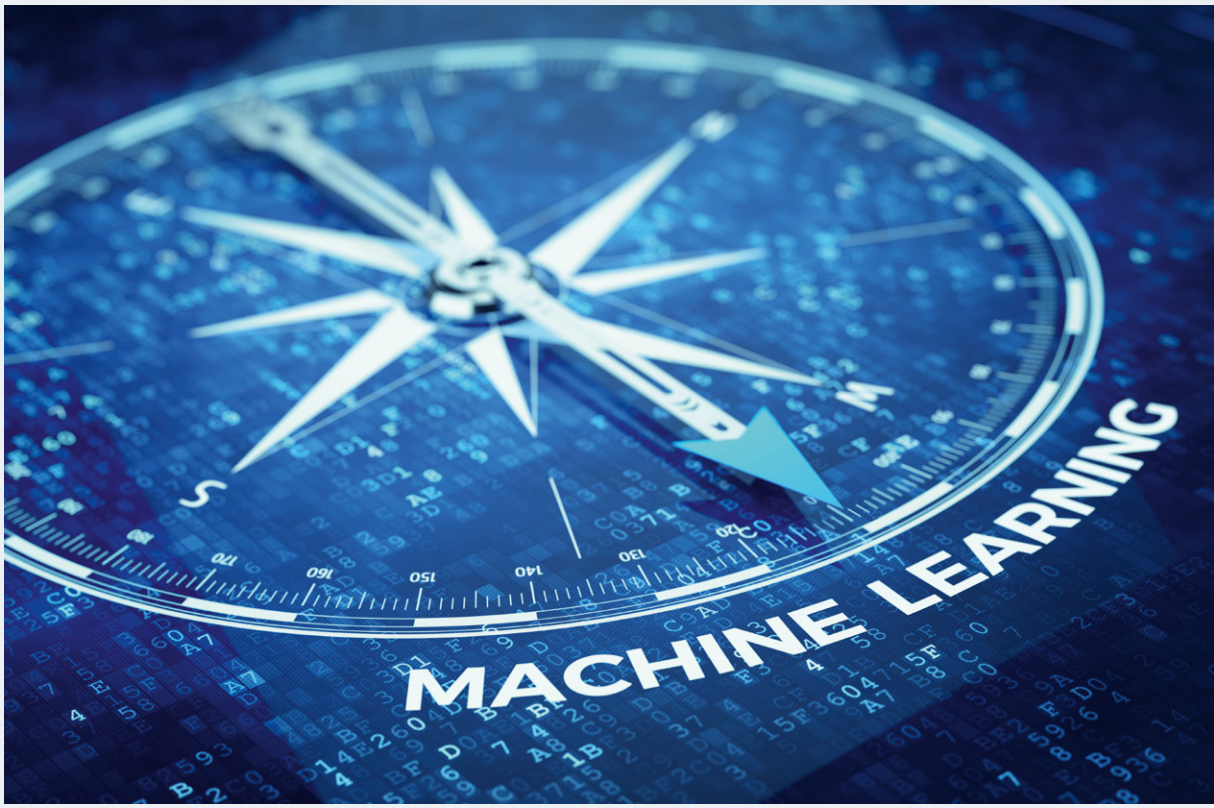
Guideline for Self-Learning Production Processes

A strategy for implementing reinforcement learning in practical industrial applications



in cooperation with

Institut für Unternehmenskybernetik

FKM Forschung im VDMA

VDMA

Forum Industrie 4.0

Editorial

Dietmar Goericke

Judith Binzer

Dear members,
Dear Sir or Madam,

Industrie 4.0 is taking on a key role in maintaining the success of the mechanical and plant engineering industry in the long term. At the same time, digitalization, networking and the integration of new information and internet technologies in products and processes are opening up new potential for business. Machine learning is an important technology for realizing the vision of the future encapsulated in Industrie 4.0. As a branch of artificial intelligence, machine learning offers exciting and new approaches for optimizing products and processes in mechanical and plant engineering.

With this in mind, these guidelines are intended to provide orientation and support. They give your company a tool for developing its own strategy for implementing the methods of machine learning and, in particular, reinforcement learning. To this end, they define basic principles and terms as well as defining guiding questions that will help your company find an implementation strategy. Accompanying toolkits assist you in answering these guiding questions.

The guidelines are the result of the InPulS project on intelligent and self-learning production processes. InPulS was executed as a precompetitive research project of VDMA Forum Industrie 4.0 in cooperation with the Assoc. Institute for Management Cybernetics (IfU) and a VDMA industrial working group supporting the project. The project was funded by the Mechanical Engineering Research Forum (*Forschungskuratorium Maschinenbau* – FKM) and VDMA between October 1, 2017 and September 30, 2019.

With these guidelines, VDMA has created an additional tool for implementing these new technologies in practical applications and has thus expanded the range of VDMA guidelines on the topic of Industrie 4.0. VDMA Forum Industrie 4.0 considers itself a trailblazer into the world of Industrie 4.0 for its member companies. At the same time, it serves as a networking platform for dialog and the exchange of experiences. With this publication, we hope that we have succeeded in offering a companion that provides orientation for mechanical engineering companies on the subject of artificial intelligence and machine learning and that can provide practical support.

We would like to thank Prof. Sabina Jeschke from the Assoc. Institute for Management Cybernetics (IfU) and her team for the scientific preparation of these guidelines. We would also like to thank the involved VDMA members for their participation in the working group accompanying the project, and in particular its chairman Dieter Herzig from AZO GmbH + Co. KG.

We hope it makes for exciting reading.
Yours,

Dietmar Goericke
Director of VDMA Forum Industrie 4.0 and the Mechanical Engineering Research Forum (FKM)

Judith Binzer
VDMA Forum Industrie 4.0
Research & Innovation

Contents

Guidelines for Self-Learning Production Processes

A strategy for implementing reinforcement learning in practical industrial applications

Prof. Sabina Jeschke

Artificial intelligence is permeating every industry sector and inspiring new technologies in a vast range of applications. This driver of innovation is also reshaping the world of industrial production – indeed, it is undeniable that AI will raise the production sites of the future to a higher level of efficiency. Machine learning is viewed as the key to the profitable realization of small batch sizes, all the way down to a batch size of one.

Two highly promising approaches are currently being pursued: supervised and unsupervised learning, which is based on large data sets, and reinforcement learning (RL), which is based on the principle of trial and error. Reinforcement learning algorithms find new and previously unknown solutions beyond the human understanding of processes. This new approach opens up enormous potential.

The impressive results of solutions such as Google's AlphaGo demonstrate what reinforcement learning is capable of. However, the ability to learn a game cannot be compared to controlling a production process. A game is a strictly monitored, structured environment, while in an industrial context, the framework conditions and uncertainties of the real world come into play and unforeseen events and malfunctions occur. Reinforcement learning enables a strategy adapted to these environmental conditions to be learned. Precisely this flexibility and adaptability is the core and strength of the reinforcement living method.

While large companies generally have R&D departments in which such procedures can be investigated, in-house developments of this nature present an enormous challenge for SMEs. For this majority of German companies, the 4.0 age poses the following key questions:

- How can German SMEs stay competitive in the age of Industrie 4.0?
- Which production processes are particularly accessible for the integration of AI in terms of the cost-benefit ratio?
- How can specialist expertise be built up in the area of AI in an efficient and sustainable manner?
- Which new business models or expansions are possible thanks to the use of AI?

The implementation of AI requires changes on all levels: Employees need training, new job profiles arise, processes change and new business models change the market. These guidelines offer an introduction into the topic of self-learning process control and serve as an orientation aid for implementing such processes.

The presented application scenario shows that, despite the special requirements, it is possible to use reinforcement learning methods in an industrial context and that these considerably raise efficiency.

Germany is investing in researching artificial intelligence like never before. It is essential that German SMEs seize this opportunity to shape the changes that will come!

Prof. Sabina Jeschke

Berlin and Strömsund, summer 2019

Management summary

Today, machine learning as a part of Industrie 4.0 is viewed as a decisive instrument for raising efficiency and an opportunity to develop new business models. In this context, however, the focus is often placed on digital fields of application; there is a lack of experience in how machine learning methods can be used in industry. In particular, the area of reinforcement learning for the autonomous control of production processes in industry has been barely tapped, if at all.

The goal of these Guidelines for Self-Learning Production Processes is therefore to provide small and medium-sized mechanical and plant engineering companies with a tool for developing their own strategy for introducing the methods of machine learning, and in particular reinforcement learning. In doing so, there will be an introduction into the terms and concepts used in the context of reinforcement learning and a description of the specific characteristics of industrial use. As such, these guidelines are not a ready-made solution for implementing industrial reinforcement learning, but rather provide support in developing an individual implementation strategy.

The goal of these guidelines is to provide companies with a tool for developing their own strategy for the industrial application of reinforcement learning.

Reinforcement learning is a subfield of machine learning which is particularly well-suited for learning an intelligent control strategy. It is recommended to first create a control strategy in a pilot project with clearly defined framework conditions, as many factors need to be taken into account for autonomous learning to be successful. These guidelines are intended to assist in the selection of such a pilot project and subsequently in the formulation of this project as a suitable problem for reinforcement learning.

The guidelines are divided into eight subsections. First, the potential of industrial reinforcement learning and the required change of perspective from plant to process control are described. There is then an introduction into the most important terms used in the context of reinforcement learning. The main part of the guidelines is made up of a list of guiding questions that need to be asked and answered in the company in order to find a suitable use case and develop an implementation strategy for this application. A toolkit is also provided, which is aimed at helping companies answer these guiding questions. The guidelines were created as part of the InPulS project on intelligent and self-learning production processes initiated by VDMA. During this project, reinforcement learning was applied for learning an autonomous assembly process and for a self-learning process on a bulk goods conveyor. The experiences, results and findings from these example applications will be summarized at the end of these guidelines.

Introduction and objective

Situation at the outset

Today's automation systems are increasingly equipped with a multitude of sensors, which are ever more closely networked with one another. The state of a system can be determined using these sensors. As the derived data enables new concepts and solutions, it harbors significant potential in industrial automation. In order to tap this potential, great importance is attached to the method of machine learning.

Machine learning is a subfield of artificial intelligence and comprises a large number of different concepts and methods, all of which use a pool of collected data in order to train a model for a desired task. They are divided into three categories: supervised learning, unsupervised learning and reinforcement learning. Supervised learning is normally used for classification and regression. The methods of unsupervised learning can be used to discover existing patterns and groups within the data and to assign the individual data points to these groups. Meanwhile, reinforcement learning is based on the principle of reward and punishment. Figure 1 shows an overview of the three principles. These guidelines deal with reinforcement learning. A detailed overview on the topic of machine learning can be found in VDMA's "Quick Guide – Machine Learning in Mechanical and Plant Engineering" (VDMA Software and Digitalization).

The named methods are suitable for use cases of various complexity. In an industrial context, there is particular potential in the areas of process monitoring, optimization and control. Figure 2 provides an overview of the three areas.

The area of **process monitoring** benefits directly from the increasing use of sensors in production plants. These sensors can be used to monitor the current status of the plant or make a simple prediction of its future state. These technologies enable increased process quality thanks to improved monitoring, reduced downtimes and a higher level of process reliability. In most cases, simple analysis methods are sufficient for realizing this monitoring.

Building on this, a process can be optimized with machine learning. This type of **process optimization** offers companies a great deal of potential in the form of raised efficiency and lowered costs. During optimization, an iterative process is used to find an optimum, e.g., an optimal task sequence, and the system moves

Machine learning		
Supervised learning	**Unsupervised learning**	**Reinforcement learning**
Classification	Cluster assignment	Interaction with the environment
Regression	Categorization	Reward principles

Figure 1: Overview of the various machine learning methods.

	Process monitoring	Process optimization	Process control
Provides	Situation detection and predictive information	Planning and decision support	Automated response to changes in the environment
Offers	Higher quality, reduced downtimes, lower shortfalls	Higher efficiency, improved usage, larger yields, more effective design	Increased production and productivity, lower labor costs, less waste
Requires	Data sources e.g., networked sensors	**Process monitoring** + Mature analytical tools	**Process optimization** + Integration of physical systems, e.g., robots
Methods	Visualization and descriptive statistics	Supervised and unsupervised methods	Reinforcement learning

Complexity

Figure 2: Use of machine learning in the various areas of automation with differing levels of complexity.

toward this optimum. Methods including those from the field of supervised and unsupervised learning are used here, which can be deployed for planning, decision support or other objectives.

The complexity increases further when the user wishes to control a process via machine learning methods, as optimization and execution on the physical system are closely interrelated. In process control, these steps are performed alternately or simultaneously. When using supervised and unsupervised learning methods in direct interaction with the process, one quickly encounters the limits of what is possible. The use of reinforcement learning is particularly promising here, as these methods require this direct interaction with the process in order to be successful.

There are already numerous example projects (see sources) that impressively demonstrate the performance of machine learning in process monitoring and optimization. However, little research has been carried out into the control of industrial processes to date due to its high level of complexity. In this context, the use of machine learning to control industrial processes enables a high degree of adaptability to unforeseen and unmodeled events, such as those caused by natural raw material fluctuations, variations in the weather and wear.

Principle of reinforcement learning

Reinforcement learning means learning through trial and error. This type of learning is similar to that employed by humans during early childhood, for example when a child learns to walk. In this case, the child knows what the target state looks like and tries it out according to the principle of trial and error until this target state has been achieved. In every attempted step, the child learns whether a behavior is constructive towards achieving this aim. At the beginning, the child's behavior is akin to haphazard experimentation, but then becomes more target-oriented over time. If we transfer this principle to a production process, various control signals are tried out within the specified scope of action of the actuators and the resulting reaction is evaluated on the basis of suitable criteria. The individual criteria are then summarized in an evaluation function that helps describe the process quality. An intelligent control strategy is thus learned over time.

Potential of reinforcement learning

The independent learning of an intelligent control strategy hold enormous potential, as it enables process optimization without modeling and thus a high level of autonomous flexibility.

Manual control strategies are based on expert knowledge gathered over the course of years. If companies lose this expert knowledge, however, they often have difficulties in finding a suitable replacement. Reinforcement learning can help in developing complex control strategies that function independently of expert knowledge.

A further advantage of strategies learned using reinforcement learning is that they leave well-trodden paths and can find wholly new solutions for known control problems, which are often more efficient than conventional strategies.

The best-known application of reinforcement learning is AlphaGo – the first computer program to beat the current world champion in the traditional Chinese game Go. Alongside the superiority of artificial intelligence, it is particularly impressive that the program won the game using a completely new strategy. As such, the solution space of AI was greater than the knowledge learned and optimized by humans over hundreds of years.

In the area of robotics, too, reinforcement learning has been applied with great success for learning a joining task (Schoettler et al., 2019) or for a drone flight (Sadeghi and Levine, 2016), among other things.

Simulations are a further tool for testing and optimizing parameter settings. However, modeling is frequently a problem here. Reinforcement learning can help with processes that are too complex to be replicated in a simulation. Using reinforcement learning, control strategies can be learned for both very complex processes and complex environmental conditions without having to explicitly model these.

Figure 3: Objective of the guidelines.

Another advantage of reinforcement learning is the possibility of determining a control strategy in real time where this would be too CPU-intensive using a simulation.

Due to the complexity of the procedure and its direct integration in the physical system, the industrial application of reinforcement learning brings with it a relatively large number of requirements at the beginning. Once these have been met, numerous examples show that self-learned control strategies are clearly superior to those created manually.

Objective and project background

These guidelines were created as part of the InPulS project on intelligent and self-learning production processes. Within the scope of this project, self-learning process control was developed using the example of a pneumatic bulk goods conveyor and a force-controlled joining process using a robot arm. InPulS was executed as a precompetitive research project of VDMA Forum Industrie 4.0 in cooperation with the Assoc. Institute for Management Cybernetics (IfU) within the Cybernetics Lab of RWTH Aachen University and a VDMA industrial working group supporting the project. The project was funded by the

Mechanical Engineering Research Forum (FKM) and VDMA between October 1, 2017 and September 30, 2019.

The goal of these guidelines is to develop a strategy for the implementation of reinforcement learning in industrial automation. They should enable the reader to recognize the potential and the necessary framework conditions for industrial application. With guiding questions and an accompanying toolkit, they are a tool for facilitating the implementation of reinforcement learning.

Target group

These guidelines are aimed at companies that wish to make their production systems more efficient using reinforcement learning and are looking for assistance with an implementation strategy alongside an orientation guide for the risks and potential.

Structure of the guidelines

The differences between a conventional control system and self-learning control with reinforcement learning will be explained in the following. This chapter will also introduce the necessary terminology and principles in the field of reinforcement learning. Following this, guiding questions and an accompanying toolkit for answering these questions will be provided in order to facilitate the selection of a suitable pilot project. There will then be an overview of the latest algorithmic approaches for reinforcement learning, which are to serve as the starting point for more detailed research. The procedure for integrating reinforcement learning will then be explained on the basis of a pilot project. A particular focus here is the question as to which of the various actors is responsible for which process step. Finally, the described recommendations for action will be illustrated using two actual example applications – an autonomous assembly process and a self-learning control system for a pneumatic bulk goods conveyor.

From plant to process control

If reinforcement learning is to be used, we must first fundamentally change the way we look at control systems. In a conventional control system, we mainly look at plant parameters in the form of the existing actuators and their respective adjustment ranges. These parameters are set on the basis of characteristic values known from experience or literature and are changed until a good control result is observed.

With control via reinforcement learning, the parameters of individual actuators are no longer considered; instead, parameters must be found that describe the process as a whole. As a result of this, the plant is initially parameter-free, with only the process parameters remaining. Therefore, it is necessary to describe what characterizes a good process. This process quality is then quantized using an evaluation function, which is known as a cost function in a reinforcement learning context.

In the learning process, this quantized process quality is for evaluating tested control strategies and reinforcing good strategies or avoiding bad ones. In this way, the system independently learns the plant parameters in such a way that an optimal process quality is achieved.

The control strategy learned through reinforcement learning can initially feel rather unusual for an expert, as completely new parameter ranges are often discovered. But this is exactly where the potential of this approach lies. In this context, expert knowledge is no longer used to set the plant parameters; instead, the new task of the expert is to create a good cost function. This must represent the process quality and ensure that a reliable and efficient behavior is learned. This task is of essential importance to the successful implementation of self-learning control and must be adapted individually for every process.

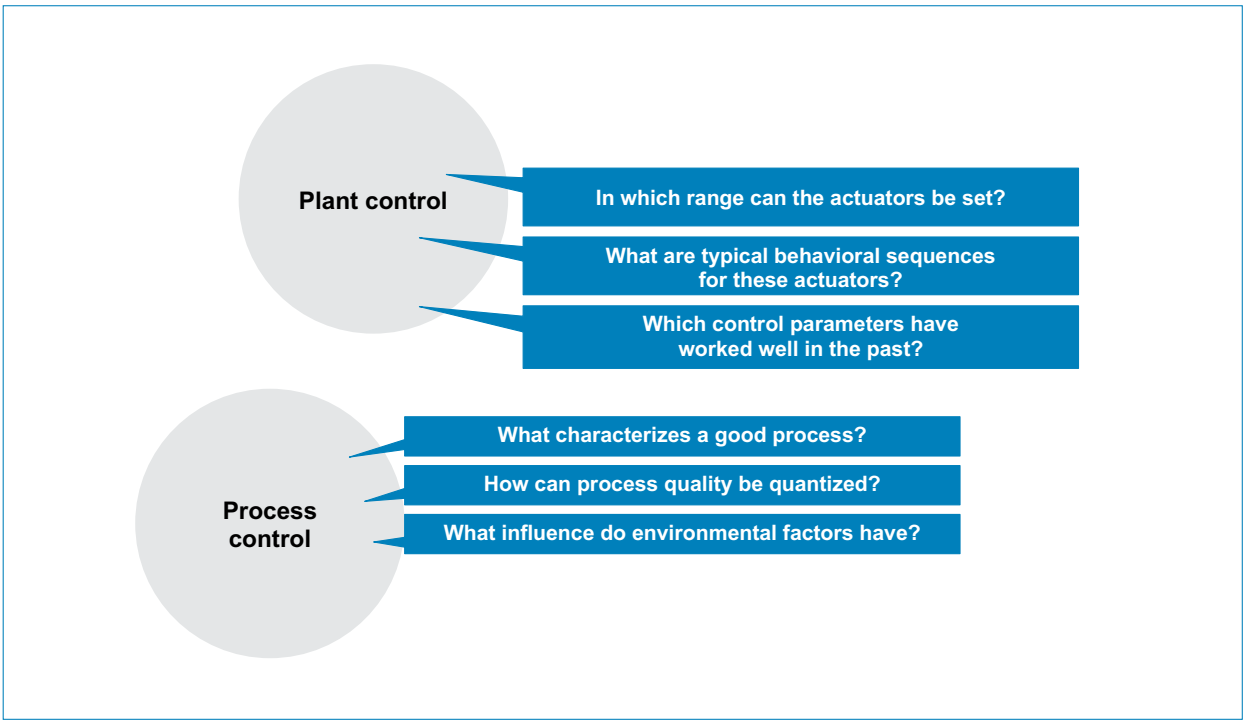


Figure 4: Reinforcement learning means a change of perspective: from plant to process control.

Reinforcement learning for industrial applications

Reinforcement learning allows a machine, i.e. a plant control system, to learn a complex relationship independently. The entire process does not need to be known in order to do this; instead, the solution is found and implemented step by step through trial and error. The principle and the necessary terminology are defined in the following:

The formal principle is shown in Figure 5. An agent influences its environment using one or more actuators. This action is then evaluated using a cost function. As feedback, the agent receives the new state and a cost value based on the cost function as an evaluation. On the basis of this, it performs another action in the next iterative step. This process is iterated until a sufficiently good result has been achieved.

Agent

The agent is an autonomous software program that assumes the role of the decision-maker in reinforcement learning. In each time step, it receives information on the current state of the environment or the system and a reward for performing the last action. Using this state and the current cost function, the agent determines the action for the next time step.

Environment

The environment is represented by the system to be controlled. This can be a production line, for instance. This system is characterized by a current state and can be directly influenced by actions of the agent.

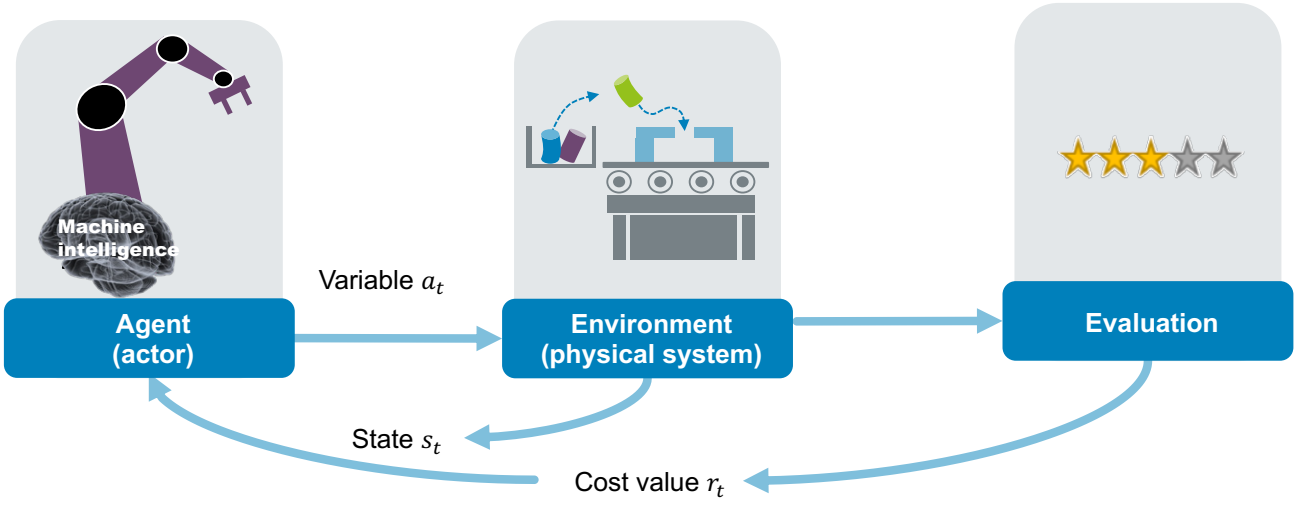


Figure 5: The reinforcement learning cycle: The agent selects an action or determines its variables, and in doing so influences its environment. It receives a new state and an evaluation of the performed action as feedback.

State and state space

The state of the system is described by means of the various sensor signals. Depending on the process, this can contain a camera on the end effector of a robot arm, or temperature, pressure, radiation or any other sensors.

The state space describes all states the system can assume. As such, it is dependent on the measuring range of the sensors among other factors.

Action / action space

An action contains the signals for all adjustable actuators of the system.

Accordingly, the action space describes the possible adjustment range of the actuators.

Cost function

The cost function describes the current process quality. The costs are not economic costs, but rather a reward or punishment for the action of an agent. The cost function is evaluated in each step. Following this, the actions of the next step are determined on the basis of the current cost function. This brings about the transformation from control of the plant parameters to the direct control of the process. The cost function must consider the process directly in order to optimize it in a targeted way. The quality of the cost function is essential for the success of self-learning control. This function makes it possible to let go of old control patterns.

Policy

In reinforcement learning, the policy describes the strategy of the agent. This depends on the current state of the system. As such, a strategy is learned that can optimally react to various states. Therefore, the term “policy” describes a kind of intelligent control strategy in a reinforcement learning context.

Training and value creation phase

In reinforcement learning, a distinction is drawn between two phases: the training phase and the value creation phase.

During training, as many process parameters as possible must be consciously chosen and checked. Therefore, the training environment should ideally be controllable and it should be possible for a process expert to explain the behavior of the system in this environment. In addition, the process expert must generate the greatest possible diversity of training data. In doing so, they should answer questions such as: How do environmental influences such as temperature, humidity, etc. affect the process? What are the different materials that could be produced/conveyed/processed? Are people / other machines also involved in the process and do these behave differently? All of these different scenarios must be depicted in the training data. If there is a sufficient level of variance in the training data, the reinforcement learning agent learns how to cover the various scenarios in its strategy.

After the training is complete, the value creation phase begins. In this phase, it is assumed that a sensible, optimal strategy has been found, which can now be applied. The exact conditions of the process must then no longer be strictly controlled in the production environment. Instead, the assumption is made that the agent’s strategy is adapted to the various process conditions and that these are recognized through the state of the system.

During the value creation phase, a reinforcement learning strategy can be transferred to various machines, sites, etc. However, if the new behavior deviates too strongly from the learned behavior, it may be necessary to retrain the process (and thus switch to another training phase). Figure 6 shows the two phases and their properties.

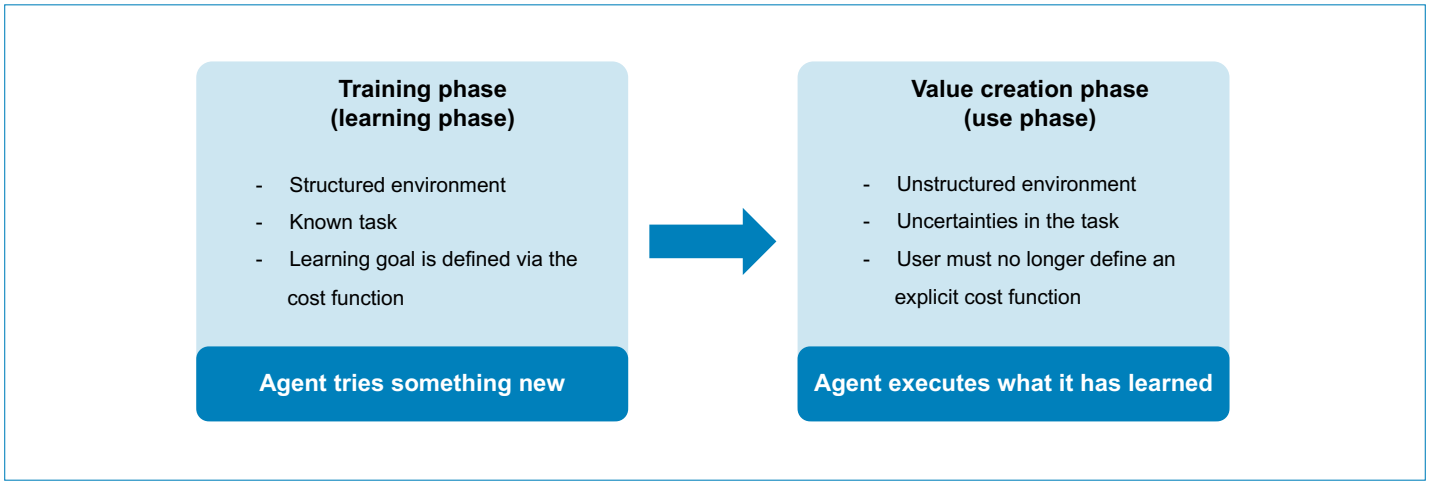


Figure 6: The application of reinforcement learning is divided into the training and the value creation phase.

What can be learned with reinforcement learning? And what cannot be learned?

Reinforcement learning means learning through trial and error. In industrial applications in particular, this has far-reaching consequences for the possible projects and the necessary framework conditions. Therefore, as illustrated in Figure 7, industrial reinforcement learning is characterized by the special requirements in terms of robustness, safety and the data efficiency of the algorithms.

A certain quantity of resources is needed in order to train a reinforcement learning model. These resources, often called training costs, should be kept as low as possible. The time expenditure is the most significant cost factor here. To train a reinforcement learning strategy, a large quantity of data needs to be generated by performing test runs of the real process. To this end, the process needs to be short and it must be possible to perform it repeatedly.

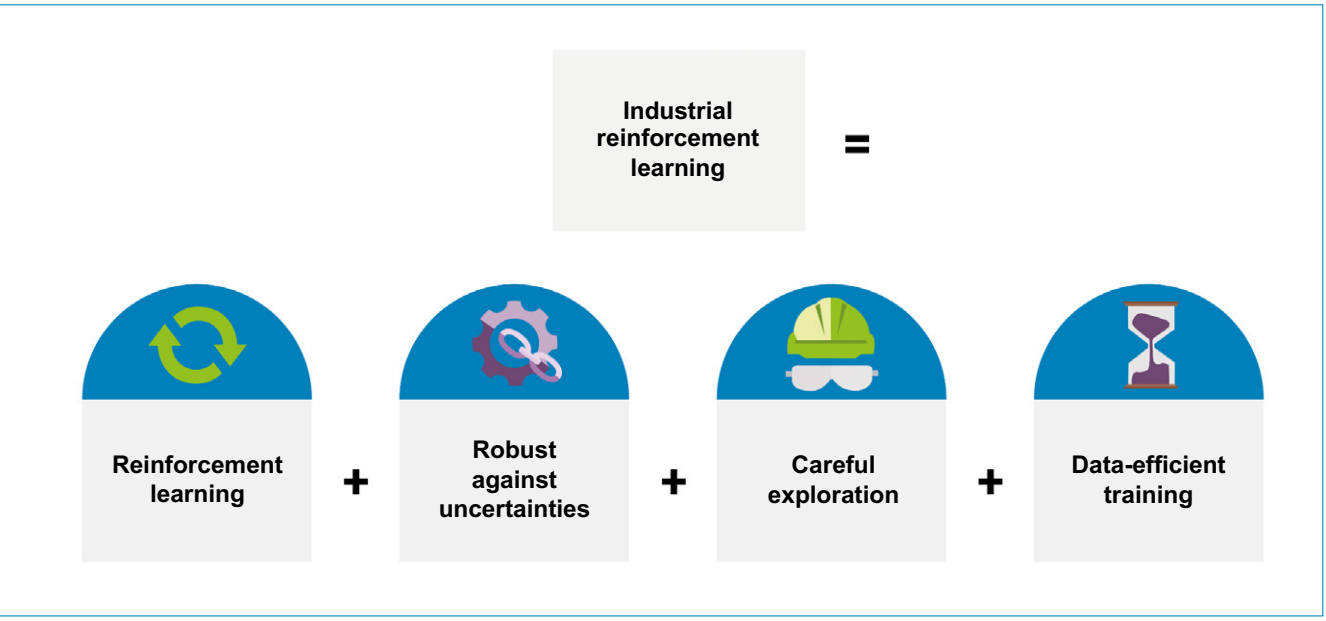


Figure 7: Industrial reinforcement learning places special demands on robustness, safety and data efficiency.

The training costs also comprise material resources such as the raw materials consumed by a process. These costs should also be kept as low as possible. On the whole, these training costs therefore place a particular demand on the data efficiency of the used algorithm.

As an alternative to the real process, training data can also be generated using a simulation. If a meaningful simulation of the process is possible, this makes it easier to generate training data. The time expenditure plays a subordinate sole in a simulation, as simulations do not necessarily have to run in real time and can be performed in parallel. Process faults do not represent a danger in a simulation and raw material costs can be ignored completely.

Furthermore, new and potentially unstable parameter settings are tested during training. In a simulation, these parameter settings can be tested without any danger. In a real system, however, the application requires careful testing of the unknown parameter settings in order to guarantee safety at all times. To this end, the real systems must be as tolerant to errors as possible, or it must be possible to detect the errors in good time and remedy them. Furthermore, the careful exploration of the environment must be taken into account even when choosing an algorithm.

The error tolerance of the system also plays an important role in its robustness against uncertainties. However, in this case it is also important to select or design the reinforcement learning method in such a way that it reacts robustly to uncertainties in its environment.

Guiding questions

Reinforcement learning is a highly promising option for the independent learning of complex control strategies. However, the implementation of such a strategy is relatively complex. It is therefore recommended to deploy this method for the first time within the scope of a suitable pilot project. This allows the necessary personnel and material resources to be built up slowly and enables the initial successes to be made visible. Important aspects for finding and executing a suitable pilot project are illustrated on the basis of the following guiding questions. The toolkit that follows in the next chapter provides assistance in answering the guiding questions introduced below.

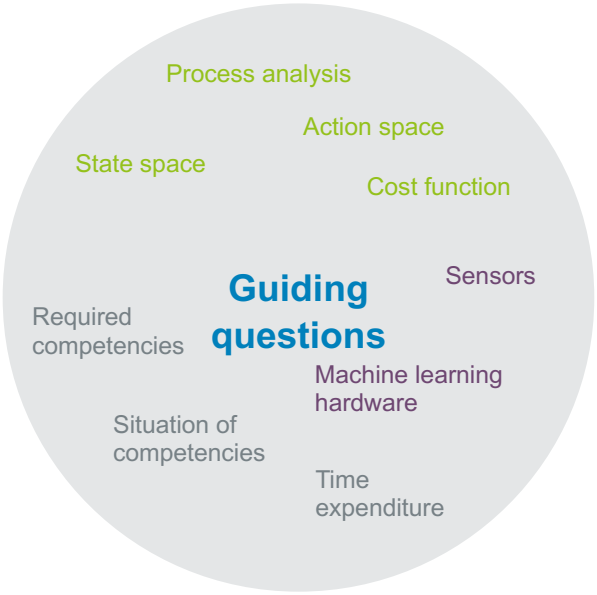


Figure 8: Overview of the important guiding questions regarding the process analysis (green), personnel resources (gray) and material resources (purple), which are important for an implementation strategy.

Process analysis: How can the existing system be characterized?

In this guiding question, it is determined whether a process is suitable for optimization with reinforcement learning. Various system categories are presented, such as the difference between discrete and continuous systems, partially and completely observable systems, and the frequent and rare feedback of quality parameters. The assignment of a process to one of these classes helps the user gain a better understanding of the optimization process through reinforcement learning.

Which target values are to be optimized in the cost function?

Once a possible process has been found, it must be examined in greater detail. Here it is important to define the target value to be optimized. The target value is part of the cost function and is crucial, as it characterizes the process quality. It must be possible to measure this value via quality parameters and influence it through the existing actuators. One example of such a target value is the flow rate in a pipe.

What are the state and action spaces of my process?

For a reinforcement learning procedure, input values are required in the form of measurement signals, as well as output values in the form of setting parameters. The inbound sensors describe the state space of the system. It must be determined whether the sensors describe the state of the system with sufficient precision to optimize the system behavior. The output signals for the existing controller describe the action space. Here, too, the existing actuators must have enough room for maneuver so that an optimum can

be found that is not restricted by the limit values of the actuators. To this end, the following questions must be answered: Can the system be controlled continuously? How precise is the controller?

Personnel resources: Which competencies are required?

After describing the process, the personnel requirements are now examined. Which personnel resources need to be present in the company in order to conduct a pilot project for the use of reinforcement learning in the respective company? A particular focus is placed on three different groups of people here. First of all, expert knowledge in the field of reinforcement learning is required. In addition, internal experts are needed who have a good knowledge of the process to be optimized. Among others, long-serving employees who have first-hand experience in manually optimizing the system are suitable for this. Finally, an experienced software technician is indispensable. In particular, this technician is responsible for providing an efficient and robust implementation for the application in the value creation phase.

Personnel resources: Where should the competencies lie?

For the three groups – reinforcement learning experts, process experts and software technicians – it must be decided where these competencies should lie.

These competencies can either be in the company itself, procured through external service providers or acquired through cooperation with universities. It is also conceivable that multiple SMEs collaborate in order to build up this knowledge together. In this connection, the following questions must be considered: How can reinforcement learning be built up as a competency in the company over the long term? Are there further possible use cases in the company in which the acquired competencies can be applied? If external experts are involved, how can it be ensured that the systems can also be operated, maintained and expanded later on?

Material resources: How can the plant be expanded?

In most use cases, special hardware is needed for training the reinforcement learning methods. The requirements for this hardware need to be specified. These are performance requirements that are dependent on the type and quantity of the data to be processed. Any real-time requirements also have to be taken into account. In addition, the communication between the existing interfaces and the new reinforcement learning module must be examined in terms of real-time requirements and system stability.

Toolkit for solving guiding questions

The background to the guiding questions posed in the previous section will be explained in the following. With this knowledge, it should be determined whether a process is suitable for reinforcement learning, or which requirements need to be put in place so that such a project can be realized successfully.

Process analysis

Before selecting the reinforcement learning method and planning the next steps of the project, the project to be optimized needs to be examined in detail. Here, the focus is on becoming aware of the process characteristics and their influence on the subsequent selection of the suitable reinforcement learning concept. This is where the aforementioned change in thinking from plant to process control has to take place. With this understanding, the process can be described in accordance with the "reinforcement learning philosophy." Figure 10 provides an overview of the most important terms used for describing the process to be analyzed.

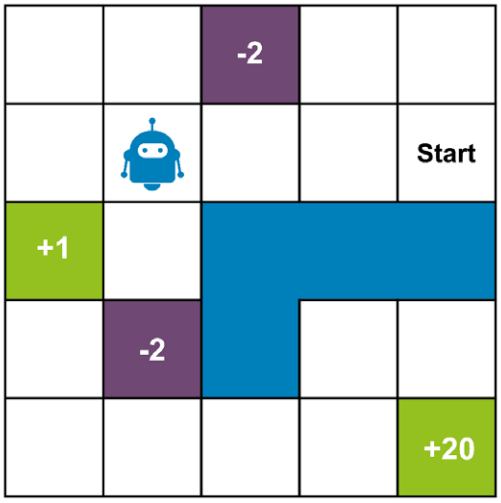


Figure 9: Discrete state and action spaces using the example of a grid world environment.

1. How can the existing system be characterized?
In the context of reinforcement learning it is important to distinguish between continuous and discrete processes and between state and action spaces. Discrete state and action spaces can be illustrated using the "grid world" shown in Figure 9.

Here, an agent is standing in a field on a grid. The agent cannot be placed freely on the grid; instead, it has (5x5) discrete options. In this context, the state of the agent consists of its current location. Because the current location can only have a discrete state, the state space of the agent is also discrete. The action space is also discrete: The available actions are "go one step up," "to the right," "to the left" or "down."

In contrast, the state of a robot arm can be described through its current position, speed and acceleration in all joints. As all of these values can be adjusted continuously, this is a continuous state space. The action space of such a robot arm often consists of a torque signal for each joint and is thus also continuous.

Another important property for characterizing processes is observability. Here, a distinction is made between partially observable systems and completely observable systems. In a partially observable system, some internal system states of the process cannot be measured directly. One example of this is a pipeline that is equipped with sensors at certain points. This means that, although the system is observable, the flow behavior is not known at all places in the pipe. The robot arm again serves as a suitable example of a completely observable system. Here, the attached sensors and the robot kinematics can be used to determine the current location and speed of every component with great accuracy.

In order to analyze the process further, the feedback of quality parameters should be examined next. Is feedback possible at certain times, or only after the process is complete? An example of this would be a casting machine, in which the quality can only be determined following the first cast and the subsequent cooling phase. Conversely, if the flow through a bulk goods conveyor is to be monitored, the conveyed product can be checked continuously using scales. Of course, there are also examples in which feedback of quality parameters takes place regularly, but is not possible on a continuous basis.

A further aspect to be considered is possible dead time of the system. Depending on the process in question, this can range from a few seconds to several hours. This time has an important influence on the time expenditure for training the system and should therefore be low. If there is a significant amount of dead time, it is even more important to select especially data-efficient reinforcement learning methods.

2. Which target values are to be optimized?

Following a precise analysis of the process characteristics, the target values to be optimized must be determined in the next step. All target values describe the process quality. On the basis of these values, it can be ascertained whether a process is well parametrized for the current task. These target values are then summarized in the cost function and optimized using a reinforcement learning approach.

In doing so, it is important to know the exact relationship between a target value and the process. A reinforcement learning method encourages what was specified through the target value and not necessarily what was intended by the process expert. For example, if a large amount of material is to be conveyed using a bulk goods conveyor, one can initially assume that a high conveying speed of the product leads to a large amount of conveyed material and, accordingly, to a good process. If the conveying speed is specified as a target value, however, it is possible that only a few particles

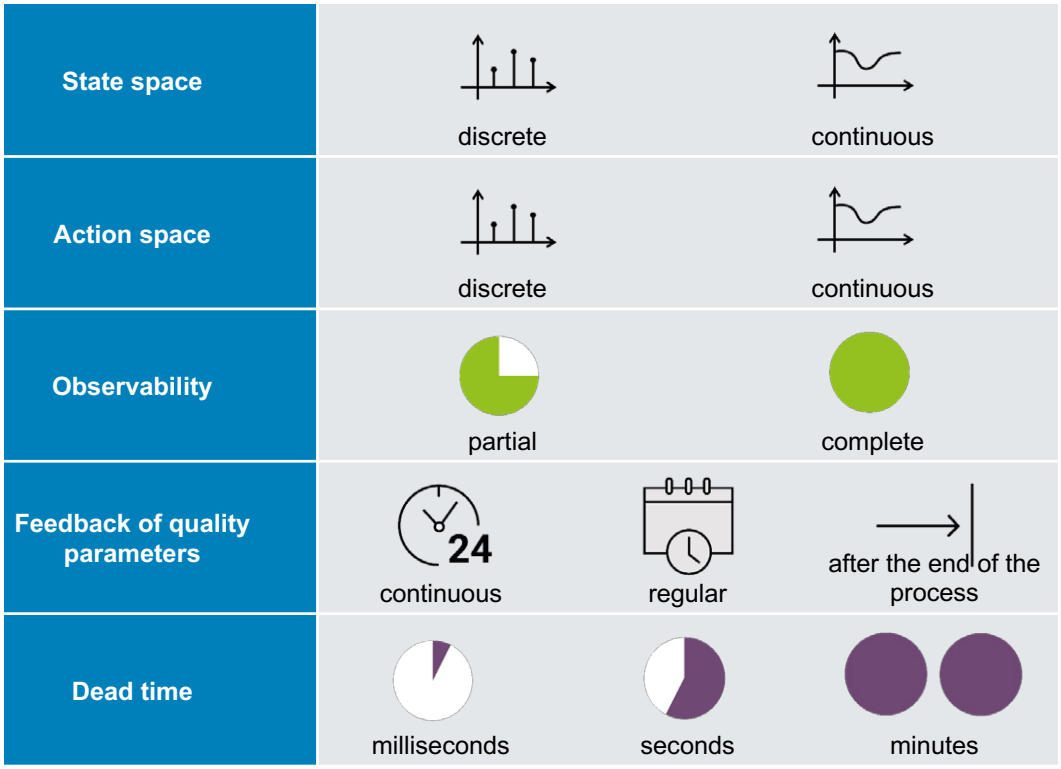


Figure 10: Toolkit for analyzing the process.

of the material are conveyed if these also reach a very high speed. As such, it is more sensible to select the weight of the conveyed material at the outlet of the bulk goods conveyor as a target value to be optimized.

3. What are the state and action spaces of my process?
After determining the target value, the state and action space can be defined. First, the existing sensors are observed. If these do not fully describe the state space, additional sensors need to be installed. The type of the sensors must also be examined in greater detail. Here, a differentiation can be made between sensors that only collect a single data point and those that gather a vector of data points. The first category includes temperature measurement sensors that gather a single temperature value, while the second group is made up of cameras that record a large number of pixels at the same time, among others. This differentiation is important, as these characteristics greatly influence the amount of data collected and thereby the computing capacity needed.

Not all reinforcement learning methods are suitable for large state spaces. A large state space is one from a size of around 12 dimensions, that is 12 individual sensor values.

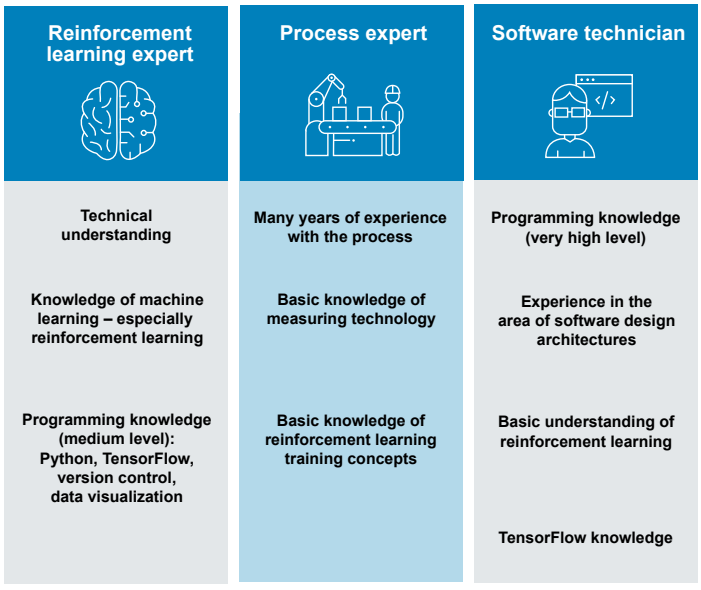


Figure 11: Toolkit for describing personnel resources.

Personnel resources

1. Which competencies are required?
To successfully conduct a pilot project with reinforcement learning in an industrial application, a broad range of different competencies are required. Figure 11 provides an overview of the necessary personnel resources and their competencies.

First of all, a **reinforcement learning expert** is required. This person should have a knowledge of mathematics, control engineering, optimization and statistics. Moreover, they need basic technical understanding tailored to the respective use case. In the area of machine learning, this expert must have in-depth knowledge in the areas of neural networks and reinforcement learning in particular, including knowledge of data visualization and interpretation. Reinforcement learning methods are often heavily dependent on their hyperparameters. These configuration variables define the exact architecture of the neural network and the training process.
For example, the number of layers and neurons in the neural network are hyperparameters, as is the number of training iterations. A suitable reinforcement learning expert should already have experience in adjusting these hyperparameters. Alongside knowledge of machine learning, an intermediate level of programming expertise is also indispensable for the reinforcement learning expert. Although a software technician is primarily responsible for the design and implementation of a good software architecture, the reinforcement learning expert must also have good knowledge in this area, as machine learning and its implementation in program code go hand-in-hand. This includes proficiency in version control in software projects. The reinforcement learning expert should also know about conventional software frameworks for machine learning, such as TensorFlow or PyTorch.

A **software technician** is essential for software that can be maintained and used. This individual is responsible for designing the software architecture and implementing it in the form of a deployable software solution. Therefore, the software technician needs to have a very high level of expertise in programming and the design of software architectures. Furthermore,

this person should have good knowledge in the field of algorithms and data structures in order to implement efficient reinforcement learning algorithms.

The **process expert** has the necessary knowledge of the plant and the process. Process experts often have many years of experience of the process and have a particularly good knowledge of the plant parameters. This knowledge is required for determining the state and action space, finding a sensible cost function and defining an initial policy. For this purpose, the process expert must acquire a basic understanding of reinforcement learning in order to view the known process in line with the reinforcement learning philosophy. Furthermore, as the process expert is responsible for selecting suitable sensors, they also need expertise in measuring technology.

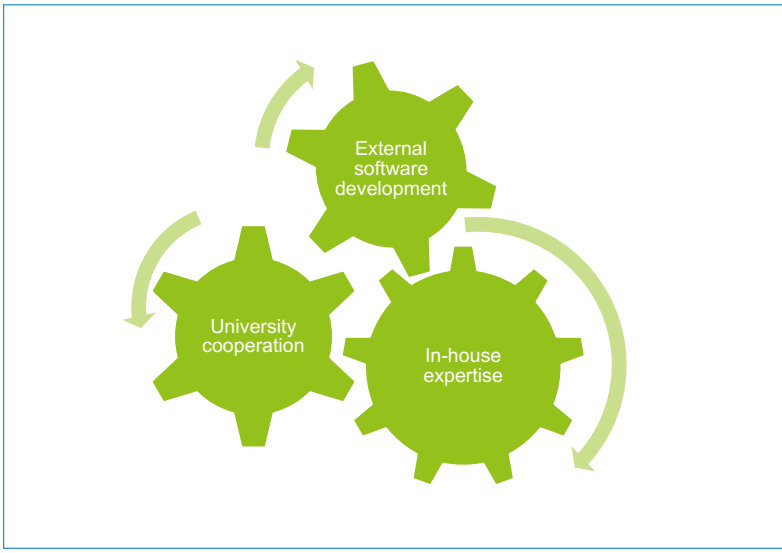


Figure 12: The various competencies can also be brought into the company through a cooperation with universities or by involving external software developers.

2. Where should the competencies lie?
Not all competencies need to be in the company itself. Depending on the internal company strategy, it can be prudent to outsource some competencies to cooperation partners.

In most cases, the **reinforcement learning expert** is not yet employed by the company itself. In this case, it is prudent to obtain external assistance in the form of a cooperation with a university or an external service provider.
If further reinforcement learning projects are planned following the pilot project, it can be sensible to build up these competencies in the company itself.

If the company already has a **software technician**, their competencies should be used where possible. If not, there are numerous ways to obtain this expertise. If there is an existing cooperation with a university, this can be acquired in the areas of both reinforcement learning and software engineering. Alternatively, there are several options for commissioning external service providers in the field of software engineering.

The competencies of the **process expert** lie within the company. The necessary knowledge of the plant, its parameters and its requirements can only be found here.

The time expenditure of the three groups of people named above (reinforcement learning expert, process expert and software technician) for implementing the project must be estimated. It may be advantageous to obtain external expert knowledge here. It is recommended to always estimate the feasibility and cost of external experts, even if all three areas of expertise are present in the company itself.

Material resources

A special hardware architecture is often needed for training neural networks. The following section will provide assistance in selecting this hardware. After choosing this hardware, the interfaces between the hardware and the plant control system also have to be defined. This interface must be examined, defined and implemented individually on a case-by-case basis.

1. What hardware do I need for a self-learning process?
On a home computer, computing processes are carried out on the central processing unit (CPU). Today, neural networks are increasingly trained on the graphic card, the GPU (graphics processing unit). The difference between CPUs and GPUs is easy to explain: CPUs

Figure 13: In order to use reinforcement learning, special hardware is needed. In particular, the interfaces between this hardware and the plant control system need to be defined.

are able to execute a few, but very complex, calculations. The advantage of GPUs, on the other hand, is that a large number of simple calculations can be performed in parallel. This is enormously beneficial for training neural networks in terms of speed, making them ideally suited to this application. Other processors can also be used for training neural networks. For example, Google tensor processors, or TPUs, were developed specifically for machine learning applications. In this case, a suitable architecture needs to be selected depending on the use case, the reinforcement learning method and the data quantity. In general, GPUs are well-suited for training neural networks, especially in the context of reinforcement learning.

2. What are the hardware requirements?
The performance requirements are dependent on the respective use case and cannot be universally specified.

As the GPU normally performs the largest number of calculations, special attention should be paid to using a powerful GPU. In the case of very complex calculations, TPUs (tensor processing units) optimized for the training of neural networks can be used. It is currently possible to rent processing time on TPUs. Furthermore, some of the latest architectures possess a small number of TPUs.

Alongside the actual performance of GPUs, the graphics memory on the graphics card is an important factor. Although this does not directly accelerate the calculations, a larger graphics memory allows more data to be processed on the GPU at the same time. In addition, it should be ensured that the GPUs have a high memory bandwidth and clock rate, as these make a key contribution to data transparency between the memory modules.

Even though the CPU does not normally perform the main calculations, it still takes on important tasks in the background. For instance, it loads the data to the main and graphics memory. A certain amount of computing power is needed here so that no bottleneck arises. As a rough guideline, mid- to high-class end user CPUs are currently sufficient.

Algorithmic approaches for self-learning production processes

Reinforcement learning methods generally begin with a data set. This data set contains the current state, the action performed and the associated costs for every time step of a training episode. The data set can then be used in various ways in order to optimize the intelligent control strategy – or the policy – learned by the algorithm. Standard reinforcement learning methods calculate at least one of the following values: a direct estimate of the current policy, an estimate of the value function, or an estimate of the system dynamics. The following section will introduce these terms and concepts and explain some of the methods based on these concepts.

Policy search

The policy search, or direct policy optimization, attempts to learn a parametrized policy on the basis of the collected data in an iterative process. To this end, the parameters of the policy are iteratively changed in such a way that the cost function is minimized. This procedure can be regarded as a numerical optimization problem. Figure 14 shows a schematic representation of a direct policy search. One decisive disadvantage of the direct policy search is the limitation in the number of parameters. The direct policy search currently only produces satisfactory results for policies with fewer than 100 parameters (Deisenroth et al., 2013). This limited complexity also restricts complexity in the task to be learned.

In the field of policy searches, a differentiation is made between procedures that use a derivative of the policy and those that do not need a derivative. The first are called **policy gradient** methods, while the second group are known as **derivative-free optimization** methods.

Value function

A second concept upon which several reinforcement learning methods are based is known as the value function. This value function should be differentiated from the cost function. It provides an estimate of the expected costs for every time step up to the completion of the training episode. These minimal, expected future costs are unknown and are thus estimated using a neural network. The estimated value function can be used to determine the optimal action for every state, i.e. the action that minimizes the expected costs. So that the value function converges more quickly, mathematical approaches can be used that enable the estimated costs to be adapted after each step in order to obtain a more accurate value function. This method is based on the principle of dynamic programming.

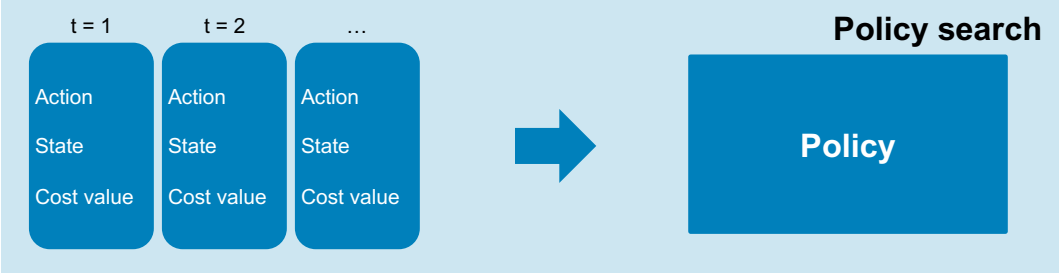


Figure 14: Direct policy search.

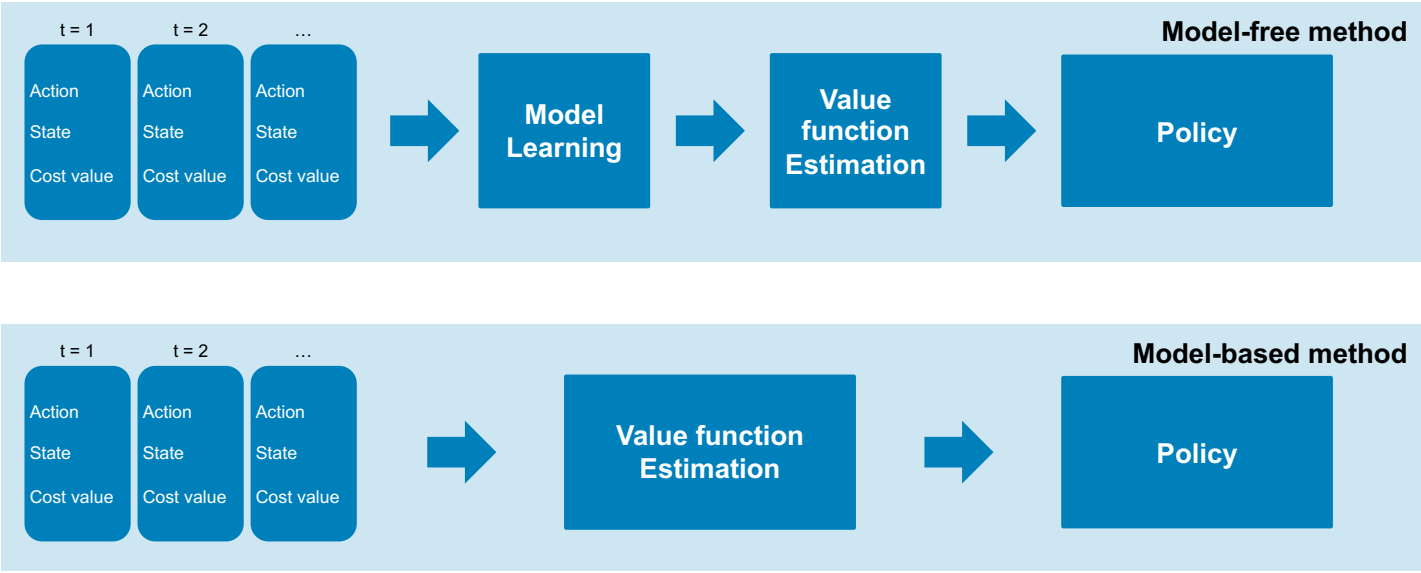


Figure 15: Difference between model-free and model-based reinforcement learning methods.

Model-free/model-based procedure

In the context of reinforcement learning, a distinction is drawn between model-based and model-free methods. The schematic representation in Figure 15 shows a comparison of model-based and model-free procedures.

The model describes the dynamic of the system, i.e. how the chosen action affects the system in each state of the agent. In a continuous system, this can be interpreted as the probability of a transition from a state A to a state B when a certain action is performed. This transition behavior is shown in Figure 16. One discrete example of this is chess. A proficient chess player has gathered knowledge of how their opponent will react and can thus go through the expected course of play in their head and choose the best scenario. Therefore, a model-based procedure will first learn the system dynamics. Using this knowledge, the reaction of the system can then be anticipated.

Model-free procedures do not use such a description of the dynamics; they are usually based on an estimate of the value function. However, this also means that they cannot use any knowledge of the dynamics.

Model-based procedures are generally far more data-efficient than model-free reinforcement learning procedures. As data efficiency is a key criterion for the industrial use of reinforcement learning, model-based procedures are often preferable in this context.

Methods

Most reinforcement learning procedures are based on these principles. However, not all of them can be clearly assigned to a category; more complex methods frequently combine more than one of these principles. Two highly developed groups of algorithms are the actor-critic method and the guided policy search method. Both of these approaches combine the policy search concept with a value function.

Actor-critic method

Actor-critic methods are classified as model-free procedures. They combine the principle of a direct policy search using a policy gradient and a value function. They consist of two parts, with each represented by a neural network. Figure 17 shows the structure of such an actor-critic method.

The task of the first neural network, known as the critic network, is to learn an estimate of the value function. This can provide an estimate of the minimum required cost up to the end of the episode for each state and time.

The second network, also known as the actor network, uses the current estimate of the value function to minimize the costs of the existing policy using a gradient procedure. In a training episode, as usual, a data set consisting of actions, states and costs is collected for the training episode. First, the critics are adapted on the basis of this data. Once the critics have determined an up-to-date estimate of the value function, the actor improves the policy, which is then forwarded to the environment.

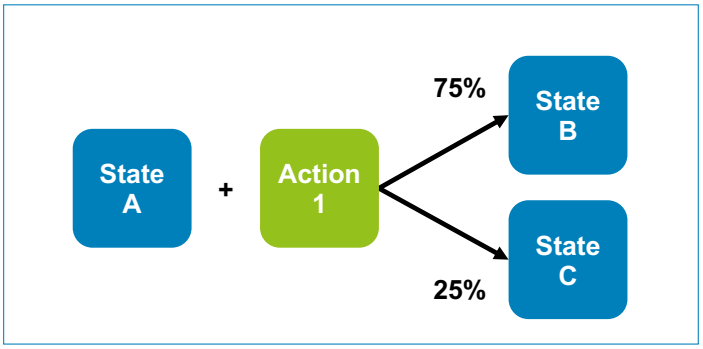


Figure 16: The dynamic model provides the probabilities of a transition from one state to another. If the agent is in state A and performs action 1, there is a 75% chance it will end up in state B and a 25% chance it will finish in state C.

Guided policy search

Guided policy search methods are among the model-based procedures. As described above, a description of the dynamic transition behavior is first learned. A stochastic dynamic is a prerequisite for this. This means that an action executed in a certain state does not always lead to the same next state, but can instead lead to different states with a certain degree of probability.

The special feature of a guided policy search method is that multiple local solutions can be learned for different training conditions. These training conditions can be, for example, different starting points for a learned movement of a robot arm, or different climate conditions when operating a production plant. A solution for a specific training condition is then called the local solution or the local policy. Learning a policy for a condition is a much more simple task than immediately training a policy that applies for all the different settings. After training the local solutions, a global solution is learned by means of a supervised training method. This global policy generalizes, which also represents a control strategy for the states outside the training conditions.

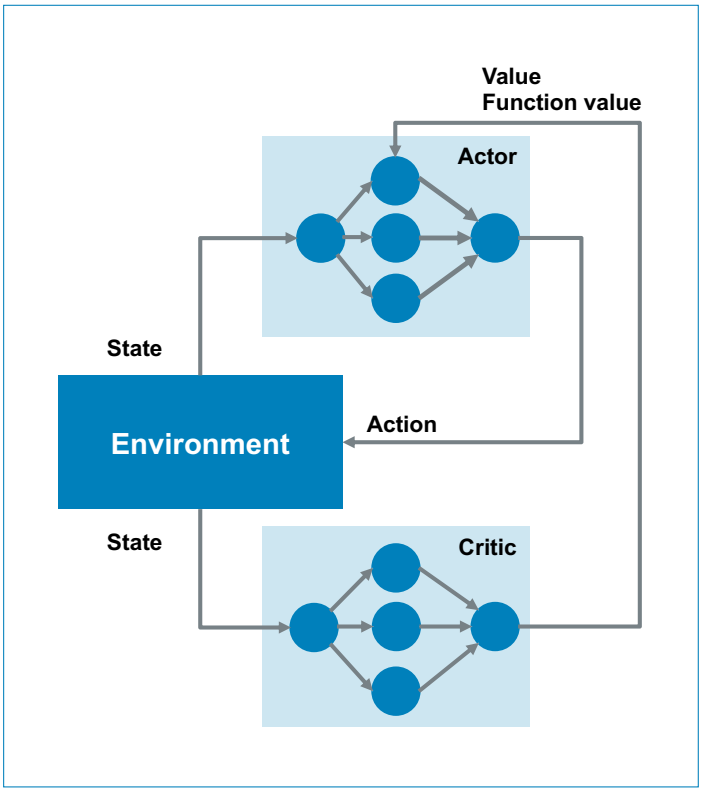


Figure 17: An actor-critic method consists of two neural networks, which are always trained alternately.

Procedure for integrating a reinforcement learning method

The use cases for reinforcement learning in industry are varied. In most cases, however, the procedure for integrating such a method follows a clearly defined pattern. This procedure can be divided into two phases, the planning and the realization phase. All in all, the integration can be split into eight successive steps. Figure 18 shows a schedule for the integration process.

1. Identification of a pilot project
First of all, a suitable pilot project needs to be found. The special requirements for industrial application must be considered in particular here. These include the robustness of the algorithms and the process, safety during training and the data efficiency of the reinforcement learning approach. The potential of a reinforcement learning method for the pilot project must also be evaluated.

2. Process analysis
During this step, the process is examined in detail. This is mainly within the process expert's area of responsibility, as sound knowledge of the plant and the sensors is required. The guiding questions and the developed toolkit can be used for assistance here. It is particularly important to view the process with the new "reinforcement learning philosophy." To this end, the agent, the environment and the accompanying state and action spaces must be defined. The optimization goal must then be defined and the components that are relevant for the cost function identified. Next, the currently available sensors and their quality must be examined.

3. Selection of the reinforcement learning approach
After analyzing the process, this knowledge can be used to choose a suitable reinforcement learning approach. This is the task of the reinforcement learning expert. The expert must take into account the special requirements for industrial use described in these guidelines. The algorithmic approaches described above can provide a starting point here.

4. Coordination
The process analysis is mainly performed by the process expert, while the selection of a reinforcement learning approach is the task of the reinforcement learning expert. Once these two tasks have been performed, the experts need to consult each other again. When doing so, the data necessary for the reinforcement learning approach and the existing sensors and actuators must be considered in particular. During this step, the requirements and prerequisites must be iteratively adapted until a match has been found, at which point the planning is completed and the realization phase can begin.

5. Implementation
The first step in the realization phase is the implementation. To this end, the reinforcement learning expert and the software technician need to be involved. It can also be prudent to involve a fourth expert, a control technician, who has detailed knowledge of the current plant. Together with the software technician, this person can define and implement the interface between the existing plant and the reinforcement learning hardware. The reinforcement learning expert and the software technician then work together to devise an initial prototype of the reinforcement learning method.

6. Prototypical learning cycles
Once an initial software design has been created, the first prototypical learning cycles can be executed. The data from these cycles must then be visualized and interpreted. The hyperparameters of the neural network can then be adapted on the basis of the visualization during the remainder of the process. Moreover, an initial evaluation and a possible adaptation of the cost function can be performed in this step.

Following the prototypical learning cycles, initial successes should confirm the choice of reinforcement learning algorithm, state and action space and cost function.

7. Code cleanup
Code cleanup comes at the end of the realization phase. Here, it is the task of the software technician to simplify the program code created during the course of the development and optimize it in line with efficiency criteria. In doing so, it is especially important to ensure that the documentation is of high quality. This documentation makes it easier to maintain the system and adapt the program code at a later stage.

8. Learning cycles
After code cleanup, learning cycles can again be run using the created software tool. With the data generated in this manner, the process and reinforcement learning experts can jointly evaluate the performance of the intelligent control strategy that has been learned. The increase in efficiency compared to the original behavior should be referred to as a criterion here. In addition, the reinforcement learning expert should analyze the convergence behavior during training.

Once a sufficient control strategy performance has been validated, it is possible to move from the training phase to the value creation phase. In order to do this, the control strategy must first be given a suitable format. Afterwards, this strategy can be applied in the value creation environment, which is not as strictly controlled as the training environment. The performance of the control strategy should be evaluated again in this environment, too.

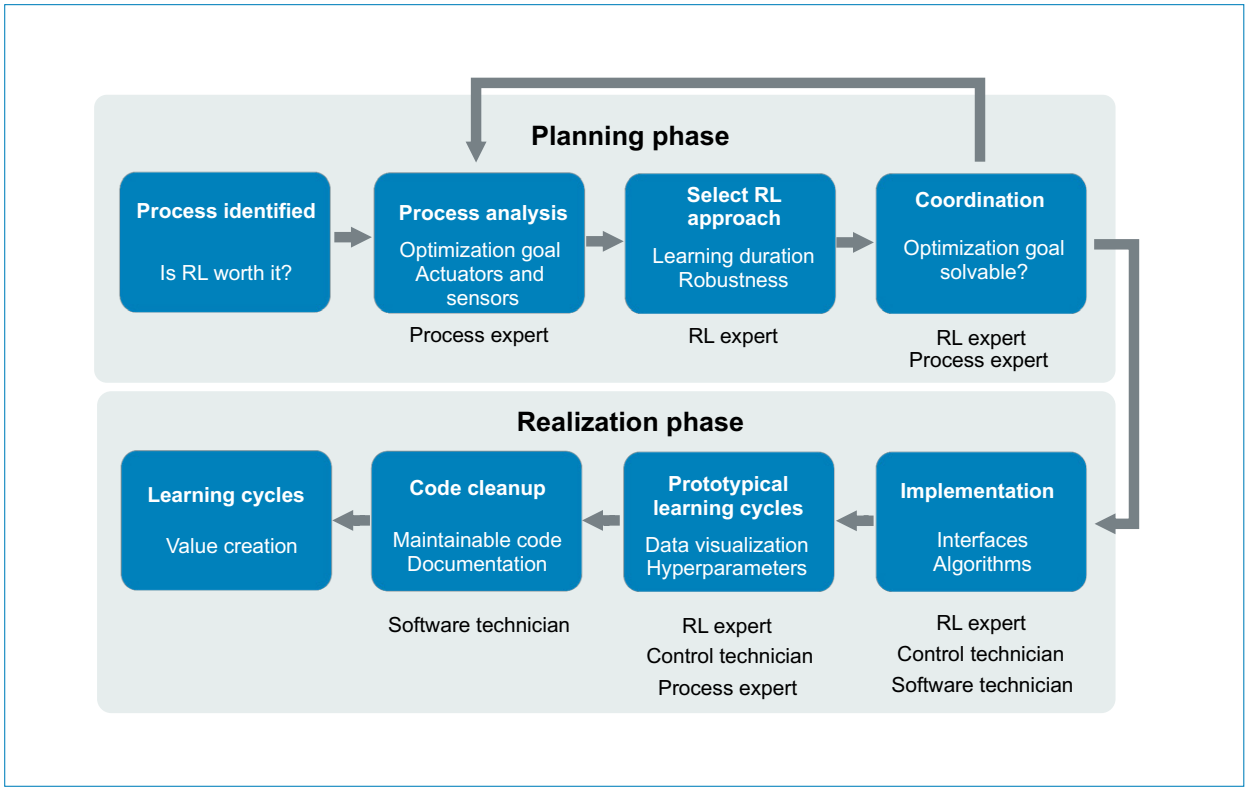


Figure 18: The process for integrating reinforcement learning consists of eight steps and is split into a planning and a realization phase in particular.

Example application

Autonomous assembly process

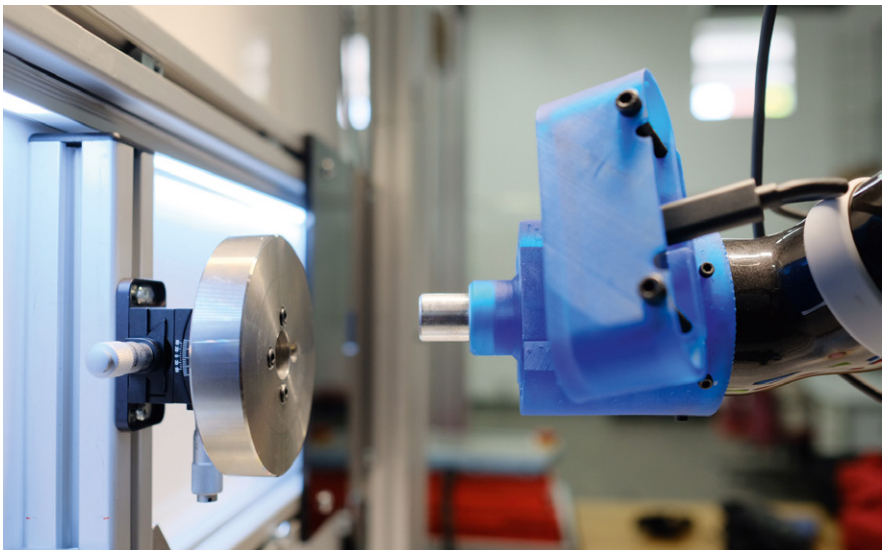


Figure 19: Autonomous assembly process on the basis of a pin-in-hole task.

Within the scope of the InPulS project, a scientific demonstrator was built at the Assoc. Institute for Management Cybernetics, which uses reinforcement learning to learn an autonomous, force-controlled assembly process. The long-term objective is to develop an autonomous assembly cell for non-plannable assembly situations. In this assembly cell, a robot independently learns an assembly movement without requiring an exact kinematic and dynamic description of the gripper system or the component. As an example of this assembly process, a joining task was examined, which is among the classic evaluation scenarios in robotics. Joining tasks entail a large number of contacts, and are thus complex learning tasks that often cannot be learned in simulations. In industrial applications, these pin-in-hole tasks often require a higher level of positioning accuracy than is possible with the latest robots.

Ø hole (mm)	20	20	20	20	20	20	20	20	20
Ø pin (mm)	19.9	19.8	19.7	19.6	19.5	19.4	19.3	19.2	19.1
Success rate	**3/10**	**8/10**	**8/10**	**8/10**	**10/10**	**10/10**	**10/10**	**10/10**	**10/10**

Table 1: Success rate of the autonomous assembly process with various pin sizes.

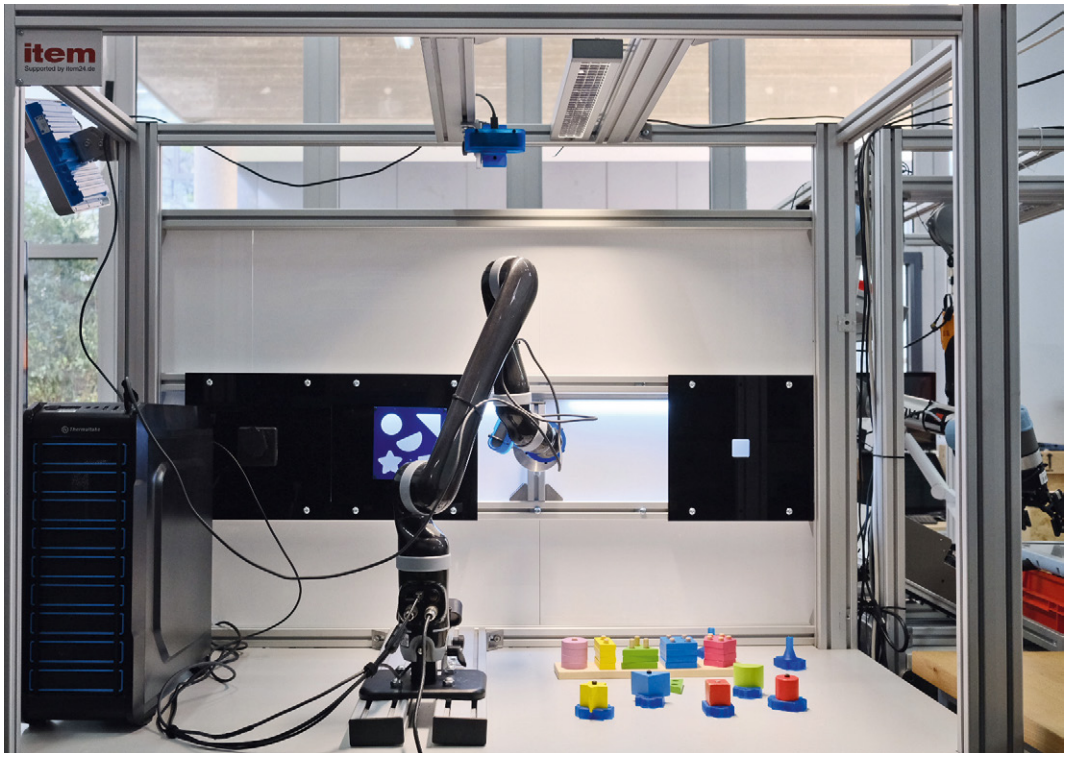


Figure 20: Autonomous joining process with differently formed objects.

Reinforcement learning setting

In the scientific demonstrator, a six-axis robot arm is used to learn how to insert a cylindrical pin in a hole. The state space of the robot consists of six joint angle settings and six accompanying joint angle speeds. Accordingly, the action space contains six torques, one for each joint. The task is to move the end effector of the robot arm to a predefined target coordinate. Accordingly, the cost function is defined by the distance between the end effector and the target point.

Learning process

The procedure for learning the assembly process corresponds to that of a child who learns to put a building block in the correct hole. To learn the movement, a method based on the guided policy search concept is applied. First of all, five random movement trajectories are tried out. In this case, the term “trajectory” describes the movement performed by the robot along a path from a specified starting point to a destination. Following this, the cost function is used to determine what is currently the best trajectory and a corresponding policy. New movements are then tested in the next iteration on the basis of the current best trajectory.

Results

The scenario was carried out with a hole diameter of 20 mm and various pin sizes in order to test the precision during assembly. The results of the test runs with pin diameters of between 19.1 and 19.9 mm are shown in Table 1. The success rate falls considerably with a pin diameter of 19.9 mm. Therefore, a precision of around 0.2 mm is possible with this method. It should be noted that a robot arm with a high position deviation was used for this scenario. A robot with a lower starting deviation may be capable of even greater precision.

Example application

Self-learning process on a bulk goods conveyor

Figure 21: Pneumatic bulk goods conveyor in the development center of AZO GmbH + Co. KG.

The second application scenario was also realized within the scope of the InPulS project in the development center of AZO GmbH + Co. KG. In this scenario, a process engineering problem was addressed using the example of a pneumatic bulk goods conveyor.

Application scenario

In industry, pneumatic conveyors are always used where a product, or bulk goods to be more precise, needs to be transported from one production location to the next. Bulk goods can be flour, sand or plastic powder, for example. The bulk goods are conveyed using a gas, typically air. In doing so, a distinction is drawn between pressure and vacuum conveying. In pressure conveying, the flow of air is created by generating pressure at the inlet, while in vacuum conveying this effect is achieved by generating a vacuum at the outlet (Hilgraf, 2019). In the use case in question, the air flow was created by a fan at the outlet. The fan speed and thereby also the air volume flow could be controlled by means of a frequency converter. The material flow at the inlet takes place via a metering screw, the metering capacity of which can be controlled via the rotation speed. The size of such industrial bulk goods conveyors can range from a few meters to several thousand meters (Hilgraf, 2019). The test plant examined in this use case has a conveying distance of 40 m.

A process plagued by uncertainties

The conveying of bulk goods is a process subject to a large number of uncertainties. Firstly, the various conveyable products have very different properties – for example, they can be coarse or as fine as dust, or can have diameters ranging from just a few micrometers to several centimeters. Moreover, when processing natural products such as nuts, there are often geometric differences between one product batch and the next. The material and flow properties of many bulk goods are also often dependent on weather conditions such as temperature and humidity. The dynamic process behavior brings further uncertainties; for example, there is a risk of a pipe suddenly becoming blocked during conveyance. This always happens when more material is fed in than can be transported away via the current air flow. Such blockages often occur unexpectedly and in most cases cannot be remedied automatically. In many cases, the pipe can only be unblocked again through manual intervention.

Due to the large number of uncertainties and the high maintenance effort in the event of a blockage, until now the conveying process was designed in a very conservative manner. As such, a robust flow process could be guaranteed even under unfavorable conditions. With the use of artificial intelligence, efforts are now being made to always keep the flow process in the optimal operating point and thus achieve a significant increase in efficiency in terms of the conveyed quantity.

The objectives of a self-learning control system for the bulk goods conveyor can thus be summed up as follows:

- The control strategy must be capable of independently adapting to new materials and environmental conditions.
- The control strategy must be as close to the optimal operating point as possible.
- The control strategy must have robust behavior in an environment characterized by uncertainties.

Reinforcement learning setting

As described in the guiding questions, the state space and action space will first be defined.

The continuous **state space** consists of a total of 18 sensors. These include

- 8 pressure sensors at various locations in the pipe
- 1 temperature sensor
- 1 humidity sensor
- 4 sensors for measuring the air and product speed at various locations in the pipe
- 1 virtual blockage sensor

The virtual blockage sensor assumes that a blockage is present when the air speed inside the pipe is zero.

The **action space** is also continuous and consists of the fan and the metering screw at the plant inlet. Both actuators are subject to technical constraints comprising a minimum and maximum speed and a safety mechanism.

In order to define a **cost function**, a good process behavior needs to be described in the next step. This can be defined using three criteria. Firstly, the actuators must behave as smoothly as possible in order to prevent unnecessary wear. In addition, the highest possible material throughput should be conveyed and blockages should be prevented.

A **model-based reinforcement learning** method was used for this. The model-based approach was chosen due to its data efficiency in particular. Specifically, a guided policy search method was used.

Learning process

Like the autonomous assembly process, the learning process is structured into episodes. This means that three different training runs are conducted and the state, action and cost values recorded. Using this training data, the system learns a current policy. This policy is then used for the next training cycle and is overlaid by noise. The addition of noise to the policy is especially important so that previously unknown states can also be reached. The conveying process of the bulk goods conveyor can be divided into three phases: the startup phase, the conveying phase and the deceleration phase. The entire process takes 90 seconds. As the individual training runs always have to start with the same initial state, the pipe is first cleared after a training run. At the beginning of training, an initial policy is selected as the starting point. When choosing this policy, robustness is prioritized ahead of the performance of the plant; however, an acceptable flow behavior should still have been achieved.

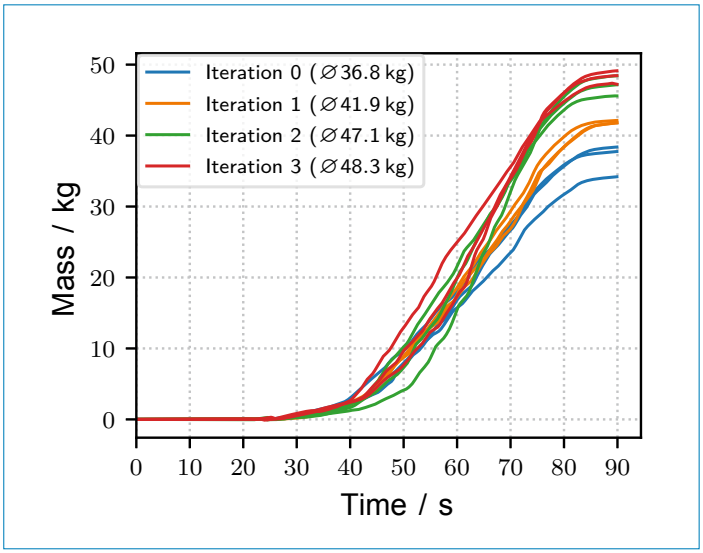


Figure 22: Conveying a plastic granulate. The blue trajectories show the conveying behavior of the initially chosen policy. The efficiency was increased by 31% during the course of the training.

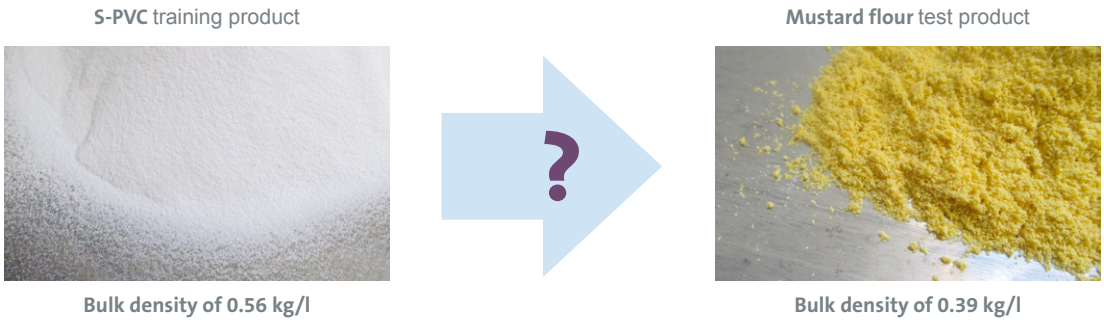


Figure 23: The policy learned for the training product is now applied to the mustard flour test product.

Results

First of all, a scenario for conveying a plastic powder (S-PVC) was tested. At the beginning of training, an initial policy was selected; this is shown in blue in Figure 22. This was followed by four training iterations. After this training, an average gain in mass flow of 31% was achieved. This result illustrates the enormous potential of reinforcement learning in industry.

After learning the process for the S-PVC powder, the learned policy was applied to the mustard flour raw material. Mustard flour has a different bulk density and, unlike plastic powder, is an oily product. As a result, its flow behavior is very different. One important factor for pneumatic conveyance and therefore also for the maximum possible mass flow is the bulk density of a product: A lower bulk density means that the maximum mass flow is lower. Therefore, the factor of bulk density must be deducted from the performance. If this factor is eliminated, the policy for the S-PVC powder, applied to the conveyance of mustard flour, can achieve a similarly good conveying behavior to the original training process. Between 75 and 100% of the original conveying performance is achieved using this method, as shown in Figure 24. When applying this policy, the conveying time was also extended from the original 90 seconds to 180 seconds while retaining the same level of efficiency. This shows that the learned policy also results in robust and efficient control of the bulk goods conveyor outside the training time.

All in all, the application on the bulk goods conveyor made it possible to highlight the potential of the reinforcement learning of complex and efficient control strategies. This control strategy can also be applied to new and unknown bulk goods and is capable of reacting to temperature and humidity fluctuations in a robust manner.

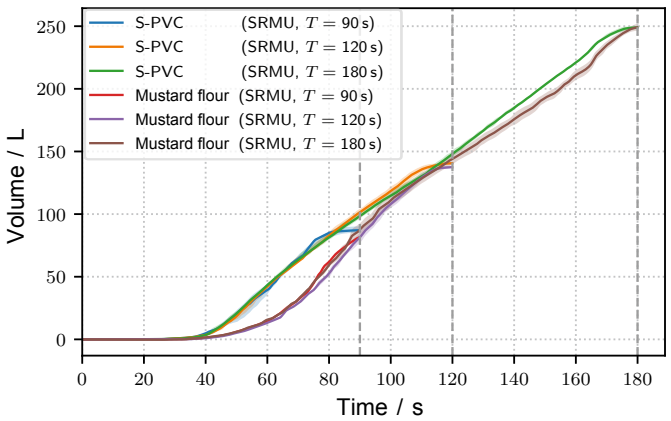


Figure 24: Application of the policy for plastic powder to mustard flour. The blue trajectory shows the originally learned policy for plastic powder. Using this policy, the material was initially conveyed over a longer period of 120 or 180 seconds.

Summary and outlook

The application on the bulk goods conveyor has confirmed the enormous potential of industrial reinforcement learning. Reinforcement learning makes it possible to control processes for which modeling with conventional methods would be too complex. As well as increasing efficiency compared to conventional control systems, the often time-consuming manual setting of the plant parameters is also no longer necessary. However, experience also shows that this kind of self-learning control either offers a great increase in efficiency or fails completely. Accordingly, reinforcement learning is not a simple introduction to the world of artificial intelligence.

Today, the potential of reinforcement learning as part of machine learning is only being discovered slowly. As a result, this topic is not yet being taught to a sufficient degree at universities and there is still a need for research. For the application on the bulk goods conveyor, an especially robust variant of a guided policy search algorithm was developed. The question remains as to whether such an algorithm can be applied to further scenarios. Further application-oriented fundamental research is needed here in order to develop new fields of application and corresponding algorithms for reinforcement learning. In particular, the special requirements of industry regarding safety and robustness must be met when developing these algorithms.

In industry, high expectations have been attached to big data over the last few years. Today, however, there is a trend away from big data and towards smart data. Contrary to the widespread belief that large quantities of data are available in industry, there are still many scenarios in which no comprehensive data collection is possible, for example in the area of special mechanical engineering. Especially data-efficient methods are required for these applications. Reinforcement learning fits in with this trend as only selected data that is tailored to the specific application needs to be collected.

Another trend is that away from central data processing in the cloud and towards increasingly decentralized processing. This bring advantages in the area of security and data protection, as well as in enabling existing real-time requirements to be met. The first industrial PCs are now available with an integrated machine learning chip, on which already trained models can be calculated. The focus here is usually on image recognition models with a direct camera interface, but special reinforcement learning modules are also conceivable in the future.

Project partners / imprint

Publisher
VDMA
Forum Industrie 4.0
Lyoner Str. 18
60528 Frankfurt am Main
Germany
Phone +49 69 6603-1810
E-Mail industrie40@vdma.org
Internet industrie40.vdma.org

FKM Forschungskuratorium Maschinenbau e.V.
Lyoner Str. 18
60528 Frankfurt am Main
Germany
Phone +49 69 6603-1681
E-Mail info@fkm-net.de
Internet www.fkm-net.de

Institut für Unternehmenskybernetik e.V.
[Assoc. Institute for Management Cybernetics]
Dennewartstr. 27
52068 Aachen
Germany
Internet www.ifu.rwth-aachen.de/en

Project management
VDMA Forum Industrie 4.0, Judith Binzer

Contributions
Institut für Unternehmenskybernetik e.V.
[Assoc. Institute for Management Cybernetics]
Philipp Ennen
Emma Pabich
Robin Kupper
Dr. Pia Benmoussa
Dr. René Vossen
Contact pia.benmoussa@ifu.rwth-aachen.de

Involved VDMA members from the InPuIS working group:
AZO GmbH + Co. KG
Festo AG & Co. KG
FIBRO GmbH
Hans Weber Maschinenfabrik GmbH
Karl Mayer Textilmaschinenfabrik GmbH,
Competence Center Parts & Components
Lenze SE
MAHLE Behr GmbH & Co. KG
Oskar Frech GmbH + Co. KG
Schaeffler Technologies AG & Co. KG
SchuF-Armaturen und Apparatebau GmbH
SMC Deutschland
TE Connectivity Germany GmbH a
TE Connectivity Ltd. Company
THEEGARTEN-PACTEC GmbH & Co.KG
Voith GmbH & Co. KGaA
Volkswagen AG
Weidmüller Interface GmbH & Co. KG
ZIMMER GmbH

Design and layout
VDMA DesignStudio / VDMA Verlag GmbH

Year of publication
2020

Copyright
VDMA, Institut für Unternehmenskybernetik e.V.

Image credits
Cover image: iStock / Olivier Le Moal
Page 1: VDMA
Page 3: Institut für Unternehmenskybernetik e.V. [Assoc. Institute for Management Cybernetics]
Page 26, 27: Institut für Unternehmenskybernetik e.V. [Assoc. Institute for Management Cybernetics]
Page 28, 30: AZO GmbH + Co. KG

Graphics
Institut für Unternehmenskybernetik e.V.
[Assoc. Institute for Management Cybernetics]

References

Deisenroth, Marc Peter; Neumann, Gerhard; Peters, Jan (2013): A survey on policy search for robotics. In: Foundations and Trends® in Robotics 2 (1–2), p. 1–142.

Hilgraf, Peter (2019): Grundlagen der pneumatischen Förderung. In: Peter Hilgraf (ed.): Pneumatische Förderung. Grundlagen, Auslegung und Betrieb von Anlagen, vol. 40. Berlin, Heidelberg: Springer Berlin Heidelberg, p. 109–232.

Sadeghi, Fereshteh; Levine, Sergey (2016): CAD2RL: Real Single-Image Flight without a Single Real Image. Available online at http://arxiv.org/pdf/1611.04201v4.

Schoettler, Gerrit; Nair, Ashvin; Luo, Jianlan; Bahl, Shikhar; Ojea, Juan Aparicio; Solowjow, Eugen; Levine, Sergey (2019): Deep Reinforcement Learning for Industrial Insertion Tasks with Visual Inputs and Natural Rewards. Available online at http://arxiv.org/pdf/1906.05841v1.

VDMA Software und Digitalisierung: Quick Guide. Machine Learning im Maschinen- und Anlagenbau 2018.

Note
The distribution, duplication and public communication of this publication requires the consent of VDMA and its partners. Excerpts from the publication can be used within the scope of the right of citation (§ 51 of the German Act on Copyright and Related Rights (UrhG)), provided that a reference to the source is included.

VDMA
Forum Industrie 4.0
Lyoner Str. 18
60528 Frankfurt am Main
Germany
Phone +49 69 6603-1810
Email industrie40@vdma.org
Internet industrie40.vdma.org

FKM Forschungskuratorium
Maschinenbau e.V.
Lyoner Str. 18
60528 Frankfurt am Main
Germany
Phone +49 69 6603-1681
Email info@fkm-net.de
Internet www.fkm-net.de

Institut für Unternehmenskybernetik e.V.
[Assoc. Institute for Management Cybernetics]
Dennewartstr. 27
52068 Aachen
Germany
Internet www.ifu.rwth-aachen.de/en

industrie40.vdma.org