

DETERMINING APPROPRIATE SEQUENCES OF ACTIONS TO TAKE UPON OPERATING STATES OF INDUSTRIAL PLANTS

5 **FIELD OF THE INVENTION**

The invention relates to the monitoring of industrial plants and to the determining of actions to take in response to particular operating states, in particular in response to abnormal operating states.

10

BACKGROUND

15

The intended operation of many industrial plants is controlled by distributed control systems, DCS, that adjust set-points of low-level controllers in order to optimize one or more key performance indicators, KPI. The plant is also continuously monitored for any abnormal conditions, such as equipment failure or malfunction. This monitoring may be at least partially integrated with control by the DCS.

20

If an abnormal condition is detected, it is not always possible to remedy it automatically by means of the DCS. In every plant, there are abnormal situations that need to be remedied by an operator executing a certain sequence of actions. WO 2019/104 296 A1 discloses an alarm management system that assists an operator with identifying high priority alarms. However, the outputting of the alarm to the operator is only the first step towards remedying the abnormal condition. It is also necessary that the operator executes the correct sequence of actions.

25

30

US 2020/026 976 A1 discloses a method for harmonizing industrial machines with an intelligent industrial assistant having a set of pre-defined commands. States of a proprietary I/O interface of a machine are mapped to pre-defined commands. Based on this mapping, a customized interface for the machine is generated.

35

US 2021/097 401 A1 discloses a neural network system for the processing of input data values. Input data values are encoded by an encoder network, and the encoded products are aggregated with an aggregator. A combination of the aggregated output and a target input value is decoded by a decoder to yield the final output.

US 2014/237 487 A1 discloses an event processing system and a method of operation. Based on observations derived from measurements, the occurrence of an event is detected. Based on the event, a to-be-performed state transition and a to-be-performed action are determined.

5

OBJECTIVE OF THE INVENTION

It is therefore an objective of the invention to aid a plant operator in remedying an abnormal situation by computing, from information gained during the monitoring of an industrial plant or a part thereof, a sequence of actions to take.

10

This objective is achieved by a first method for determining an appropriate sequence of events according to a first independent claim, and by a second method according to a second independent claim for training the machine-learning networks that are used in the first method. Further advantageous embodiments are detailed in the respective dependent claims.

15

DISCLOSURE OF THE INVENTION

The invention is defined by the appended claims. Embodiments and examples not covered by the claims are presented to illustrate, and facilitate the understanding of, the claimed invention.

20

The invention provides a computer-implemented method for determining an appropriate sequence of actions to take during operation of an industrial plant or part thereof.

25

In the course of this method, values of a plurality of state variables that characterize an operational state of the plant or part thereof are obtained. For example, these values

may be combined in a vector or tensor. The concrete set of state variables that is needed to characterize the state of a plant or part thereof is plant-specific. Examples of such state variables include a pressure, a temperature, a mass flow, a voltage, an electrical current, a fill level, and a concentration of a substance in a mixture of substances.

The plurality of state variables (e.g., the vector or tensor comprising the values of the state variables) is encoded, by at least one trained state encoder network, into a representation of the operating state of the plant or part thereof. In particular, such a representation may have a much lower dimensionality than the original plurality of state variables. I.e., the representation may depend on a much smaller number of variables than the original vector with the values of state variables.

Examples for state variables include, but are not limited to:

- variables that are at least indicative of process variables, such as pressures, mass flows, voltages or currents;
- events that denote discrete changes in the plant, such as the switching on or off of a motor, a signal crossing an alarm limit, the opening or closing of a valve; and
- a combination of process variables and events.

If the state variables include both process variables and events, the state encoder network may be a combined state encoder network that takes both process variables and events as inputs simultaneously. But a combination of two state encoder networks, wherein the one encodes the process variables and the other encodes the events, may also be used.

By means of a trained state-to-action network, the representation of the operating state is mapped to a representation of a sequence of actions to take in response to the operating state. In particular, just like a small number of variables may encode a complex operating state of the plant that depends on very many state variables, a small number of variables may also encode a complex sequence of actions comprising many different actions.

The representation of the sequence of actions is decoded to the sought sequence of actions to take by a trained action decoder network. The so-determined sequence of actions may then be used in any possible manner in order to remedy an abnormal situation. For example, to the extent that the actions in the sequence can be performed automatically, the automatic performing of these actions may be triggered. For actions that can only be performed by an operator, the operator may be notified in any suitable manner to perform them. Also, any suitable actions may be taken to aid the operator in performing such action. For example, if the operator needs to operate some particular control element in a human-machine interface of the DCS, this may be highlighted. If an action requires the operation of a physical control (such as a button, switch or knob) and this is control guarded by a cover to prevent accidental or inadvertent operation, the cover may unlock and/or open. Likewise, if the action requires physical access to certain equipment, access doors to the location of the equipment may be unlocked, and/or the location of the equipment may be rendered safe for the operator to enter,

- 4 -

such as by stopping equipment that might harm the operator. Hybrid solutions in the sense that actions available in the HMI of a DCS are executed automatically, without human intervention or optionally at least under human supervision (i.e., the human observes as the sequence of actions proposed by the system is executed automatically) are also possible.

Examples for actions in the sequence of actions include:

- enabling or disabling a piece of equipment in the plant or part thereof;
- opening or closing a valve in the plant or part thereof; and
- changing a set-point value of at least one low-level controller in the plant or part thereof.

The main evaluation of the operating state of the plant happens when the representation of this operating state is mapped to a representation of a sequence of actions. Since the representations usually have a much lower dimensionality than the operating state, respectively the sequence of actions, this means that the evaluation merely comprises a mapping between two low-dimensional spaces. This makes the state-to-action network a lot easier to train than a network that directly gets high-dimensional operating states and directly outputs high-dimensional action sequences. In particular, that the state-to-action network is easier to train comprises that the training requires a lesser quantity of training samples.

Moreover, distinct tasks in the overall processing chain, namely

- going from a high-dimensional input to a low-dimensional representation,
- processing this representation into another representation in another low-dimensional space, and
- going from that second low-dimensional representation space into a second high-dimensional output space,

are assigned to distinct networks that can specialize on their respective jobs. This improves the overall accuracy of the finally determined output compared with the use of a single network that has to perform all tasks at once and may have to make a trade-off between conflicting goals. This is somewhat akin to the philosophy of Unix tools, such as sed, awk or grep. Each tool is built to do exactly one single simple job, and excel at that one single simple job. For more complex jobs, the output of one tool is piped into the other as input.

- 5 -

In a particularly advantageous embodiment, the state encoder network is chosen to be an encoder part of an encoder-decoder arrangement that first maps a plurality of state variables to a representation of the operating state of the plant or part thereof and then reconstructs a plurality of state variables from this representation. Such an encoder-decoder arrangement may be trained in a self-supervised manner. That is, a training sample may be inputted into the arrangement, and the outcome of the arrangement may be evaluated as to how well it is in agreement with the original input. For this training to work, the training samples do not need to be labelled with "ground truth". In machine learning applications, obtaining "ground truth" is the most expensive part of the training.

Moreover, training for a reconstruction in this manner trains the encoder-decoder arrangement to force the part of the input that is most salient for the reconstruction through the "information bottleneck" of the low-dimensional representation. This eliminates less important portions of the input, such as noise. Therefore, the encoder also gets some de-noising functionality by virtue of the training.

Alternatively or in combination to this, in another particularly advantageous embodiment, the action decoder network is chosen to be a decoder part of an encoder-decoder arrangement that first maps a sequence of actions, and/or a processing result derived from this sequence of actions, to a representation, and then reconstructs a sequence of actions, and/or a processing result derived therefrom, from this representation. The advantages are the same.

Reconstructing a "processing result" means that, for example, the encoder-decoder arrangement may be trained to predict, based on a part of a sequence of actions, the next action in this sequence of actions.

Examples of networks that may be used as encoder networks and/or decoder networks for operating states, and/or for sequences of actions, include recurrent neural networks, RNNs, and transformer networks. In a recurrent neural network, the output is fed back as input, and the network is executed for a predetermined number of iterations. A transformer neural network comprises a sequence of multiple coding units, each of which comprises an attention head that computes correlations between

- 6 -

different parts of the input. Both architectures are particularly useful for processing sequences as inputs.

5 The state-to-action network may, for example, comprise a convolutional neural network. In a particularly advantageous embodiment, the state-to-action network comprises a fully connected neural network. This architecture provides the greatest flexibility for training, at the price that it contains many parameters per size of the input and output. As discussed before, the dimensionality of the representation of the operating state, as well as the dimensionality of the representation of the sequence of
10 actions, is rather low. Therefore, the higher number of parameters that a fully connected neural network has may be accommodated.

Exemplary plants where the method is particularly advantageous, especially to remedy abnormal situations, include continuous or process plants that are configured to emit
15 alarms and events data. For example, the method may be used to remedy abnormal situations in:

- a waste incineration plant;
- a hydrocarbon separation plant;
- a reinjection system for injecting water into a hydrocarbon well;
- 20 • a hydrocarbon exploitation facility; and/or
- a deglycol regeneration plant.

These plants have in common that abnormal situations frequently pose safety issues. In case of a high priority alarm, if a wrong sequence of actions is commenced, or if
25 errors are made when executing the correct sequence (such as omitting a step or swapping the order of two steps), this may make the abnormal situation worse, possibly forgoing the last opportunity to get the plant under control again. Also, because safety-critical abnormal situations are fortunately rare, there is only little training data for these situations available. It is therefore advantageous that the method
30 can work with lesser training data because the main reasoning is, as discussed before, done between two spaces of rather low dimensionality.

The invention also provides a method for training an arrangement of networks for use in the method described above.

35

- 7 -

In the course of this method, a pre-trained first encoder-decoder arrangement of an action encoder network and an action decoder network is obtained. Also, a pre-trained second encoder-decoder arrangement of a state encoder network and a state decoder network is obtained.

5

Samples of training data are obtained. Each such sample comprises values of a plurality of state variables that characterize the operating state of the plant or part thereof. These state variables are input data for the to-be-trained arrangement. Each sample also comprises sequence of actions taken in response to this operating state. This sequence is the “ground truth” label attached to the operating state of the sample.

10

The values of the state variables in each sample are encoded into a representation of the respective operating state by the pre-trained state encoder network. The resulting representation of the operating state is mapped to a representation of a sequence of actions. The sequence of actions encoded in this representation is the sequence of actions that the arrangement of networks is proposing given the operating state characterized by the state variables.

15

Correspondence of this sequence of actions to the “ground truth” that is attached to the sample is measured by means of a cost function. This can be accomplished by any one, or a combination, of the following two manners:

20

According to the first manner, the loss function measures to which extent the representation of the sequence of actions is in agreement with a representation obtained by encoding, by the pre-trained action encoder network, the sequence of actions in the training sample;

25

According to the second manner, the loss function measures to which extent a sequence of actions obtained by decoding, by the pre-trained action decoder network, the representation of the sequence of actions is in agreement with the sequence of actions in the training sample.

30

In the course of the training, parameters that characterize the behavior of the to-be-trained state-to-action network are optimized such that, when further training samples are processed, the rating by the loss function is likely to improve.

35

The state variables may be obtained during a real execution of a process on the plant, or after such execution from a plant historian, or from execution of simulation with a process simulator producing the same state-variables as the process on the plant. The
5 usage of a process simulator is particular beneficial for newly commissioned plants with little historical data and when the model is first trained on simulated data capturing the general process dynamics and later with the limited amount of data from the real process execution. Likewise, the sequences of actions may be monitored during execution of the process, or it may be obtained after such execution from an action log,
10 or for initial training be obtained from simulation experiments with real plant operators or predefined actions sequences. Since both the state variables in the plant historian and the actions in the action log are timestamped, they can be correlated to each other. The training may thus be understood to “mine” a workflow with which the plant operator reacts to a particular situation and teach the network arrangement to propose this
15 workflow when this, or a substantially similar, situation occurs again. In this manner, even knowledge that is present in the operator’s mind but hard to put in words, or to communicate to another operator, may be put to use.

For example, if an operator has learned to “open a valve slowly if the flame turns
20 bluish”, each operator may perceive the moment at which the flame turns bluish slightly differently. Also, different operators may have different notions of opening a valve “slowly”. The present training method allows to capture knowledge in an automated manner that leaves no room for interpretation.

25 In a particularly advantageous embodiment, the obtaining of a first pre-trained encoder-decoder arrangement of an action encoder network and an action decoder network comprises:

- obtaining training samples of sequences of actions;
- providing the sequence of actions in each training sample, and/or a processing
30 result derived therefrom, to the to-be-trained action encoder network, thereby obtaining a representation;
- providing this representation to the to-be-trained action decoder network, thereby obtaining a sequence of actions, and/or a processing result derived therefrom;

- measuring, by means of a predetermined loss function, to which extent this sequence of actions, and/or this processing result, is in agreement with the sequence of actions, and/or the processing result, in the training sample; and
- optimizing parameters that characterize the behavior of the to-be-trained action encoder network and the to-be-trained action decoder network such that, when further training samples are processed, the rating by the loss function is likely to improve.

The training samples used for this pre-training may have training samples in common with the main training discussed above, but it may also be performed on a set of training samples disjoint from the one used for the main training. For example, the pre-training may be performed in a generic manner once for a particular type of plant. For each instance of the plant that is subsequently installed, the pre-trained encoder-decoder arrangement may then be used in the training of the state-to-sequence network. Optionally, when moving from the generic training to a concrete instance of the plant, the pre-training of the encoder-decoder arrangement may be refined using further training samples that have been acquired from this instance of the plant.

Pre-training might be done with a variety of tasks for which “ground truth labels” can be easily produced from the available process state data and the actions sequences. Examples of such tasks are reconstruction of the input (plant state variables or action sequences), predicting the next n elements of the sequences (plant state variables or action sequences), identifying the correct next sequence segments among a number of presented sequence segments, identifying the correct previous sequence segments among a number of presented sequence segments, identifying whether presented sequence are neighbors in the overall sequence, etc. Such tasks might also be combined in parallel or in sequence which is beneficial because it further increases the amount of training data for pretraining and also prevents overfitting the pre-trained model to a single task.

The same advantages apply, in an analogous manner, to a further particularly advantageous embodiment where the obtaining of a second pre-trained encoder-decoder arrangement of a state encoder network and a state decoder network comprises:

- obtaining training samples comprising values of a plurality of state variables that characterize an operating state of the plant or part thereof;
- providing the values of state variables in each training sample to the to-be-trained state encoder network, thereby obtaining a representation;
- 5 • providing this representation to the to-be-trained state decoder network, thereby obtaining values of state variables;
- measuring, by means of a predetermined loss function, to which extent these values are in agreement with the values in the training sample; and
- 10 • optimizing parameters that characterize the behavior of the to-be-trained state encoder network and the to-be-trained state decoder network such that, when further training samples are processed, the rating by the loss function is likely to improve.

15 In a further particularly advantageous embodiment, the to-be-trained action encoder network and the to-be-trained state encoder network are combined in one single network architecture. This single network architecture may depend on fewer parameters than a combination of two individual architectures, resulting in an easier training. Also, since the tasks performed by both networks have something in common, the two trainings may benefit from each other to some degree by “sharing” knowledge
20 in the one single network architecture.

25 In a further particularly advantageous embodiment, the obtaining of training samples for the training of the first encoder-decoder arrangement, and/or the second encoder-decoder arrangement, and/or the state-to-action network, comprises aggregating training samples obtained in multiple industrial plants. This improves the overall variability in the set of training samples, resulting in a better performance of the final arrangement of networks in terms of accuracy. As discussed before, abnormal situations that pose safety risks tend to occur very rarely in any given plant. Because of the safety risks, it is usually not practical to provoke such a situation just for the
30 purpose of acquiring training samples. But in a larger fleet of plants, there will be enough instances of the abnormal situations occurring on their own that a reasonable amount of training samples may be gathered.

35 As discussed before, the methods are computer-implemented. The invention therefore also relates to one or more computer programs with machine-readable instructions

that, when executed on one or more computers and/or compute instances, cause the one or more computers to perform a method described above. In this context, a virtualization platform, a hardware controller, network infrastructure devices (such as switches, bridges, routers or wireless access points), as well as end devices in the network (such as sensors, actuators or other industrial field devices) that are able to execute machine readable instructions are to be regarded as computers as well.

The invention therefore also relates to a non-transitory storage medium, and/or to a download product, with the one or more computer programs. A download product is a product that may be sold in an online shop for immediate fulfillment by download. The invention also provides one or more computers and/or compute instances with the one or more computer programs, and/or with the one or more non-transitory machine-readable storage media and/or download products.

DESCRIPTION OF THE FIGURES

In the following, the invention is illustrated using Figures without any intention to limit the scope of the invention. The Figures show:

Figure 1: Exemplary embodiment of the method 100 for determining an appropriate sequence of actions to take during operation of an industrial plant;

Figure 2: Exemplary embodiment of the method 200 for training an arrangement of networks for use in the method 100;

Figure 3: Illustration of the two manners in which the state-to-action network 4 may be trained.

Figure 1 is a schematic flow chart of an embodiment of the method 100 for determining an appropriate sequence of actions to take during operation of an industrial plant 1.

In step 110, values of a plurality of state variables 2 that characterize an operational state of the plant 1 or part thereof are obtained.

- 12 -

In step 120, the plurality of state variables 2 are encoded into a representation 2a of the operating state of the plant 1 or part thereof by at least one trained state encoder network 3.

5 According to block 121, the state encoder network 3 may be chosen to be an encoder part of an encoder-decoder arrangement that first maps a plurality of state variables 2 to a representation 2a of the operating state of the plant 1 or part thereof and then reconstructs a plurality of state variables 2 from this representation 2a.

10 In step 130, the representation 2a of the operating state is mapped to a representation 6a of a sequence 6 of actions to take in response to the operating state by a trained state-to-action network 4.

15 In step 140, the representation 6a of the sequence 6 of actions is decoded to the sought sequence 6 of actions to take by a trained action decoder network 5.

According to block 141, the action decoder network 5 may be chosen to be a decoder part of an encoder-decoder arrangement that first maps a sequence 6 of actions, and/or a processing result derived from this sequence 6 of actions, to a representation 20 6a, and then reconstructs a sequence 6 of actions, and/or a processing result derived therefrom, from this representation 6a.

Figure 2 is a schematic flow chart of an embodiment of the method 200 for training an arrangement of networks 3, 4, 5 for use in the method 100 described above.

25 In step 210, a pre-trained first encoder-decoder arrangement of an action encoder network 5# and an action decoder network 5 is obtained. In the example shown in Figure 2, this obtaining comprises the additional steps of:

- obtaining, according to block 211, training samples of sequences 6 of actions;
- 30 • providing, according to block 212, the sequence 6 of actions in each training sample, and/or a processing result derived therefrom, to the to-be-trained action encoder network 5#, thereby obtaining a representation 6a;
- providing, according to block 213, this representation 6a to the to-be-trained action decoder network 5, thereby obtaining a sequence 6' of actions, and/or a processing result derived therefrom;
- 35

- measuring, according to block 214, by means of a predetermined loss function 9, to which extent this sequence 6' of actions, and/or this processing result, is in agreement with the sequence 6 of actions, and/or the processing result, in the training sample; and
- 5 • optimizing, according to block 215, parameters 5a#, 5a that characterize the behavior of the to-be-trained action encoder network 5# and the to-be-trained action decoder network 5 such that, when further training samples are processed, the rating 9a by the loss function 9 is likely to improve. The finally optimized state of the parameters 5a#, 5a is labelled with the reference sign
- 10 5a#*, 5a*.

In step 220, a pre-trained second encoder-decoder arrangement of a state encoder network 3 and a state decoder network 3# is obtained. In the example shown in Figure 2, this obtaining comprises the additional steps of:

- 15 • obtaining, according to block 221, training samples comprising values of a plurality of state variables 2 that characterize an operating state of the plant 1 or part thereof;
- providing, according to block 222, the values of state variables 2 in each training sample to the to-be-trained state encoder network), thereby obtaining a
- 20 representation 2a;
- providing, according to block 223, this representation 2a to the to-be-trained state decoder network 3#, thereby obtaining values 2' of state variables;
- measuring, according to block 224, by means of a predetermined loss function
- 25 10, to which extent these values 2' are in agreement with the values 2 in the training sample; and
- optimizing, according to block 225, parameters 3a, 3a# that characterize the behavior of the to-be-trained state encoder network 3 and the to-be-trained state decoder network 3 such that, when further training samples are
- 30 processed, the rating 10a by the loss function 10 is likely to improve. The finally optimized state of the parameters 3a, 3a# is labelled with the reference sign
- 3a*, 3a#*.

In step 230, samples 7 of training data are obtained. Each sample 7 comprises values of a plurality of state variables 2 that characterize the operating state of the plant 1 or

35 part thereof, and a sequence 6* of actions taken in response to this operating state.

According to block 231, the obtaining 230 of training samples 7 for the training of the first encoder-decoder arrangement, and/or the second encoder-decoder arrangement, and/or the state-to-action network 4, comprises aggregating training samples 7
5 obtained in multiple industrial plants 1.

In step 240, the values of the state variables 2 in each sample 7 are encoded into a representation 2a of the respective operating state by the pre-trained state encoder network 3.
10

In step 250, the representation 2a of the operating state is mapped to a representation 6a of a sequence 6 of actions.

In step 260, it is measured, by means of a predetermined loss function 8, to which extent
15

- the representation 6a of the sequence of actions is in agreement with a representation 6a* obtained by encoding, by the pre-trained action encoder network 5#, the sequence 6 of actions in the training sample 7; and/or
- a sequence of actions 6* obtained by decoding, by the pre-trained action
20 decoder network 5, the representation 6a of the sequence 6 of actions is in agreement with the sequence 6* of actions in the training sample 7;

In step 270, parameters 4a that characterize the behavior of the to-be-trained state-to-action network 4 are optimized such that, when further training samples 7 are
25 processed, the rating 8a by the loss function 8 is likely to improve. The finally optimized state of the parameters 4a is labelled with the reference sign 4a*.

Figure 3 illustrates the two manners in which the state-to-action network 4 may be trained.
30

Values of one or more state variables 2 from a training sample 7 are encoded to a representation 2a of the operating state of the plant 1. By means of the state-to-action network 4, this representation 2a is mapped to a representation 6a of a sequence 6 of actions. This representation 6a needs to be compared with a "ground truth" sequence

of actions 6^* in the training sample 7, so as to measure to which extent the representation 6a outputted by the state-to-action network 4 is correct.

5 In a first manner, the “ground truth” sequence 6^* of actions is encoded, by the pre-trained action encoder network 5#, into a “ground truth” representation $6a^*$. The loss function 8 measures to which extent the representation 6a outputted by the state-to-action network 4 is in agreement with o the “ground truth” representation $6a^*$.

10 In a second manner, the representation 6a outputted by the state-to-action network 4 is decoded by the pre-trained action decoder network 5, so as to obtain a sequence $6'$ of actions. The loss function 8 then measures to which extent this sequence 6 of actions is in agreement with the “ground truth” sequence 6^* of actions.

List of reference signs:

	1	industrial plant
5	2	state variables characterizing operational state of plant 1
	2'	state variables decoded during encoder-decoder training
	2a	representation of operational state of plant 1
	3	state encoder network
	3a	parameters, characterize behavior of network 3
10	3a*	finally optimized state of parameters 3a
	3#	state decoder network
	3a#	parameters, characterize behavior of network 3#
	3a#*	finally optimized state of parameters 3a#
	4	state-to-action network
15	4a	parameters, characterize behavior of network 4
	4a*	finally optimized state of parameters 4a
	5	action decoder network
	5a	parameters, characterize behavior of network 5
	5a*	finally optimized state of parameters 5a
20	5#	action encoder network
	5a#	parameters, characterize behavior of network 5#
	5a#*	finally optimized state of parameters 5a#
	6	sequence of actions
	6'	sequence decoded during encoder-decoder training
25	6*	sequence of actions in training sample 7
	6a	representation of sequence 6 of actions
	6a*	representation encoded from sequence 6*
	7	training samples for training state-to-action network 4
	8	loss function for training state-to-action network 4
30	8a	rating by loss function 8
	9	loss function for action encoder-decoder arrangement 5#, 5
	9a	rating by loss function 9
	10	loss function for state encoder-decoder arrangement 3, 3#
	10a	rating by loss function 10
35	100	method for determining appropriate sequence 6 of actions

- 17 -

110 obtaining state variables 2
120 encoding state variables into representation 120
121 choosing state encoder 3 from encoder-decoder arrangement
130 mapping representation 2a to sequence representation 6a
5 140 decoding representation 6a to sought sequence 6
141 choosing action decoder 5 from encoder-decoder arrangement
200 method for training arrangement of networks 3, 4, 5
210 obtaining pre-trained first encoder-decoder arrangement 5#, 5
211 obtaining training samples of sequences 6
10 212 providing training sequence 6 to action encoder network 5#
213 providing representation 6a to action decoder network 5
214 rating decoded sequence 6' with loss function 9
215 optimizing parameters 5a#, 5# of networks 5a, 5
220 obtaining pre-trained second encoder-decoder arrangement 3, 3#
15 221 obtaining training samples of state variables 2
222 providing state variables 2 to state encoder network 3
223 providing representation 2a to state decoder network 3#
224 rating values 2' with loss function 10#
225 optimizing parameters 3a, 3a# of networks 3, 3#
20 230 obtaining samples 7 of training data
231 aggregating training samples 7 across plants
240 encoding training state variables 2 into representation 2a
250 mapping state representation 2a to sequence representation 6a
260 rating sequence representation 6a/sequence 6 with loss function 8
25 270 optimizing parameters 4a of state-to-action network 4

Claims:

1. A computer-implemented method (100) for determining an appropriate sequence (6) of actions to take during operation of an industrial plant (1) or part thereof, comprising the steps of:
 - obtaining (110) values of a plurality of state variables (2) that characterize an operational state of the plant (1) or part thereof;
 - encoding (120), by at least one trained state encoder network (3), the plurality of state variables (2) into a representation (2a) of the operating state of the plant (1) or part thereof, wherein this representation (2a) of the operating state depends on less variables than the operating state of the plant (1);
 - mapping (130), by a trained state-to-action network (4), the representation (2a) of the operating state to a representation (6a) of a sequence (6) of actions to take in response to the operating state, wherein this representation (6a) of the sequence (6) of actions depends on less variables than the sequence (6) of actions; and
 - decoding (140), by a trained action decoder network (5), the representation (6a) of the sequence (6) of actions to the sequence (6) of actions to take.
2. The method (100) of claim 1, wherein the state encoder network (3) is chosen (121) to be an encoder part of an encoder-decoder arrangement that first maps a plurality of state variables (2) to a representation (2a) of the operating state of the plant (1) or part thereof and then reconstructs a plurality of state variables (2) from this representation (2a).
3. The method (100) of any one of claim 1 or 2, wherein the action decoder network (5) is chosen (141) to be a decoder part of an encoder-decoder arrangement that first maps a sequence (6) of actions, and/or a processing result derived from this sequence (6) of actions, to a representation (6a), and then reconstructs a sequence (6) of actions, and/or a processing result derived therefrom, from this representation (6a).
4. The method (100) of any one of claims 1 to 3, wherein the state encoder network (3), and/or the action decoder network (5), comprises a recurrent neural network, RNN, and/or a transformer neural network.

5. The method (100) of any one of claims 1 to 4, wherein the state-to-action network (4) comprises a convolutional neural network, and/or a fully connected neural network.
- 5 6. The method (100) of any one of claims 1 to 5, wherein the state variables (2) comprise one or more of: a pressure, a temperature, a mass flow, a voltage, an electrical current, a fill level, and/or a concentration of a substance in a mixture of substances.
7. The method (100) of any one of claims 1 to 6, wherein the actions comprise one or more of:
 - 10 • enabling or disabling a piece of equipment in the plant (1) or part thereof;
 - opening or closing a valve in the plant (1) or part thereof; and
 - changing a set-point value of at least one low-level controller in the plant (1) or part thereof.
8. The method (100) of any one of claims 1 to 7, wherein the plant (1) or part thereof is a continuous or process plant that is configured to emit alarms and events data.
- 15 9. The method (100) of any one of claims 1 to 8, wherein the plant (1) or part thereof comprises one or more of:
 - 20 • a waste incineration plant;
 - a hydrocarbon separation plant;
 - a reinjection system for injecting water into a hydrocarbon well;
 - a hydrocarbon exploitation facility; and/or
 - a deglycol regeneration plant.
10. A computer-implemented method (200) for training an arrangement of networks (3, 4, 5) for use in the method (100) of any one of claims 1 to 9, comprising the steps of:
 - 25 • obtaining (210) a pre-trained first encoder-decoder arrangement of an action encoder network (5#) and an action decoder network (5);
 - obtaining (220) a pre-trained second encoder-decoder arrangement of a state encoder network (3) and a state decoder network (3#);
 - 30 • obtaining (230) samples (7) of training data, wherein each sample (7) comprises values of a plurality of state variables (2) that characterize the operating state of the plant (1) or part thereof, and a sequence (6*) of actions taken in response to this operating state;

- encoding (240), by the pre-trained state encoder network (3), the values of the state variables (2) in each sample (7) into a representation (2a) of the respective operating state, wherein this representation (2a) of the operating state depends on less variables than the respective operating state;
 - 5 • mapping (250), by the to-be-trained state-to-action network (4), the representation (2a) of the operating state to a representation (6a) of a sequence (6) of actions, wherein this representation (6a) of the sequence (6) of actions depends on less variables than the sequence (6) of actions;
 - 10 • measuring (260), by means of a predetermined loss function (8), to which extent
 - the representation (6a) of the sequence of actions is in agreement with a representation (6a*) obtained by encoding, by the pre-trained action encoder network (5#), the sequence (6*) of actions in the training sample (7); and/or
 - 15 ○ a sequence of actions (6) obtained by decoding, by the pre-trained action decoder network (5), the representation (6a) of the sequence (6) of actions is in agreement with the sequence (6*) of actions in the training sample (7),
thereby obtaining a rating (8a); and
 - 20 • optimizing (270) parameters (4a) that characterize the behavior of the to-be-trained state-to-action network (4) such that, when further training samples (7) are processed, the rating (8a) by the loss function (8) is likely to improve.
11. The method (200) of claim 10, wherein the obtaining (210) of a first pre-trained encoder-decoder arrangement of an action encoder network (5#) and an action decoder network (5) comprises:
- 25 • obtaining (211) training samples of sequences (6) of actions;
 - providing (212) the sequence (6) of actions in each training sample, and/or a processing result derived therefrom, to the to-be-trained action encoder network (5#), thereby obtaining a representation (6a);
 - 30 • providing (213) this representation (6a) to the to-be-trained action decoder network (5), thereby obtaining a sequence (6') of actions, and/or a processing result derived therefrom;
 - measuring (214), by means of a predetermined loss function (9), to which extent this sequence (6') of actions, and/or this processing result, is in

agreement with the sequence (6) of actions, and/or the processing result, in the training sample; and

- optimizing (215) parameters (5a#, 5a) that characterize the behavior of the to-be-trained action encoder network (5#) and the to-be-trained action decoder network (5) such that, when further training samples are processed, the rating (9a) by the loss function (9) is likely to improve.

12. The method (200) of any one of claims 10 to 11, wherein the obtaining (220) of a second pre-trained encoder-decoder arrangement of a state encoder network (3) and a state decoder network (3#) comprises:

- obtaining (221) training samples comprising values of a plurality of state variables (2) that characterize an operating state of the plant (1) or part thereof;
- providing (222) the values of state variables (2) in each training sample to the to-be-trained state encoder network (3), thereby obtaining a representation (2a);
- providing (223) this representation (2a) to the to-be-trained state decoder network (3#), thereby obtaining values (2') of state variables;
- measuring (224), by means of a predetermined loss function (10), to which extent these values (2') are in agreement with the values (2) in the training sample; and
- optimizing (225) parameters (3a, 3a#) that characterize the behavior of the to-be-trained state encoder network (3) and the to-be-trained state decoder network (3#) such that, when further training samples are processed, the rating (10a) by the loss function (10) is likely to improve.

13. The method (200) of claims 11 and 12, wherein the to-be-trained action encoder network (5#) and the to-be-trained state encoder network (3) are combined in one single network architecture.

14. The method (200) of any one of claims 10 to 13, wherein the obtaining (230) of training samples (7) for the training of the first encoder-decoder arrangement, and/or the second encoder-decoder arrangement, and/or the state-to-action network (4), comprises aggregating (231) training samples (7) obtained in multiple industrial plants (1).

15. One or more computer programs, containing machine-readable instructions that, when executed on one or more computers, cause the one or more computers to perform a method (100, 200) of any one of claims 1 to 14.

16. A non-transitory storage medium and/or a download product with the computer program of claim 15.
 17. One or more computers, configured to perform a method (100, 200) of any one of claims 1 to 14 by means of the computer program of claim 15, and/or by means of the non-transitory storage medium and/or download product of claim 16.
- 5