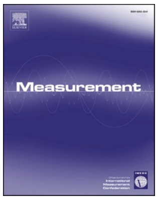
Contents lists available at ScienceDirect

Measurement

journal homepage: www.elsevier.com/locate/measurement



Reinforcement Learning for Statistical Process Control in Manufacturing

Zsolt J. Viharos [a,b,*], Richárd Jakab [a]

[a] Institute for Computer Science and Control (SZTAKI), Centre of Excellence in Production Informatics and Control, Eötvös Loránd Research Network (ELKH), Center of Excellence of the Hungarian Academy of Sciences (HAS), H-1111 Budapest, Hungary
[b] John von Neumann University, H-6000 Kecskemét, Hungary

ARTICLE INFO

Keywords:
Statistical Process Control
Optimal Control
Reinforcement Learning
Production Trends
Disturbance Handling

ABSTRACT

The main concept of the authors is to place Reinforcement Learning (RL) into various fields of manufacturing. As one of the first implementations, RL for Statistical Process Control (SPC) in production is introduced in the paper; it is a promising approach owing to its adaptability and the continuous ability to perform. The widely used Q-Table method was applied for get more stable, predictable, and easy to overview results. Therefore, quantization of the values of the time series to stripes inside the control chart was introduced. Detailed elements of the production environment simulation are described and its interaction with the reinforcement learning agent are detailed. Beyond the working concept for adapting RL into SPC in manufacturing, some novel RL extensions are also described, like the epsilon self-control of exploration–exploitation ratio, Reusing Window (RW) and the Measurement Window (MW). In the production related transformation, the main aim of the agent is to optimize the production cost while keeping the ratio of good products on a high level as well. Finally, industrial testing and validation is described that proved the applicability of the proposed concept.

1. Introduction

Artificial Intelligence (AI) and Machine Learning (ML) approaches are spreading across all areas in our live, it is also valid for technical fields, e.g., for the manufacturing sector as well. Nowadays, the increase in the speed of this expansion is growing, consequently, the intensity of changes and novel challenges require more and more attention with exhaustive research & development activities. Moreover, the frequently arising, novel AI and ML techniques have to be continuously adopted to the given domain to achieve the best match. This mission is valid also to manufacturing, while the well-known Industry 4.0 global initiative (called also as Industrial Internet or Cyber Physical Production Systems (CPPS)) supports, facilitates, moreover, incorporates these directions, consequently, the actual situation in this sector is quite promising.

There are various areas of the AI discipline (e.g., machine learning, search techniques, multicriteria optimization, inference and expert systems, graph modelling and traversal…), nowadays, the so-called Deep Learning (DL) receives the highest level of attention, making it to the most fashionable solution, while sometimes some may forget the other very important areas of AI. In general, ML is one of the key, basic foundations in AI, originally this branch started with the two directions of supervised and unsupervised learning. However, in the 80s, the early, pioneering results of Sutton [1] with his professors and colleagues extended this range to the Reinforcement Learning (RL). Currently there are also further combinations of such techniques like semi-supervised learning.

The spread of various artificial intelligence and machine learning techniques in manufacturing is valid for reinforcement learning as well. However, as described in the next paragraph, the reviewing the *scientific literature mirrors* that the domain specific *adaptation of reinforcement learning to various production fields concentrates mainly, only to production scheduling and robotics*. This state-of-the-art status provoked the motivation to extend and adapt reinforcement learning to further potential fields of manufacturing. So, *the current paper introduces novel RL based, Statistical Process Control (SPC) in manufacturing, with various additional novel components*:

- *The main contribution of the paper is the adaptation concept of reinforcement learning to the process control in manufacturing.*
- *A novel, general* (manufacturing independent), *dynamic Q table handling for RL is described*, even if it was motivated by the production adaptation challenges.
- The specialties of industrial process control led to the *introduction of the so-called Reusing Window (RW) in RL based SPC in manufacturing.*
- To compare the efficiencies of various RL solutions in production SPC, *the Measuring Window (MW) is introduced.*

[*] Corresponding author.

https://doi.org/10.1016/j.measurement.2021.109616
Received 15 February 2021; Received in revised form 2 April 2021; Accepted 10 May 2021
Available online 6 June 2021

- The features and behaviour of the related, *SPC in production, specific simulation* are described.
- A *novel, dynamic*, (manufacturing independent) *self-adaptation method is presented to control its own exploration–exploitation rate by the agent itslef*.
- *Industrial testing and validation is decribed that proves the applicability* of the method proposed.

The paper is organized as follows. After the current introduction, the actual status about the reinforcement learning in production field is summarized. The third paragraph reviews the current SPC solutions in manufacturing followed by the novel approach to introduce RL in production, especially for SPC assignments. Its specialized extensions are introduced in the next paragraph, after the introduction of the Reusage Window (RW) and Measurement Window (MW) the proving results of an Industrial Test and Validation are presented. Conclusions with Outlook, Acknowledgements and References close the paper.

2. Reinforcement learning in production

Despite the given, highly potential situation, the state-of-the art literature mirrors that RL applications in manufacturing concentrate mainly only on two fields: *production scheduling* and *robotics*.

In production *scheduling*, the state-of-the-art for dynamic scheduling shows a growing increase for the use of RL algorithms. Several papers use Q-learning [2,3,4], deep Q-learning [5] and adapted versions of Q-learning [6,7]. Most cases focus on value-based algorithms [2,3,4,5,8], however a few papers like [4,7] are policy-based. Some researchers used the epsilon-greedy method [3,4,5], whereas Bouazza et al. [2] used it in addition to the machine selection rule. While Kuhnle et al. [7,8] considered the architecture of a RL algorithm framework, Qu et al. [10] analysed the optimal assignment of multi-skilled workers. In papers [2,5,10] multi-agent architectures were realized. In general, researcher applied a simulation tool to test and validate their approach. E.g., Kardos et al. introduced a Q learning based RL architecture into the scheduling/dispatching decisions in productions systems and proved on simulation basis that their solution significantly reduces the average lead time of production orders in a dynamic environment [9]. Moreover, it was shown that as the complexity of the production environment increases, the application of RL for dynamic scheduling becomes more and more beneficial; that makes the future production systems more flexible and adaptive.

In the field of *robotics* applications of RL, Nair et al. presented a system with a RL component for solving multi-step tasks [11]. The report by Plappert et al. [12] introduced a suite of challenging continuous control tasks and also a set of concrete research ideas for improving RL algorithms. Zhu et al. combined reinforcement and imitation learning for solving dexterous manipulation tasks from pixels [13]. Kahn et al. presented a high-performing RL algorithm for learning robot navigation policies [14]. Long et al. optimized a decentralized sensor level collision avoidance policy with RL [15] and Johannink et al. studied the combination of conventional feedback control methods [16].

There are some first trials for applying RL also in the field of process control in manufacturing, however, such *papers are particular and much rare* than those for scheduling and robotics, so, these results have to be introduced in more detail.

Large variety is one of the challenges in brine injection into bacon food products, requiring an adaptive model for control. Injection pressure and injection time can be controlled by an adaptable Deep Deterministic Policy Gradient (DDPG) reinforcement learning model presented by Andersen et al. [43]. The DDPG could adapt a model to a given *simulated environment that is based on 64 conducted real experiments*. With a target setpoint of mass increase of 15% it was capable of producing a mean of 14.9% mass increase with a standard deviation of 2.6%.

The main contribution of Beruvider et al. was the design and implementation of a reinforcement learning architecture for default pattern identification in multi-stage assembly processes with non-ideal sheet-metal parts [44], in which *the failure patterns are generated by simulation*. The architecture was composed by three modes (knowledge, cognitive and executive modes) combining an artificial intelligence model library with a Q learning algorithm. The results presented by three methods (MLP, SOM and MLP + GAs) achieved high precision level regarding the different measurement parameters generated especially for this training and validation. The architecture extension by a reinforcement learning algorithm (Q-learning in this case) resulted in further benefits because its helps to determinate parameters for the models that enabled better adjustment to the different changes of the experimental data and to the different work regiments.

The paper of Guo et al. [45] introduced a professional, hybrid framework for optimizing and control of an injection moulding process for lens production. *The model is pre-trained by a simulation framework establishing the initial knowledge that can be promptly applied for on-line optimization and control*. Finally, the complete process was controlled staying inside the prescribed quality control chart. This paper is excellent, and the approach is robust. In comparison to genetic algorithm and fuzzy inference based optimization and control methods, the proposed reinforcement learning based solution outperformed them significantly.

Kader and Yoon [3] proposed a methodology for the stencil process parameter estimation in the surface mount technology (SMT) of circuit boards. The aim was to build an optimal adaptive controller to enhance the solder paste volume transfer during the process. Printing speed, printing pressure and separation speed in a discretized coding formed the state space of reinforcement learning and two prediction models are built to estimate the reward functions, which are average (avg.) and standard deviation (std.) of solder paste volume transfer efficiency (TE). To estimate the immediate reward, after predicting the avg. and std. of volume TE, *the capability indices C_{pk} and C_{pkm} were calculated for each component, in this aspect the results are really harmonized with the SPC techniques. If these values are above a threshold, the reward was 1, if not, the reward was* -1. The *simulation*-based testing results showed that the agent successfully arrived at the terminal state with taking only few actions.

Li et al. proposed a novel neural network [46], called a Reinforcement Learning Unit Matching Recurrent Neural Network (RLUMRNN), with the aim of resolving the problem that the generalization performance and nonlinear approximation ability of typical neural networks are not controllable, which is caused by the experience-based selection of the hidden layer number and hidden layer node number. This was the main contribution of the paper, and so, the *simulation* of the training data is easy. *In the application side, a discriminator was constructed for dividing the whole state degradation trend of rolling bearings into three kinds of monotonic trend units: ascending unit, descending unit and stationary unit.* By virtue of reinforcement learning for the recurrent neural network, its hidden layer number and hidden layer node numbers was fitted to a corresponding monotone trend unit that was selected to enhance its generalization performance and nonlinear approximation ability.

By taking advantages of the concept *a new state trend prediction method for rolling bearings was proposed*. In this prediction method, *the moving average of the singular spectral entropy was used* first as the state degradation description feature, and then this feature was inputted into the model to accomplish the state trend prediction for rolling bearings. The same concept was copied and re-published by Wang et al. [47], the only differences were that it optimizes the structure of a deep neural network and the test environment is a test-rig and a locomotive bearing.

In the paper of Ramanathan et al. [48] a reinforcement learning based smart controller was proposed, and its performance was demonstrated by using it to control the level of liquid in a non-linear conical tank system. The advantage is that a standalone controller is designed on its own without prior knowledge of the environment or the system. Hardware implementation of the designed unit showed that it controlled the level of fluid in the conical tank efficiently, and rejected random disturbances introduced in the system. This controller provided an edge over PID, fuzzy, and other neural network-based solutions, by eliminating the need for linearizing non-linear characteristics, tuning PID parameters, designing transfer functions, or developing fuzzy membership functions. *A significant advantage of this concept was that every component of the reinforcement learning, and its environment was connected to a real application, and real measurements were feed-back to the simulation model* and updated the knowledge base. This makes the paper and the results superior. *In the real application the agent performed only exploitations, while the simulation served with the mixed exploration–exploitation background.* First, the initial Q matrix was obtained by running simulation in MATLAB, without interfacing directly with the real time system. Later, the Q matrix was updated by many trials in the experiments.

After reviewing various cases a special aspect, the number of the required reinforcement learning training steps, moreover, the simulation background to support it have to be considered. *Production scheduling* is performed by a computer attached to the manufacturing system, and it is natural to “play forward” the future states of the complete system (plant, process steps, etc.), partly because the required basis information is available also for the scheduling itself. Similar is the situation at *robotic applications* when the (e.g., 3D) simulation of the robot movements is an available, basic service in the movement (e.g., trajectory) planning. Consequently, *both fields naturally serve with a valid simulation component that makes the application of reinforcement learning easier*. Also, *each presented process control examples apply a simulation component, that are based either on a physical model of the environment or on a fitted model to a smaller number of real experimental data.* This simulation requirement is inherited in the RL applications because of the typically high number of required training steps. Moreover, the exploration steps of RL would cause too high costs, risks or disconformities in real manufacturing environments when. *The current paper applies also the simulation-based approach* while the basic information of the SPC is inherited from real manufacturing. Data and human experts-based knowledge are included as well, so, the basis is a validated, mixed human–machine information source.

Considering the RL adaptations to various industrial/manufacturing fields, there are many open issues and challenges, this is valid also for bringing forward the RL application to the field of SPC in production. Beyond the rare and particular cases presented as process control in the given industrial assignments, *the current paper proposes a generalized, widely applicable Reinforcement Learning for Statistical Process Control (RL for SPC) framework* to keep the production between the prescribed Lower Tolerance Limit (LTL) and Upper Tolerance Limit (UTL) of the related Statistical Control Chart (SCC), moreover, *important novel extensions are introduced as well.*

3. Statistical process control in manufacuring

Statistical Process Control (SPC) in manufacturing is addressed in the scientific literature around the phrase of Control Chart Pattern (CCP), so, control charts and trend behaviour play a key role in this field of production control. The paper of Ranaee and Ebrahimzadeh [19] differentiates in six types of trends that typically arise in SPC charts as presented in Fig. 1.

As another classification, Lavangnananda and Khamchai defined nine variants of patterns [20] (Fig. 2.), where the final one represents that typically there is a mixture of effects, consequently, superposition of patterns are to be faced in the industry.

Considering the various control chart patterns in Fig. 1. and Fig. 2. it is still an open challenge what can be considered as “Normal” behavior, what distribution with what parameters, even if at all it can be described with a formal statistical distribution. It is an issue to determine what level of noise is superposed on it, what distribution the noise has, even if at all the noise and the basic signal trend can be separated [21]. On the other hand, it is also a significant challenge to identify and separate the different trend types and their parameters based on a real SPC measurement signal.

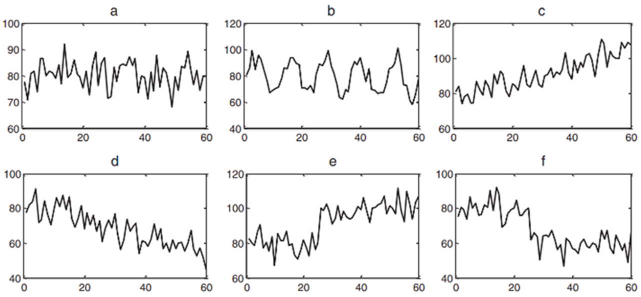


Fig. 1. Six common types of generalized Control Chart Patterns (CPPs): (a) normal, (b) cyclic, (c) upward trend, (d) downward trend, (e) upward shift and (f) downward shift [19]. This structuring is very useful for defining situation detection assignments for improving the progress supervision.

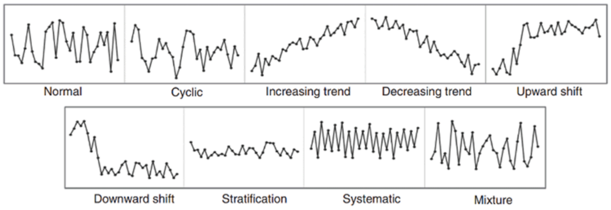


Fig. 2. Nine common types of generalized Control Chart Patterns (CPPs): top: normal, cyclic, upward (increasing) trend, downward (decreasing) trend, upward shift; bottom: downward shift, stratification (novel), systematic (novel) and mixture [20]. Together with Fig. 1. it represents that beyond the basic CPP types, additional ones can be defined, moreover, in practice a supercomposition of them receives the highest importance.

Köksal et al. reviewed the quality management related applications of various data mining techniques in manufacturing industry [22]. They grouped the quality related assignments into four groups: product/process quality description, predicting quality, classification of quality, and parameter optimization. They proved the increasing importance of such research and application techniques and their relevance to industry.

El-Midany et al. used ANNs to recognize a set of sub-classes of multivariate abnormal patterns [23] in machining of a crank case as one of the main components of the compressor. They used a simulated and a real-world data set as well; furthermore, they can identify the responsible variable(s) on the occurrence of the abnormal pattern. Ranaee and Ebrahimzadeh used a hybrid intelligent method [19] to recognize whether a process runs in its planned mode or it has unnatural patterns. This method includes three modules: a feature extraction module, a multi-class Support Vector Machine (SVM)-based classifier module (MCSVM) and an optimization module using genetic algorithm. They tested the algorithm on synthetically generated control charts. CCPs with different levels of noise were analyzed by Lavangnananda and Khamchai [20]. They implemented and compared three different classifiers: Decision Tree, ANN, and the Self-adjusting Association Rules Generator (SARG) for process CCPs that were generated by predefined equations of GARH (Generalized Autoregressive Conditional Heteroskedasticity) Model for $\overline{X}$ chart. Pelegrina et al. used different Blind Source Separation (BSS) methods in the task of unmixing concurrent control charts to achieve high classification rates [24]. Gutierrez and Pham presented a new scheme to generate training patterns for Machine Learning (ML) algorithms: Support Vector Machine (SVM) and Probabilistic Neural Network (PNN) [25]. Yang et al. proposed a hybrid approach that integrates extreme-point symmetric mode decomposition with extreme learning machine (ELM) to identify typical concurrent CCPs [26]. Motorcu and Güllü constructed X-R control charts for each production line on the data obtained from shop floor to provide high quality production by eliminating key problems: undesirable tolerance limits, poor surface finish or circularity of spheroidal cast iron parts during machining [27].

Huybrechts et al. applied standardization, trend modelling, and an autoregressive moving average (ARMA) model to determine short-term correlation between subsequent measurements [28]. The out-of-control observations can be determined precisely with the Dijkstra model on the cumulative sum chart of the corrected residuals between the measured and predicted values. Milk yield data from two automatic milking system farms and one farm with a conventional milking system were used for the case study.

Viharos and Monostori presented an approach, already in 1997 for optimization of process chains by artificial neural networks and genetic algorithms using quality control charts [29]. It was shown that the control of "internal" parameters (temporal parameters along the production chain) is a necessity, by this way, early decisions can be made whether to continue the production of a given part or not. Continuous optimization of the production system is possible using the proposed solution as well. Survey on neuro-fuzzy systems and their applications in technical diagnostics and measurement was presented by Viharos and Kis [30], further machine learning techniques supported the supervision of the milling tool life of cutting ceramics in the selection of appropriate features of the vibration signals [31].

Concerning the applied techniques, the most prevalent approaches are based on statistical methods, such as autoregression, moving average and their combinations: autoregressive integrated moving average model (ARIMA) [32] with use of linear regression analysis, quasi-linear autoregressive model [33] or Markov chain models (MCM) [34]. These methods are based on historical production or time series data for modelling and prediction.

Another approach has appeared with the evolution of artificial intelligence, such us modelling with artificial neural networks (ANN), support vector machines (SVM) or nearest neighbour approaches, based on pattern sequence similarity [35]. There are several curve-fitting methods in this field for small sample data, such as genetic algorithm [36]. By using artificial neural networks combined with statistical methods to compensate drawbacks of the separate approaches in trend forecasting lead to better classification and approximation results.

A mixed, physical model integrating real process measurements was presented by R. Paggi et. al. for computing process uncertainties beyond their prognosis values [37]. Various physical modelling techniques, like finite element methods, analytical equations can represent the known dependencies. Francesco et. al. [38] used effective measurements derived from the conformity tests to improve the accuracy of the Remaining Useful Life (RUL) evaluation.

The importance for applying appropriate SPC control charts is mirrored through the results of Wu et al [49]. A new method combining the ensemble-integrated empirical mode decomposition (EEMD) algorithm and the EWMA control chart was proposed to identify and alert global navigation satellite system (GNSS) deformation information. The experimental results show that the recognition accuracy of the EWMA control chart method and the modified EWMA method is higher than that of the cumulative sum control chart method. The use of a modified EWMA control chart improves the accuracy of deformation identification and early warning, which reduces the false alarm rate and missing alarm rate.

An extended SPC control chart was introduced by Aslam et al. and it has been compared with the existing plan using simulated data generated from neutrosophic COM-Poisson distribution [50]. The practical implementation of the suggested chart has also been expounded using the data from the manufacturing of the electric circuit boards. Overall, the results demonstrate that the suggested chart will be a proficient addition in the control chart literature.

Research on variety types of control charts' applications is widespread. Xbar-R charts for Material Removal Rate (MRR) and Ra control were applied by Kavimani et al. for to evaluate the measurement uncertainty of Wire Electric Discharge Machining (WEDM) [52]. To investigate wind speed range by state and by month, the R chart was adopted by Aquila et al. [53]. Control limits were obtained by the specific equation related to the examined wind energy field. Through the R graphic, it was possible to observe that the higher amplitudes, regarding wind average speeds measured for each month for four years, occurred in the state of Bahia during summer months. A Bayesian Control Chart is a graphical monitoring tool that shows measurements of process samples over time by using the Bayes' theorem to update information on a process state. It was applied by Mba et al. [54] in the proposed, novel fault detection and classification system based on the integration of stochastic resonance and the hidden Markov modelling (HMM) of vibration data tested by simulated and real-life gearbox applications. The measurement error results of diameter and thickness of O-rings are presented on a quality control chart by Peng et al. when improving the human inspection-based quality control by vision systems [55]. The proposed method by Wu et al. can adjust the train rail defect detection (control chart) thresholds according to the design law to improve the detection accuracy. In addition, a data pre-screening perceptron classification and learning model is proposed for defect features of end face, weld reinforcement and lower crack of screw hole, which can transform the defects detection problem into a contour classification problem, and the detection performance can be improved by learning sample images [56]. Gopal and Prakash applied quality control charts for fabrication of magnesium hybrid metal matrix composite by reinforcing silica rich E-waste CRT panel glass and BN particles through powder metallurgy route [57]. A novel Al/Rock dust composite was fabricated successfully by Prakash et al. through stir casting technique for varying reinforcement parameters i.e. Rock dust particle size and weight percent [58]. Alam et al. describes a lossy compression technique with a quality control mechanism for PPG monitoring applications [59].

The review of the literature and the related applications mirror that there are various methods for statistical process control in manufacturing, including also many machine learning techniques, however, the advances of reinforcement learning are not yet exploited in this area of production. This status served with the scientific motivation to adopt RL to SPC in production as introduced in the next paragraphs.

3.1. RL for SPC in manufacturing

In a general reinforcement learning approach the central component is an agent (or a set of agents) that senses its environment and acts through actions to its environment, moreover, it receives rewards from the environment evaluating externally and independently the actions taken. Series of such interactions serve with continuous information to the agent and it can learn in an uninterrupted manner from its environment, moreover, because also it takes actions parallelly, it performs in the same time valuable tasks as well. This *a very important advantage of reinforcement learning over supervised or unsupervised techniques because the learning component can be continuously applied to perform the given assignment parallel to its training that is especially important in manufacturing.*

As a rough description, in the proposed framework for applying RL for SPC in manufacturing the agent walks through the sensed time series, as a moving window. Each moving window will be quantized and become a state. The agent considers only the actual moving window from the past as information source when deciding which action to select. The generated actions act on the production environment with which it can influence the trend (the Control Chart Patterns) inside (or sometimes unfortunately outside) the prescribed manufacturing tolerance range. The agent receives the related reward from the environment, according to the taken (production influencing) action. The proposed external reward system is defined so that every action has a (real) cost, and the reward is inversely proportional to the cost. In addition, it incurs and extra penalty if the trend (produced products) goes out to the out-of-control range. To introduce the proposed RL for SPC in manufacturing concept more precisely, the various individual components of RL have to be defined exactly as described in the next sections about the state, actions, learning method, reward, events in the environment, knowledge representation and learning method.

3.2. Temporal difference learning

There are various kinds of learning strategies in reinforcement learning, *Temporal Difference (TD) learning is one of the most popular and effective methods, so, it was selected for the proposed solution*, however, other learning types can be applied here as well. In the TD case, after the agent receives the reward from the environment, the chosen action's (Q) value will be updated using the Temporal Difference Learning equation (eq. 1), where $Q_{s,a}$ is the value of the action of row s (state) of the Q-Table (detailed later). $0 \leq \alpha \leq 1$ is a constant step-size parameter, which influences the speed of learning, therefore it is called as learning rate. γ is a parameter, $0 \leq \gamma \leq 1$, it is called as discount rate/factor. The discount factor determines the present value of future rewards: a reward received k time steps in the future is worth only γ^{k-1} times less what it would be worth if it were received immediately [39]. In this case it is applied to estimate the value of the next state where the agent lands after taking action a in state s. The R marks the related received reward.

$$Q_t(s,a) = Q_{t-1}(s,a) + \alpha\left(R(s,a) + \gamma \max_{a^{'}} Q(s^{'},a^{'}) - Q_{t-1}(s,a)\right)$$

This update rule is an example of a so-called temporal-difference learning method, because its changes are based on the difference $Q_t(s,a) - Q_{t-1}(s,a)$, so on Q values at two successive steps/states.

The values of α and γ has recently been treated as a hyperparameters of the optimisation e.g., by OpenAI and Xu et al. [40,41]. After numerous tests of the authors of this paper, it was found that the step-size parameter should be around 0.3 and the discount rate should be around 0.75 in the analysed manufacturing environment.

Optimal action selection is a widely researched area in the RL, it is well known as the "Exploration-Exploitation Trade-off" [41]. In RL cases, where a complete model of dynamics of the problem is not available, it becomes necessary to interact with the environment to learn by trial-and-error the optimal policy that determines the action selection. The agent has to explore the environment by performing actions and perceiving their consequences. At some point in time, it has a policy with a particular performance. In order to see whether there are possible improvements to this policy, sometimes the agent has to try out various, in some cases not optimal actions to see their results. This might result in worse performance because the actions might (probably) also be less good than the current policy. However, without trying them, it might never find possible improvements. In addition, if the world is not stationary, the agent has to do exploration in order to keep its policy up to date. So, in order to learn, it has to explore, but in order to perform well, it should exploit what it already knows. Balancing these two things is called the exploration–exploitation problem.

3.3. Environment: Validated production simulation

As it was widely reviewed in the sections before, the actual state-of-the-art research and applications of reinforcement learning in production (almost) always incorporate a simulation component as in the proposed concept as well. *In the current approach a simulation environment was built up that emulates the production trend behaviour and generates a time series signal similar as it is produced by the manufacturing environments and production plants.* In this environment the RL agent is able to learn while the complete known signal behaviour (the individual details) is inside the environment simulator that is able to generate time series of any length. As the simulation generates the time series, the proposed reinforcement learning based agent takes an action at each time series point, so, after each manufactured product that affect back the time series itself in its next points. In addition, as in every production environment, independent events (many times called as "changes and disturbances") can occur with certain probabilities that affect the evolutions of the time series as well. The time series' data points are generated step-by-step, at first, they are generated noiselessly, namely they are formed by individual linear movements/trends. So, the starting position inside the control chart, the steepness of the linear line and its length simulates the production trend evolution. Due to the complexity and natural noisiness of the real time series, noise is added to the data point after it is generated, so the new data point will be sampled from a gaussian distribution, where its mean is the value of the original noiseless (linear trend) point and the measure of the noise is its standard deviation. In many efficient production environments, the size of the noise is less or between 5% and 10% of the interval formed by the two out-of-control boundaries of SPC (LTL, UTL).

As Fig. 3. shows two time series one is the original trend the other is the final time series, having additional noise. The learning uses only the noisy time series, but if required it is simple and easy to generate various noise levels based on the original time series as well.

A special "trick" is the length of the generated trend. In reality, each trend has only one step lengths, because the RL agent selects and action at each product produced, so, at each time series point the selected action redefines the trend and the noise level as well, so, these components of the trends are continuously redefined at each point. However, there is a special type of the action, the so called "No action" when no changes are implemented in the actual behaviour of the actual trend, noise, etc. In such a case the size of the original trend becomes much longer (than one). Fortunately, after some learning steps and also in the reality the "No action" is far the most frequent one, so much longer trends are also emerging.

It has to be emphasized that in the given case all of the elements of the simulator are inherited a validated through a real manufacturing environment similar to some of the introduced solution before in the review of the state-of-the-art. It is valid for the possible events, actions, their effects, various noise levels, as well for frequencies of events, effects of actions, etc., even if the figures in the paper are distorted.

3.4. Events

In manufacturing unexpected events happen, like tool failure or equipment failure, etc., consequently, the simulation has to be able to emulate this behaviour as well. For that reason, it is needed to specify the events' frequencies and their impacts, too. The impacts can be divided into three different parts. Events have impact on the mean of the trend, so, where

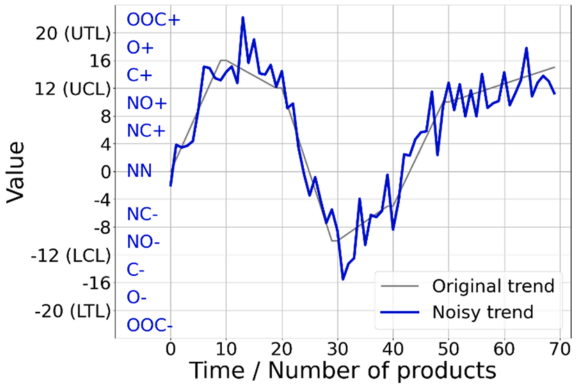


Fig. 3. Generated time series, showing the original (grey) and the extended noisy (blue) trend (noise = 3, Upper Tolerance Limit (LTL) – Nominal value = 20 = Nominal Value – Lower Tolerance Limit (LTL)). The effect of the added noise is clearly represented as deviation of the time series. The figure shows also the Lower and Upper Tolerance Limits (LTL, UTL) and the commonly applied Lower and Upper Control Limits (LCL and UCL) as well.

the next time series point starts on the y axis (see Fig. 5.), inside (or sometimes outside) the control chart. These mean-changing (trend shift in Fig. 1.) information are regularly given as distributions, which are usually not uniform. So, it is necessary to provide the intervals for the mean changes and each of the subintervals' probabilities. Second, the events may affect the time series' trend (steepness), meaning that they can start e.g., a long-term, low-intensity trend which slowly brings the mean outward (or inward). Outward means the trend is slowly going towards the out-of-control stripes, inward means the trend is slowly going towards the central, normal stripe (stripes are discretised intervals of the control chart as described in detail later and shown in Fig. 4.). Finally, the events effect the time series' noise, with which the individual data points are emulated. If multiple events occur at the same time, their effects are cumulated, more precisely, the mean/starting value of the next starting trend is the average of the starting points of the individual events, trend steepness are summed, however, their effect on the noise is selected otherwise: the new noise will be equal to the largest noise among all the actual events. *The same structure is applied to describe the effects of the actions as well.*

3.5. Reward of production

In the proposed architecture the *rewards are defined as costs of the actions* taken: they are negative, and more the action costs, so lower the reward is. It is assumed in the simulation, that *if a trend point (=product) goes out of tolerances, it will incur additional penalties (cost of waste product)*. It is possible to distinguish when the time series goes out at the bottom limit (LTL) and when it goes out at the upper limit (UTL). The logic behind this is that the former is more undesirable in some cases, because if the product's length is less than it should be, then in most cases it cannot be corrected and it becomes waste, but in that case when the product's length is longer than prescribed, it can be remanufactured/corrected. Thereby, the penalty in the former case can be much higher than in the latter case, but in both cases, it is typically higher than when the produced products are inside the prescribed LTL, UTL range. If the production is inside the prescribed tolerance range and no action occurs (so, always the "No action" is selected at each product), no penalty is given as reward to the agent (penalty is zero).

3.6. State representation

Before starting to use of deep neural networks, which are popular nowadays everywhere, thus here as well, the hidden dynamics of the production SPC as time series were analysed. For this aim, at first a simple *Q-Table was used owing to its white box nature, meaning that the concealed processes are visible.*

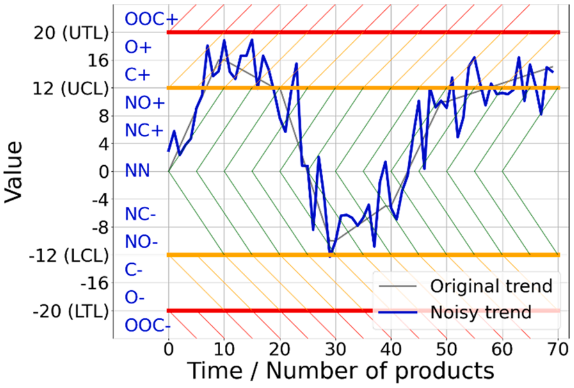


Fig. 4. Time series with stripes. Green in the middle is the normal range (optimal), yellows are the control ranges (warning) and the reds are the out-of-control ranges (resulting in product failure).

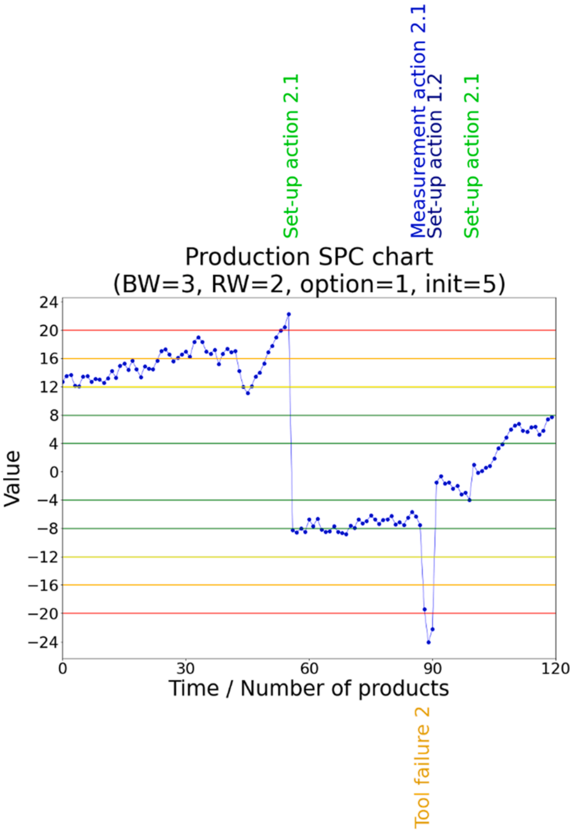


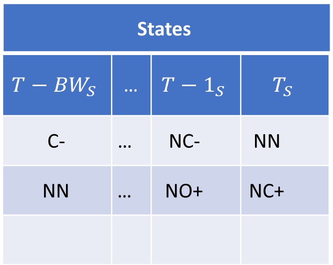
Fig. 5. Simulated production time series with generalised stripe boundaries, external manufacturing changes and disturbances, represented as events (bottom) and selected production control actions (top). Horizontal green lines are the normal range limits (optimal), yellows are the control range borders (warning) and the reds lines are the out-of-control limits (failure). The figure mirrors the behaviour of the production (quality) data (blue) incorporating noise and trends, together with the effects of the disturbing external events (bottom: yellow) and of the applied manufacturing control actions (top: green, blue). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

The value range of the signal was divided into fix stripes of the control chart. The time series values in the same stripe get the same quantized value/coding. This is necessary to the Q-Table, because with continuous values the length of the table would grow exponentially. A fix interval was arbitrarily selected for the value range between −20 and 20, and it was divided it into eleven stripes – two out-of-control (under −20 and above 20), same control stripes (between (-20, −10] and [10, 20)), and some normal stripes (between (-10, 10)), as Fig. 4. shows. This approach is inherited naturally from the industrial SPC approaches. These numbers are fully arbitrary, it is possible to choose other numbers or other boundaries. Of course, the real ranges are normalized to the given simulator and vice-versa, so, any control chart measures with any magnitudes and units can be handled with the proposed state representation.

The main structure of the first part of the Q-Table is shown in Fig. 6., where states consist of quantized values, measured in T, T-1, T-2, …, order. T is the current time/actual product, consequently *the proposed solution considers BW_s number of measurements/values from the past.* OOC refers to the Out-Of-Control range (red), C refers to the Control range (yellow in Figs. 4 and 5.) and N refers to the Normal range (green in Figs. 4 and 5.). The minus and plus signs are shown due to the symmetry (above or under the Normal range).

3.7. Actions for SPC in manufacturing

In manufacturing, different actions are distinguishable, such as



States			
$T-BW_S$	...	$T-1_S$	T_S
C-	...	NC-	NN
NN	...	NO+	NC+

Fig. 6. A part of the typical Q table and its content in the RL for SPC in manufacturing approach. States for the RL agents are represented as BW long series of stripe codes according to the allocations of the production data in the different stripes in the recent BW number of points.

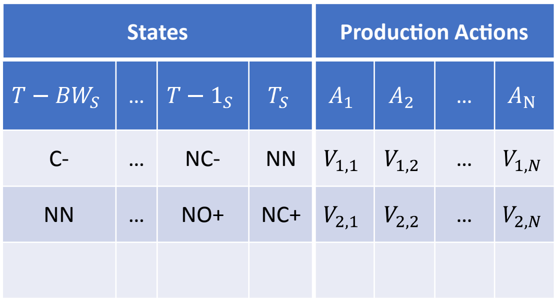
Maintenance actions, Measurement actions, Set-up actions and No action as well. There are actions whose impacts on the trend are outstanding and need to be considered, and there are action whose impacts are negligible. No action stands out from the rest, because taking it means the time series is controlled and the near future of the products looks promising, so, no intervention is needed. In general, an action can have the same effect on the production trend as an evet: it can influence the starting position of a trend, the steepness and the noise level after taken. There are two strategies for action selection which are needed to be balanced: exploration and exploitation. When the agent does not have sufficient information about its environment, it is advisable to choose the exploration strategy when the agent chooses barely chosen actions with intent to explore its environment and so the state-space. In contrast, when the agent does have enough knowledge about the environment mostly it is acceptable to choose the action with the highest goodness.

3.8. Knowledge representaion: Q table

The Q table incorporates naturally also the possible actions (A_1, …, A_N) of the RL agent as shown in Fig. 7. The values $V_{(i,j)}$ under them are the estimated goodness (values/expected reward) of the actions [1], in respect to the actual state (row), they are the so-called Q values (es referred later on).

The table in Fig. 7. is structured as follows: the states are chosen for the rows, and the actions are chosen for the columns. The states consist of quantized values, its length depends on how many previous values are taken into account in the action selection (BW_s). Concerning the actions, they are the possible activities performed by the operators, experts or a control system to control the given process, so, in the RL, when the best action is searched at a state, it means in reality that the best production intervention is searched.

According to the applied, well-known Q table representation the production related knowledge is stored in the $V_{(i,j)}$ values. Later, it will be substituted by one (or more) regression techniques, like deep neural networks.



States				Production Actions			
$T-BW_S$	...	$T-1_S$	T_S	A_1	A_2	...	A_N
C-	...	NC-	NN	$V_{1,1}$	$V_{1,2}$	...	$V_{1,N}$
NN	...	NO+	NC+	$V_{2,1}$	$V_{2,2}$	...	$V_{2,N}$

Fig. 7. A part of the typical Q table and its content in the RL for SPC in manufacturing approach. Rows are the different, already visited states and the possible, selectable actions are shown as A1, …, A_N, while the Q table values ($V_{i,j}$) store the knowledge of the RL agent.

4. Important RL for SPC in manufacturing extensions

The next paragraphs introdues some extension elements of the proposed solution that enable a harmonized fit or RL for SPC in manufacturing.

4.1. Dynamic Q-Table

A major problem with Q-Table is its memory requirement. For example, for a table with A columns where each column takes B different values, the size of the table could reach to B^A rows. With large A and B, the generation, storage and handling of the table could cause problems, moreover it is unnecessary to allocate the memory for the empty table before it is used. Therefore, a technique was introduced, called *dynamic Q-Table, where only as much memory is allocated as required and only when it is required*. It the proposed concept, at the beginning the Q table is empty. When the algorithm reaches a new state, it adds it to a list which represents the rows of the Q-Table and its initial actions' values are selected randomly. (In the future research the order of the Q table rows will be much more optimized for exploiting the characteristics of the production SPC – e.g., normal states are much more frequent.) If the actual state is already in the Q table, then the chosen action will be updated as it is detailed in the next chapter. As a research outlook, this concept could be transferred to the Artificial Neural Network (ANN) based solution, so, at the beginning the ANN could have really limited knowledge and later on it can learn continuously new states, however, this is a challenge for a future research.

At the beginning, when the agent mostly explores its environment, it often meets states in which it has not been before, so they are added to (the end of) the Q table. As it explores, its knowledge about the environment grows, therefore it meets more and more times visited states, where it only updates the relevant action values of the state, so in most of such cases no new row is needed. As a result, the length of the table grows logarithmically, as it shown in the upper part of Fig. 8. In the given, presented case in the bottom part of Fig. 8., it may be rational to stop the learning after ~ 2000 learning steps, because even though the table is still growing, that is not increasing the in the given, particular case considered recognition rate for production trend forecast significantly. It is a very important result proving that there exists a rational limit for the Q table, beyond it, the size, calculation time and other performance requirements grow significantly but it does not bring valuable additional knowledge to the given SPC assignment. Consequently, with the proposed dynamic Q table solution the IT background requirements can be limited and kept under control.

4.2. State extension with past actions

In manufacturing environments, it is crucial to consider what were the last actions taken for control the given production process, because it is not worth to do the same (e.g. *expensive*) *action repeatedly in short time*. To handle this requirement also, the state space was extended with the last 'n' actions from the past (in the current analysis n = 3), this extension can be seen in Fig. 9.

4.3. Dynamic exploration & exploitation through ε self-control

In the learning, the most popular ε-greedy algorithm is applied in which ε controls the ratio of exploration vs. exploitation. Its zero value means that there is exploitation only and the value one means full exploration [1]. There are various strategies how to adapt the value of ε during learning, typically it starts with a high value for wide exploration and it is decreased over time to enforce exploitation while the learnt knowledge of the agent is continuously increasing. This is a really valuable feature of RL but on the other hand it is an additional

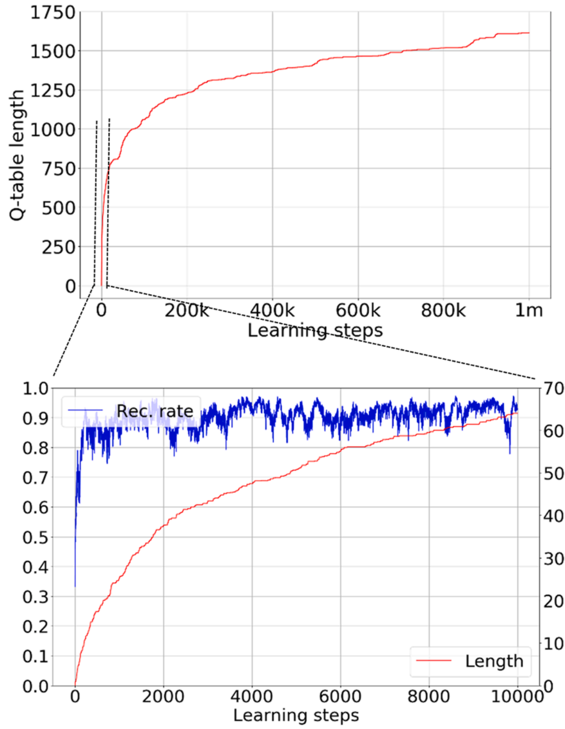


Fig. 8. Length of the Q-Table (red line) during a very long learning (top) and in short term (bottom). Length can be compared to the recognition rate (blue line) for statistical process control trend forecast (bottom) [51], representing that a relative high recognition rate for production trend direction forecast can be achieved already at early stage with smaller number of Q table rows. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

States							Production Actions			
$T-BW_S$	...	$T-1_S$	T_S	$T-3_A$	$T-2_A$	$T-1_A$	A_1	A_2	...	A_N
C-	...	NC-	NN	A_3	A_5	A_5	$V_{1,1}$	$V_{1,2}$	...	$V_{1,N}$
NN	...	NO+	NC+	A_6	A_N	A_8	$V_{2,1}$	$V_{2,2}$	...	$V_{2,N}$
				...						

Fig. 9. Extension of the state vector with the last ‘n’ actions taken (‘n’=3 in this particular case) as shown in the middle of the table by the columns T-3$_A$, T-2$_A$, T-1$_A$. Consequently, not only the results of the earlier actions are considered through the past stripes in the actual state vector, but the recently taken ‘n’ actions themselves as well.

parameter that has to be defined and controlled. *This requirement motivated the authors to build up a self-control solution for setting the ε dynamically, moreover, by the RL agent itself.* The novel introduced “trick” is to apply an extended Q table with actions for selecting the appropriate ε value becoming valid after the actual state. This selection considers the same state space as the production actions and receives the same rewards from the environment. The only, but significant difference is that in the selection of the next ε value only exploitation is applied, so, always the ε value having the highest (best) Q table value is selected. It means that there isn’t cumulated exploration, so, there is no exploration in the selection of the value for exploration. *As result, the final, extended Q table is presented in Fig. 10, possible ε values are the E_1, …, E_M with the related Q values as $E_{(1,1)}$, …, $E_{(x,M)}$.*

The usefulness of this strategy was validated through various experiences as well, the applied RL algorithm continuously controlled its own exploration ratio during the episodes. *Especially promising is the behaviour that was experienced when an unexpected, external event occurred, then the agent automatically sets higher exploration ration for handling the novel condition while in other cases it keeps it small, so in the range of almost full exploitation.*

4.4. Exploration control rule

Thanks for the beneficial feature of the reinforcement learning technique the agent realizes a balance between exploration and exploitation. This part of the proposed solution is relevant only for the exploration phase, so, when the agent decides to explore the environment. *Three different types of exploration options were defined, analysed and evaluated*: at the first option, if the agent decides to explore its environment the production action is randomly selected from all possible actions with a uniform distribution, without considering the agent’s self-knowledge about the prognosed/expected effect of the selection. The second option means that the action is selected according to the agent’s self-knowledge, so based on the current Q values of the possible actions given at the current state, namely, the higher (better) the value (=expected reward) the action has, the more likely it is to be selected. Finally, the third option serves with that the action is selected according to the external (known) reward, namely, the higher the reward of the action has, the more likely it is to be selected. This last option is only for theoretical testing because in the real environment the agent does not know the reward in advance, moreover, this information is party stored in the Q table as an estimation, so, this option is party incorporated into the second one. The choice for the best option is unknown, so, it has to be tested as described later on.

4.5. Initial exploration level

During the interaction with the environment, beyond the selection of a production action (as introduced before) the agent chooses for itself ε values as well (in Fig. 10.: Epsilon Actions) for its next activities. For the production action the actual ε controls the exploration–exploitation ratio, however for the action of selecting the future ε value, only

States							Production Actions				Epsilon Actions			
$T-BW_S$	...	$T-1_S$	T_S	$T-3_A$	$T-2_A$	$T-1_A$	A_1	A_2	...	A_N	E_1	E_2	...	E_M
C-	...	NC-	NN	A_3	A_5	A_4	$V_{1,1}$	$V_{1,2}$	...	$V_{1,N}$	$E_{1,1}$	$E_{1,2}$	...	$E_{1,M}$
NN	...	NO+	NC+	A_6	A_8	A_8	$V_{2,1}$	$V_{2,2}$	...	$V_{2,N}$	$E_{2,1}$	$E_{2,2}$	...	$E_{2,M}$
				...										

Fig. 10. Extension of the Q table with the action and knowledge set for specifying the ε value for the next iterations. The table represents the final solution, with the ε related extension, too, beyond the production related Q table (that is shown in Fig. 9.). The state inside the given control chart is described by the past measured production (e.g. quality) values coded to series of stripes together with the recent actions (“States” part) and the agent takes double decisions in the given state: 1. It selects an action that influences directly the production (among the “Production Actions”); 2. It selects also a novel ε value that will be applied when visiting this state again. The later selection realizes the automatic self-control of the exploration–exploitation ratio for the reinforcement learning agent themselves.

exploitation was defined (this solution performed as best). However, *there is a small exception, when the agent meets a new, previously not visited state it has to select the initial, first value of the ε for this new state*. This choice determines the staring exploration–exploitation ratio. In the given proposed concept and in the related manufacturing experiments, fixed, concrete, discretized values for ε are applied only (0.0: 0 - as exploitation only, 1.: 0.05, 2.: 0.15, 3.: 0.25, 4.: 0.5 and 5.: 1.0 - as full exploration), consequently, the selection one from these initial values determine the starting exploration–exploitation ratio in the new state. So, in this initial stage there is a small level of exploration also for the exploration itself (as an exception). The optimal selection for the first value is unknown, but it was tested as described afterwards. One can recognise that the possible values for ε are not distributed equally, because the experiences mirrored that the well working ratio of exploitation is much higher than exploration, consequently, smaller ε values are needed more than larger ones.

Comprehensive experiment was performed for selecting the best/optimal exploration control rule and the related optimal/best initial exploration level. A complete and so, long RL training was performed at all combinations of these factors (at each exploration control rule with initial exploration levels of 0.: 0.0, 3.: 0.25 and 5.: 1.0). The final cumulated costs, the ratio of good products, the unit price of a product, the self-controlled, final exploration–exploitation ratio and the size of the Q table were measured at each combination for finding the optimal set-up. The Fig. 11. shows the experimental results for these indicators, resulting in the optimal selection for the exploration control rule to the second option (the action is selected randomly but in relation according to the current Q values of the possible action list given at the current state) and the initial exploration level to 5.: $\varepsilon = 1.0$-as full exploration.

4.6. Reusage Window (RW) and measurement Window (MW)

In production environment each measured value, inherited from many sources, incorporates significant related cost of collection, e.g., cost of the equipment, training of the personnel, doing the measurement activities, establishment of the IT background for data collection, IT communication and storage, software to handle the measurements, continuous re-calibration of the measuring devices, their maintenance, etc. Consequently, the measured values in manufacturing incorporate high business and technical value and so, they have to be exploited as much as possible. *In the production control environments of today, this ideal situation is far not yet approached, the data asset is significantly higher than its usage and exploitation.*

The main problem with the novel method introduced in the previous paragraphs that it lacks data reusability similar to the cases in the related scientific literature. In manufacturing, such time series are measurements, in which every measurement is a data about a produced component/product made by a process/cycle, which is, or is always considered as expensive, at least because of the extreme cost push of the market. Consequently, using all measurement values only once seems to be very wasteful. Therefore, it is desirable to reuse measurement values many times, so, *the Reusage Window (RW) concept is introduced, which specifies how many times an individual measured value is re-used in the proposed RL for SPC in manufacturing concept.*

4.7. Reusage Window (RW)

As opposite to the original concept, where the agent walks only once through the time series, with RW it is repeated RW times, as follows: at first, an interval, which length is RW, is selected from the time series and the agent goes through this selected interval once, sampling and quantizing states from it. This is one learning iteration. Then, the RW moves one step (one product or one measurement) on the time series ahead, and it starts again, until the end when the RW meets the actual end of the time series. It means that one data is (re)used so many times as long the length of the RW is. As Fig. 12. represents, the RW moving window goes

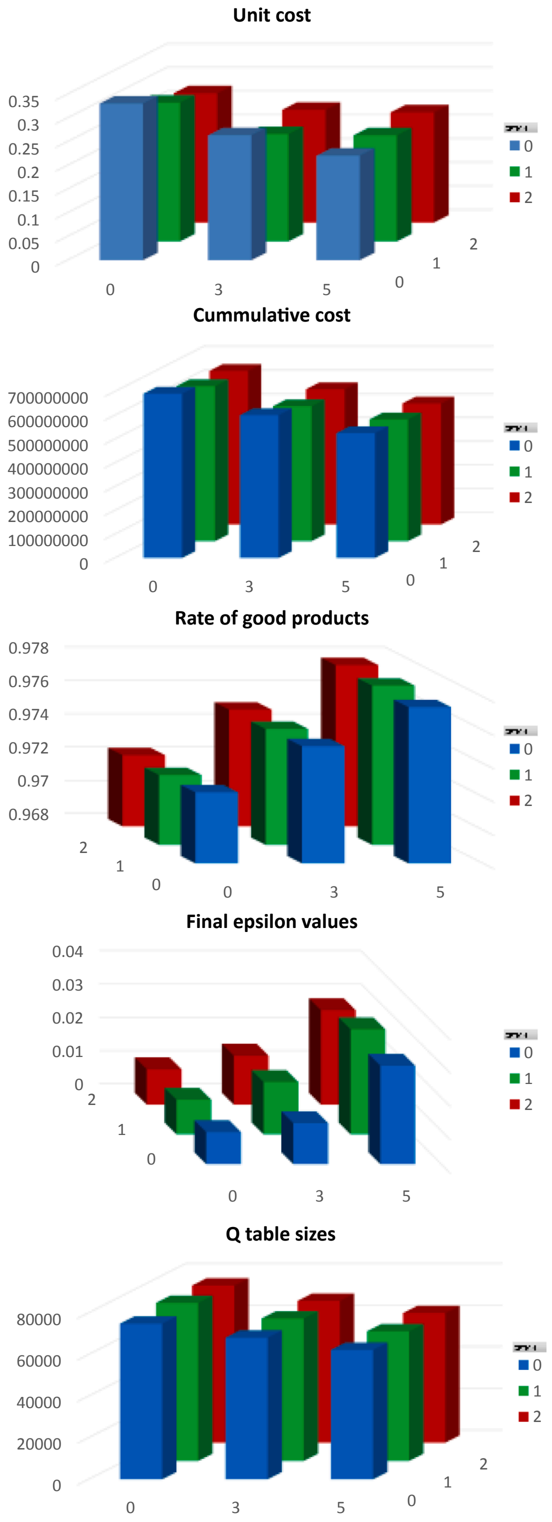


Fig. 11. Performance parameter values of the proposed solution after long runs. The two horizontal axes show the possible *Options* (0 – equal chance for action selection, 1 – action selection proportionally to the actual own estimated reward, 2 – action selection proportionally to the known external reward) and the *Initial exploration level* applied for new states (0 – as exploitation only, 3 – balanced exploitation and exploration, 5 – exploration only) and the vertical axes represents the various evaluation measures. As result, the Option 1 with (the action is selected randomly but in relation according to the current Q values of the possible action list given at the current state) and the Initial exploration level of 5.: $\varepsilon = 1.0$ - meaning full exploration was identified as optimal.

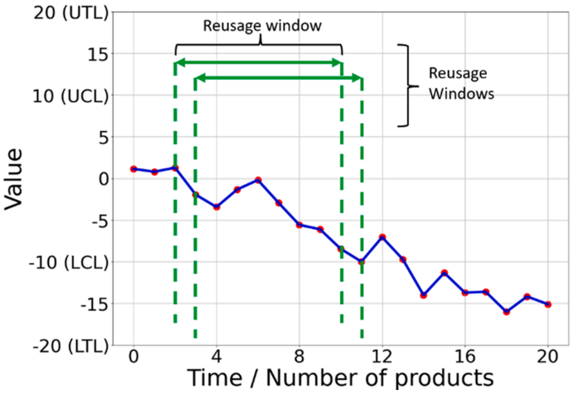


Fig. 12. The RW (here, it is 8) determines a continuously shifted interval within the time series which is a cycle that is circulated as a repeated window, consequently, it determines how many times a measured (expensive) production data is re-used by the learning. The red points connected by the bule lines presents the original measured values [51].

step-by-step through the time series and specifies an interval (marked in green). Within that interval, the agent goes through that, and processes each interval as described in the epsilon greedy algorithm mentioned above. When it reaches the end of the green interval (RW), the whole green interval moves one component/product/process/cycle ahead and the whole process starts again. The RW goes all the way to the end of the time series. Finally, it results that all measured production data are used RW times [51].

The preliminary results [51] mirrored that MW has an optimal value, so, its selection is one of the settings of the introduces concept as highlighted in Fig. 13.

4.8. Measurement Window (MW)

The performance of different agents' reactions are evaluated after every learning iteration by the received rewards, by the ratio of good products, by the unit cost and by the averages of the self-controlled ε-s. *The concept of the MW was introduced in the previous paper of the authors* [51] for allowing *fair* comparisons among the various process control solutions (for production trend evolution forecast) having different RWs. The comparisons of the performance parameters received by using different RW mirrored high fluctuation and unreliability, so, there was a requirement to define an independent time window that specifies the number of recent, past data as the basis for evaluate the actual performance of the agent. As result, the Fig. 14. shows a comparison [51] for different RWs evaluated by the same MW. As result, with the proposed MW concept the performances are equal that was not the case without the MW concept.

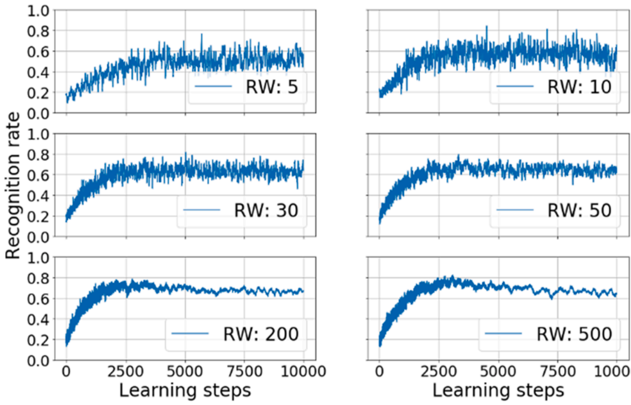


Fig. 13. Comparison among various production trend forecast recognition rates with different RWs. (noise = 10, MW = 150) [51]. It is mirrored that the RW has an optimal level because above that the production trend forecast recognition rate does not increase but the calculation need grows significantly (above 30 in this case).

Finally, the preliminary research results mirrored that both the RW and MWs have optimal values, consequently they can be specified as *meta*-parameters in the practical experiments.

4.9. Industrial test and validation

During the development and validation of the concept, a concrete simulation, based on a real environment was created. Millions of simulated industry data were generated on the one side, and other thousands of real values were analysed to determine the hidden dynamics of the real time series for the simulation. *The whole process for emulating the environment of the agent is fully automatic*. Events are sampled from Gaussian distributions determined on real data and expert know-how. As described before, events affect the time series, they may move the time series' mean outward, increase its noise or it may immediately let jump the time series into another stripe. Events like "tool failure" or "equipment failure", etc. are continuously generated according to their real manufacturing distributions. *The algorithm's goal is to minimize the production cost and as far as it is possible to keep the time series in the allowed range (between LTL and UTL). To achieve this, it effects on the time series with actions like "maintenance action", "set-up action", etc. and there is a special but very important action called "No action", when no invention is applied.*

Detailed descriptions of the events' and actions' effects are described in the sections above, a typical series of evolution is presented in Fig. 15., it represents a relative well-trained agent. It has to be mentioned again that the agent takes decision at each product produced but in majority of the cases (fortunately and similar to in the real plant) the appropriate action is "No action" that is not highlighted separately in the Fig. 15. Additionally, after each action and produced product the agent receives the related reward (in the current, proposed SPC control the related cost) from the environment.

The performance measures of the system are presented through the evolution of the *cumulated production cost*, the *rate of good products* (without any failure, being inside the tolerance range), the *unit cost of a product* (it is the derivative of the cumulative cost) and the (by the agent self-controlled) *exploration ratio*. Fig. 16. mirrors the continuous increase of the process control effectiveness, thanks to the proposed concept applying the agent's reinforcement learning capability. During the episodes, the knowledge of the agent increases continuously, so, the selection of the production control actions became better and better, the agent adapts itself to the various, changing production circumstances. It

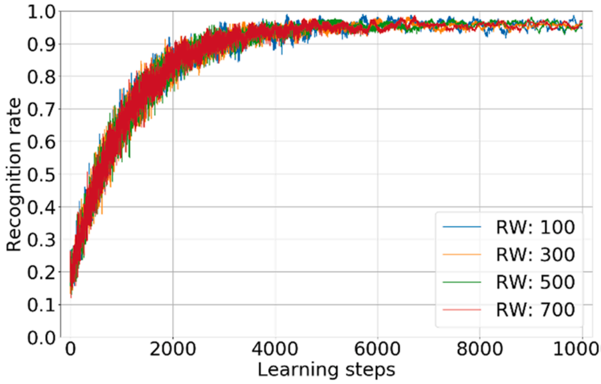


Fig. 14. Comparison of different recognition rates of production trend forecast with different RWs. (noise = 0, MW = 150) [51]. The figure shows that the MW of 150 is applicable because this length of performance measurement serves with the same data independently from the data reusage amount (RW).

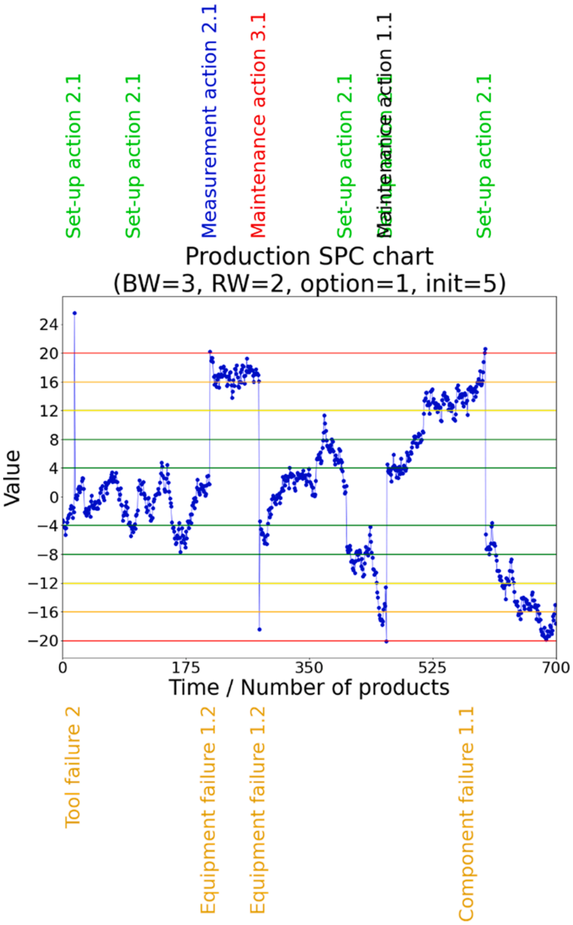


Fig. 15. Production trend evolution (blue points) influenced by external production events (bottom in yellow) and by the reinforcement learning agent's selected production actions (top in various colours) for keeping the manufacturing inside the prescribed tolerance range. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

was only possible because continuous exploration and exploitation is served by the agent's RL system.

The manufacturing related performance comparison of the continuously training and performing agent is realized by analysing the distribution of selected production action frequencies and additionally the unit prize together with the rate of good products, the later measures are the two most important Key Performance Indicators (KPIs). *The production performance of the learning agent is considered satisfactory* when the average of the unit prize (black line in the Fig. 16. bottom) and the rate of good products (blue line in the third diagram in the Fig. 16. from the top) is constant, so, when their moving average is inside a prescribed (small) tolerance range. Similarly, *the accumulated knowledge level of the agent is considered satisfactory (enough training steps were performed)* when the moving average of the self-controlled exploration–exploitation ratio (ε) is constant, so, is inside a prescribed (small) tolerance range (moving average(s) of the red line(s) in Fig. 16.). As shown in Fig. 16., the level of the ε ratio (red line) is continuously decreasing mirroring that during the learning the agents continuously decreases its own exploration demand. After a certain number of learning steps it is kept on a really small but greater than zero level, representing that the agent exploits its own knowledge for selecting the best possible production action almost always and so, it explores the environment in very rare cases. Consequently, the production process is close to its optimum as shown also by the constant curves of unit prize and level of good products. Having achieved this performance level, the frequencies of selected production actions were compared to their frequencies in the real manufacturing environment to compare the performance of the proposed concept numerically. The most important KPI was the ratio of the selected "No action" type action because in realty at the majority of the produced products the manufacturing is running automatically without any need of external intervention, consequently, the ratio of real "No action" type actions is very close to 1. The same level was achieved also by the proposed concept, moreover in the best set-ups this ratio was higher, namely, the agent proposed around 10–30% less production intervention actions than it happens in the production shopfloor. Significant difference was experienced among the distributions of the selected other actions because some actions were selected much rare (less than 50%+) or much more frequently (more than 50%+) than in the practice, so, this phenomenon shall be analysed further.

An important difference has to be expressed at this stage: in the mass production practice, controlled by quality control charts, production intervention is necessary only when

- there are quality problems and also
- at the fixed, prescribed periods, when it seems to be worth by the experts, however,

in the proposed concept every workpiece is considered, and *an action is selected by the agent for each of them. Consequently, the level of supervision is significantly higher using the proposed reinforcement learning agent.* It is possible only because the cost of the proposed RL based automatic solution is much-much lower than the supervision by manufacturing experts. This positioning gives especial importance to the experienced ratio of the "No action" type.

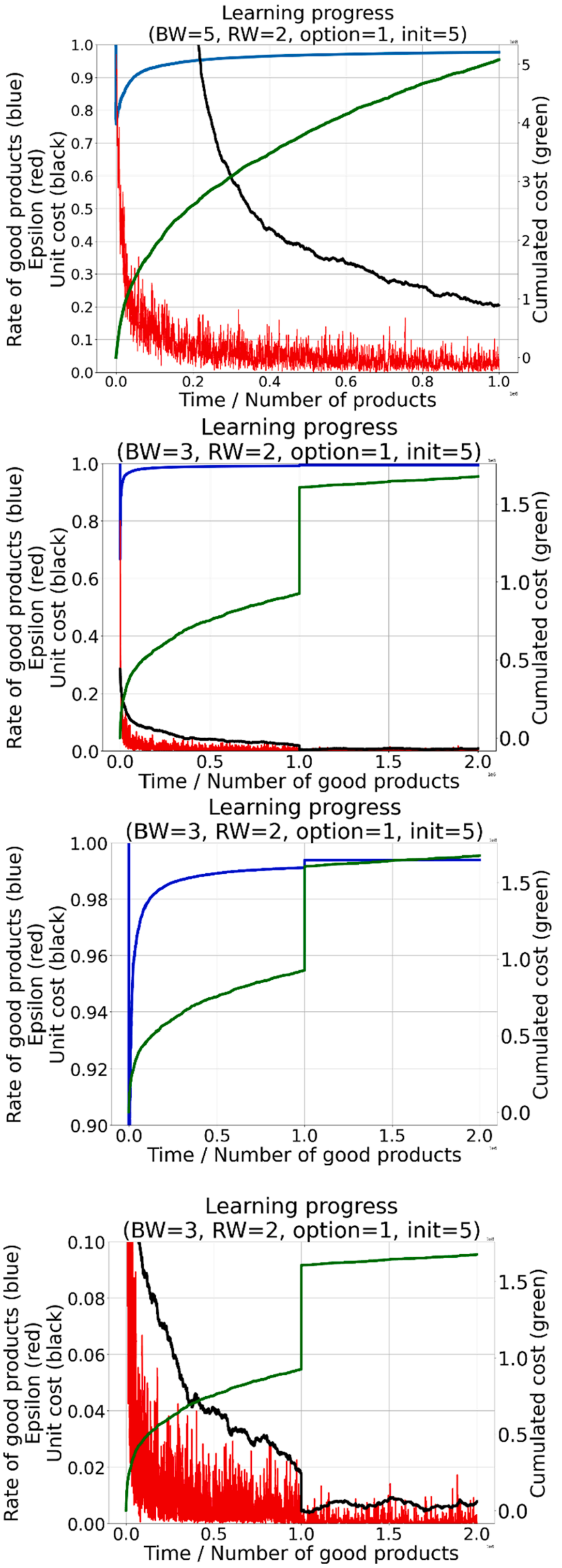
Finally, it was proven that the introduced agent was able to learn and perform at the same time, to adapt its behaviour to the environmental changes and disturbances, moreover, the minimal unit cost and maximal ratio of good products are achieved. Consequently, the introduced concept for realizing Reinforcement Learning (RL) for Statistical Process Control (SPC) in manufacturing was approved.

5. Conclusions and outlook

The paper introduced the concept and the solution to place Reinforcement Learning (RL) into Statistical Process Control (SPC) in manufacturing. This introduction is novel for manufacturing because the actual state-of-the-art literature mirrors typical applications of RL (only) in the fields of production scheduling and robotics, this is the core motivation of the research reported here. RL is an advantageous approach owing to the adaptability and continuous application capability. To adopt RL to SPC in manufacturing all of its necessary elements were defined (states, actions, rewards, etc.) in the paper. The well-known Q-Table method [1] was applied for get more stable, predictable and easy to overview results, therefore, quantization of the values of the time series and Quality Control Charts (QCC) to stripes was required. Series of recent stripes in which the production trend values arose together with the recently selected production actions formed the state vector. Typically applied manufacturing interventions for keeping the measured production values inside the tolerance range formed the action list of RL. The formulated manufacturing goal was to minimize the production unit cost while keeping the ratio of good products on a high level. The RL reward solution consists of the cost of the applied actions and additionally the cost of failure products.

A novel technique, called dynamic Q-Table was introduced, in which only as much memory is allocated as required, and only when it is required, it a beneficial approach from practical applications' viewpoints as well.

Furthermore, two additional concepts were introduced, the Reusing Window (RW) and the Measurement Window (MW). The RW is a sliding window that determines how many times one measured value of the time series will be reused during the RL repeatedly, while the MW is



(caption on next column)

Fig. 16. Key Performance Indicators (KPIs) of the RL for SPC in manufacturing agent: cumulated production cost (green), the rate of good products (blue), unit cost (black) and exploration ratio (red) along the production cycles (BW: length for considering the past, RW: reusing window, option 1.: action selection proportionally to the actual own estimated reward, init = 5: maximal exploration in new states). The figure on the top represents a first stage of the training process for mirroring the learning when the KPIs' values change quickly. The next three figures show the same, but significantly longer training (and performing) process where the left half of the figures presents the starting period of the agent-event interaction and the right side of the figure presents the "end" of the training when the unit cost, ratio of good products and the exploitation ratio is stable, almost constant (between them a long period was cut out). Consequently, these right half parts show the final performance of the trained agent. The figure below the top one shows the KPIs of a full training, the third one is the same but with zooming into the top (vertical axis) region to show the final ratio of good products (~99.6% in this example, in blue), the last figure on the bottom zooms to the constant, final unit cost (0.0075 in this example in black) and the stable but fortunately not zero ε (average of ~ 0.005 in this example in red) values. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

defined for enabling the fair comparison of learnings with different RWs by sampling them with the same evaluation frequencies. This extension of the traditional RL is necessary in the given manufacturing SPC environment, considering the cost of a measurement value and the precise evaluation requirement about the performance of the production system.

Beyond the working concept for adapting RL into SPC in production, some novel RL extensions are described, like the epsilon self-control of exploration and exploitation, and the optimisation of some *meta*-data of the training.

The manufacturing related performance comparison of the continuously training and performing agent was realized by analysing the distribution of selected action frequencies and additionally the unit prize together with the rate of good products, the later measures are the two most important Key Performance Indicators (KPIs). In the validation experiment the production process is considered close to its optimum with constant and stabilized unit prize and level of good products. Having achieved this performance, the frequencies of selected production actions were compared to their frequencies in the real manufacturing environment to measure the performance of the proposed concept numerically. The most important KPI was the ratio of the selected "No action" type action. In the best set-ups this ratio was higher than in the practice, namely, the agent proposed around 10–30% less production intervention actions than it happens in the manufacturing shopfloor. *Finally, industrial testing and validation proved the applicability of the proposed method.*

As next step, the future research has to answer numerous open challenges, like more efficient state coding of the past production history, involving the real-time evaluation of C_p and C_{pk} values of the analysed production process. Additionally, the related simulation could be extended to generate more frequently new sates to be explored to increase the speed of learning.

CRediT authorship contribution statement

Zsolt J. Viharos: Conceptualization, Methodology, Validation, Formal analysis, Investigation, Resources, Data curation, Writing - original draft, Writing - review & editing, Visualization, Supervision, Project administration. **Richárd Jakab:** Methodology, Software, Validation, Formal analysis, Investigation, Resources, Data curation, Writing - original draft, Writing - review & editing, Visualization.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

First of all, special thanks have to be expressed to the Opel Szentgotthárd Kft., Hungary, working in the automotive sector; especial for the experts Jenő Csanaki, Csaba Péntek and László Cserpnyák for supporting the research with continuous consultations and validations as well as with relevant data and bi-directional knowledge exchange.

The research in this paper was partly supported by the European Commission through the H2020 project EPIC (https://www.centre-epic.eu/) under grant No. 739592., by the Hungarian ED_18-2-2018-0006 grant on a "Research on prime exploitation of the potential provided by the industrial digitalisation" and by the Ministry of Innovation and Technology NRDI Office within the framework of the Artificial Intelligence National Laboratory Program.

References

[1] A.G. Barto, R.S. Sutton, P. Brouwer, Associative search network: A reinforcement learning associative memory, Biological Cybernetics 40 (1981) 201–211.

[2] W. Bouazza, Y. Sallez, B. Beldjilali, A distributed approach solving partially flexible job-shop scheduling problem with a Q-learning effect, IFAC PapersOnline 50–1 (2017) 15890–15895.

[3] N. Khader, S.W. Yoon, Online control of stencil printing parameters using reinforcement learning approach, Procedia Manufacturing 17 (2018) 94–101.

[4] Y.-C. Wang, J.M. Uscher, Application of reinforcement learning for agent-based production scheduling, Engineering Applications of Artificial Intelligence 18 (2005) 73–82.

[5] B. Waschneck, A. Reichstaller, L. Belzner, T. Altenmüller, T. Bauernhansl, A. Knapp, A. Kyek, Optimization of global production scheduling with deep reinforcement learning, Procedia CIRP 72 (2018) 1264–1269.

[6] Schneckenreither M.; Haeussler S.: Reinforcement Learning Methods for Operations Research Applications: The Order Release Problem. In: Nicosia G., Pardalos P., Giu rida G., Umeton R., Sciacca V. (eds) Machine Learning, Optimization, and Data Science, Part of the Lecture Notes in Computer Science book series (LNCS, volume 11331), 2019, pp. 545-559.

[7] Al Kuhnle, L. Schäfer, N. Stricker, G. Lanza, Design, Implementation and Evaluation of Reinforcement Learning for an Adaptive Order Dispatching in Job Shop Manufacturing Systems, Procedia CIRP 81 (2019) 234–239.

[8] A. Kuhnle, N. Röohrig, G. Lanza, Autonomous order dispatching in the semiconductor industry using reinforcement learning, Procedia CIRP 79 (2018) 391–396.

[9] Kardos, Cs.; Laflamme, C.; Gallina, V.; Sihn, W.: Dynamic scheduling in a job-shop production system with reinforcement learning, Procedia CIRP, 8th CIRP Conference of Assembly Technology and Systems, 29 Sept. – 1. Oct., Athens, Greece, 2020., in print.

[10] S. Qu, J. Wang, S. Govil, J.O. Leckie, Optimized Adaptive Scheduling of a Manufacturing Process System with Multi-SkillWorkforce and Multiple Machine Types: An Ontology-Based, Multi-Agent Reinforcement Learning Approach, *Procedia CIRP* 57 (2016) 55–60.

[11] A. Nair, B. McGrew, M. Andrychowicz, W. Zaremba, P. Abbeel, Overcoming Exploration in Reinforcement Learning with Demonstrations, in: 2018 IEEE International Conference on Robotics and Automation (ICRA), 2018, pp. 6292–6299.

[12] Plappert, M.; Andrychowicz, M.; Ray, A.; McGrew, B.; Baker, B.; Powell, G.; Schneider, J.; Tobin, J.; Chociej, M.; Welinder, P.; Kumar, V.; Zaremba, W.: Multi-Goal Reinforcement Learning: Challenging Robotics Environments and Request for Research, ArXiv, (2018), abs/1802.09464.

[13] Y. Zhu, Z. Wang, J. Merel, A. Rusu, T. Erez, S. Cabi, S. Tunyasuvunakool, J. Kramár, R. Hadsell, N. Freitas, N. Heess, Reinforcement and Imitation Learning for Diverse Visuomotor Skills, *Proceedings of Robotics: Science and Systems*, Pennsylvania, Pittsburgh, 2018, p. 10.

[14] G. Kahn, A. Villaflor, B. Ding, P. Abbeel, S. Levine, Self-Supervised Deep Reinforcement Learning with Generalized Computation Graphs for Robot Navigation, in: 2018 IEEE International Conference on Robotics and Automation (ICRA), 2018, pp. 5129–5136.

[15] P. Long, T. Fan, X. Liao, W. Liu, H. Zhang, J. Pan, Towards Optimally Decentralized Multi-Robot Collision Avoidance via Deep Reinforcement Learning, in: 2018 IEEE International Conference on Robotics and Automation (ICRA), 2018, pp. 6252–6259.

[16] Johannink, T.; Bahl, S.; Nair, A.; Luo, J.; Kumar, A.; Loskyll, M.; Ojea, J.A.; Solowjow, E.; Levine, S.: Residual Reinforcement Learning for Robot Control, 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 2019, pp. 6023-6029.

[19] V. Ranaee, A. Ebrahimzadeh, Control chart pattern recognition using a novel hybrid intelligent method, Applied Soft Computing 11 (2011) 2676–2686.

[20] K. Lavangnananda, S. Khamchai, Capability of Control Chart Patterns Classifiers on Various Noise Levels, Procedia Computer Science 69 (2015) 26–35.

[21] Viharos, Zs. J.; Csanaki, J.; Nacsa, J.; Edelényi, M.; Péntek, Cs.; Kis, K. B.; Fodor, Á.; Csempesz, J.: Production trend identification and forecast for shop-floor business intelligence, ACTA IMEKO, The Open Access e-Journal of the International Measurement Confederation (IMEKO), Vol. 5. No. 4., 2016., pp. 49-55. ISSN: 2221-870X.

[22] G. Köksal, I. Batmaz, M.C. Testik, A review of data mining applications for quality improvement in manufacturing industry, *Expert Systems with Applications*, Elsevier 38 (2011) 13448–13467.

[23] T.T. El-Midany, M.A. El-Baz, M.S. Abd-Elwahed, A proposed framework for control chart pattern recognition in multivariate process using artificial neural networks, Expert Systems with Applications 37 (2010) 1035–1042.

[24] G.D. Pelegrina, L.T. Duarte, C. Jutten, Blind source separation and feature extraction in concurrent control charts pattern recognition: Novel analyses and a comparison of different methods, Computers & Industrial Engineering 92 (2015) 105–114.

[25] Gutierrez, H. De la T.; Pham, D.T.: Estimation and generation of training patterns for control chart pattern recognition, Computers & Industrial Engineering, Vol. 95, 2016., pp. 72-82.

[26] W.-A. Yang, W. Zhou, W. Liao, Y. Guo, Identification and quantification of concurrent control chart patterns using extreme-point symmetric mode decomposition and extremelearning machines, Neurocomputing 147 (2015) 260–270.

[27] A.R. Motorcu, A. Güllü, Statistical process control in machining, a case study for machine tool capability and process capability, Materials and Design 27 (2006) 364–372.

[28] T. Huybrechts, K. Mertens, J. De Baerdemaeker, B. De Ketelaere, W. Saeys, Early warnings from automatic milk yield monitoring with online synergistic control, *American Dairy Science Association*, Vol.v97 (2014) 3371–3381.

[29] Viharos, Zs. J.; Monostori, L.: Optimization of process chains by artificial neural networks and genetic algorithms using quality control charts, Proceedings of Danube - Adria Association for Automation and Metrology, Dubrovnik, 1997., pp. 353-354.

[30] Viharos, Zs. J.; Kis K. B.: Survey on Neuro-Fuzzy Systems and their Applications in Technical Diagnostics and Measurement, Measurement, Vol. 67., 2015., pp. 126-136.

[31] Móricz, L.; Viharos, Zs. J.; Németh, A.; Szépligeti, A.; Büki, M.: Off-line geometrical and microscopic & on-line vibration based cutting tool wear analysis for micro-milling of ceramics, Measurement, Vol. 163., 2020., online available.

[32] J. Xie, Y. Wang, X. Zheng, Q. Yang, T. Wang, Y. Zou, J. Xing, Y. Dong, Modeling and forecasting Acinetobacter baumannii resistance to set appropriate use of cefoperazone-sulbactam: Results from trend analysis of antimicrobial consumption and development of resistance in a tertiary care hospital, American Journal of Infection Control 43 (2015) 861–864.

[33] M. Gan, Y. Cheng, K. Liu, G. Zhang, Seasonal and trend time series forecasting based on a quasi-linear autoregressive model, Applied Soft Computing 24 (2014) 13–18.

[34] C.R. Clarkson, J.D. Williams-Kovacs, F. Qanbari, H. Behmanesh, M.H. Sureshjani, History-matching and forecasting tight/shale gas condensate wells using combined analytical, semi-analytical, and empirical methods, Journal of Natural Gas Science and Engineering 26 (2015) 1620–1647.

[35] I. Koprinska, M. Rana, A.T. Lora, F. Martínez-Álvarez, Combining pattern sequence similarity with neural networks for forecasting electricity demand time series, *The*, International Joint Conference on Neural Networks (IJCNN) 2013 (2013) 1–8.

[36] V.K. Semenychev, E.I. Kurkin, E.V. Semenychev, Modelling and forecasting the trends of life cycle curves in the production of non-renewable resources, Energy 75 (2014) 244–251.

[37] R. Paggi, G.L. Mariotti, A. Paggi, A. Calogero, F. Leccese, Prognostics via Physics-Based Probabilistic Simulation Approaches, *Proc. of Metrology for Aerospace, 3rd IEEE International Workshop* 21–23 (2016) 130–135.

[38] Ed. Francesco, De; De Francesco, Ett.; De Francesco, R.; Leccese, F.; Cagnetti, M.: Improving Autonomic Logistic analysis by including the production compliancy status as initial degradation state, Proc. of Metrology for Aerospace, 3rd IEEE International Workshop, Firenze, Italy, June 21-23, 2016., pp. 371 - 375.

[39] R. Sutton, A.G. Barto, Reinforcement Learning: An Introduction, The MIT Press, Book, 2018.

[40] OpenAI. Openai five. https://blog.openai.com/openai-five/, 2018.

[41] Xu, Z.; van Hasselt, H.; Silver, D.: Meta-gradient reinforcement learning. arXiv, preprint arXiv:1805.09801, 2018.

[43] Andersen, R.E., Madsen, S., Barlo, A.B.K., Johansen, S.B., Nør, M., Andersen, R.S., Bøgh, S., 2019. Self-learning Processes in Smart Factories: Deep Reinforcement Learning for Process Control of Robot Brine Injection. Procedia Manufacturing 38, 171–177. DOI: 10.1016/j.promfg.2020.01.023.

[44] G. Beruvides, A. Villalonga, P. Franciosa, D. Ceglarek, R.E. Haber, Fault pattern identification in multi-stage assembly processes with non-ideal sheet-metal parts based on reinforcement learning architecture, Procedia CIRP 67 (2018) 601–606, https://doi.org/10.1016/j.procir.2017.12.268.

[45] F. Guo, X. Zhou, J. Liu, Y. Zhang, D. Li, H. Zhou, A reinforcement learning decision model for online process parameters optimization from offline data in injection molding, Applied Soft Computing 85 (2019), 105828, https://doi.org/10.1016/j.asoc.2019.105828.

[46] F. Li, Y. Chen, J. Wang, X. Zhou, B. Tang, A reinforcement learning unit matching recurrent neural network for the state trend prediction of rolling bearings, Measurement 145 (2019) 191–203, https://doi.org/10.1016/j.measurement.2019.05.093.

[47] R. Wang, H. Jiang, X. Li, S. Liu, A reinforcement neural architecture search method for rolling bearing fault diagnosis, Measurement 154 (2020), 107417, https://doi.org/10.1016/j.measurement.2019.107417.

[48] P. Ramanathan, K.K. Mangla, S. Satpathy, Smart controller for conical tank system using reinforcement learning algorithm, Measurement 116 (2018) 422–428, https://doi.org/10.1016/j.measurement.2017.11.007.

[49] H. Wu, Y. Dai, C. Wang, X. Xu, X. Jiang, Identification and forewarning of GNSS deformation information based on a modified EWMA control chart, *Measurement*, Volume 160, ISSN 107854 (2020) 0263–2241, https://doi.org/10.1016/j.measurement.2020.107854.

[50] M. Aslam, G. Srinivasa Rao, A. Shafqat, L. Ahmad, R.A.K. Sherwani, Monitoring circuit boards products in the presence of indeterminacy, *Measurement*, Volume 168, ISSN 108404 (2021) 0263–2241, https://doi.org/10.1016/j.measurement.2020.108404.

[51] Viharos, Zs. J.; Jakab, R. B.: Reinforcement Learning for Statistical Process Control in Manufacturing, 17th IMEKO TC 10 and EUROLAB Virtual Conference: "Global Trends in Testing, Diagnostics & Inspection for 2030", October 20-22., 2020., ISBN: 978-92-990084-6-1, pp. 225-234.

[52] V. Kavimani, K.S. Prakash, T. Thankachan, Multi-objective optimization in WEDM process of graphene – SiC-magnesium composite through hybrid techniques, Meas. J. Int. Meas. Confed. 145 (Oct. 2019) 335–349, https://doi.org/10.1016/j.measurement.2019.04.076.

[53] Aquila, G.; Peruchi, R.S.; Rotela, P.; Rocha, Jun.L.C.S.; de Queiroz, A.R.; de O. Pamplona, E.; Balestrass, P.P.: Analysis of the wind average speed in different Brazilian states using the nested GR&R measurement system, Meas. J. Int. Meas. Confed., vol. 115, pp. 217–222, Feb. 2018, doi: 10.1016/j.measurement.2017.10.048.

[54] C.U. Mba, V. Makis, S. Marchesiello, A. Fasana, L. Garibaldi, Condition monitoring and state classification of gearboxes using stochastic resonance and hidden Markov models, Meas. J. Int. Meas. Confed. 126 (Oct. 2018) 76–95, https://doi.org/10.1016/j.measurement.2018.05.038.

[55] G. Peng, Z. Zhang, W. Li, Computer vision algorithm for measurement and inspection of O-rings, Meas. J. Int. Meas. Confed. 94 (Dec. 2016) 828–836, https://doi.org/10.1016/j.measurement.2016.09.012.

[56] F. Wu, Q. Li, S. Li, T. Wu, Train rail defect classification detection and its parameters learning method, Meas. J. Int. Meas. Confed. 151 (Feb. 2020), 107246, https://doi.org/10.1016/j.measurement.2019.107246.

[57] P.M. Gopal, K.S. Prakash, Minimization of cutting force, temperature and surface roughness through GRA, TOPSIS and Taguchi techniques in end milling of Mg hybrid MMC, Meas. J. Int. Meas. Confed. 116 (Feb. 2018) 178–192, https://doi.org/10.1016/j.measurement.2017.11.011.

[58] K.S. Prakash, P.M. Gopal, S. Karthik, Multi-objective optimization using Taguchi based grey relational analysis in turning of Rock dust reinforced Aluminum MMC, Meas. J. Int. Meas. Confed. 157 (Jun. 2020), 107664, https://doi.org/10.1016/j.measurement.2020.107664.

[59] S. Alam, R. Gupta, J. Bera, Quality controlled compression technique for Photoplethysmogram monitoring applications, Meas. J. Int. Meas. Confed. 130 (Dec. 2018) 236–245, https://doi.org/10.1016/j.measurement.2018.07.091.