Computer Science • 15 (4) 2014 http://dx.doi.org/10.7494/csci.2014.15.4.411

Gabriel Rojek

AGENTS MODELING EXPERIENCE APPLIED TO CONTROL OF SEMI-CONTINUOUS PRODUCTION PROCESS

Abstract

The lack of proper analytical models of some production processes prevents us from obtaining proper values of process parameters by simply computing optimal values. Possible solutions of control problems in such areas of industrial processes can be found using certain methods from the domain of artificial intelligence: neural networks, fuzzy logic, expert systems, or evolutionary algorithms. Presented in this work, a solution to such a control problem is an alternative approach that combines control of the industrial process with learning based on production results. By formulating the main assumptions of the proposed methodology, decision processes of a human operator using his experience are taken into consideration. The researched model of using and gathering experience of human beings is designed with the contribution of agent technology. The presented solution of the control problem coincides with case-based reasoning (CBR) methodology.

Keywords agent technology, industrial control, case-based reasoning

1. Introduction

The main genesis of the work presented here is the observation that many decision problems in the real world are difficult to solve with known computational techniques, but are well resolved by humans using experience, intuition, and other attributes of personal skills. According to this motive, the scope of the research presented here is an analysis of processes which occur in the human mind, which enables us to make decisions in situations characterized by the unknown relation between the undertaken action and its result. This analysis leads to a model of human decision making that complies to specifics of human thinking about some happenings, events, or incidents being some autonomous and distributed episodes referenced in the mind. The perceived autonomy of episodes is the reason for our application of agent technology at the design and implementation of a computer system using the researched model of human decision making. As presented in [16, 15], the main characteristic of an intelligent agent is autonomy, which indicates usefulness of a multi-agent approach in problems of a distributed nature.

The control of an industrial process is the background for our presented analysis of the decision-making model. The oxidizing roasting process of sulphide zinc concentrates is chosen as an exemplary industrial process. This industrial process is one of a group that consists of processes being difficult to control with known computational techniques. The nature of the chosen process prevents us from obtaining proper values of parameters by computing from determined dependences in the form of mathematical equations. Proper rules of its control are also difficult to formalize, which handicaps the building of a knowledge base, and as a consequence, is difficult at constructing an expert system. Presented in [13], the proposition of our solution uses a neural net in order to predict results of hypothetical control. This neural net takes, as input, values of all parameters of production without taking into account the big difference of frequency of parameters measuring (once a second versus once a batch – a day period). Such an approach leads to adding missing data with the use of interpolation techniques, which can be a source of faults and errors in the case of processes with nonuniform frequency of signal measuring. The oxidizing roasting process of sulphide zinc concentrates is an example of a production that is organized into batches. But every batch is continuously controlled, so it is one of the semi-continuous (or semi-batch) processes [4]. Interpolation of parameters, which are measured at the time of different production batches, leads to adding nonexistent and perhaps distorted values of parameters, which is the main disadvantage of the solutions presented in [13].

Due to the problems mentioned with automatic control of the oxidizing roasting process of sulphide zinc concentrates, this control is still performed by human operators in known industrial factories. The most important factor of the work of an operator is his experience, as stated by representatives of industrial plants. These observations leads to the conclusion that this industrial process is a desirable example for modeling of human decision making based on experience. In the next sections of the article, the following issues are presented: the afore-mentioned industrial process; an analysis of the experience model and case-based reasoning methodology that coincides with the model of experience; the design of the agent system modeling experience; remarks on its implementation; and obtained results.

1.1. Goal and motivations

The main motivation of the research presented here is an effort to design a methodology that can be a universal method of resolving problems concerning situations characterized by an a-priori unknown relationship between an undertaken action and its result. Design of the so-characterized methodology and an investigation of its application possibilities are the main goals of the research presented here. This methodology can lay a groundwork for the design of decision-support systems in many domains of the application of computer systems: customer service, e-commerce, help-desk systems, scheduling of production or control, and optimization of an industrial process. The decision support system obtained by the application of the research methodology should substitute a human employee, whose goal is to resolve problems too difficult to solve with other known computational techniques. According to this conclusion, the researched methodology is oriented on an analysis of human processes occurring in the mind of a worker performing goals by using and gathering experience. This analysis can lead to the formalization of the analyzed processes using the formalism of case-based reasoning (CBR), as presented in section 4.

The presented research also has a minor motivation concerning the control of the semi-continuous industrial process. In the case of unknown analytical models, industrial processes are often controlled with the use of an artificial neuron net. The neuron net usually predicts parameters of production on the base of all measured signals without taking into account various frequencies of signals sampling (as presented in [13]). Such a proposition can lead to faults according processes with a non-uniform frequency of signal sampling. Specifically, semi-continuous processes are controlled on the base of signals, which can be measured just once per batch or in a continuous way during a single period of a batch. The presented methodology of using and gathering of experience applied to a semi-continuous industrial process should be a pattern for engineers, who cope with the design of a control system of a semi-continuous industrial process.

Presented in section 5, the design of the system with the application of the research methodology uses agent technology. Motivation of the use of the agent technology is oriented on obtaining the structure of the system, in which the main parts (components) are related to their denotation in the research conception. The presented idea of design assumes that each experience item (referenced in CBR methodology as a case) is represented by an individual agent. This idea is supported by the autonomy of a case and is a widely-used application of the agent approach to computational intelligence techniques. The presented conception of design is similar to the conception of an evolutionary multi-agent system, where every potential solution of a resolved problem is represented by an individual agent, as presented in [8, 6].

Application of this conception to the work presented here concerning methodology of human decision making enables an easy addition of mechanisms, which allows for the management of population – reducing the agent population in the case of an application domain that is related to problems with a too-large quantity of experience items.

2. The oxidizing roasting process of sulphide zinc concentrates

The production of zinc from sulfide concentrates is realized in the industry mainly through hydro-metallurgical processes. The first stage of this production is the transformation of metal sulfides to oxides (called the roasting process) and is carried out in fluidized bed furnaces. As the result of roasting the zinc sulfide concentrates, zinc oxide is obtained with the maximum content of sulphide sulfur near 1.0%. During the roasting process, the aim is to obtain a minimal amount of sulphide sulfur in the composition of the product.

The oxidizing roasting of zinc sulfide concentrates is in the sphere of our interest due to the problems with automatic control (as mentioned in the introduction). There is a lack of analytical models of this process that would enable us to compute optimal values of process parameters. This process is also sensitive to different parameters, which are measured with very different frequencies – some parameters are measured with frequency near a second or a minute, but others are measured only once per production batch (a day). This remark is the grounds for classifying this process as a semi-continuous or semi-batch process – some parameters are constant during a batch, but other parameters are measured or set frequently during a single batch.

2.1. Input signals

The basic analysis and design of a computer control system requires the identification of input signals that influence the controlled industrial system. All input signals that are measured or set can be classified into one of three main groups: independent signals, controllable signals, or dependent signals [13]. From the point of view of measuring frequency, these signals can be classified as constant or changing during a batch.

2.1.1. Independent signals

All parameters that cannot be modified or changed during the production cycle are independent signals. These signals are independent of other process parameters, so it is impossible to change their values in a direct or indirect way. Independent signals usually have an influence on the production results obtained. In the case of the analyzed oxidizing roasting process, all independent signals I are measured once for a whole batch and indicate the chemical composition of raw materials:

$$I = [i_S, i_{Zn}, i_{Pb}, i_{Fe}] \tag{1}$$

where i_S is measured concentration of sulfur, i_{Zn} is measured concentration of zinc, i_{Pb} is measured concentration of lead and i_{Fe} is measured concentration of iron. Independent signals I are constant for a whole day of production (a batch period) – these signals are measured just once at the start of each batch period.

2.1.2. Dependent signals

All measured parameters that cannot be directly modified and which are changing during a batch are dependent signals. In the case of the analyzed oxidizing roasting process, dependent parameters are measured with a frequency near a second or a minute, so dependent parameters X measured for a whole batch consist of many single measurements:

$$X = [X_1, X_2, \ldots, X_N] \tag{2}$$

where N is the number of measurements for a single batch period. X_n is a result of one measurement that consists of many parameters according temperature, pressure, and concentration of SO_2:

$$X_n = [x_n^{t1}, x_n^{t2}, x_n^{t3}, x_n^{t4}, x_n^{t5}, x_n^{t6}, x_n^{c}, x_n^{p1}, x_n^{p2}] \tag{3}$$

where $x_n^{t1}, x_n^{t2}, \ldots, x_n^{t6}$ are temperatures measured in different places of furnace, x_n^c is concentration of SO_2, x_n^{p1} and x_n^{p2} are pressures measured in different places in the furnace.

The value of every dependent signal is a hypothetical function of other production parameters and a possible time delay. This function is unknown in the case of the analyzed oxidizing roasting process.

2.1.3. Controllable signals

All directly set, changed, or updated parameters are controllable signals. Only these signals can be directly controlled in order to obtain products characterized by the desired properties, so the controllable signals are decision-making variables. Controllable signals U are a set with the same frequency as dependent signals X. Controllable parameters U set and saved for a whole batch consist of many single settings:

$$U = [U_1, U_2, \ldots, U_N] \tag{4}$$

where N is the number of settings for a single batch period. U_n is a single setting consisting of:

$$U_n = [u_n^{m1}, u_n^{m2}, u_n^{a1}, u_n^{a2}, u_n^{p1}, u_n^{p2}, u_n^{f}] \tag{5}$$

where u_n^{m1} and u_n^{m2} are amounts of feed material, u_n^{a1} and u_n^{a2} are air flows, u_n^{p1} and u_n^{p2} are air pressure and u_n^f is fan speed.

2.2. Evaluation of products

The aim of the industrial process control is to achieve products that are characterized by optimal properties. In the case of the analyzed oxidizing roasting process, the goal is to obtain a minimal concentration of sulphide sulfur in the roasted products. This concentration is usually measured several times per batch. Because the number of these measurements differ, and these measures are not related to specific periods of controllable or dependent signal measures, the quality evaluation for a batch period Q is a single parameter related to the average concentration of sulphide sulfur in the roasted products.

A formal definition of the decision problem can be presented in the form of statement: how to choose values of controllable signals U knowing the values of independent I and dependent signals X in order to obtain the best possible quality evaluation Q. In other words, the quality evaluation is a hypothetical function of all signals:

$$Q = f(U, I, X) \tag{6}$$

However, during the process control, only controllable signals U can be directly changed. Controllable signals U should be adjusted to the measured values of all other signals in order to maximize quality criterion Q. Such a statement indicates that the final evaluation of quality can be influenced by the composition of raw materials (independent signals I), so the desirable quality for different input materials can differ. It is also assumed that the way of control of the analyzed process can be different for different values of independent signals I in order to obtain the desired quality Q. From the point of view of optimization, the oxidizing roasting process is a nonlinear and multidimensional process.

2.3. Run of the process

A run of the oxidizing roasting process of sulphide zinc concentrates is presented in Figure 1. At the beginning of the current batch, independent signals I are measured that indicate the chemical composition of raw materials. Those signals are measured only once per a batch, before the start of the production. The production of a current batch is controlled in a continuous way – dependent signals X are measured and controllable signals U are set with a frequency near one second. After the production of a current batch is stopped, all quality measures are known and the quality measure Q is possible to calculate. The quality measure Q is equal to the average concentration of sulphide sulfur in the roasted products during the current batch.

The oxidizing roasting process of sulphide zinc concentrates is a semi-continuous (or semi-batch) processes [4], because the production is organized into batches and some signals are measured only once per batch. However, the process of production during a batch is continuously controlled. Controllable signals U are set with a frequency that is sufficient to ensure continuous control of the analyzed process.

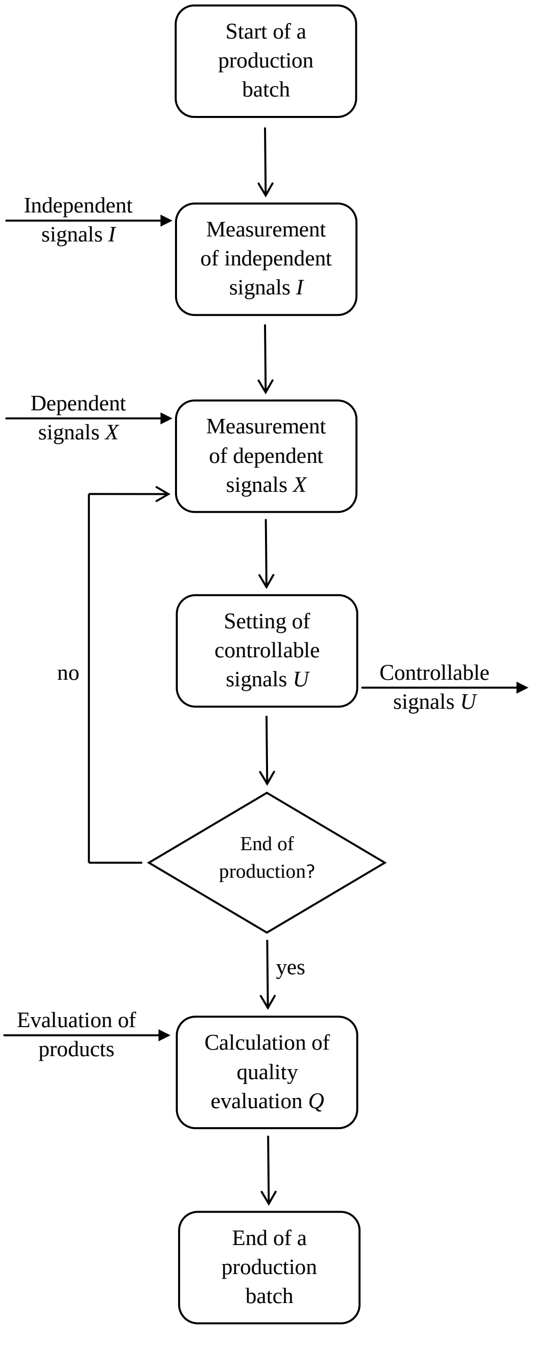

Figure 1. Run of the oxidizing roasting process of sulphide zinc concentrates.

3. Human decision making

Having a goal to follow the decision processes which take place in the mind of an operator of the industrial process (presented in the previous section), we should analyze his every day work. At the beginning of a batch (a production day), the operator knows the values of independent signals I; this means he knows the chemical composition of the raw materials used for production. The operator assumes that this chemical composition is constant for a whole current batch due to the frequency of independent signals measured, which is done only once per batch. Before the start of process control, the operator should decide how to control this process. This means how to set present values of controllable signals U knowing the currently-measured values of dependent signals X. This decision is based on his experience. It is assumed that setting controllable values U is done with the same frequency as reading the values of dependent parameters X (e.g., several times per minute).

The assumed model of the work of an operator indicates experience is his source of knowledge according to the control of the process. This experience contains many episodes from past production, which are referred to as cases. Each case in the experience contains information concerning:

- description of solved problem – how to control the industrial process assuming known independent signals I (chemical concentration of raw materials),
- description how this problem was solved – the means, how the process was controlled (the way of controllable signals U setting taking into consideration presently measured values of dependent signals X),
- description how production was evaluated in the form of measured concentration of sulphide sulfur in the roasted products (referenced as evaluation criterion Q).

As mentioned in the first paragraph of this section, the operator should decide how to control this process with the use of his experience before the start of process control. Referring to the afore-mentioned influence of independent signals I on obtained quality, the human operator first searches his experience for cases that concern the same (or similar) problem and then chooses one that brings the best effect described by the value of evaluation criterion Q. After such a selection, the human operator is trying to control the process in the way he did it in the past (as remembered by him as the solution in the chosen case). This means the human operator is trying to follow the way he has set values of controllable signals U knowing the measured values of dependent signals X (noticed at the chosen case of experience). This stage of experience utilization goes on until the end of the current production period.

When the current production is ended, the human operator obtains information concerning the evaluation of the made products. So, this time it is possible to update his experience with the case that concerns the production period that just ended in order to use this experience in the future. This phenomenon enables learning on the base of past-made production control and its results.

The presented model of using and gathering experience coincides with case-based reasoning methodology presented in the next section. The assumptions presented above also indicate a distribution of cases that are autonomous items of human experience. The autonomy of experienced items is the main reason for using the agent technology (as presented in [16, 15]) at the design and implementation of this model, which is shown further in this work.

4. Case-based reasoning

From the most general point of view, case-based reasoning (CBR) methodology relies on experiences made in the past during the solving of concrete problem situations, instead of using only general knowledge related to a problem domain [1]. The main conception of CBR methodology is focusing on the solving of a current problem by reusing previous situations similar to the current problem. A CBR decision system uses a collection of past-made and stored experience items, called past cases, or cases. Each time a new problem has to be solved, first a past case relevant (similar) to the present problem is selected, and next, this selected case has to be adopted to the current situation. When the current problem is solved, the new experience is retained in order to be available for future reasoning concerning future problem situations. The retention of made experiences enables incremental learning that is closely related to problem solving and its results.

As presented in [5], application areas of a CBR approach are help-desk and customer service, advisory systems in e-commerce, knowledge, and experience management. There are also known medical applications, applications in image processing, applications in law, technical diagnosis, design, planning, and human entertainment (computer games, music).

4.1. Cases as experience items

The main assumption of case-based reasoning is the notion of a case as a representation of an experience item, also referred to as an episode. An episode covers a problem (a problem situation) that was resolved with a solution, so a case is usually described as par $(\mathrm{problem}, \mathrm{solution})$, as can be found in [2]. The existence of a particular case denoted as $c_i = (p_i, s_i)$ is related to a concrete episode in the past that concerns a problem denoted as p_i, which was resolved with a solution denoted as s_i.

Taking into account the previously-discussed decision making of a human operator, each case of an experience item should also relate to the evaluation of the effects of a particular solution applied to a particular problem. This remark is related specifically to the control of an industrial process – a human operator tends to choose experience items that brought about the best effects in the past. In consequence, concerning control of an industrial process, a case should be described as a triple $(\mathrm{problem}, \mathrm{solution}, \mathrm{effects})$, as can be found in [10]. As mentioned earlier, the problem is related to control of the analyzed industrial process that should adjust to specific values of known independent signals I. The solution is the way the process was controlled, which is consistent to the manner of controllable signals U setting, taking into consideration the presently-measured values of dependent signals X. The effects are related to the evaluation of products Q in the form of measured concentration of sulphide sulfur in the roasted products. These remarks lead to definition of a case.

Definition 1. A case (as an item of experience used by a human operator of the analyzed industrial process) is defined as a triple:

$$C = (I, (X, U), Q)$$

where I are independent signals measured for a particular batch, (X, U) are dependent and controllable signals determining control of the process for a batch, and Q are effects of production within the period of a batch in the form of quality measure.

A single case represents one experience item of a human operator. Taking into account this definition, a single case is related to one particular batch of production. A reasoning system should use knowledge related to many batches of production (a set of cases), which leads to the following definition.

Definition 2. A case base is a finite set of cases:

$$\Delta = \{C^1, C^2, \ldots, C^N\}$$

where C^i is a case and N is the number of cases in the case base of a reasoning system.

4.2. The CBR cycle

The CBR cycle is the main and common point of all CBR systems, despite the different domains of application or the use any additional different techniques (e.g., induction, fuzzy logic or database technology) as presented in [14]. The CBR cycle is the main algorithm performed by every CBR system in order to solve new problems and supplement the case base with experiences made during system functioning. The CBR cycle is an algorithm that consists of four steps – sequential processes, which are called also phases [1]:

1. Retrieve the most similar case or cases
2. Reuse the information and knowledge in that case to solve the problem
3. Revise the proposed solution
4. Retain the parts of this experience likely to be useful for future problem solving

The CBR cycle starts when a new problem (also called a current problem) has to be solved. In the first step, the most-similar case or cases to the new problem are retrieved by browsing the case base consisting of all past-made experience items (cases). In the second step, a solution to the current problem is proposed – the retrieved case is combined with the current problem; in other words, the solution contained in the retrieved case is reused to solve the current problem. In the third step, the proposed solution is tested for success, usually by being applied to the real world environment or evaluated by a teacher. In this step, the proposed solution can be also repaired (if necessary and possible). In the fourth step, the current problem and the revised solution are retained for future use. In this step, the case base is updated by a new learned case, or by modification of some existing cases.

4.2.1. Retrieve phase

The main task of the first retrieve phase is to select one or more past cases from the case base that are relevant to the specified current problem. By notion 'relevant case' a past case is meant that is usually similar to the current problem, what is performed by searching of k-nearest-neighbor considering a specific similarity measure (e.g., inverse Euclidean or Hamming distance). In the analyzed domain of the industrial control, the evaluation of searched past cases also should be taking into consideration – it is highly desirable to take a pattern from a solution that was well-evaluated in the past. So, modeling the decision making of a human operator, the main goal of the retrieve phase is to find a past case that concerns a problem similar to the current problem and contained in this case solution was well-evaluated in the past. It is proposed to choose first a small number of past cases representing similar problems (using k-nearest neighbor algorithm), and next, to select among them only the one that has the best evaluation. Finally, the task of the retrieve phase can be done in two steps:

1. Choose a number of cases from the case base with the highest similarity rate; the similarity is measured as the inverse Euclidean distance between values of independent signals for the current problem and the solved problems included in the case base,
2. Select among chosen cases only one that is evaluated to have the most-desirable value of quality measure.

The notion of similarity plays an important role in CBR, especially in the retrieve phase where similarity is the base for choosing relevant cases. Similarity is usually formalized as a function $sim : P \times P \to [0,1]$, which compares descriptions of two problems from P and produces a similarity assessment as a real value from $[0,1]$ [5]. Taking into consideration control of the analyzed industrial process, the problem is described by independent signals $I = [i_S, i_{Zn}, i_{Pb}, i_{Fe}]$ measured for a particular batch. In consequence, the inverse Euclidean distance is proposed as the similarity measure between two independent signals I^x and I^y:

$$sim(I^x, I^y) = \frac{1}{1 + \sqrt{(i^x_S - i^y_S)^2 + (i^x_{Zn} - i^y_{Zn})^2 + (i^x_{Pb} - i^y_{Pb})^2 + (i^x_{Fe} - i^y_{Fe})^2}} \quad (7)$$

Algorithm of the retrieve phase

Step 1. Input a new problem specified by values of independent signals $I^p = [i^p_S, i^p_{Zn}, i^p_{Pb}, i^p_{Fe}]$.

Step 2. For every case $C^i = (I^i, (X^i, U^i), Q^i)$ in the case base $\Delta = \{C^1, C^2, \ldots, C^N\}$ compute the similarity $sim(I^p, I^i)$.

Step 3. On the base of the computed similarity choose a set of cases, which are most similar to the new problem: $\Delta^R = \{C^1, C^2, \ldots, C^{N_R}\}$, where N_R is constant number of chosen elements.

Step 4. Select among the chosen cases from the set Δ^R only one case $C^R = (I^R, (X^R, U^R), Q^R)$, which is evaluated with the most optimal value of quality measure represented by the factor Q^R.

Step 5. Return C^R.

The above-stated algorithm relates to a part of the decision-making processes of a human operator. At the start of each batch, the operator has to decide which experience item is relevant to the current problem of production control. The human operator first searches his experience for cases that concern the same or similar problem and, next, chooses one that brought about the best effect in the past. The case C^R returned by the algorithm of the retrieve phase should relate to the experience item chosen by the operator.

4.2.2. Reuse phase

When one or several similar cases are selected in the retrieved phase, solutions contained in these cases are reused in order to solve the current problem (which takes place at the reuse process). This process can be very simple when the solution is returned unchanged as the solution for the current problem, but some application domains require an adaptation of the solution. Two main ways to adapt the retrieved past cases to the current problem exist: (1) transform the past case; (2) reuse the past method that constructed the solution, as presented in [1]. From the point of view of the presented model of decision making of a human operator, the method described as reusing the past method seems to be the most proper. The human operator remembers the method and how he controlled the process, and he uses this method for the current production batch.

Taking into consideration the result of the previous retrieve phase, the main conception of the reuse phase is to control the analyzed industrial process during the current batch in the same way, how the process was controlled in the relevant case $C^R = (I^R, (X^R, U^R), Q^R)$. It means that controllable signals U^P of present production should be set on the base of currently-measured values of dependent signals X^P with the same manner as controllable signals U^R were set according to dependent signals X^R. This problem can be resolved with approximation, which involves two steps in the reuse phase:

1. approximate how measured values of dependent signals X^R influence values of controllable signals U^R that are set,
2. use approximation in order to set present values of controllable signals U^P on the base of presently measured dependent signals X^P.

An artificial neuron net (ANN) is proposed to be used as the approximator, which coincides with remarks presented in [7]. The net should take as the input presently-measured dependent signals X^P and should return as the output values of controllable signals U^P. In further-presented tests of the proposed solution, a multilayer perceptron is used as the ANN approximating production signals. The course of algorithm of the reuse phase should, first, train the ANN, and next, use the trained ANN to predict controllable signals of present production.

Algorithm of the reuse phase

Step 1. Input the relevant case $C^R = (I^R, (X^R, U^R), Q^R)$ that is returned by the algorithm of the retrieve phase.

Step 2. Extract in the relevant case values of dependent and controllable signals: $X^R = [X_1^R, X_2^R, \ldots, X_N^R]$ and $U^R = [U_1^R, U_2^R, \ldots, U_N^R]$, where N is the number of measures or settings of a single batch.

Step 3. Create the multilayer perceptron, which as the input takes one single measure of dependent signals X_n and as the output returns one single setting of controllable signals U_n, where $1 \leq n \leq N$.

Step 4. Train the created multilayer perceptron with pars $(X_1^R, U_1^R), (X_2^R, U_2^R), \ldots, (X_N^R, U_N^R)$.

Step 5. Use the trained net in order to predict present values of controllable signals U_n^P on the base of the presently-measured dependent signals X_n^P. This step continues till the end of the present batch.

The last step of the algorithm of the reuse phase goes on until the end of the current batch, as the control of production is performed continuously during the whole period of the batch. The prediction is made with the same frequency as the frequency of dependent-signal measuring and controllable-signal setting.

4.2.3. Revise phase

At the revise phase, the solution generated at the reuse process is evaluated. In the case of an undesired evaluation, it is possible to repair the solution of the current case using domain-specific knowledge. This phase can consist of two tasks: an evaluation of the solution and a fault repair [1]. The evaluation task uses results from applying the suggested solution to the real environment, which can happen by asking a teacher or performing the task in the real world. This task is usually performed outside the CBR system and makes necessary to link the CBR system with the real world domain, which concerns the solved problem. Fault repair involves the detection of errors in the current solution and using failure explanation to modify the solution in order to prevent errors from occurring.

Taking into consideration the analyzed industrial process, the evaluation of the solution is in the form of the concentration of sulphide sulfur in the roasted products. Because the final results of this evaluation are known after the end of the present batch, a fault repair is not possible. The only outcome of the revise phase is the value of evaluation criterion Q^P, which relates to production determined by independent signals I^P, dependent signals X^P, and controllable signals U^P.

Algorithm of the revise phase

Step 1. Obtain measured values of concentration of sulphide sulphur in the roasted products.

Step 2. Return Q^P, which is equal to the average of the obtained values.

4.2.4. Retain phase

The retain phase at the CBR cycle concerns learning by retaining of the current experience, what usually occurs by simply adding the revised case to the case base [5]. Thanks to this adding, the revised solution becomes available for a reuse at future problem solving. As a result of the retain process the CBR system gains new experience due to and together with regular solving of current problems.

The algorithm of the retain phase has to create a new case, which relates to the current batch. This is possible after the end of the revise phase, when evaluation of products is known. The created case is added to the case base in order to be available for future runs of the CBR cycle.

Algorithm of the retain phase

Step 1. Input independent signals I^P, dependent signals X^P, controllable signals U^P and value of evaluation criterion Q^P for the current batch.

Step 2. Create a new case $C^P = (I^P, (X^P, U^P), Q^P)$.

Step 3. Add the case C^P to the case base Δ.

4.3. Related work concerning control of an industrial process

Presented in [12], the implementation of CBR methodology to control of combustion control of blast furnace stoves seems analogous to the formalization of experience presented above, using and gathering at control of the oxidizing roasting process of sulphide zinc concentrates. The main problems shown in [12] application of CBR methodology are related to the definition of a case, the case base, and all phases of the CBR cycle. In the retrieve phase, similar past cases to the current one are searched with methods analogous to those presented in section 4.2.1. The reuse phase is much simpler compared to the solution presented in section 4.2.2, because a case represents only one moment of time, and the solution represented in the relevant case is just taken directly as the final control decision. The method proposed in this scope indicates that the industrial process analyzed in [12] is a continuous process with uniform measuring frequency of all signals, which is big difference compared to solutions of the research presented here. In our research a case represents a batch, during which signals are continuously measured and set. Such notion of a case induces the use of approximation methods at the reuse phase, what is different to solutions presented in [12]. The revise and retain phases are presented in [12] very shortly. It is assumed that cases are evaluated later and are added to the case base for future problem solving, which is analogous to the research presented in sections 4.2.3 and 4.2.4.

5. Design of the agent system modeling experience

As presented in the third section, the basis for the decision making of a human is a set of episodes, cases from the past. This assumption is also consistent with CBR methodology, which relates to a case base as the source for decision making according a solution to a current problem. The case base consists of past cases that relate to previously noticed episodes related to the domain of use of a CBR system.

Considering the autonomy of every past case, the case base is proposed to be designed as the set of autonomous agents. Each agent in this set (called Past Episode Agent) should contain all data relating to the episode represented by him – a past case of control of the industrial process. The Past Episode Agent should also provide that data to other agents existing in the system.

The existence of Past Episode Agents is not enough to solve the problem of current control of the analyzed industrial process. A second type of agent is proposed, called the Control Agent. The Control Agent should perform all processes that are related to choosing one relevant case from the past, using this case as the pattern for the current control, and updating the experience according to results of the current production. Referring to CBR methodology, the Control Agent should perform all four phases of the CBR cycle, cooperating with all Past Episodes Agents representing the case base, as presented in Figure 2.

As presented in Figure 2, during the first retrieve phase, the Control Agent communicates with all Past Episode Agents in order to select one relevant Past Episode Agent representing the past case that is relevant to controlling the current batch. The Control Agent next moves to the reuse phase, during which it obtains data according a solution (the way of control) from the chosen relevant Past Episode Agent and uses that data at the control of production of the current batch. The reuse phase lasts until the end of the current production batch. During the third, revise phase, the Control Agent obtains an evaluation of products made in the period of the current batch. During the last retain phase, the Control Agent creates a new Past Episode Agent, which represents all information according the batch period that just ended. The created Past Episode Agent becomes a part of the case base used in next run of the system. At the end of the retain phase, the Control Agent terminates. A detailed description of the general view of agent functioning is presented in next subsections.

Taking into consideration the remarks presented above, the set of agents acting in the whole system contain:

$$Ag = \{PEA^1, PEA^2, \ldots, PEA^N, CA\}$$

where each Paste Episode Agent PEA^i represents one case C^i ($1 \leq n \leq N$, N is number of cases in present case base Δ), and the Control Agent CA performs four steps of the CBR cycle, which is presented in the previous section.

Choosing agent technology as the main paradigm of software design is supported by the many advantages of such a solution. There are many possibilities which accord

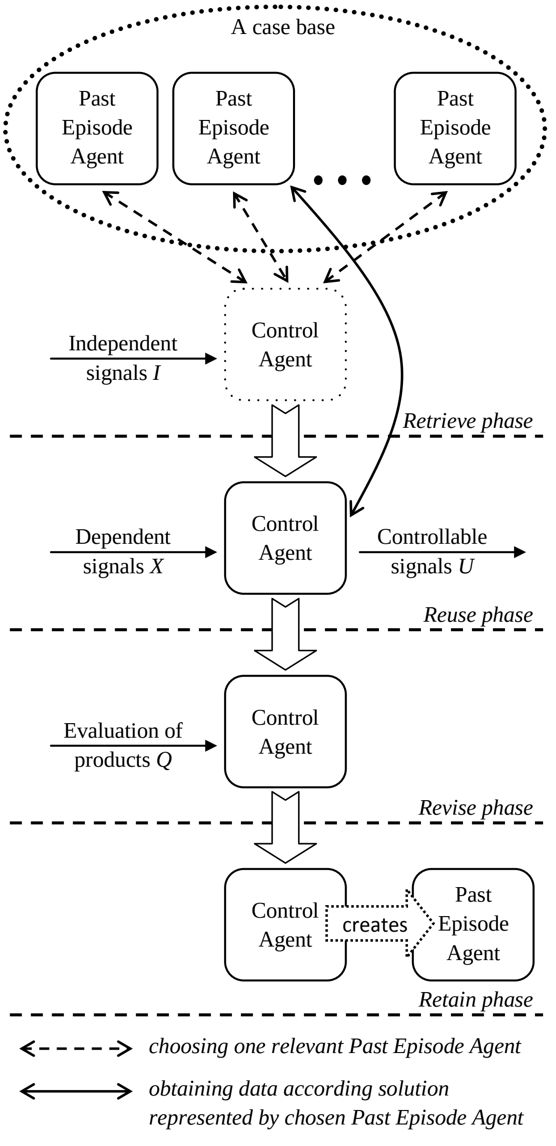

Figure 2. The Control Agent cooperating with Past Episode Agents.

future evolution of the system presented here in the base version. One such possibility can be oriented on the aggregation of similar past cases. This aggregation can be realized through adding some functionality to Past Episode Agent, which provides the formation of agents-aggregates on the base of similarities found in episodes represented by them. The agent technology was also successfully used in the design and implementation of industrial systems, as presented in e.g. [9, 11].

5.1. Past Episode Agent

As previously mentioned, a Past Episode Agent represents one past case related to an atomic part of experience concerning the domain of control of the analyzed industrial process. A case is a triple (problem, solution, effects) defined as $C = (I, (X, U), Q)$, where I are independent signals measured for a particular batch, (X, U) are dependent and controllable signals determining control of the process for this batch, and Q are effects of production in the form of quality measure within the period of this batch. The problem is specified by measured independent signals I (chemical composition of the input concentrate). The solution is the run of the control used to production characterized by specified dependent and controllable signals (X, U). The effects are represented by the quality measure Q equal to the average measure of concentration of sulphide sulfur in the products made during the batch period.

5.1.1. Data structures of a Past Episode Agent

According to the main assumption that a Past Episode Agent should represent one past case, an agent of this type has to contain data structures related to the definition of a case. Assuming a Past Episode Agent PEA^i represents a case $C^i = (I^i, (X^i, U^i), Q^i)$, it should contain data structures as follows:

- a single value of independent signal for the whole batch: $I^i = [i^i_S, i^i_{Zn}, i^i_{Pb}, i^i_{Fe}]$ related to the chemical composition of the input concentrate,
- an array of values of dependent and controllable signals registered during the considered batch: $(X^i, U^i) = ([X^i_1, X^i_2, \ldots, X^i_N], [U^i_1, U^i_2, \ldots, U^i_N])$, where N is the number of measures or settings for the batch period, X^i_n and U^i_n are specified by equations (3) and (5),
- a single value of the quality measure Q^i of products made during the considered batch.

Each Past Episode Agent (having such data structures filled with proper data) models one case of past production. The set of all Past Episode Agents corresponds to the case base Δ considering CBR methodology. The case base Δ, as the notion of CBR methodology, is a set of all past cases.

Example 1. Table 1 presents a small fragment of industrial data registered during one batch period in which the control of production was done manually by a human operator. The registered data is the source of knowledge according one past case represented by one Past Episode Agent.

The Past Episode Agent, which represents considered batch period (data presented in Table 1), should contain data structures filled with data:

- the single value of independent signal for the whole batch: $I = [34.50, 56.59, 2.54, 5.29]$,

- the array of values of dependent and controllable signals registered during the considered batch: $(X, U) = ([X_1, X_2, \dots, X_{1351}], [U_1, U_2, \dots, U_{1351}])$, where values of X_1, X_2, X_{1351}, U_1, U_2 and U_{1351} are presented in Table 1,
- the single value of the quality measure $Q = 0.687$ of products made during the considered batch.

Table 1

Fragment of industrial data registered during one batch period.

i_s	34.50								
i_{Zn}	56.59								
i_{Pb}	2.54								
i_{Fe}	5.29								
X_1	963	977	972	973	936	355	7.50	-0.10	18.22
U_1	65	65	18349	20.07	369	12.48	847		
X_2	963	977	970	972	937	357	7.50	-0.07	18.23
U_2	65	65	18342	20.12	357	12.65	847		
...	...	...	...	...	...	...	...	...	...
X_{1351}	945	959	951	952	944	361	8.39	-0.06	17.52
U_{1351}	56	57	18193	19.69	378	12.08	857		
Q	0.687								

5.1.2. Interactions of a Past Episode Agent

Every Past Episode Agent interacts with the Control Agent by solving of the current problem of production control. This interaction occurs through message passing. A Past Episode Agent PEA^i can receive messages:

- $RSVIS$ – request of sending back value of independent signal,
- $RSVDCS$ – request of sending back array of values of dependent and controllable signals,
- $RSVQ$ – request of sending back value of quality measure.

As a replay of above stated messages, a Past Episode Agent PEA^i sends back proper messages:

- VIS – information containing values of independent signals I^i (as a replay to a $RSVIS$ message),
- $VDCS$ – information containing array of values of dependent and controllable signals (X^i, U^i) (as a replay to a $RSVDCS$ message),
- VQ – information containing value of quality measure Q^i (as a replay to a $RSVQ$ message).

Each Past Episode Agent functions until the whole system stops functioning, immediately sending replays for received messages.

Example 2. Past Episode Agent filled with data presented in Example 1:

- when will receive $RSVIS$ message, it sends back VIS message with data $I = [34.5, 56.59, 2.54, 5.29]$,
- when will receive $RSVDCS$ message, it sends back $VDCS$ message with data $(X, U) = ([X_1, X_2, \ldots, X_{1351}], [U_1, U_2, \ldots, U_{1351}])$, where values of X_1, X_2, X_{1351}, U_1, U_2 and U_{1351} are presented in Table 1,
- when will receive $RSVQ$ message, it sends back VQ message with data $Q = 0.687$.

5.2. Control Agent

The goal of the Control Agent is to control the current production process. This goal is obtained by the Control Agent through execution of the CBR cycle, which is presented in section 4.2. The Control Agent starts functioning at the beginning of a batch, when values of independent signals which characterize the chemical composition of raw materials used in the current production are known. After the start of its functioning, the Control Agent sequentially performs phases of the CBR cycle: retrieve, reuse, revise, retain.

5.2.1. Retrieve phase of a Control Agent

The main goal of the retrieve phase of the Control Agent is to select one relevant case according to the control of the current batch. In order to obtain this goal, the Control Agent performs the algorithm of the retrieve phase (presented in section 4.2.1). Because the case base is designed as the set of Past Episode Agents, the retrieve phase of the Control Agent is realized through interaction between agents. The interaction scenario related to the retrieve phase is presented as follows (with references to the algorithm in section 4.2.1 and messages stated in section 5.1.2):

1. the Control Agent gets values of independent signals I^P (according to Step 1.),
2. the Control Agent sends $RSVIS$ to all Past Episode Agents,
3. the Control Agent receives VIS,
4. the Control Agent chooses Past Episode Agents, which represent cases that are most similar to the present problem of production (according to Step 2. and 3.) and sends $RSVQ$ to all of the chosen agents,
5. the Control Agent receives VQ,
6. the Control Agent chooses one Past Episode Agent PEA^R that represent retrieved case C^R (according to Step 4.),
7. the Control Agent indicates Past Episode Agent PEA^R to be used in the reuse phase (according to Step 5.).

Example 3. Let us assume that the agent system consists of 5 Past Episode Agents and one Control Agent $Ag = \{PEA^1, PEA^2, PEA^3, PEA^4, PEA^5, CA\}$. Past Episode Agents are filled with data, a fragment of which is presented in Table 2.

Table 2
Independent signals and quality measure of 5 exemplary Past Episode Agents.

Agent	Independent signal $I = [i_S, i_{Zn}, i_{Pb}, i_{Fe}]$	Quality measure Q
PEA^1	[35.00, 53.10, 2.29, 7.65]	0.645
PEA^2	[34.10, 55.60, 2.09, 4.81]	0.754
PEA^3	[34.50, 56.59, 2.54, 5.29]	0.687
PEA^4	[34.60, 59.57, 1.95, 5.22]	0.621
PEA^5	[33.70, 58.60, 1.95, 4.16]	0.599

An example of an interaction scenario related to the retrieve phase can be seen as follows:

1. the Control Agent gets values of independent signals $I^P = [34.80, 55.90, 2.34, 4.99]$,
2. the Control Agent sends $RSVIS$ to all Past Episode Agents,
3. the Control Agent receives messages VIS from all Past Episode Agents together with independent signals represented by them (values presented in Table 2),
4. the Control Agent chooses Past Episode Agents that represent the most-similar cases to the present problem:
   - it computes the similarity for every Past Episode Agent: $sim(CA, PEA^1) = 0.2054$, $sim(CA, PEA^2) = 0.5490$, $sim(CA, PEA^3) = 0.5452$, $sim(CA, PEA^4) = 0.2126$, $sim(CA, PEA^5) = 0.2465$,
   - the most similar are PEA^2 and PEA^3, so the Control Agent sends $RSVQ$ to PEA^2 and PEA^3,
5. the Control Agent receives VQ from agents PEA^2 ($Q = 0.754$) and PEA^3 ($Q = 0.687$),
6. the Control Agent chooses among PEA^2 and PEA^3 one agent, which represents the best quality measure, so PEA^3 represents the smallest (best) value of quality measure and is chosen as PEA^R,
7. the Control Agent indicates the Past Episode Agent $PEA^R = PEA^3$ to be used in the reuse phase.

5.2.2. Reuse phase of a Control Agent

In the reuse phase, the solution represented by Past Episode Agent PEA^R (selected in the retrieve phase) is applied to the current control of the industrial process. Having the goal to reuse the solution, the control represented by Past Episode Agent PEA^R has to be approximated, and next, has to be used in the control of the present batch of production (which is done with the use of an artificial neuron net). The Control Agent follows the algorithm of the reuse phase (with references to the algorithm presented in section 4.2.2):

1. the Control Agent sends to the indicated Past Episode Agent PEA^R request of sending back an array of values of dependent and controllable signals $RSVDCS$ (according to Step 1.),
2. the Control Agent receives $VDCS$ – information containing an array of values of dependent and controllable signals (X^R, U^R) (according to Step 2.),
3. the Control Agent creates a neuron net (according to Step 3.),
4. the Control Agent trains the created neuron net (according to Step 4.),
5. the Control Agent uses the trained net to predict present values of controllable signals on the base of currently-measured dependent signals (according to Step 5.).

The reuse phase continues until the end of the current production batch. During Step 5, values of dependent and controllable signals should be saved in order to be used in the retain phase.

Example 4. Continuing example 3. the Control Agent now has the goal of using data associated with relevant agents $PEA^R = PEA^3$ at the control of production during the present batch:

1. the Control Agents sends $RSVDCS$ message to the PEA^3 agent,
2. the Control Agents receives $VDCS$ message from the PEA^3 agent; lets assume that data structures of PEA^3 are presented in Table 1; $VDCS$ message contains an array of values of dependent and controllable signals $(X^R, U^R) = ([X_1, X_2, \ldots, X_{1351}], [U_1, U_2, \ldots, U_{1351}])$, where values of X_1, X_2, X_{1351}, U_1, U_2 and U_{1351} are presented in Table 1,
3. the Control Agent creates a neuron net that, as the input takes one measure of dependent signal X_n and as the output returns one setting of controllable signals U_n,
4. the Control Agent trains the created neuron net with pars (X_1, U_1), (X_2, U_2), ..., (X_{1351}, U_{1351}), where values of X_1, X_2, X_{1351}, U_1, U_2 and U_{1351} are presented in Table 1,
5. the Control Agent uses the trained net in order to predict the present values of controllable signals on the base of currently-measured dependent signals, which means that the operation stated below is repeated until the end of the current batch:
   - the Control Agents obtain presently-measured dependent signal X_P (eg. $X_P = [947, 962, 953, 954, 942, 361, 8.40, 0.16, 18.11]$) and returns present controllable signal U_P (eg. $U_P = [62, 62, 18281, 20.49, 427, 12.94, 890]$); U_P is computed with the use of the trained neuron net.

5.2.3. Revise phase of a Control Agent

During the revise phase, the Control Agent obtains evaluations of products made in the period of the batch, during which it has controlled the industrial process (according to the algorithm presented in section 4.2.3). This evaluation is in the form of value Q^P, which is a single value of the average concentration of sulphide sulfur in the products. The result of the evaluation can not influence control done by this agent, because production was ended before obtaining the results of quality measures.

Example 5. In the case of the presented industrial process, this quality measure is done manually by production staff, and its effects are known after the end of production. So, the Control Agent obtains quality measure Q^P after the end of production performed in the current batch. Let assume $Q^P = 0.686$.

5.2.4. Retain phase of a Control Agent

This phase starts when the current problem was solved and the evaluation of this solution is known. At that time, the Control Agent knows the description of the problem (independent signals I^P), its solutions (dependent signals X^P and controllable signals U^P), and the obtained results (evaluation criterion Q^P). Its goal now is to retain this information by adding the past case that relates to the batch of production that just ended. Because every past case is represented by a Past Episode Agent, the Control Agent should create a new agent of Past Episode Agent type. The Control Agent follows the algorithm of the revise phase (with references to the algorithm presented in section 4.2.4):

1. the Control Agent inputs independent signals I^P, dependent signals X^P, controllable signals U^P, and value of evaluation criterion Q^P for the current batch (according to Step 1.),
2. the Control Agent creates the new case $C^P = (I^P, (X^P, U^P), Q^P)$ (according to Step 2.),
3. the Control Agent creates the new Past Episode Agent PEA^P, which represents the case C^P (according to Step 3.).

After creating the Past Episode Agent, the Control Agent is terminated.

Example 6. Continuing our previous examples, the Control Agent knows all signals of the batch that contain controlled production:

- independent signals $I^P = [34.80, 55.90, 2.34, 4.99]$ (step 1 in Example 3),
- dependent X^P and controllable signals U^P (step 5 in Example 4),
- quality measure $Q^P = 0.686$ (Example 5).

Now, the Control Agent creates new case $C^P = (I^P, (X^P, U^P), Q^P)$ and new Past Episode Agent PEA^6, which represents case C^P. After creation of the PEA^6 agent, the Control Agent is terminated.

6. System implementation and testing

In order to predict the capabilities of the presented approach to the industrial process control, an agent system is implemented according to remarks on design (as presented in the previous section). The implemented system functions as a test application that operates on the archival data of an industrial plant. The lack of deployment of the newly-created application into the currently-running real industrial process disables to obtain real products made under control of the developed system. The lack of real products results in the lack of evaluation of production results, what is the reason for problems related to implementation of the revise and retain phases of the Control Agent. As it is presented in subsections 5.2.3 and 5.2.4, the revise and retain phases require the real evaluation of products made in the current batch. Despite the aforementioned problems, the whole system gives a solution for control of the current batch.

6.1. Implementation of agents

The Past Episode Agent is implemented according to its design presented in section 5.1. By implementation, the Java programming language and JADE (Java Agent DEvelopment framework) are used. JADE is a flexible agent platform that simplifies the creation of agent-based systems [3]. JADE is used in the presented implementation mainly as a code library – the Past Episode Agent extends the Agent class from the jade.core package. The Past Episode Agent communicates with other agents in the created system via communication mechanisms provided by JADE – the ACLMessage class from the jade.lang.acl package is used. The yellow pages service is also used by the Past Episode Agent. This service supports a discovery mechanism that enables an agent to be discovered by other agents existing in the system. The Past Episode Agent is composed of 3 behaviors that are objects of the CyclicBehavior class from the jade.core.behaviors package. A single behavior is responsible for receiving and replaying a single message type ($RSVIS$, $RSVDCS$, or $RSVQ$, as presented in section 5.1.2).

The Control Agent is implemented alongside the Java programming language and JADE. The Control Agent extends the Agent class from the jade.core package and is composed of one behavior that extends the Behavior class from the jade.core.behaviors package. This behavior represents the task of the Control Agent, whose task includes selecting the relevant case and reusing that case at control of the present production batch (as presented in section 5.2). As mentioned at the beginning of the present section, the Control Agent does not perform the revise and retain phases due to the lack of the evaluation of real products. The Control Agent also uses the yellow pages service in order to discover agents (Past Episode Agents) in the system. The Control Agent communicates with Past Episode Agents via JADE communication mechanisms – the ACLMessage class from the jade.lang.acl package is used.

The Control Agent in the reuse phase uses the artificial neural network, as presented in section 5.2.2. Creation, training, and utilization of the neural net is implemented with the use of Neuroph (Java Neural Network Framework), which provides a Java neural network library containing ready-to-use Java classes for different types of neural networks. By implementation of the neural network of the Control Agent, the MultiLayerPerceptron class is used from the org.neuroph.nnet package. The structure of the implemented neural network is presented in Figure 3.

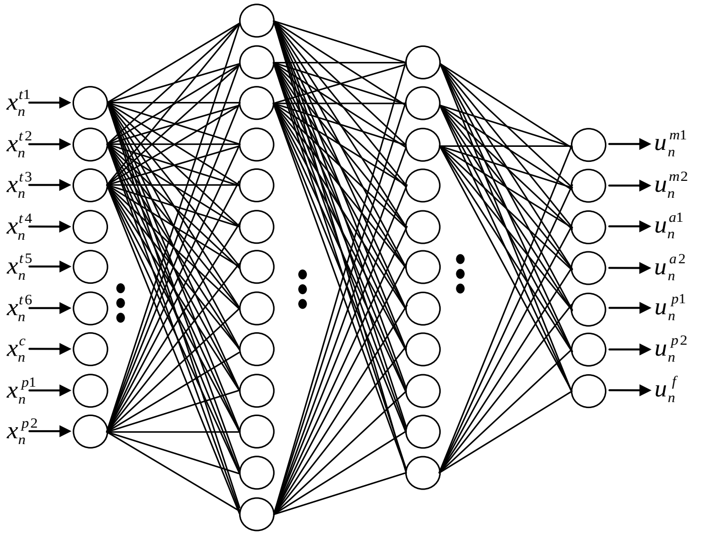

Figure 3. The structure of the neural network used by the Control Agent (dots represent connections, which are not shown).

As presented in Figure 3 the neural network (which is used in the reuse phase of the Control Agent functioning) is a multilayer perceptron. All neurons are located in 4 layers composed of 9, 13, 11, 7 neurons. The input layer is composed of 9 neurons and inputs one single measure of dependent signals X_n (referring to section 2.1.2). The output layer is composed of 7 neurons and returns one single setting of controllable signals U_n (referring to section 2.1.3). All neurons in the neuron net use the Sigmoid neuron-transfer function – the constant SIGMOID of the TransferFunctionType class from the org.neuroph.util package is used at the creation of the neural network object. By training of the neural network, the learn() method of the MultiLayerPerceptron class is used (which performs supervised learning).

6.2. Obtained results

The presented approach to control the industrial process requires data in order to be used as the case base – the set of experience items, which is the basis for processes that reflect the use and gathering of the human experience. The available industrial data concerns 19 full batches of production related to the oxidizing roasting of zinc sulfide concentrates, which is presented in section 2. Table 3 presents the main characteristics of the used data in the form of quality result for each batch. The available data is transformed to the case base of the implemented system. As presented in section 5.1, each past case is represented by one Past Episode Agent, so 19 Past Episode Agents were created according to the available industrial data.

Table 3

Measured quality for 19 past batches of real production.

Batch number	1	2	3	4	5
Measured quality	0.698	0.733	0.637	0.680	0.637
Batch number	6	7	8	9	10
Measured quality	0.645	0.670	0.687	0.621	0.599
Batch number	11	12	13	14	15
Measured quality	0.672	0.685	0.652	0.649	0.666
Batch number	16	17	18	19	—
Measured quality	0.838	0.808	0.775	0.754	—
Average quality	0.690				

Ten tests of the implemented system were made. Each test followed the same scenario, as stated below:

1. creation of 19 Past Episode Agents (according to available industrial data),
2. creation of one Control Agent,
3. sending to the Control Agent randomly-selected independent signals I,
4. sending to the Control Agent randomly-selected dependent signals X (after the agent has trained the neural network),
5. receiving from the Control Agent values of controllable signals U,
6. repeating steps 4. and 5. 1000 times.

During such a test, values of all signals are saved in order to be evaluated. The evaluation of each test is done with the external application. This application uses a neural network that, as the input, takes values of independent I, dependent X, and controllable U signals and predicts the average concentration of sulphide sulfur in products equal to quality measure Q. The neural network predicting quality is the multilayer perceptron and is trained with the same industrial data as that used to create the case base (represented by all Past Episode Agents). The use of the external application predicting product quality has to substitute a real deployment of the presented agent system into the control of production. Table 4 presents results of evaluations from each performed test with the external application.

Table 4

Results obtained for 10 runs of the implemented system.

Test number	1	2	3	4	5
Estimated quality	0.633	0.629	0.613	0.614	0.625
Test number	6	7	8	9	10
Estimated quality	0.603	0.606	0.620	0.645	0.641

As presented in Table 3, the average quality result for 19 archival batches is equal to 0.690. Taking into consideration that the quality measure is equal to the concentration of sulphide sulfur in products and the goal is to obtain a minimal concentration of sulphide sulfur in the roasted products, it can be stated, that the results presented in Table 4 are better than the average result for 19 archival batches that were controlled manually by a human operator. The estimated quality for each test is better than the average result of production made in the past.

6.3. Computing time and remarks on robustness

All tests whose results were presented in the previous subsection were done with the use of a computer with AMD Athlon 64 Dual Core Processor and 4 GB RAM. Computational time is presented in Table 5.

Table 5

Time of computation: t_{full} – time of full test computation, t_{comm} – time of communication of agents, t_{lear} – time of neuron net training.

Test number	t_{full} [ms]	t_{comm} [ms]	t_{lear} [ms]
1	33203	16	32438
2	77890	40	77032
3	42672	38	41891
4	70859	31	70141
5	165094	32	164312
6	67657	31	66937
7	76172	32	75421
8	61969	47	61219
9	174344	31	173593
10	186250	31	185500
average	95611	32.9	94848.4

As can be seen in Table 5, the average time of computation of one whole test is nearly 90 seconds. The largest part of the time of computation of the whole test corresponds to the training of the neuron net, which is trained and next used by the Control Agent in the reuse phase. All computations (except the training of the neuron net) last about one second. The average time of communications among interacting agents in the system is around 33 milliseconds. The time of communications can grow when the case base of the system includes more cases. Each case is represented by an individual Past Episode Agent, so more communicating agents mean more messages sent and received. The problem of too many Past Episode Agents can be resolved by using some additional agent techniques – Past Episode Agents can interact in order to select and remove those that are not usable (in other words, those that do not bring the desired effect on the control of a hypothetical batch). Such selection occurring among agents representing past cases is similar to selection in evolutionary multi-agent systems [8, 6] and is seen as the solution for problems with a growing collection of past cases. This proposition for future system modernization is also one of motivations that indicate the use of agent technology and support obtaining a system open for future changes and research.

In order to research the ability of the developed system to cope with errors during its execution, some Past Episode Agents are removed during system functioning. This abnormality of calculations does not cause the system to stop; however, the system uses only part of the knowledge related to experiences made in the past. Such robustness of the system to continue operating when the case base is partially damage is achieved by the proper use of agent techniques. Formulating remarks on system robustness, it has to be emphasized that the developed system should be seen as an illustration for the proposed approach that combines control of a semi-continuous (semi-batch) production process with learning based on production results. Deployment of the proposed methodology into a running production can involve rebuilding of the implemented system in order to fulfill conditions of a concrete industrial environment (it is possible that a different programming language should be used). Such a system rebuild should be deployed into a running production with carefulness – in the initial step of its functioning, the system can (or even should) be directly supervised by human workers.

7. Conclusions

The presented remarks on human decision making (using models and gathering experience) are formalized according to main notions of case-based reasoning methodology. This model can be used in many fields of problems that are still resolved by humans using experience. One such field is the control of industrial processes that are difficult to control with other known techniques. Deliberations presented in the article focus on semi-continuous industrial processes in which production is organized into individual batches while control performed during a single batch is done in a continuous manner. Control of such types of processes (in the case of the unknown analytical model) can involve interpolation of signals that are measured once per batch in order to obtain the frequency of signal measuring equal to other continuously-measured signals. Such interpolation can be undesired due to adding nonexistent and perhaps distorted values of parameters.

The presented model of using and gathering experience allows us to avoid undesired interpolation of signals in the case of control of the researched semi-continuous processes. The methodology proposed in the article also enables us to avoid the need of specifying rules according proper control. These features are main advantages according control of an industrial processes with the application of our proposed solutions. This particular application of the human decision making model to control a semi-continuous industrial process is a novel achievement that is unique in the known scientific literature. The presented conception can be a pattern for the design of a computer control system of an industrial process to which other known computational techniques do not bring the desired effects.

The researched conception of using and gathering experience of a human being assumes autonomy of cases, how individual items of experience are referred. The autonomy of experience items is the reason for using the agent technology at the design and implementation of a system that matches the presented model of using and gathering experience. The use of the agent technology at the design level enables us to obtain clear structure of a system in which the main components are agents performing activities associated with their meanings in the researched conception of human decision making. Implementation of the system is also easy due to the use of additional tools which enable quick and simply creation of multi-agent systems (JADE).

The implemented system is used as the test application in order to confirm the correctness of the presented model of human decision making and its application to the exemplary industrial process. The oxidizing roasting process of sulphide zinc concentrates was chosen as an example of a semi-continuous industrial process. This process is still controlled by human operators in known industrial plants. Results obtained during tests of the implemented system are better than results related to the available industrial data, which indicates usefulness and correctness of the proposed solutions in this article.

Acknowledgements

Financial support of the Ministry of Science and Higher Education (AGH UST, project no. 11.11.110.300) is acknowledged.

References

[1] Aamodt A., Plaza E.: Case-based Reasoning: Foundational Issues, Methodological Variations, and System Approaches. *AI Communications*, vol. 7(1), pp. 39–59, 1994.

[2] Atanassov A., Antonov L.: Comparative analysis of case based reasoning software frameworks jCOLIBRI and myCBR. *Journal of the University of Chemical Technology & Metallurgy*, vol. 47(1), pp. 83–90, 2012.

[3] Bellifemine F. L., Caire G., Greenwood D.: *Developing Multi-Agent Systems with JADE*. Wiley, 2007.

[4] Bequette B.: *Process Control: Modeling, Design, and Simulation*. Prentice Hall Press, Upper Saddle River, NJ, USA, 2002.

[5] Bergmann R., Althoff K. D., Minor M., Reichle M., Bach K.: Case-Based Reasoning – Introduction and Recent Developments. *Künstliche Intelligenz: Special Issue on Case-Based Reasoning*, vol. 23(1), pp. 5–11, 2009.

[6] Byrski A.: Tuning of agent-based computing. *Computer Science*, vol. 14(3), pp. 491–512, 2013.

[7] Hornik K., Stinchcombe M., White H.: Multilayer Feedforward Networks Are Universal Approximators. *Neural Networks*, vol. 2(5), pp. 359–366, 1989.

[8] Kisiel-Dorohinicki M.: Evolutionary Multi-Agent Systems in Non-Stationary Environments. *Computer Science*, vol. 14(4), pp. 563–575, 2013.

[9] Leitao P.: Agent-based distributed manufacturing control: A state-of-the-art survey. *Engineering Applications of Artificial Intelligence*, vol. 22(7), pp. 979–991, 2009.

[10] Richter M. M.: Introduction. In: M. Lenz, B. Bartsch-Spörl, H. D. Burkhard, S. Wess (eds.): Case-Based Reasoning Technology, From Foundations to Applications, pp. 1–16. Springer-Verlag, London, UK, 1998.

[11] Shen W., Norrie D.H., Kremer R.: Developing Intelligent Manufacturing Systems Using Collaborative Agents. *Proceedings of the 2nd International Workshop on Intelligent Manufacturing Systems*, pp. 157–166. 1999.

[12] Sun J.: CBR Applications in Combustion Control of Blast Furnace Stoves. In: *Proceedings of the International MultiConference of Engineers and Computer Scientists 2008*, pp. 25–28. Newswood Limited, 2008.

[13] Sztangret Ł., Rauch Ł., Kusiak J., Jarosz P., Małecki S.: Modeling of the oxidizing roasting process of zinc sulphide concentrates using the artificial neural networks. *Computer Methods in Materials Science*, vol. 1(11), pp. 122–127, 2011.

[14] Watson I.: Case-based reasoning is a methodology not a technology. *Knowledge-Based Systems*, vol. 12(5-6), pp. 303–308, 1999.

[15] Weiss G., ed.: *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence.* MIT Press, Cambridge, MA, USA, 1999.

[16] Woolridge M.: *Introduction to Multiagent Systems.* John Wiley & Sons, Inc., New York, NY, USA, 2001.

Affiliations

Gabriel Rojek
AGH University of Science and Technology, Department of Computer Science, Krakow, Poland, rojek@agh.edu.pl

Received: 2.06.2014
Revised: 16.09.2014
Accepted: 16.09.2014