17th IMEKO TC 10 and EUROLAB Virtual Conference
"Global Trends in Testing, Diagnostics & Inspection for 2030"
October 20-22, 2020.

Reinforcement Learning for Statistical Process Control in Manufacturing

Zsolt János Viharos[1,2], Richárd Jakab[1]

[1]*Intitute for Computer Science and Control, Budapest, Hungary, {viharos.zsolt, jakab.richard}@sztaki.hu*
[2]*John von Neumann University, Kecskemét, Hungary*

***Abstract* – The main concept of the paper is to place Reinforcement Learning (RL) into various fields of manufacturing. As a first attempt, RL for Statistical Process Control (SPC) in production is introduced in the paper; it is a promising approach owing to the adaptability and continuous application capability of reinforcement learning.**
The well-known 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 was required. The formulated goal was to predict the time series value in a certain number of production steps ahead as manufacturing trend forecast. The recent values of the analysed time series were selected as states for the RL and the future probabilities of its values being in the formulated stripes were defined as RL actions. For action update, the Bellman equation was applied and the RL reward depends on how accurate the predicting is. Furthermore, two 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 defined for enabling the comparison of learnings with different RWs by sampling them with the same evaluation frequency.

Keywords – Reinforcement Learning; Manufacturing; Statistical Process Control (SPC); Quality Trend Forecast;

I. INTRODUCTION

Artificial Intelligence (AI) and machine learning approaches are spreading across all territories in our live, it is also valid for technical fields, e.g. for 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 exploitation domain to reach the best match. This mission is valid also to manufacturing, while the well-known Industry 4.0 global initiative (called as Industrial Internet or Cyber Physical Production Systems (CPPS)) supports, moreover, incorporates these directions, consequently, the actual situation is quite promising.

There are various areas of the AI discipline (e.g. machine learning, search techniques, multicriteria optimisation, inference and expert systems, graph modelling and traversal…), nowadays the so called Deep Learning (DL) receives the highest level attention, making it to the most fashionable solution, while sometimes some may forget the other important areas of AI. In general, Machine Learning (ML) is one of the key, basic foundations in AI, originally this branch started with two directions of supervised and unsupervised learning, but the pioneering results of Sutton [1] and his professors and colleagues extended this range to reinforcement learning in 1980s, currently there are also further combinations, e.g. semi-supervised learning.

The spread of various artificial intelligence and machine learning techniques in manufacturing is valid for reinforcement learning as well. However, reviewing the literature mirrors that the domain specific adaptation to various production fields concentrates mainly to production scheduling and robotics. This state-of-the-art status led to the motivation to extend and adapt RL to furthers potential fields of manufacturing, the current paper introduces the RL based SPC with three main advances ahead:

- A novel, general, manufacturing independent, dynamic Q table handling for RL is described, even if it was motivated by the production adaptation challenges.
- The specialities of process control in manufacturing led to the introduction of the so-called Reusing Window (RW) in RL for SPC.
- To compare the efficiencies of various RL solution in production SPC the Measuring Window (MW) had to be introduced.

The paper is organised as follows. After the current introduction the actual status about the reinforcement learning in production field is shortly summarized. The third paragraph introduces the SPC in manufacturing

followed by the novel approach to introduce RL in production, especially for SPC assignments and its test and application results are described in the next paragraph. Conclusions and outlook, acknowledgements and references close the paper.

II. RL IN PRODUCTION

In spite of this high potential situation, the state-of-the art literature mirror that RL applications in manufacturing are concentrated mainly only on two fields: Production scheduling and Robotics.

In *production scheduling*, the state-of-the-art for dynamic scheduling shows a growing increase in the use of RL algorithms. Several papers use Q-learning [1][2][3], deep Q-learning [4] and adapted version of Q-learning [5][6]. Most cases focus on value-based algorithms [1][2][3][4][7], however [3][6] are policy-based. Some researchers use the epsilon-greedy method [2][3][4], whereas Bouazza et al. [1] use it in addition to the machine selection rule. While Kuhnle et al. [3][6] consider the architecture of a RL algorithm framework, Qu et al. [8] analyse the optimal assignment of multi-skilled workers. In [1][8][4] a multi-agent setting is realised. Overall, all papers except [8] use a simulation to test their approach. 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 reduced the average lead time of production orders in a dynamic environment. Moreover, it was shown that as the complexity of the production environment increases, the application of RL for dynamic scheduling becomes more beneficial that makes the future production systems more flexible and adaptive [9].

In the field of *robotics* applications of RL. Nair et al. present a system with RL to solve multi-step tasks [10]. The report by Plappert et al. [11] introduces a suite of challenging continuous control tasks and a set of concrete research ideas for improving RL algorithms. Yuke et al. combined reinforcement and imitation learning for solving dexterous manipulation tasks from pixels. [12]. Kahn et al. have presented high-performing RL algorithm for learning robot navigation policies [13]. Long et al. optimize a decentralized sensor level collision avoidance policy [14]. Johannink et al. studied the combination of conventional feedback control methods [15].

Considering the RL adoptations to various industrial/manufacturing fields, there are many open issues and challenges, the current paper is aiming to bring forward the RL application to the field of SPC in production.

III. 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). The paper of Ranaee and Ebrahimzadeh [20] differentiates in six types of trends that typically arise in SCP charts as presented in Fig. 1.

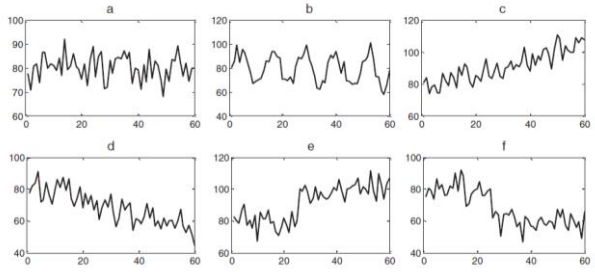

Fig. 1. Six common types of CCPs: (a) normal, (b) cyclic, (c) upward trend, (d) downward trend, (e) upward shift and (f) downward shift.

However, Lavangnananda and Khamchai defined nine variants of patterns [22] (Fig. 2.), where the final one represents that there is a mixture of effects typically, consequently, superposition of patterns can be faced in industry.

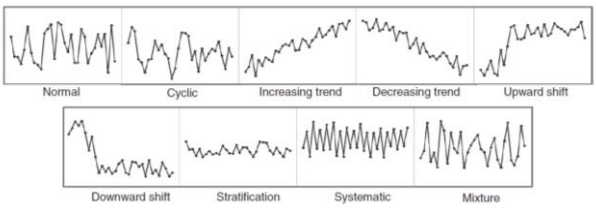

Fig. 2. Nine common types of CCPs: top: normal, cyclic, upward (increasing) trend, downward (decreasing) trend, upward shift ; bottom: downward shift, stratification (novel), systematic (novel) and mixture

Considering the various control chart patterns in Fig. 1. and Fig. 2. it is still an open challenge what can be considered as "Normal" behaviour, what distribution with what parameters it has, even if at all it can be described with a formal statistical distribution, what level of noise is superposed on it, what distribution the noise has (even if at all it has one), even if at all the noise and the basic signal trend can be separated. 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 signal measurement signal.

Köksal et al. reviewed the quality management related applications of various data mining techniques in manufacturing industry [23]. 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 in 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 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 [20] 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 SVM-based classifier module (MCSVM) and an optimization module using genetic algorithm. They tested the algorithm on synthetically generated control charts. Control Chart Patterns (CCPs) with different levels of noise were analysed by Lavangnananda and Khamchai [22]. 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 $\bar{X}$ chart. Pelegrina et al. used different Blind Source Separation (BSS) methods in the task of unmixing concurrent control charts to achieve high classification rates. [12] Gutierrez and Pham presented a new scheme to generate training patterns for ML algorithms: Support Vector Machine (SVM) and Probabilistic Neural Network (PNN) [26]. Yang et al. proposed a hybrid approach that integrates extreme-point symmetric mode decomposition (ESMD) with extreme learning machine (ELM) to identify typical concurrent CCPs [27]. 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 [28].

Huybrechts et al. applied standardization, trend modelling, and an autoregressive moving average (ARMA) model to determine short-term correlation between subsequent measurements. The out-of-control observations can be determined precisely with the Dijkstra model and cumulative sum chart of the corrected residuals between the measured and predicted values. Milk yield data from two Automatic Milking System (AMS) farms and one farm with a conventional milking system were used for the case study [29].

Viharos and Monostori presented an approach, already in 1997 [30] for optimization of process chains by artificial neural networks and genetic algorithms using quality control charts. 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. Also continuous optimization of the production system is possible using the proposed solution.

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) [31] with use of linear regression analysis, quasi-linear autoregressive model [32] or Markov chain models (MCM) [33]. These methods 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 [34]. There are several curve-fitting methods in this field for small sample data, such as genetic algorithm [35]. 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 [36]. Various physical modelling techniques, like finite element methods, analytical equations can represent the known dependencies. Francesco et. al. [37] used effective measurements derived from the conformity tests to improve the accuracy of the Remaining Useful Life (RUL) evaluation.

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

IV. RL FOR SPC IN MANUFACTURING

In the current approach a simulation environment was built up that emulates the production trend behaviour and generates time series signal as it is produced by the manufacturing environments and production plants.

A. Production simulation environment

A simulation environment is created where the RL agent is able to learn while a complete behaviour is known about the trends inside the environment. The simulation is able to generate time series of any length. The time series consist of linear trends, whose lengths are sampled from a uniform distribution. Their slope can have three types – decreasing (-45°), stagnating (0°) and increasing (45°). Due to the complexity and noisiness of the real time series, noise is added to the original trend after it is generated, as every point of the new noisy trend is sampled from a gaussian distribution, where the mean is the value of the original noiseless point and the size of the noise is the standard deviation. Usually, the size of the noise is between 1% and 10% of the interval formed by the two out-of-control boundaries of SPC.

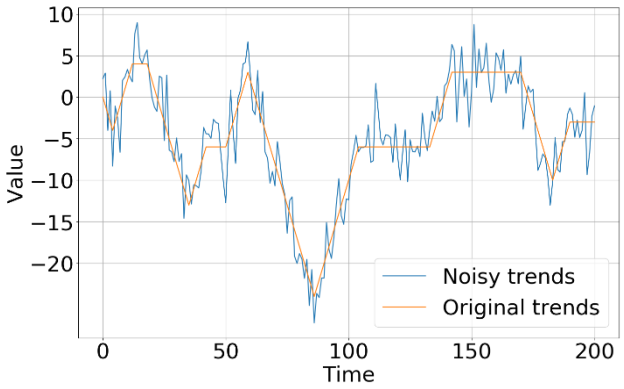

Fig. 3. Generated time series, showing the original and the noisy trend (noise=3, Upper Tolerance Limit (LTL) – Nominal value = 20 = Nominal Value – Lower Tolerance Limit (LTL))

As Fig 3. shows the time series which consists of original trends and another time series, consisting of noisy trends. The learning uses only the noisy time series, but in case requiring the same time series noisier or less noisy it is simple to generate it based on the original time series.

B. Stripes and Q-Table

Before starting to use deep neural networks, which are popular nowadays everywhere, thus here as well, we wanted to analyse the hidden dynamics of the production SPC as time series.

To predict future trends, the problem was broken down to forecast the time series value (which generates a related action in production system) in a certain number of steps forward. For this goal, Q-Table was used owing to its white box nature, meaning that the concealed processes are visible.

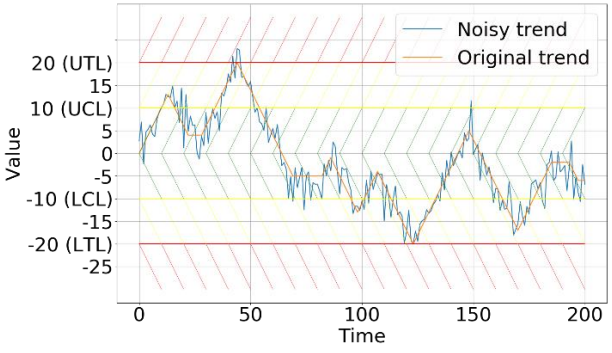

Fig. 4. Time series with stripe boundaries. Green is the normal range (optimal), yellows are the control ranges (warning) and the reds are the out-of-control ranges (failure).

The value range of the signal generated was divided into fix stripes. The time series values in the same stripe get the same quantized value. 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 divided it into five stripes – two out-of-control (under -20 and above 20), two control stripes ((-20, -10] and [10, 20)), and a normal stripe ((-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.

The main structure of the Q-Table is shown in Fig. 5., where states consist of quantized values, measured in T, T-1, T-2, …, order. T is the current time. OOC refers to the Out-Of-Control range, C refers to the Control range and N refers to the Normal range. The minus and plus signs are shown due to the symmetry (above or under the Normal range). T refers to time. The numbers under them are the goodnesses of the actions, respect to the row.

The table 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 during predicting. Concerning the actions, we had a novel idea – in each state, the actions are the stripes for prognosis, so during learning when the best action is searched in every state, in reality the predicted future one's stripe is searched. As a result, in the table each state/row has five actions/columns (according to the number of stripes), meaning the table has five columns and a changeable rows number in summary.

State				Actions				
T – 3	T – 2	T – 1	T	OOC-	C-	N	C+	OOC+
N	N	N	N	0.0	0.05	0.99	0.07	0.0
N	N	N	C+	0.0	0.001	0.5	0.89	0.21
N	N	C+	C+	0.0	0.0	0.6	0.99	0.5
C+	C+	N	C+	0.0	0.03	0.62	0.8	0.17
…								

Fig. 5. A part of the typical Q table and its content in the RL for SPC in manufacturing approach.

C. Dynamic Q-Table in simulation

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 and storage of the table could cause problems, moreover it is unnecessary to allocate the memory for the empty table before it is used. Therefore, we introduced a technique, called dynamic Q-Table, where only as much memory is allocated as required and only when it is required. When the algorithm reaches a new state, it adds it to a list which represents the rows of the Q-Table. Then its initial actions' values are selected randomly. The values of the actions represent the goodness of the actions – high value means that with numerous chance the trend will be in that stripe, which, after several steps, corresponds to the action. So, if the state was not in the Q-Table yet, then it is added to the table's end (in the future research the order of the rows will be much more optimised exploiting the characteristics of the production SPC – e.g. normal states

are much more frequent). Its action values are selected randomly, because we found that it helps exploring at the beginning. If it was already in the Q table, then the chosen action will be updated as it is detailed in the next chapter.

At the beginning, when the agent mostly explores its environment, it often meets with states whom it has not been before, so they are added to the 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 of the state, so in most of such cases new row are needed. As a result, the length of the table grows logarithmically, as it shown in Fig. 6. In the given, concrete case as the Fig 6. shows, 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 recognition rate significantly. It is a very important result proving that there exists a rational limit for the Q table where the size, calculation time, etc. requirements grow significantly but it does not bring valuable additional knowledge to the given SPC trend forecast, consequently, the IT background requirements can be limited.

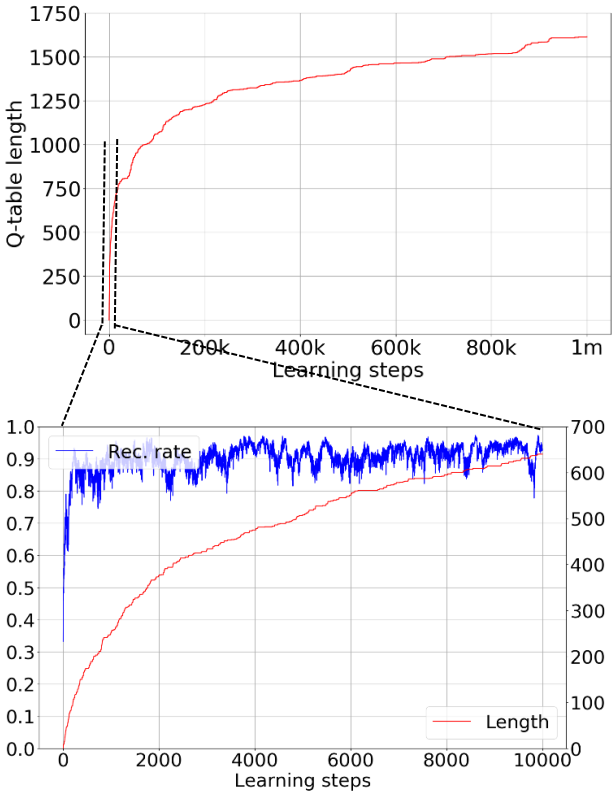

Fig. 6. Length of the Q-Table during a very long learning (top) and in short term the length compared to the recognition rate for production statistical process control forecast (bottom).

D. Learning

During learning, the agent walks through the generated time series, as a moving window. Each moving window will be quantized and become a state. The agent attempts to predict in which stripe it will be at a certain time depending on the current state and its actions' values in the Q-Table. In the next step the agent receives a reward from the environment, according to its prediction. The reward system is defined so that the agent gets 1.0 reward, if its forecast was accurate (the forecasted stripe is became true), 0.5 reward if it predicted one is one of the neighbouring stripes of the real one, and 0.0 reward, if it was more inaccurate than that, so, in all other cases.

After receiving the reward, the chosen action's Q value will be updated using the Bellman equation (eq. 1), where $Q_{s,a}$ is the value of the a action of row s of the Q-Table. γ is a parameter, $0 \leq \gamma \leq 1$, called as discount rate. The discount rate determines the present value of future rewards: a reward received k time steps in the future is worth only γ^{k-1} times what it would be worth if it were received immediately [38]. r marks the reward.

$$Q_{s,a} = Q_{s,a} + \gamma \cdot (r - Q_{s,a}), \qquad (1)$$

Action selection is a widely researched area in the RL, it is well known as the "Exploration-Exploitation Trade-off" [39]. 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. The agent has to explore the environment by performing actions and perceiving their consequences. At some point in time, it will have a policy with a particular performance. In order to see whether there are possible improvements to this policy, it sometimes has to try out various 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 explore 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.

In this paper the ε greedy algorithm is used [38] – the algorithm chooses one action based on a number of ε. The probability of choosing one action randomly is epsilon and the probability to choose the one with the largest value (best action) is 1-ε. During learning, the value of epsilon decreases exponentially, therefore the algorithm rather explores at the beginning and it gains knowledge about its environment and exploits it. This ε greedy algorithm is the most popular and widely applied in research and practice.

V. REUSAGE WINDOW (RW) AND MEASUREMENT WINDOW (MW)

In production environment each measured value has significant cost coming from many sources, e.g. cost of the equipment, training of the personnel, doing the activity, establishment if the IT background for collection, communication and storage, moreover softwares to use the measurements, moreover, continuous re-calibration of the measuring devices, their maintenance, etc. Consequently, the measured values in manufacturing have high value and has to be exploited as much us possible. In the production control environment of today, this ideal situation is far not yet reached, the data asset is higher than its usage.

The main problem with the method mentioned above is that it lacks of reusability. In manufacturing, the time series is a series of measurements, where every measurement is a produced component/product/process, which may be expensive. Consequently, using all measurement values only once seems very wasteful. Therefore, it is desirable to reuse measurement values, and this is why we introducer the Reusage Window (RW) concept, which tells how many times we reuse the individual values during learning.

A. Reusage Window (RW)

As opposit to the previous concept, where the agent walks through the time series only once, with RW it works as follows: an interval is selected from the time series the and the moving window of 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 on the time series, and it starts again, until the end of the RW meets with the end of the time series, then the learning stops for that individual time window. It means that one data is (re)used so many times as long the length of the RW is.

As Fig. 7. represents, the RW moving window goes through step-by-step the time series and selects an interval (marked in green). Within that interval, the agent goes through that, and process 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 step further and the whole process starts again. The RW goes all the way to the end of the time series.

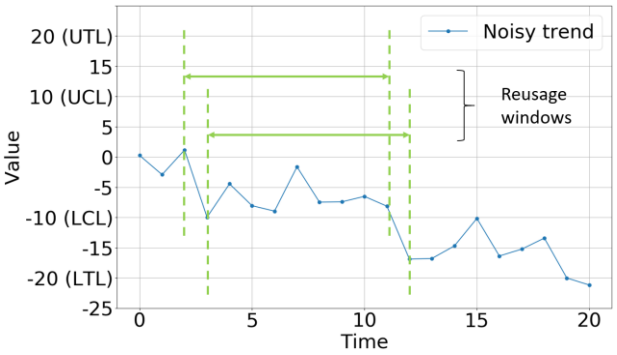

Fig. 7. The RW (here it is 10) selects an interval within the time series. The blue points are the measured values. The figure represents the repeated reinforcement learning as the cycle of the continuously shifted reusage windows.

B. Measurement Window (MW)

Accuracies of different agents with their recognition rate curves are measured – after every learning iteration, the received rewards are summarized and is divided by the length of the learning iteration (which is basically the RW). However, when recognition rates with different RWs are compared, the curve of the higher RW will be more balanced, because in that case the algorithm averages more proportions in the same time, resulting the outliers' moderation. Therefore, for a more accurate comparison it is needed to standardize the length of that interval within the performance evaluation takes place and apply it in all cases. It the established implementation during learning the agent-predicted stripe (action) and the actual stripe where the predicted point of the trend appeared are stored, therefore, it is possible to choose arbitrarily the length of the interval for evaluation, which is the MW. Fig. 8. shows a comparison case for different RW evaluated by the same MW, so, with the proposed MW concept the performances are (finally) equal that was not the case without the MW concept.

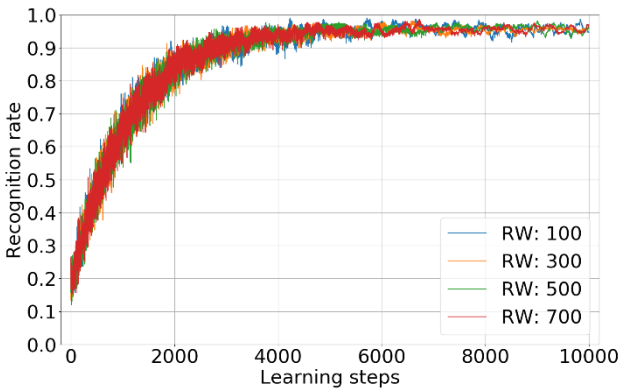

Fig. 8. Comparing different recognition rates with different RWs. (noise=0, MW=150)

VI. EXPERIMENTAL RESULTS

In this section the performance of the learning system as the production SPC recognition rates of different RWs, MWs and noise levels are compared. It is desirable to find the appropriate range of RWs where the recognition rates reach high values while the resources usage (time, calculate time) are minimalized/acceptable. It is also desirable to find the range of MWs with which the system performance is accurate and not too noisy and no information is loosed.

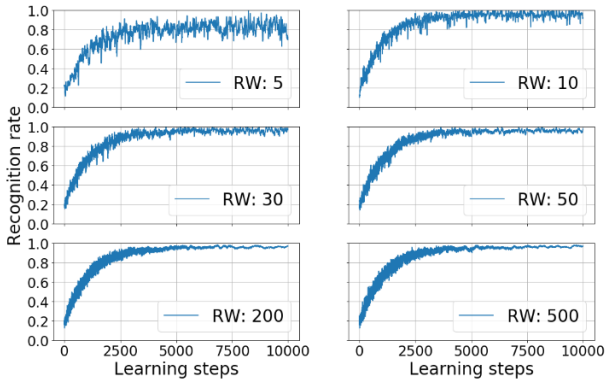

Fig. 9. Comparing different recognition rates with different RWs. (noise=0, MW=150)

As Fig. 9. shows, as RW increases the recognition rates reach higher levels and their deviations decrease. It can be seen that *RW=5* is too small, because it barely reusages the data, while *RW=500* is too large, because the difference between it and *RW=200* barely noticeable – but it requires 2.5 times more learning time for the same result. Taken in consideration that learning requires time and computing capacity, it is not meaningful using higher RWs than 200 in the given case. It means that reusing of production data is especially important and necessary, however it has a maximum reusing ratio, consequently, the RW has a maximal length, that is typically domain and situation dependent.

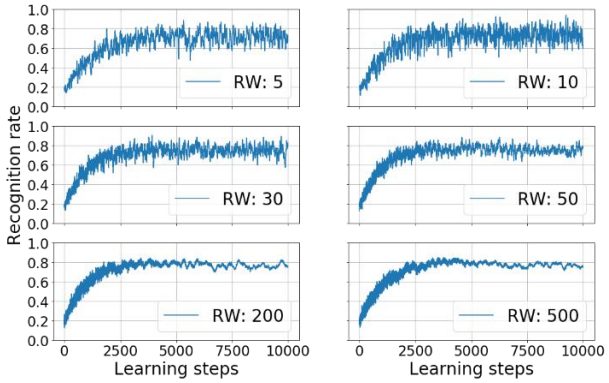

Fig. 10. Comparing different recognition rates with different RWs. (noise=5, MW=150)

At higher noise levels (Fig. 10. And Fig. 11.), the same situation is observed because the difference between the curves of *RW=200* and the curves of *RW=500* is not significant, while *RW=5* or *RW=10* are very noisy, fluctuating and unstable for appropriate and reliable evaluation.

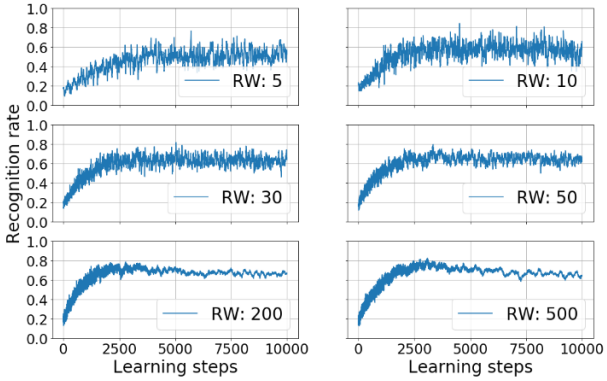

Fig. 11. Comparison among various recognition rates with different RWs. (noise=10, MW=150)

As conclusion, in the current manufacturing domain there is no need and in parallel no further potential to reuse measured data more than 200 times, so, RW = 200 is a case related appropriate choice.

MW related calculations have also computational requirements and comparison expectations, consequently, it is also worth to analyse whether it has an useful maximum value, so, maximum length, meaning, that the SPC in production has also a maximal time period to that is worth to analyse and e.g. to supervise.

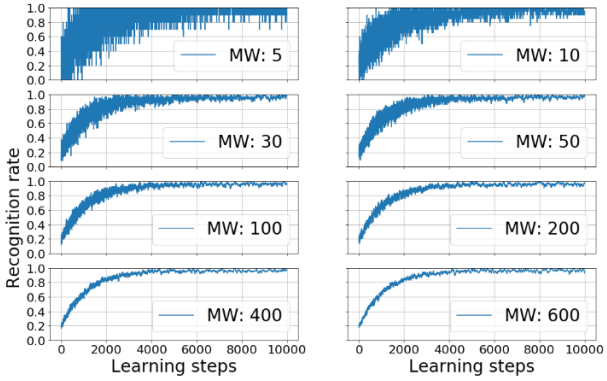

Fig. 12. Comparing RL performance with different MWs. (noise=0, RW=50)

As Fig. 12. shows, at the lower MWs the accuracy and the reliability of production system evaluation decreases – low MW means thin intervals in within the rewards are averaged, meaning that the individual rewards affect too much. High MW solves this problem, but in that case, information loss appears because each data was used 50 times (because of *RW=50*), but in the averaging process each of them was used 600 times (because of *MW=600*), therefore it creates a significantly smoother curve than the reality is.

VII. INDUSTRIAL APPLICATION

To test the concept, a simulation based on real consequences was created. Thousands of industry data was processed to determine the hidden dynamics of the real time series.

The whole process is automatic. Events are sampled from Gaussian distributions determined on the data, which affect the time series, for example it moves the time series' mean outward, increases its noise or it immediately jumps the time series into an outer stripe. We created events like "tool brokes" or "measurement error". The algorithm's goal is to keep the time series in the normal zone. To reach this, it affects on the time series with actions like "tool change", "verification" or "no action". This is shown on Fig. 13., where actions are presented (except the "no action" action). After the action, from the simulation environment the algorithm recieves a reward based on the action's cost (e.g. "tool change" is more costly than "verification") and on "how well" it moves the time series inward, meaning its distance from the normal center.

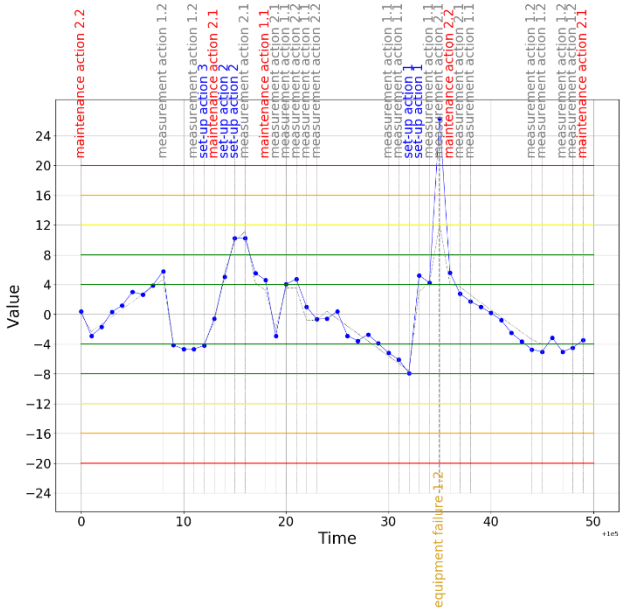

Fig. 13. Production trend behaviour, production (line) event (bottom) and suggested actions (top) by the reinforcement learning agent.

VIII. CONCLUSIONS AND OUTLOOK

The paper introduced the concept and the solution to place Reinforcement Learning (into Statistical Process Control in manufacturing. It was proved that it is a promising approach owing to the adaptability and continuous application capability of reinforcement learning.

The well-known Q-Table method 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. The formulated goal was to predict the time series value in a certain number of production steps ahead as manufacturing trend forecast.

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 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 defined for enabling the 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 in production and the precise evaluation requirement about the performance of the production system.

IX. ACKNOWLEDGMENTS

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 and by the Hungarian ED_18-2-2018-0006 grant on a "Research on prime exploitation of the potential provided by the industrial digitalisation" (https://inext.science/).

REFERENCES

[1] **Barto, A.G., Sutton, R.S., & Brouwer, P.**: Associative search network: A reinforcement learning associative memory, *Biological Cybernetics*, Vol. 40. 1981, pp. 201-211.

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

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

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

[5] **Waschneck, B.; Reichstaller, A.; Belzner, L.; Altenmüller, T.; Bauernhansl, T.; Knapp, A.; Kyek, A.;** Optimization of global production scheduling with deep reinforcement learning, *Procedia CIRP*, 72, 2018, pp. 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] **Kuhnle, Al.; Schäfer, L.; Stricker, N.; Lanza, G.:** Design, Implementation and Evaluation of Reinforcement Learning for an Adaptive Order Dispatching in Job Shop Manufacturing Systems, *Procedia CIRP*, 81, 2019, 234–239.

[8] **Kuhnle, A.; Röohrig, N.; Lanza, G.:** Autonomous order dispatching in the semiconductor industry using reinforcement learning. *Procedia CIRP*, 79, 2018. pp. 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] **Qu S., Wang, J., Govil, S., Leckie, J. O.:** 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, pp. 55–60.

[11] **Nair, A.; McGrew, B.; Andrychowicz, M.; Zaremba, W.; Abbeel, P.**: Overcoming Exploration in Reinforcement Learning with Demonstrations, *2018 IEEE International Conference on Robotics and Automation (ICRA)*, Brisbane, QLD, 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] **Zhu, Y.; Wang, Z.; Merel, J.; Rusu, A.; Erez, T.; Cabi, S.; Tunyasuvunakool, S.; Kramár, J.; Hadsell, R.; Freitas, N.; Heess, N.:** Reinforcement and Imitation Learning for Diverse Visuomotor Skills, *Proceedings of Robotics: Science and Systems*, Pittsburgh, Pennsylvania, 2018, 10 p.

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

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

[17] **Richard, S.; Andrew, B.:** Reinforcement Learning: An Introduction, 2018.

[18] **van Otterlo M., Wiering M.** Reinforcement Learning and Markov Decision Processes. In: Wiering M., van Otterlo M. (eds) Reinforcement Learning. Adaptation, Learning, and Optimization, vol 12. Springer, Berlin, Heidelberg, 2012, https://doi.org/10.1007/978-3-642-27645-3_1

[19] **Popescu, A.**: *Electrical machines and drives*, Politehnium, Iasi, 2011, p. 234.

[20] **Curtley, M, Prince, O.A.**: A synchronous detector with improved parameters, *Proceeding of the International Symposium on Circuits and Systems*, Barcelona, July 15-17, 2009, pp. 255-260.

[21] **Ranaee, V.; Ebrahimzadeh, A.**: Control chart pattern recognition using a novel hybrid intelligent method, *Applied Soft Computing*, Vol.11, 2011., pp. 2676–2686.

[22] **Lavangnananda, K.; Khamchai, S.**: Capability of Control Chart Patterns Classifiers on Various Noise Levels, *Procedia Computer Science*, Vol. 69, 2015, pp. 26–35.

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

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

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

[26] **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.

[27] **Yang, W.-A.; Zhou, W.; Liao, W.; Guo, Y.**: Identification and quantification of concurrent control chart patterns using extreme-point symmetric mode decomposition and extremelearning machines, *Neurocomputing*, Vol. 147, 2015, pp. 260-270.

[28] **Motorcu, A. R.; Güllü, A.**: Statistical process control in machining, a case study for machine tool capability and process capability, *Materials and Design*, Vol. 27, 2006., pp.364–372.

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

[30] **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.

[31] **Xie, J.; Wang, Y.; Zheng, X.; Yang, Q.; Wang, T.; Zou, Y.; Xing, J.; Dong, Y.**: 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*, vol.43, 2015, pp.861-864.

[32] **Gan, M.; Cheng, Y.; Liu, K.; Zhang, G.**: Seasonal and trend time series forecasting based on a quasi-linear autoregressive model, *Applied Soft Computing*, Vol. 24, 2014, pp.13-18.

[33] **Clarkson, C. R.; Williams-Kovacs, J. D.; Qanbari, F.; Behmanesh, H.; Sureshjani, M. H.**: History-matching and forecasting tight/shale gas condensate wells using combined

analytical, semi-analytical, and empirical methods, *Journal of Natural Gas Science and Engineering*, Vol. 26, 2015, pp.1620-1647.

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

[35] **Semenychev, V. K.; Kurkin, E. I.; Semenychev, E. V.**: Modelling and forecasting the trends of life cycle curves in the production of non-renewable resources, *Energy*, Vol.75., 2014, pp. 244-251.

[36] **Paggi, R.; Mariotti, G. L.; Paggi, A.; Calogero, A.; Leccese, F.**: Prognostics via Physics-Based Probabilistic Simulation Approaches, *Proc. of Metrology for Aerospace, 3rd IEEE International Workshop*, Firenze, Italy, June 21-23, 2016, pp. 130 - 135.

[37] **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.

[38] **Richard, S.; Andrew, B.**: Reinforcement Learning: An Introduction, *Book, The MIT Press*, 2018.

[39] **van Otterlo, M.; Wiering, M.:** Reinforcement Learning and Markov Decision Processes, *Wiering M., van Otterlo M. (eds) Reinforcement Learning. Adaptation, Learning, and Optimization*, Vol 12. Springer, Berlin, Heidelberg, 2012, https://doi.org/10.1007/978-3-642-27645-3_1