BULLETIN OF THE POLISH ACADEMY OF SCIENCES
TECHNICAL SCIENCES, Vol. 72(4), 2024, Article number: e150114
DOI: 10.24425/bpasts.2024.150114

CONTROL, INFORMATICS AND ROBOTICS

Agent-based system for continuous control and its application to activated sludge process

Jakub POŚPIECH*, **Witold NOCOŃ**, and **Krzysztof STEBEL**

Faculty of Automatic Control, Electronics and Computer Science, Silesian University of Technology, ul. Akademicka 16, 44-100 Gliwice, Poland

Abstract. This paper presents a concept of architecture and ontology layouts for the development of multiagent model-based predictive control systems. The presented architecture provides guidelines to simplify the development of agent-based systems and improve their maintainability. The proposed multiagent system (MAS) layout is split into multiple subsystems that include agents dedicated to performing assigned tasks. MAS implementation was prepared which can use provided algorithms and actuators and can react to changes in its environment to reach the best available control quality. An example of MAS based on the proposed architecture is shown in the application of dissolved oxygen (DO) concentration control in a laboratory-activated sludge setup with a biological reactor. For that application, MAS incorporates agent-based controllers from the boundary-based predictive controllers (BBPC) family. Presented experiments prove the flexibility, resilience, and online reconfiguration ability of the proposed multiagent system.

Keywords: adaptive control; agent-based control; control of dissolved oxygen; model-based predictive control; multiagent system; practical validation.

1. INTRODUCTION

Solutions based on multiagent systems (MAS) consistently gain popularity in various branches of industry. Adoption of these types of systems is considered an important part of industrial digital transformation and further development of cyber-physical systems [1].

Nowadays, research interest in multiagent systems is focused mostly on its applications to distributed systems, including cooperative, consensus, and formation control methods for MAS [2] that are applicable for purposes like UAVs or high-speed train car control [3]. Multiagent systems are also actively developed in areas of balance and fault control of microgrids [4] and human-computer interfaces [5]. Another area of intensive MAS research focuses on applications in control, fault detection, or supervisory purposes in industrial processes. Most solutions in this area of MAS research are dedicated to manufacturing systems control. One of the first major, industrial, agent-based control systems was P2000+ – a multiagent system dedicated to balancing the workload of production machines at the Daimler Chrysler plant in Stuttgart [6]. In recent years there has been a noticeable increase in research interest in agent-based solutions for flexible, reconfigurable manufacturing systems. A message protocol of inter-agent communication dedicated to facilitating the reconfiguration of Plug and Produce manufacturing systems was proposed in [7].

Industrial applications of agent-based systems also gained recognition in the process control area [8]. A hierarchical, multicontroller incremental design framework for multiagent systems was presented in [9]. This MAS implements various control algorithms as agents whose actions are switched between by the supervisor agent depending on the operating regime. That solution was then applied to control room temperature. In [10] authors developed a multiagent distributed model predictive controller in which control was derived through a cooperative game algorithm with fuzzy negotiations. Other tasks, where MAS was used to directly control continuous processes include [11] in which an agent-based system controlled the pressure of recycled gas with the presence of significant disturbances and [12] where self-organizing MAS was developed to control "black box" processes, the latter example was tested on prey-predator problem.

Provided examples suggest that interest in researching MAS dedicated to direct process control exists; however, these types of solutions are less common than in other presented areas. That state is most likely caused by lacking and outdated standardization [1, 13], the lack of common education on building agent-based systems [14], and concerns about reliable time performance of the most popular tools for building MAS [15]. On the other hand, many MAS attributes are proven to be useful in the process control domain, e.g. reconfigurability [16], ability to perform social interactions [10, 17], or decision-making using belief, desires, and intentions. Attempts to create standardized approaches to MAS implementation are taking place, but these are still in the initial stages of development [18]. Although systems built with popular MAS design platforms cannot fulfill hard real-time constraints, their time performance is sufficient to control processes with higher time tolerance [19].

One of the process types, where dynamics are slow enough to allow usage of control systems with larger time variance is

---

*e-mail: jakub.pospiech@polsl.pl

Manuscript submitted 2023-12-07, revised 2024-02-29, initially accepted for publication 2024-03-26, published in July 2024.

the control of dissolved oxygen (DO) concentration in activated sludge process used for municipal wastewater treatment plants (WWTP). In activated sludge reactors, DO concentration for aerobic processes should be kept within the range of 2–3 (mgO_2/l) which is enough to maintain the growth of microorganisms [20].

The problem of DO concentration control in bioreactors has been the subject of intensive studies over the last decades and various approaches have been proposed. Predictive control of DO concentration in a bioreactor with simulative validation is presented in [21]. A nonlinear model predictive controller with a hierarchical structure for DO concentration control is presented in [22]. Another type of predictive control for DO concentration using a two-level model predictive controller is proposed in [23]. Boundary-based predictive controller (BBPC) is proposed in [23]. For that algorithm DO concentration value is kept between set maximum and minimum values with an ON-OFF pump, using predictions of future process behavior based on a dynamical model of a process for varying prediction horizon.

Although WWTP processes are considered to have large potential regarding the introduction of MAS [14], aside from agent-based expert systems designed to assist operators in control of activated sludge process [24] and agent-oriented preliminary work [25, 26], any multiagent control system dedicated more or less specifically for activated sludge process control and especially for DO concentration control have not been proposed in the literature.

This paper introduces a new multiagent control system architecture that provides the possibility to use the system in multi-input scenarios and gives the ability to alter MAS configuration during runtime through additions and removals of specific agents. This new system is then used alongside a control algorithm from the BBPC family in DO concentration control to verify its usability.

The novelty of this article mainly results from the following contributions:

1. Application of MAS which uses a dynamical model of DO concentration, prediction algorithms, and multiagent social interactions to operate multiple aeration pumps with various characteristics used to keep process value precisely within the desired range and proposition of design framework for multiagent systems dedicated to model-based control algorithms. The novelty of the proposed framework is expressed in dividing controller operations into simple tasks that are distributed between multiple agents grouped in subsystems, instead of encapsulating control algorithms inside an individual agent as in [18]. This approach encourages the reusability of agents and simplifies their development, which considering the mentioned lack of common education on building agent-based systems and shortage of specialists in agent-based development [14], is especially important.
2. Presentation of an evaluation mechanism for agents, which is used as an indication tool to find agents, whose calculation results do not match results from real process measurements.
3. Introduction of fault detection algorithm that uses the provided knowledge about the system to detect malfunctioning agents.
4. Example of practical implementation of described multiagent system with the evaluation mechanism and the fault detection algorithm included and experimental validation during aeration of real, laboratory scale setup of activated sludge process.
5. Design of generic, lightweight ontology schema dedicated to real-time control of continuous processes and its adoption to multiagent predictive control system presented in this article.

The paper is organized as follows: in Section 2 details regarding the application of MAS to DO concentration control are described. Section 3 contains a detailed description of the developed MAS architecture. Ontology for agent-based predictive control is presented in Section 4 and is followed by the description of an example implementation of the prepared architecture in Section 5. Section 6 presents preliminary simulation results and is followed by Section 7, which presents experimental verification results obtained on activated sludge laboratory setup. Lastly, conclusions, discussion, and authors' remarks are described in Section 8.

2. APPLICATION OF MAS TO DISSOLVED OXYGEN CONCENTRATION CONTROL

2.1. Definition of control goal

The control goal for the proposed system is to keep the DO concentration precisely within the desired range of $[SP_{\min}, SP_{\max}]$. Control values should be derived by dedicated agents and supplementary data, e.g. information about model parameters should be provided by separate agents. For this specific task of DO concentration control MAS control value calculations should be based on the BBPC algorithm. Process controlled by the MAS needs to reach both its lower-value boundary $SP_{\min}$ and its upper-value boundary $SP_{\max}$ with similar precision. However, as mentioned in [24], a single input system based on an ON-OFF pump and BBPC control algorithm is not able to reliably fulfill that requirement. To achieve that, a smaller variable speed pump was added to create a multi-input control system as shown in Fig. 1. Similar multiple blower systems composed of both ON-OFF and variable speed pumps are also used in large-scale WWTPs [23]. The ON-OFF pump can be turned ON only for a fixed time during the aeration cycle. Due to this technical limitation, both pumps operate in 30 (s) aeration cycles. Variable speed pump can work with shorter aeration cycle times, but due to the slow dynamics of the aeration process, reduction of aeration cycle time would not provide significant improvements. Therefore, the authors decided to use identical aeration cycle times for both pumps. The designed MAS needs to operate both aeration pumps to reach boundaries as closely as possible. Additionally, aeration should be mainly performed by the ON-OFF pump due to its higher efficiency and the variable speed pump should be used only for reduction of remaining control error. In order to avoid another layer of complexity to the control problem regarding complex dynamics of airflow, an assumption is made that both pumps cannot be in the ON state simultaneously.

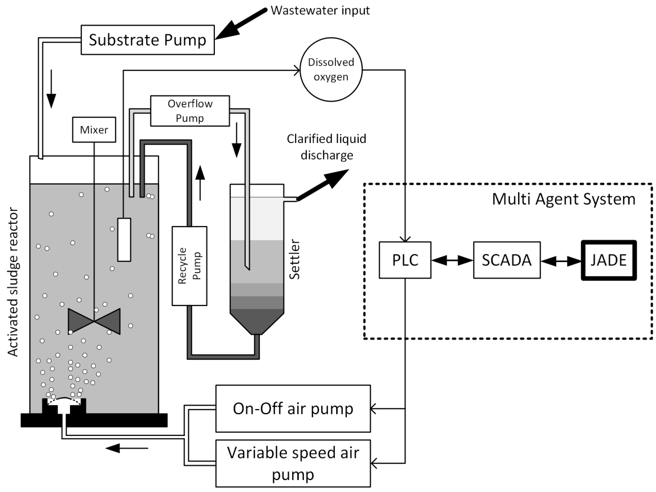

Fig. 1. Schematic diagram of DO concentration control system

2.2. Controlled plant description

The bioreactor used to verify the designed MAS is presented in Fig. 2.

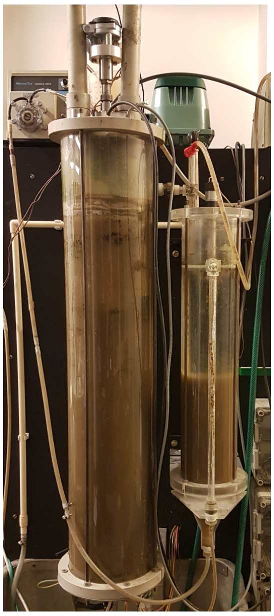

Fig. 2. Overview of a laboratory setup with activated sludge bioreactor (on left)

The reactor is part of a continuously operated activated sludge laboratory setup, which includes a well-mixed fed-batch bioreactor with constant liquid level, where biological removal of organic waste is performed. DO is measured with the sampling time 1 (s) using a Hach Lange SC200 meter with a dissolved oxygen probe. Biomass inside the bioreactor is fed with substrate transported by the peristaltic pump, substrate flow can be manipulated to induce changes in process load. During the experiments organic substrate prepared from 6 (g) of peptone per 1000 (ml) of water is fed to the reactor by the 150 (ml/h) peristaltic pump. Sensors and most of the actuators are connected to the NI-9074 CompactRIO controller, variable speed pump is connected to the separate NI-9074 CompactRIO controller. The remaining parts of the experimental setup are a desktop PC that exchanges data with controllers using LabVIEW Network-Published Variables and publishes variables using the OPC UA server created with LabVIEW OPC UA Toolkit and another PC connected through the network where MAS should run:

$$\frac{\mathrm{d}DO(t)}{\mathrm{d}t} = k_La(t)\,(DO_{\mathrm{sat}} - DO(t)) - OUR(t), \tag{1a}$$

$$T_{k_La}\frac{\mathrm{d}k_La(t)}{\mathrm{d}t} = -k_La(t) + u_{\mathrm{air}}(t - T_0). \tag{1b}$$

Control algorithms use a dynamical model of DO changes (1), where $k_La(t)$ (1/h) represents the transfer coefficient of oxygen from air bubbles to liquid, and DO_{sat} is DO saturation level (mg O_2/l). DO_{sat} is considered constant because all the experiments are performed using an indoor reactor where external conditions, especially temperature can be considered constant at 20°C. However, in practice DO saturation level can change depending on, e.g. pressure, air humidity, and primarily water temperature [27]. The oxygen demand of organisms inside the reactor is represented by $OUR(t)$ (oxygen uptake rate) (mgO_2/lh), this value depends on substrate flow and its changes are the main source of process disturbances. The first-order plus dead time model in (1b) contains the following parameters: time constant T_{kLa} (h), delay time T_0 (h), and control action represented by term $u_{\mathrm{air}}(t)$, for ON-OFF pump value of this term is switched between 0 (OFF mode) and u_A (ON mode). For aeration with variable speed pump values of this term depend on the parameters of the running pump. The process is operated with constant liquid level and under unvarying mixing conditions so values of parameters T_{kLa} and T_0 are considered constant.

3. MULTIAGENT SYSTEM ARCHITECTURE

The concept of the proposed MAS architecture focuses on the division of the process control task into simpler, independent sub-tasks assigned to the subsystems. Figure 3 presents a general UML component diagram of the architecture of the MAS. In this diagram, MAS consists of multiple subsystems, each subsystem has defined interfaces that are used to transfer data over the system. Every subsystem is a logically separated part of the system that should realize the assigned task. Tasks that should be performed by every proposed subsystem are described further in this chapter. As presented in the component diagram of an exemplary subsystem in Fig. 4, subsystems contain software agents that together realize a task dedicated to their subsystem. Every agent in the system must be assigned to exactly one subsystem, the assignment is permanent. Agents communicate with each other by exchanging messages. Details regarding the methods of transfer of the messages and detection of the specific agent that another agent should communicate with are left to be decided during the implementation phase. In the proposed architecture, agents that are assigned to the same subsystem can communicate and form relationships with other agents without any constraints. Examples of communication and relationships

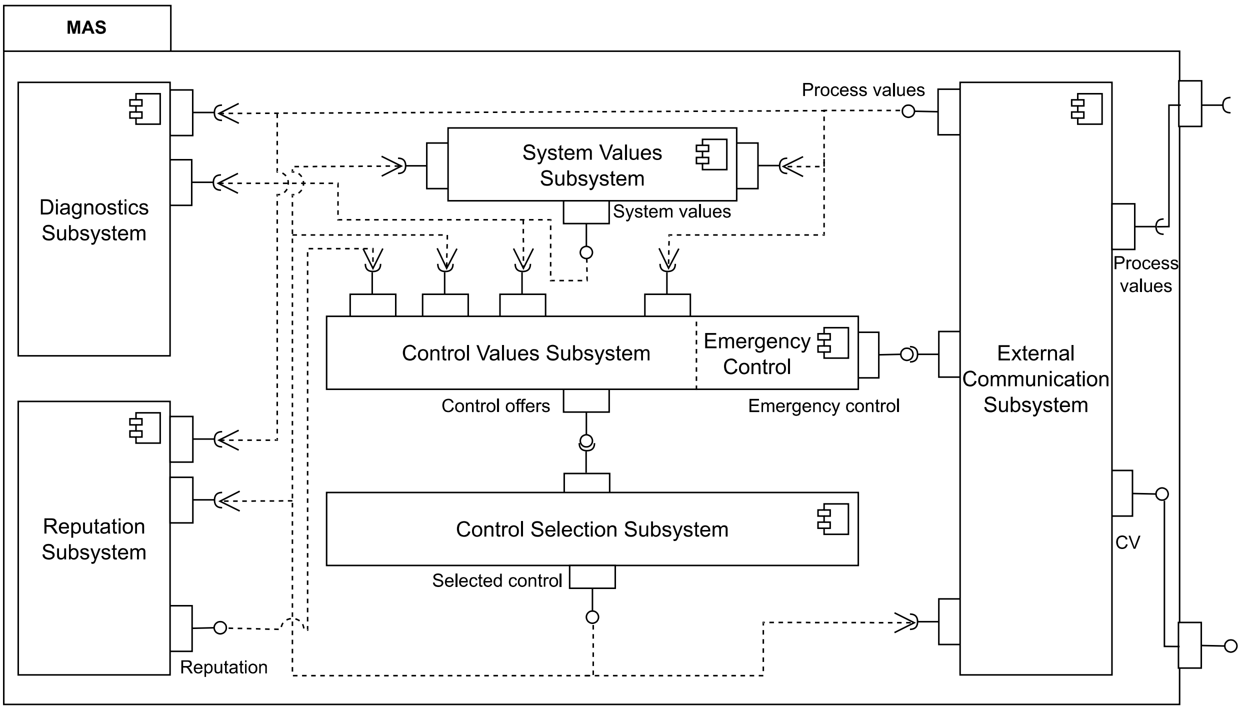

Fig. 3. Layout of proposed MAS architecture with information flow

between agents in the subsystem are presented in Fig. 4. Agent 1 needs to communicate with Agent 2 to deliver Output 1, Agent 2 cooperates with Agent 3 to calculate Output 2, Agent M competes with Agent N in "better" production of Output N.

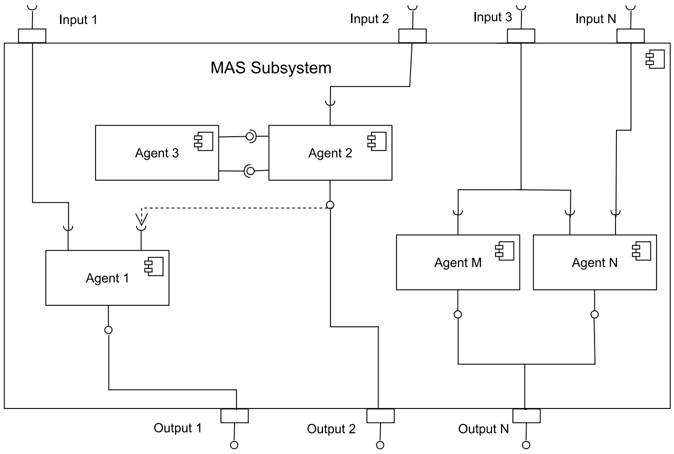

Fig. 4. Subsystem component diagram

On the other hand, communication between agents from different subsystems is limited to the exchange of data specified by the subsystem ports. Receiving or passing any other type of data to agents outside the subsystem shall be forbidden.

The presented approach should simplify the development of MAS and improve its maintainability. Successively splitting complex tasks into simple, independent sub-tasks during the design phase facilitates the development of agents dedicated to performing the sub-task. On the other hand, increasing the number of agents might complicate system integration and raise the risk of the occurrence of unexpected emergent behaviors. A proper distinction of sub-tasks and constraints imposed on communication between submodules should address this issue – agents need to be integrated only with other agents within the same subsystem making this task more manageable. It should be possible to modify, replace, and in some cases even remove individual agents inside a subsystem without the need to alter the logic or procedures of agents inside any other module of the system. This could reduce the time and costs needed for any modification of the control system.

In the presented architecture, MAS is divided into six subsystems, four of which are directly involved in the calculation of new control values (CV).

- External communication subsystem – is responsible for receiving and sending data outside of MAS, e.g. sending CV directly to the actuator or receiving data from sensors through some type of network protocol.
- System values subsystem – is dedicated to calculating or estimating *system values* based on available *process values* and *current control*. *System values* and *process values* are described in the next section.
- Control values subsystem – contains agents that use available knowledge about the process to calculate the next control value. Agents from the subsystem may generate more than one control value. In such cases, all new *CV*s are sent as *control offers* to the control selection subsystem where the most suitable *CV* is selected. This subsystem may contain a separate group of agents that work as a security layer and take immediate *emergency control* over the process when an emergency occurs.

- Control selection subsystem – is dedicated to gathering available *control offers* and selecting the most suitable *CV*. The selected *CV* is then passed to the external communication subsystem. In cases where the control values subsystem generates a single *CV*, this subsystem can be omitted.

The remaining two subsystems are dedicated to performing supportive tasks aimed at improving control quality and system resilience:

- Evaluation subsystem – agents from this module evaluate performance and assign *reputation* grades to agents from the control values subsystem. The purpose of the assignment is to identify agents that, at the specific moment, do not provide reliable information (e.g. their predictions regarding controlled process behavior do not match reality) and prevent them from taking part in future process control. As a result, the control quality of the system should improve over the time. Additionally, the system should be able to adapt to changing conditions.
- Diagnostics subsystem – its task is to detect faults that occur inside the MAS. After any fault is detected, diagnostics subsystem should try to remove the detected issue and or mitigate negative effects caused by the fault e.g. through informing the process operator about the incident, performing self-repair actions on faulty elements, or in case critical issues turn into safe mode.

4. ONTOLOGY FOR AGENT-BASED MULTIPLE INPUT PREDICTIVE CONTROL

Modern ontologies usually form large schemas with hundreds of classes, relations, and thousands of instances. These domains are used in large services alongside dedicated ontological engines to precisely describe concepts from broad areas. However, for a case presented in this article, such a large ontology is not necessary. Systems dedicated to controlling plants in real-time usually need knowledge from the area limited to the characteristics of controlled processes, actuators, sensors, and external factors that may influence the control quality. Based on that, the authors assessed that the usage of available tools for ontology management would cause unnecessary complexity and decided that the designed ontology should not rely on any available ontology engine and instead use more lightweight tools to manage the generated knowledge base. As a result, the authors decided that the main objective is to design a generic schema with a relatively small number of classes that do not require a dedicated full-scale ontology engine but can describe the used concepts with satisfactory precision. The layout of the prepared ontology in the form of a class diagram is presented in Figs. 5 and 6. Dashed lines with a circle at one end that are presented in Fig. 5, indicate relations in which one object contains the other's name. The ontology representation is divided into three groups called domains. Each domain contains classes dedicated to specific purposes.

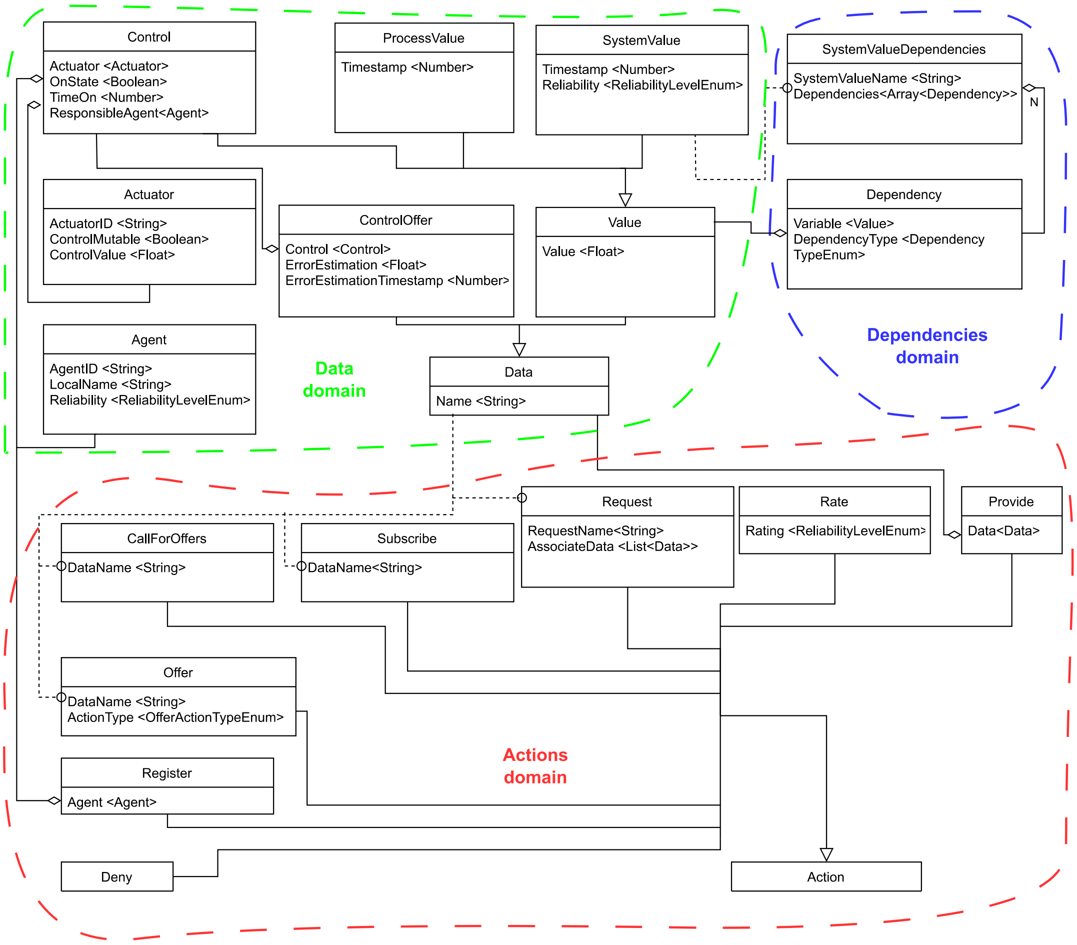

Fig. 5. Proposed ontology schema

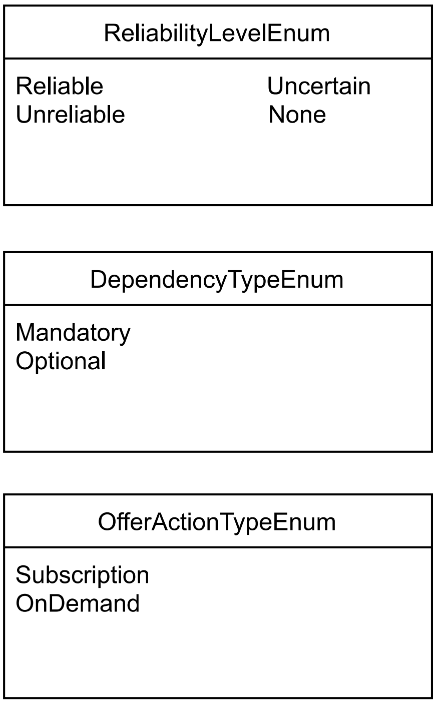

Fig. 6. Ontology additional types

- Data domain can be considered as a main domain of the ontology. Classes inside this group represent concepts of information used by agents to perform their actions. The root class for most of the concepts in the domain is a class called *Data*, which is a generic representation of data and contains information about its name. In the presented layout this root class has two descendants, one of which is called *Value* and is an abstraction that represents any concept possible to describe with a single numeric value. The class *Value* has three descendants: *ProcessValue* represents all the values gathered directly from the process (i.e. sensor readings), and the *SystemValue* class describes values that are not gathered by sensors but derived from *ProcessValues* through some type of calculations. The last descendant is the *Control* class which is a concept dedicated to describing control values. For the control system presented in the paper, this class has additional information about an actuator that will be used and an agent that prepared the control value. Such an idea of data and value concepts hierarchy is universal and can be reused in applications that adopt many other control systems and algorithms. The *Data* domain contains also the *ControlOffer* class that represents concepts more specific to the control algorithm described in this article. *ControlOffer* instances contain information about the proposed control value and predicted response of the process when this control will be applied.
- Actions domain is based on FIPA-ACL agent communication language with the addition of new concepts and modification of existing ones. This domain contains classes that represent interactions between agents with *Action* as a root class. *CallForOffers*, *Subscribe*, and *Request* are classes that describe various stages and methods to gather data by agents. Using *Subscribe* to get data results in receiving messages with specified information periodically. On the other hand, *Request* is used to get data only once. The *AssociateData* field can be used to pass additional information that will be used by the receiving agent to deliver requested data. Information (i.e. instanced of classes from the data domain) is exchanged between agents through *Provide* action. *Register* action is used to inform other agents about the presence of a new agent in the system. *Rate* action is used by evaluation agents in the evaluation mechanism and *Deny* represents a negative response to agents' requests.
- Classes in the Dependencies domain are used in the example fault detection algorithm and are elaborated in the next chapter that describes the practical implementation of the *Diagnostics Subsystem*.

Information in the form of instances of the presented ontology concepts is passed between agents in messages. In practical implementation presented in the following chapter messages sent between agents contain XML nodes which are serialized from and can be deserialized to Java objects using XML serialization and deserialization tool called Jackson.

Data, control, and value concepts hierarchy form a universal core of the prepared ontology that can be reused in other control systems. Other classes can be modified, added, or removed to comply with the specific needs of other applications. Additionally, the number of classes used in ontology allows for manual modifications without the need to use any tools dedicated to large-scale ontologies.

5. PRACTICAL IMPLEMENTATION OF MULTIAGENT SYSTEM ARCHITECTURE

The multiagent system runs on a PC connected through a local network to the SCADA PC which is directly connected to two CompactRIO NI-9074 PLCs that send signals to the actuators and receive data from the sensors.

Data transfer between MAS and PC with SCADA deployed is performed using OPC UA protocol as presented in Fig. 7. Practical implementation of the proposed multiagent system architecture for DO concentration control was created using the JADE framework.

For the proposed MAS architecture, presented subsystems realize the following tasks:

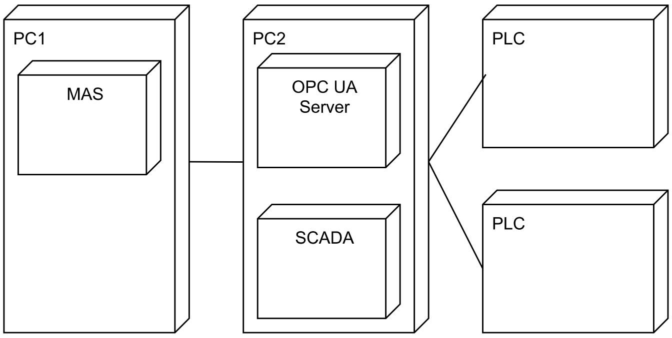

Fig. 7. Deployment diagram of example implementation

- External communication subsystem - contains agents that operate OPC UA client, which is based on an open-source implementation of OPC UA communication protocol called Eclipse Milo. For DO concentration control only one *process value* is used – current DO concentration level reading from the dissolved oxygen probe. Additionally, the subsystem sends to the OPC UA server *CV* values that are prepared by the MAS.
- System values subsystem – two *system values* are distinguished – $k_{La}(t)$ that is calculated by agents in the subsystem using transformed (1b). Another *system value* is $OUR(t)$. Although $OUR(t)$ directly describes the oxygen uptake rate of the controlled process, its measurements are not available in this case, thus its value is estimated by agents from the subsystem using methods described in [24].
- Control values subsystem – subsystem contains agents that realize predictive algorithm to obtain new *CV* for ON-OFF pump and agents that calculate *CV* to reduce remaining error using variable speed pump.

$$DO_{\text{diff}} = \begin{cases} \left|SP_{\max} - \hat{Y}_{\max}\right|; & CV = ON, \\ \left|SP_{\min} - \hat{Y}_{\min}\right| & CV = OFF. \end{cases} \tag{2}$$

- Control selection subsystem – selection of the most suitable *CV* is based on three factors. The first factor is the prediction of the expected result that will be observed, if offered *CV* is applied. This information is expressed as DO_{diff} – a distance between the expected extreme DO value and set boundary (2). The next factor is the agent's *reputation* which is assigned by the evaluation subsystem. Offers from agents with higher *reputations* are preferred over ones from agents that are considered unreliable. The last factor is information about the pump used for control in the *control offer*, as described in Section 2. Aeration should be mainly performed by the ON-OFF pump. Therefore, in a situation where multiple *control offers* from agents with reliable *reputations* declare satisfactory DO_{diff} values, the offer that uses the ON-OFF pump will be selected.
- Evaluation subsystem – agents located in the evaluation subsystem can assign three possible *reputation* grades: *reliable*, *uncertain*, and *unreliable*. A grade *reliable* indicates that other agents should trust declarations provided by the agent with this grade, on the other hand, declarations from agents with a grade *unreliable* should not be trusted. *Uncertain* is a transitional grade that indicates that the agent's reputation may soon change to another grade, agents with this grade are treated by others the same as *reliable* agents. Reputation grades are assigned based on verification of the expected control result that the agent declared in the *control offer*. If the agent's declaration matches the observed result, its reputation is raised by one grade until it reaches a *reliable* grade, otherwise, its reputation is dropped by one grade until it reaches an *unreliable* grade.
- Diagnostics subsystem – uses knowledge about dependencies between *process values* and *system values* to detect faults in the system. These dependencies are represented in the ontology (Figs. 5 and 6). Every *system value* may have one or more dependencies. The diagnostic process is started manually or by agents when unexpected behavior is detected, e.g. an agent requested a new value of some *system value* but did not receive any answer. When that happens, the agent sends a request to the diagnostic subsystem to start diagnostics on potentially faulty values. The diagnostic subsystem begins the diagnostic process by sending a request to provide the potentially faulty value. From that point, one of three possible scenarios might occur: a positive reply when the requested value is received, such reply indicates that the system works correctly, and thus no fault is detected. The diagnostic subsystem may also receive a negative reply which indicates that agents responsible for providing requested value work correctly but due to outside factors are unable to provide the requested value. In such a case, the diagnostic subsystem uses its knowledge base to get dependencies of the requested value and start diagnostic processes for these dependencies. The last scenario assumes that no reply is received which indicates that a fault occurred in the agents responsible for providing the requested value. The result of the diagnostic process is presented as a GUI message to process operators who should react and start adequate procedures.

6. PRELIMINARY SIMULATION RESULTS

At the beginning, initial measurements to estimate u_A for the ON-OFF pump and u_{air} values range for the variable speed pump were performed using the approach described in [24]. For the ON-OFF pump u_A was estimated at 2.52 (1/h), for the variable speed pump u_{air} is represented as a range of values achievable by the pump and is estimated at $[0, 2.34]$ (1/h), $u_{\text{air}} = 0$ (1/h) corresponds to the situation when the pump is turned OFF and $u_{\text{air}} = 2.34$ (1/h) means that the variable speed pump is switched to 100% of its capacity. Simulation validation was performed using the presented multiagent control system and LabVIEW environment where the aeration process described with (1) was simulated using previously estimated u_{air} values. Other parameters were set to the following values: $DO_{\text{sat}} = 10$ (mg O_2/l), $T_{kLa} = 88$ (s), $T_0 = 27$ (s), $OUR(t)$ was set at a constant value during the simulation equal to 2.4 (mg O_2/l h). Process simulation was using LabVIEW OPC UA Toolkit to communicate with the created MAS.

Firstly, two control methods were analyzed and compared through simulation. In the first approach, the simulated process was controlled by a conventional BBPC algorithm. For the next approach, control was performed by the multiagent system that used both the ON-OFF pump and the variable speed pump. For MAS, control of the ON-OFF pump was realized using the BBPC algorithm, and a variable speed pump was used in the last aeration cycle to eliminate the remaining error as described in Section 2.1. The results of the simulation are presented in Fig. 8.

Obtained results show that for both approaches $SP_{\min}$ value was reached precisely, but the precision of reaching the $SP_{\max}$ value varies between them. For *BBPC* significant overshoots over the $SP_{\max}$ boundary are visible (around 0.1 (mgO_2/l)),

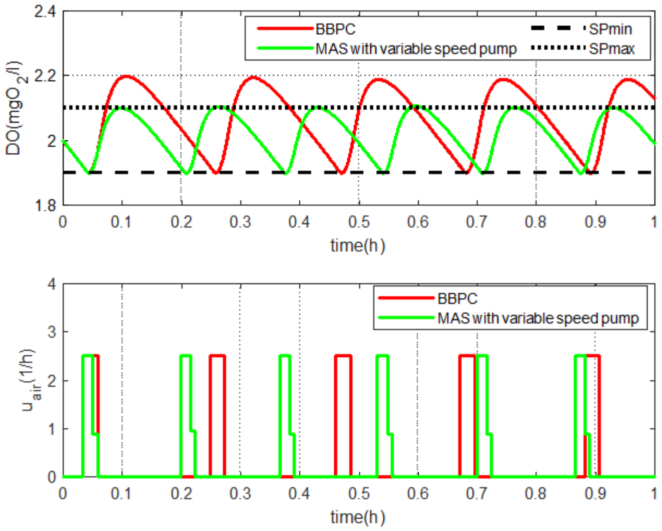

Fig. 8. Simulation results of DO concentration control using different control systems

this behavior is expected and matches the results and analysis presented in [24]. MAS with variable speed pump control can achieve significantly better precision in reaching $SP_{\max}$. Differences between peaks of DO values and $SP_{\max}$ for the second approach were not larger than 0.006 (mgO$_2$/l). CV values in Fig. 8 illustrate how MAS control with variable speed pump functions. A drop of the u_{air} from 2.52 (1/h) to other non-zero values indicates the moment when MAS decides to turn on the variable speed pump tuned to precisely reach the $SP_{\max}$ boundary. For standalone BBPC u_{air} value can only be switched between 0 (1/h) and 2.52 (1/h), because of that, during the last aeration cycle u_{air} must stay on 2.52 (1/h) or 0 (1/h) which usually results in either noticeable overshoot or maximum value not reaching $SP_{\max}$.

During the next simulation, MAS reputation and reconfiguration mechanisms were verified. The simulation was started using the MAS system with a variable speed pump, but the parameters of this pump model were intentionally set incorrectly. Later, agents with the correct variable speed pump model parameters were added to MAS. Results of the simulation presented in Fig. 9 visualize the influence of evaluation subsystem actions on process control and MAS ability for online reconfigurations. For the first two aeration periods, MAS uses an incorrectly modelled variable speed pump. This can be noticed by the visible difference between the maximum DO declaration and the real maximum DO value. Because of that, in each of these aeration periods, the reputation of agents responsible for variable speed pump control is dropped by one grade down to an *unreliable* grade. After two aeration periods, control offers graded as *unreliable* from agents controlling variable speed pump are ignored and as a result BBPC algorithm based only on the ON-OFF pump is realized. That results in visible gaps between $SP_{\max}$ and maximum DO value but expected maxima now match measured values, proving that agents responsible for this control are *reliable* and their offers should be used over *unreliable* ones. The moment when new agents were added to MAS is indicated by the vertical line. Newly added agents were seamlessly included in the MAS control flow and the next aeration period their control offer was already in use resulting in immediate improvement in the precision of reaching $SP_{\max}$ value. This proves that the MAS can adopt new agents to its structure during runtime. Therefore, it is possible to provide online improvements and corrections to MAS and perform its reconfigurations.

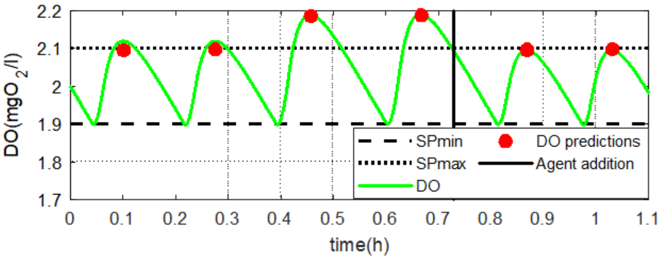

Fig. 9. Simulation verification of reputation mechanism and MAS reconfiguration

Lastly, the MAS ability to diagnose its faults was verified. For this experiment, the agent responsible for $OUR(t)$ calculations was modified to impair its communication capabilities and thus become a fault point of the MAS. When the control system was started, the agent responsible for diagnostic capabilities was requested to diagnose the state of CV calculation. The agent presented every event that might indicate a fault in the system. Such events occurred during the CV test, control-forward test (in the MAS implementation control-forward is the name of the selected control), and lastly in the $OUR(t)$ test where the nonresponding agent was detected.

7. EXPERIMENTAL RESULTS

Experimental validation was made using a real activated sludge laboratory setup presented in Figs. 1 and 2 with a preliminary off-line estimation of the parameters of the model (1) based on measurement data. Assuming constant saturation concentration $DO_{\text{sat}} = 10$ (mg O$_2$/l), parameter values were estimated as follows: $T_{kLa} = 50$ (s), $T_0 = 14$ (s), and $u_A = 2.52$ (1/h). The first experiment was focused on the comparison of the presented multiagent control system with the BBPC algorithm. During experimental measurements, the $OUR(t)$ value was estimated at $OUR(t) = 2.7$ (mg O$_2$/l h).

As can be seen in Fig. 10, overshoots generated by the BBPC algorithm are visibly smaller than ones obtained through simulation. Even though, in comparison to simulation results, overshoots during BBPC control for these conditions are significantly smaller, the proposed agent-based solution is still able to outperform the former controller and reach the $SP_{\max}$ boundary precisely during every aeration cycle by properly adjusting the CV value for the last aeration cycle in the sequence. CV graph in Fig. 10 shows that for some aeration cycles, there was no need to use the variable speed pump to achieve high control precision. MAS recognized these cases and used only the ON-OFF pump.

As it is not possible to prepare an organic substrate batch that would ensure identical $OUR(t)$ for each experiment, slight differences in $OUR(t)$ values can be noticed through descending *DO* slopes not being parallel to each other in Fig. 10. However, the influence of those differences on final results are negligible.

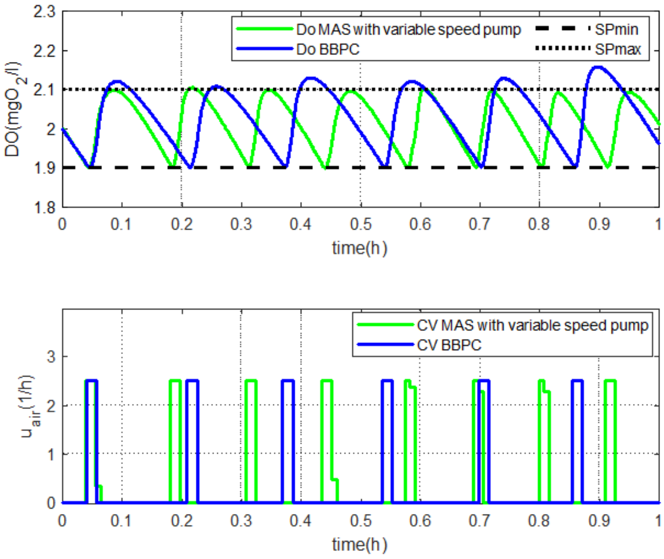

Fig. 10. *DO* concentration control for $OUR = 2.7$ (mg O_2/lh) (above), corresponding *CV* values (below)

The goal of the second experiment was to verify the implemented MAS response to the addition and removal of new controllers. The experiment was divided into three phases. Firstly, agents responsible for variable speed pump control were absent, forcing the control system to use only an agent-based version of the BBPC algorithm. After some time, agents that can control the variable speed pump were added to the system. Then after 1.5 (h), those agents were removed, forcing the system to once again rely only on the BBPC algorithm.

Results in Fig. 11 show that the control system can switch to more precise control the moment it becomes available. When the capabilities of the control system are limited by removing specific agents, the MAS is also able to immediately switch to less precise control without any disruptions.

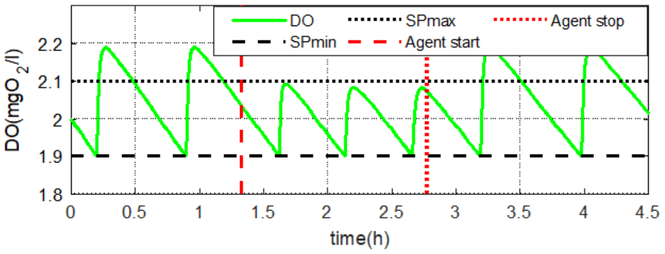

Fig. 11. DO concentration control during addition and removal of supplementary controller agent

8. CONCLUDING REMARKS

This paper presented the concept of a novel multiagent, multicontroller system for model-based predictive control of continuous processes.

The proposed MAS design was verified in a practical implementation of a boundary-based control of DO concentration in a biological reactor aerated by ON-OFF and variable speed aeration pumps. For that implementation, a novel, agent-based control algorithm was proposed and tested in both simulation environments as well as in real activated sludge laboratory setups. A practical comparison of the novel control system with the BBPC algorithm proved the superiority of MAS, which was able to reach set boundaries, especially the $SP_{\max}$ boundary with significantly better precision. Further experiments additionally verified that parts of the presented MAS can be added or removed from the system at runtime with a seamless switch in control. MAS is also able to reconfigure its structure and switch between available control algorithms based on a reputation mechanism that compares agents' declarations with real measurements and assigns reputation grades based on that knowledge. Additionally, a fault detection algorithm that uses knowledge about the system to detect faulty agents is proposed and verified. These attributes improve MAS maintainability, and using such MAS-based control systems might reduce downtime and costs spent for maintenance in industrial use cases.

Ontology structure for multiagent systems and suggested design of MAS for model-based predictive control of continuous processes are generic and can be applied to the control of various continuous processes using many types of control laws. The architecture of the proposed MAS was designed to address issues that impede further introduction of MAS in industrial solutions, which were pointed out in other research work. Most importantly, it should simplify the implementation of MAS by preparing agents dedicated to performing simple tasks which should be easier to develop.

Further work on this topic might focus on further development of the proposed MAS architecture, analysis of agent-based diagnostic and evaluation mechanisms, analysis regarding usage of this architecture in other industrial areas as well as further development of a family of balance-based predictive controllers and their introduction in other types of processes.

ACKNOWLEDGEMENTS

Authors were financed by the grant – subsidy for maintaining and developing the research potential in 2023, Jakub Pośpiech under grant 02/060/BKM23/0051, Witold Nocoń and Krzysztof Stebel under grant 02/060/BK_23/0043.

REFERENCES

[1] S. Karnouskos, P. Leitao, L. Ribeiro, and A.W. Colombo, "Industrial Agents as a Key Enabler for Realizing Industrial Cyber-Physical Systems: Multiagent Systems Entering Industry 4.0," *IEEE Ind. Electron. Mag.*, vol. 14, no. 3, pp. 18–32, Sept. 2020, doi: 10.1109/MIE.2019.2962225.

[2] L. Ding, Q. Han, X. Ge, and X. Zhang, "An Overview of Recent Advances in Event-Triggered Consensus of Multiagent Systems," *IEEE Trans. Cybern.*, vol. 48, no. 4, pp. 1110–1123, April 2018, doi: 10.1109/TCYB.2017.2771560.

[3] D. Huang, Y. Chen, D. Meng, and P. Sun, "Adaptive Iterative Learning Control for High-Speed Train: A Multi-Agent Approach," *IEEE Trans. Syst., Man Cybern.-Syst.*, vol. 51, no. 7, pp. 4067–4077, July 2021, doi: 10.1109/TSMC.2019.2931289.

[4] P. Qaderi-Baban, M.B. Menhaj, M. Dosaranian-Moghadam and A. Fakharian, "Intelligent multi-agent system for DC microgrid energy coordination control," *Bull. Pol. Acad. Sci. Tech. Sci.*, vol. 67, no. 4, pp. 741–748, Aug. 2019, doi: 10.24425/bpasts.2019.130183.

[5] D. Seredyński *et al.*, "Agent-based approach to the design of a multimodal interface for cyber-security event visualisation control," *Bull. Pol. Acad. Sci. Tech. Sci.*, vol. 68, no. 5, pp. 1187–1205, Oct. 2020, doi: 10.24425/bpasts.2020.134662.

[6] S. Bussmann and K. Schild, "An agent-based approach to the control of flexible production systems," in *ETFA 2001. 8th International Conference on Emerging Technologies and Factory Automation. Proceedings (Cat. No.01TH8597)*, Antibes-Juan les Pins, France, 2001, vol.2, pp. 481–488, doi: 10.1109/ETFA.2001.997722.

[7] T. Arai, Y. Aiyama, Y. Maeda, M. Sugi, and J. Ota, "Agile Assembly System by 'Plug and Produce," *CIRP Annals*, vol. 49, no. 1, pp. 1–4, Jan. 2000, doi: 10.1016/S0007-8506(07)62883-2.

[8] M. Metzger and G. Polakow, "A Survey on Applications of Agent Technology in Industrial Process Control," *IEEE Trans. Ind. Inform.*, vol. 7, no. 4, pp. 570–581, Nov. 2011, doi: 10.1109/TII.2011.2166781.

[9] A.J.N. van Breemen and T.J.A. de Vries, "Design and implementation of a room thermostat using an agent-based approach", *Control Eng. Practice*, vol. 9, no. 3, pp. 233–248, 2001, doi: 10.1016/S0967-0661(00)00111-8.

[10] M. Francisco, Y. Mezquita, S. Revollar, P. Vega, and Juan F. De Paz, "Multi-agent distributed model predictive control with fuzzy negotiation," *Expert Syst. Appl.*, vol. 129, pp. 68–83, Sept. 2019, doi: 10.1016/j.eswa.2019.03.056.

[11] Y.N. Guo, J. Cheng, D. Gong, and J. Zhang, "A Novel Multi-agent Based Complex Process Control System and Its Application," in *Intelligent Control and Automation: International Conference on Intelligent Computing, ICIC 2006*, Kunming, China, pp. 319–330. doi: 10.1007/978-3-540-37256-1_39.

[12] S. Videau, C. Bernon, P. Glize, and J.L. Uribelarrea, "Controlling Bioprocesses Using Cooperative Self-organizing Agents," in *Advances on Practical Applications of Agents and Multiagent Systems*, Berlin, Germany, 2011, pp. 141–150. doi: 10.1007/978-3-642-19875-5_19.

[13] S. Karnouskos and P. Leitão, "Key Contributing Factors to the Acceptance of Agents in Industrial Environments," *IEEE Trans. Ind. Inform.*, vol. 13, no. 2, pp. 696–703, Apr. 2017, doi: 10.1109/TII.2016.2607148.

[14] V. Mařík and J. Lažanský, "Industrial applications of agent technologies", *Control Eng. Practice*, vol. 15, no. 11, pp. 1364–1380, 2007, doi: 10.1016/j.conengprac.2006.10.001.

[15] G. Polaków, "JADE environment performance evaluation for agent-based continuous process control algorithm," in *2016 21st International Conference on Methods and Models in Automation and Robotics (MMAR)*, Międzyzdroje, Poland, 2016, pp. 571–576, doi: 10.1109/MMAR.2016.7575199.

[16] G. Polaków, P. Laszczyk, and M. Metzger, "Agent-based approach to model-based dynamically reconfigurable control algorithm," in *2015 20th International Conference on Process Control (PC)*, Strbske Pleso, Slovakia, 2015, pp. 375–380, doi: 10.1109/PC.2015.7169992.

[17] D. Choiński, W. Nocoń, and M. Metzger, "Multi-Agent System for Hierarchical Control with Self-organising Database," in *Agent and Multi-Agent Systems: Technologies and Applications-KES-AMSTA 2007*, Wroclaw, Poland, 2007, pp. 655–664. doi: 10.1007/978-3-540-72830-6_68.

[18] M. Senik and D. Choiński, "Distributed control systems integration and management with an ontology-based multi-agent system," *Bull. Pol. Acad. Sci. Tech. Sci.,* vol. 66, no 5, pp. 613–620, Oct. 2018, doi: 10.24425/bpas.2018.124277.

[19] L. Ribeiro, S. Karnouskos, P. Leitão, J. Barbosa, and M. Hochwallner, "Performance Assessment Of The Integration Between Industrial Agents And Low-Level Automation Functions," in *2018 IEEE 16th International Conference on Industrial Informatics (INDIN)*, Porto, Portugal, 2018, pp. 121–126, doi: 10.1109/INDIN.2018.8471927.

[20] G. Tchobanoglous, F.L. Burton, and H.D. Stensel, *WastewaterEngineering: Treatment and Reuse*. New York, NY, USA: McGraw-Hill, 2003.

[21] P. Łaszczyk, "Predictive functional control of dissolved oxygen with online estimation of oxygene uptake rate." in *Proceedings of the 20th International Conference on Methods and Models in Automation and Robotics (MMAR)*, Miedzyzdroje, Poland, 2015, pp. 602–607. doi: 10.1109/MMAR.2015.7283943.

[22] R. Piotrowski, H. Sawicki, and K. Żuk, "Novel hierarchical nonlinear control algorithm to improve dissolved oxygen control in biological WWTP," *J. Process Control*, vol 105, pp. 78–87, Sep. 2021, doi: 10.1016/j.jprocont.2021.07.009.

[23] R. Piotrowski, M.A. Brdys, K. Konarczak, K. Duzinkiewicz, and W. Chotkowski, "Hierarchical dissolved oxygen control for activated sludge processes," *Control Eng. Practice*, vol. 16, no. 1, pp. 114–131, Jan. 2008, doi: 10.1016/j.conengprac.2007.04.005.

[24] K. Stebel, J. Pospiech, W. Nocon, J. Czeczot and P. Skupin, "Boundary-Based Predictive Controller and its Application to Control of Dissolved Oxygen Concentration in Activated Sludge Bioreactor," *IEEE Trans. Ind. Electron.*, vol. 69, no. 10, pp. 10541–10551, Oct. 2022, doi: 10.1109/TIE.2021.3123629.

[25] M. Sànchez, U. Cortés, J.Lafuente, I.R. Roda and M. Poch, "DAI-DEPUR: an integrated and distributed architecture for wastewater treatment plants supervision," *Artif. Intell. Eng.*, vol. 10, no. 3, pp. 275–285, Aug. 1996, doi: 10.1016/0954-1810(96)00004-0.

[26] J. Pospiech, "Multi-Agent System for Closed Loop Model-Based Control of Dissolved Oxygen Concentration," in *2021 25th International Conference on Methods and Models in Automation and Robotics (MMAR)*, Międzyzdroje, Poland, 2021, pp. 145–149, doi: 10.1109/MMAR49549.2021.9528445.

[27] M. Czyżniewski, R. Łangowski, and R. Piotrowski, "Respiration rate estimation using non-linear observers in application to wastewater treatment plant," *J. Process Control*, vol 124, pp. 70–82, Apr. 2023, doi: 10.1016/j.jprocont.2023.02.008.