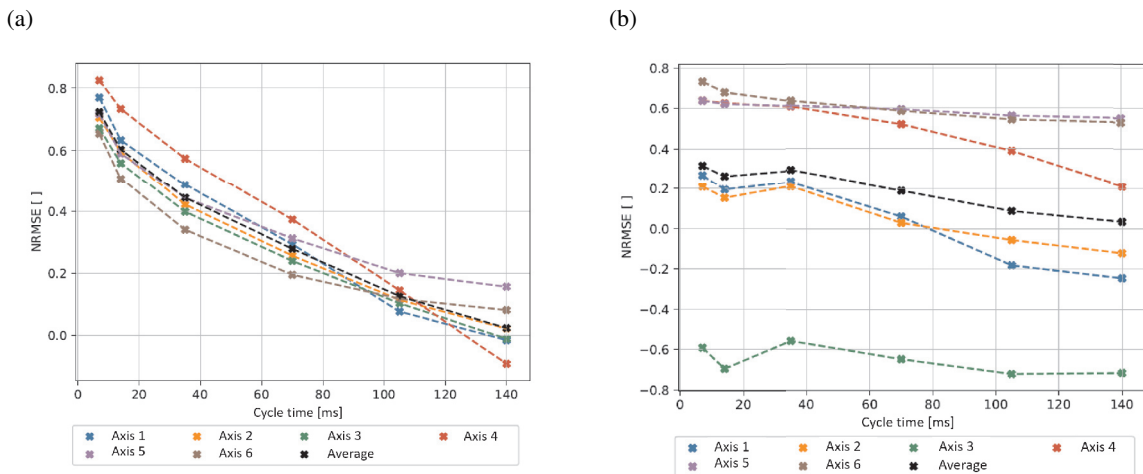


Fig. 2. NRMSE of (a) velocity and (b) position at different virtual cycle times in simulation.

4.2. Model accuracy and control performance for AMTEC industrial robot

In the robotic experiments, axes 1, 4, 5, and 6 were observed to be linear, so a state-space model could be identified. Therefore, only these axes were considered in the experiments. With a mean cycle time of 70 ms, the NRMSE of the velocities was again better than the NRMSE of the positions (see Table 1), with mean values of 0.83 and 0.23, respectively. Small-scale bootstrapping showed certainty of the model parameters, with all certainties at least one order of magnitude smaller than the mean value of the respective parameter, except for the parameters identified as zero.

Since the identified models of axes 1, 4, 5 and 6 are assumed to be sufficiently reliable, an arbitrary data set was selected from the seven collected data sets for the physical experiments. The model identified on the basis of this data set was selected for the controller performance experiments, the plots of which are shown in Fig. 3. The steady-state error for axes 1 and 5 was less than 1% and for axes 4 and 6 it was less than 4% (see Table 1). The control performance is exemplarily depicted in Fig. 3a.

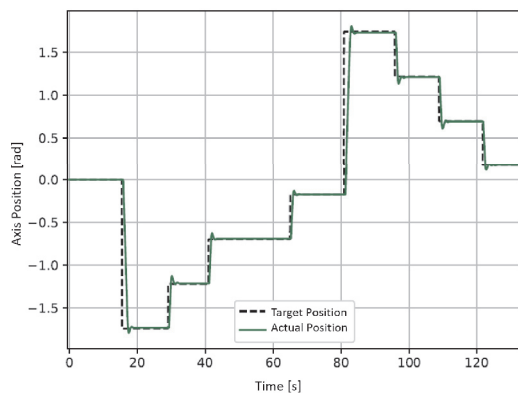
4.3. Model accuracy and control performance for waterworks

The water system proved to be linear over a range of 60% to 120% of the pump output. At an average cycle time of 70 ms, the NRMSE of the modelled system pressure was 0.78 with high confidence in the model parameters.

Table 1. Average and standard deviation of NRMSE for robot axes velocity model and MPC performance indicators for robot axes position control

	Axis 1	Axis 2	Axis 3	Axis 4	Axis 5	Axis 6
Velocity NRMSE Average	0.84	-0.14	0.52	0.81	0.84	0.81
Velocity NRMSE Std Dev	0.002	1.1	0.23	0.009	0.009	0.05
Position NRMSE Average	-0.11	-8.17	-0.87	-0.11	0.72	0.45
Position NRMSE Std Dev	0.41	17	0.55	0.37	0.10	0.20
Position Steady State Error	0.4%	–	–	3.9%	0.9%	2.1%
Position Overshoot	5.4%	–	–	1.9%	14.5%	9.4%
Position Rise Time	1.0s	–	–	2.4s	0.5s	0.4s

(a) AMTEC Robot Axis 1



(b) Waterworks Pressure Change

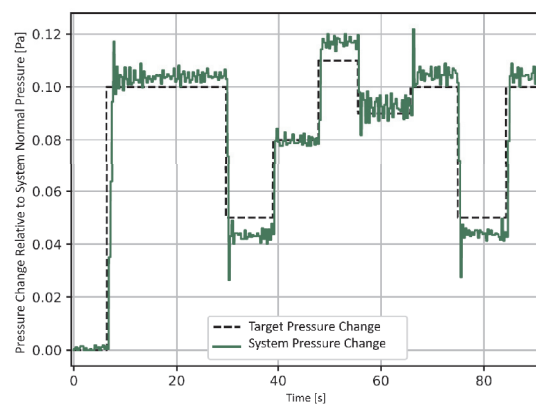


Fig. 3. Position control for Amtec industrial robot and pressure control for waterworks using MPC service with derived models from identification service.

The waterworks models' parameters had a standard deviation at least one magnitude smaller than their mean. This is valid in the considered range of pump performance between 60% and 120%. The simulations of the validation trajectory using the identified models also indicate a good quality of system identification. The average NRMSE of the pressure modelling was 0.78 with a low standard deviation of 0.016.

The average steady state error was 0.3%. The average overshoot of 54% was much higher than in the experiments on the Amtec robot, and in particular the values of overshoot after reaching the second, sixth and seventh control targets were very high with 47%, 219% and 45%, respectively. The reason for the strong overshooting control, on the one hand, is due to the long cycle time of 309 ms on average, on the other hand due to the missing possibility of controlling the pumps with negative values. The control performance is depicted in Fig. 3b.

5. CONCLUSIONS

Overall, subspace identification and model predictive control have been shown to be universally applicable to linear systems, even with high measurement noise and in network environments with latencies up to 140 ms. Stable control is achieved at even higher cycle times, but performance degradation occurs. To achieve optimal control performance, several technical problems must be solved. The main cause of insufficient control performance in the experiments was the non-optimal setting of the MPC weights, especially when switching between models with different cycle times or between models based on noise-free or noisy data sets. Automated determination or analytical derivation of the MPC parameters would help to achieve the optimal, or at least comparable, control performance with each identified model.

Using the above designed and implemented architecture, web service-based subspace identification and model predictive control were shown to be feasible. Within the limits set, the service combination for identification and the MPC were demonstrated to be comparable and in some cases even competitive to conventional, first-principle methods of modelling and feedback control.

ACKNOWLEDGEMENTS

The publication costs of this article were covered by the Estonian Academy of Sciences and Tallinn University of Technology.

REFERENCES

1. Vick, A. and Krüger, J. Using OPC UA for distributed industrial robot control. In *Proceedings of the 50th International Symposium on Robotics, ISR 2018, Munich, Germany, June 20–21, 2018*. VDE Verlag, 501–506.
2. Yen, C.-T., Liu, Y.-C., Lin, C.-C., Kao, C.-C., Wang, W.-B. and Hsu, Y.-R. Advanced manufacturing solution to industry 4.0 trend through sensing network and cloud computing technologies. In *Proceedings of the 2014 IEEE International Conference on Automation Science and Engineering (CASE)*. IEEE, 1150–1152.
3. Vick, A., Guhl, J. and Krüger, J. Model predictive control as a service: concept and architecture for use in cloud-based robot control. In *Proceedings of the 2016 21st International Conference on Methods and Models in Automation and Robotics (MMAR)*. IEEE, 607–612.
4. Rudorfer, M. and Krüger, J. Industrial image processing applications as orchestration of web services. *Procedia CIRP*, 2018, **76**, 144–148.
5. Lambrecht, J., Steffens, E. J., Geitz, M., Vick, A., Funk, E. and Steigerwald, W. Cognitive edge for factory: a case study on campus networks enabling smart intralogistics. In *Proceedings of the 2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), Zaragoza, Spain, September 10–13, 2019*. IEEE, 1325–1328.
6. Wassermann, J., Vonásek, V. and Vick, A. Distributed industrial robot control using environment perception and parallel path planning cloud services. In *Proceedings of the 2018 IEEE 23rd International Conference on Emerging Technologies and Factory Automation (ETFA), Torino, Italy, September 4–7, 2018*. IEEE, **1**, 1263–1266.

7. Vick, A., Kilinc, A. and Guhl, J. Variable-latency networked P-PI and MPC controller performance for industrial robots. In *Proceedings of the 2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, Zaragoza, Spain, September 10–13, 2019. IEEE, 1501–1504.
8. Briese, C., Vick, A. and Krüger, J. Cloud-based active disturbance rejection control for industrial robots. In *Proceedings of the 2018 IEEE 23rd International Conference on Emerging Technologies and Factory Automation (ETFA)*, Torino, Italy, September 4–7, 2018. IEEE, 1, 559–565.
9. Gillespie, M. T., Best, C. M., Townsend, E. C., Wingate, D. and Killpack, M. D. Learning nonlinear dynamic models of soft robots for model predictive control with neural networks. In *Proceedings of the 2018 IEEE International Conference on Soft Robotics (RoboSoft)*, Livorno, Italy, April 24–28, 2018. IEEE, 39–45.
10. Vick, A., Horn, C., Rudorfer, M. and Krüger, J. Control of robots and machine tools with an extended factory cloud. In *Proceedings of the 2015 IEEE World Conference on Factory Communication Systems (WFCS)*, Palma de Mallorca, Spain, May 27–29, 2015. IEEE, 1–4.
11. Jamaludin, I. W., Wahab, N. A., Khalid, N. S., Sahlan, S., Ibrahim, Z. and Rahmat, M. F. N4SID and MOESP subspace identification methods. In *Proceedings of the 2013 IEEE 9th International Colloquium on Signal Processing and its Applications*, Kuala Lumpur, Malaysia, March 8–10, 2013. IEEE, 140–145.

Teenus tööstuslike tootmiseseadmete universaalseks identifitseerimiseks ja kontrolliks

Verena Tessaro, Axel Vick ja Jörg Krüger

Käesolevas artiklis esitatakse universaalne lähenemisviis tootmiseseadmete identifitseerimiseks ja suletud ahela kontrolliks, mida osutatakse veebiteenuste kaudu OPC UA abil. Kiirprototüüpimine ning vanade süsteemide moderniseerimine ja digitaliseerimine vajavad sageli suletud ahelaga kontrolliteerimise projekteerimist ja rakendamist. Selliste süsteemide analüüs ja modelleerimine on töömahukas ja vajab protsesside mõistmist. Praegused identifitseerimis- ja kontrollivahendid nõuavad sisend-/väljundandmete süstemaatilist ettevalmistamist, mudelite muutmist ja häälestamist ning klassikaliste PID-kontrolliteerimise nõuetekohast projekteerimist. Artiklis esitatud lähenemisviis kasutab nõudmisel põhinevat teenust, võimaldades identifitseerimist ja kontrolli võrguühenduse kaudu, millel on otsene juurdepääs kontrollitavale süsteemile. Simulatsioon ja laboratoorsed katsed näitasid häid tulemusi erinevate süsteemide kontrollimisel. Selline lähenemisviis annab hea võimaluse süsteemi identifitseerimiseks ja ennustava kontrolli rakendamiseks tootmises.