2022

Deep Residual Policy Reinforcement Learning as a Corrective Term in Process Control for Alarm Reduction: A Preliminary Report

Ammar N. Abbas
Software Competence Center Hagenberg, Austria

Georgios C. Chasparis
Software Competence Centre Hagenberg, Austria

John Kelleher
Technological University Dublin, john.kelleher@tudublin.ie

Funder: Science Foundation Ireland (SFI) Research Centres Program (Grant No. 13/RC/2106 P2).

Deep Residual Policy Reinforcement Learning as a Corrective Term in Process Control for Alarm Reduction: A Preliminary Report

Ammar N. Abbas
Data Science, Software Competence Center Hagenberg, Austria. E-mail: ammar.abbas@scch.at

Georgios C. Chasparis
Data Science, Software Competence Center Hagenberg, Austria. E-mail: georgios.chasparis@scch.at

John D. Kelleher.
ADAPT Research Centre, Technological University of Dublin, Ireland. E-mail: john.d.kelleher@tudublin.ie

Conventional process controllers (such as proportional integral derivative controllers and model predictive controllers) are simple and effective once they have been calibrated for a given system. However, it is difficult and costly to re-tune these controllers if the system deviates from its normal conditions and starts to deteriorate. Recently, reinforcement learning has shown a significant improvement in learning process control policies through direct interaction with a system, without the need of a process model or the system characteristics, as it learns the optimal control by interacting with the environment directly. However, developing such a black-box system is a challenge when the system is complex and it may not be possible to capture the complete dynamics of the system with just a single reinforcement learning agent. Therefore, in this paper, we propose a simple architecture that does not replace the conventional proportional integral derivative controllers but instead augments the control input to the system with a reinforcement learning agent. The agent adds a correction factor to the output provided by the conventional controller to maintain optimal process control even when the system is not operating under its normal condition.

Keywords: Deep Reinforcement Learning (DRL), Residual Policy Learning (RPL), process control, optimal control, and alarm management.

1. Introduction

Industrial processes control has become autonomous with the advent of sophisticated control strategies such as *Proportional Integral Derivative* (PID) or *Model Predictive Control (MPC)* Efheij et al. (2019), based on look-ahead optimization. However, one of the major disadvantages of such control laws is that their implementation requires an explicit understanding of the system dynamics and sometimes also knowledge of the environment. Furthermore, once the controller is tuned to the specific model or setpoints of the system it only provides the optimal control under set system specificities. If the system deteriorates or the environmental conditions and setpoints drift from the normal conditions, the controller starts deviating and provides suboptimal control strategies and sometimes can fail to control the process at all. In these cases, it becomes necessary to optimize the controller performance by re-tuning the controller parameters and re-identification the system, tasks that lead to process shutdowns, and massive time consumption Spielberg et al. (2019).

Recent developments in model-free *Deep Reinforcement Learning (DRL)* have demonstrated the feasibility of replacing such controllers with fully autonomous controllers that interact with the environment in an online setting and create their understanding of the model of the environment, thereby eliminating the need for system re-identification Spielberg et al. (2019). Reinforcement Learning (RL) is a branch of machine learning that learns through interaction with the environment without having prior knowledge of the data set Sutton and Barto (2018). Most of the work on DRL for process control replaces conventional controllers entirely with the DRL controller, as suggested in Spielberg et al. (2019);

Proceedings of the 32nd European Safety and Reliability Conference.
Edited by Maria Chiara Leva, Edoardo Patelli, Luca Podofillini and Simon Wilson

ISBN: 981-973-0000-00-0 :: doi: 10.3850/981-973-0000-00-0_output

Nian et al. (2020); McClement et al. (2021); Conradie and Aldrich (2001); Mageli (2019). Such an approach is well suited for simpler control problems. However, developing a controller for sophisticated control scenarios generally requires either proper domain knowledge or a very complex DRL algorithm structure that is not easily generalizable.

Process control is a critical optimization problem that needs to consider optimizing every time step to be able to run the process smoothly because if it fails, at any instant, then the process trips (shutdown), and this may lead to catastrophic failures. DRL was developed to solve an optimization problem without considering the path that optimal policy takes to achieve the maximum cumulative reward. Therefore, it is not necessarily appropriate to replace conventional control with DRL, as the trajectory a process follows can have a major impact on the process control. Hence, we argue that it is best to use DRL in a hybrid setting with the conventional controllers, as also recommended by Shin et al. (2019).

Therefore, we propose a methodology that merges the conventional controller with a DRL-based correction factor applied to each output of the controller, an approach called *Residual Policy Learning* (RPL) Silver et al. (2018) in process control. The corrected signal is then fed as an input to the plant. This correction factor aids the adjustment of the control in the case of system disturbances or when the controller requires retuning. DRL interacts with the process in real-time and generates an additional control signal that rectifies the output provided by the conventional controller (PID/MPC) and results in optimal control with reduced alarm scenarios and operator burden.

2. Related Literature

A model-free adaptive and self-learning DRL controller is proposed Spielberg et al. (2019). The proposed controller learns while interacting with the process in real-time; therefore, it is a data-based approach. The proposed system uses an actor-critic architecture Konda and Tsitsiklis (1999) for the DRL agent based on the Deep Policy Gradient (DPG) Lillicrap et al. (2015). To make the DRL agent aware of the system dynamics, the state is defined as the current state, as well as the previous states, and the current control action taken by the RL agent, as well as the previous control actions, up to a predefined number of the previous time steps. Additionally, the state also incorporates the current deviation from the setpoint defined by the system. The approach is validated on the setpoint tracking problem in control theory where the controller has to reach the predefined setpoint with minimal oscillations and time while reducing the error caused by the deviation of the system state from the defined setpoints. The performance of the DRL controller is evaluated through simulation experiments with several use cases, including (i) a paper machine, (ii) a distillation column, and (iii) a heating, ventilation, and air conditioning (HVAC) system.

A multi-criteria decision-making control process using DRL has been implemented He et al. (2021) and has been evaluated using the case study of a textile manufacturing process. Process optimization for the textile industry includes various parameters that must be tuned simultaneously, and DRL is well suited for such multi-objective optimization.

Panzer and Bender (2021) provide a review of the literature on the use of DRL in production systems. The research reviewed was applied across several case studies, such as the liquid level control of multiple connected tanks, single- and multi-input and -output processes, and chemical-mechanical polishing Noel and Pandian (2014); Spielberg et al. (2017); Yu and Guo (2020). In all reviewed case studies, DRL was used to replace the conventional controller, and the DRL-based controller achieved optimal performance with reduced maintenance and cost along with increased process stability relative to conventional control strategies.

Mageli (2019) used a DRL agent to replace regular controllers in a case study of tank-level regulation. The DRL controller was compared with a *Proportional* controller, a type of PID controller where only the first component Proportional (P) is used. The results showed that the P-controller performed better with stable controller

output changes, whereas DRL with larger output changes resulted in system oscillation. This research shows that replacing a conventional controller with a DRL agent does not always result in improved performance, particularly in relatively simple control scenarios where the complexity of a DRL agent may not be necessary. However, in more complex scenarios with potentially nonlinear system dynamics that require the controller to have the ability to accommodate the deviation of the system from the standard operating conditions for which the controller was originally tuned DRL has great potential.

A generalizable approach to process control using DRL is used McClement et al. (2021). The approach can be integrated within existing control structures and used to tune the PID or MPC controllers, or it can be used as an independent controller without the aid of any other existing control. For example, DRL is used as a setpoint decision-maker Hernández-del Olmo et al. (2018) in a wastewater treatment plant, where the suggested setpoint is then controlled using a PID controller.

Shin et al. (2019) presents a brief introduction to RL and its use in process control, followed by its limitations and comparison with conventional controllers. They argue that model-based/mathematical programming-based controllers such as MPC are limited in their ability to incorporate stochasticity of the environment and that RL can overcome these issues. Furthermore, they identify three strategies for implementing RL in process control: (i) replacing the conventional control with RL, (ii) hybrid RL and conventional controller, and (iii) RL to manage the control systems (PID tuning or MPC gain adjustments). In this paper, the second method of using a hybrid model is followed and an instantiation of this strategy is proposed.

Residual reinforcement learning (or residual policy learning (RPL) is effective in the context of robot control that involves stochastic events (uncertain and random) as shown by Johannink et al. (2019); Silver et al. (2018). Robot control includes dynamics that can not be easily computed with the first-order physical modeling and therefore is difficult to tune manually. They show the effectiveness of such an approach of superpositioning two control signals (RPL). A part of the control is solved by a conventional control and the residual is solved by RL. The effectiveness is proven in a real-world block assembly task performed by the robot. Kulkarni et al. (2022) discusses that learning a control strategy for a robot through RL is data-inefficient, time-consuming, and involves high risk. In contrast, the conventional untrained classical controllers are nearly optimal and reliable. However, the real-world environment is stochastic and does not allow the classical control to achieve optimality. Therefore, to get the best of both worlds, a strategy is proposed by the authors that combine classical control with a recurrent RL with a time-varying weighted sum that achieves accurate and robust control of the system.

Liu et al. (2022) use RPL for the control of a blimp. They point out the inherent non-linear dynamics and the time-delayed response of the blimp structure that makes a PID-controller difficult to tune and demonstrated the ability of the RPL system to robustly control the blimp even in the case of disturbances such as the windy conditions.

The approach proposed in this paper is inspired by the *deep residual policy reinforcement learning* strategy used by Zhang et al. (2019) and that will be described in more detail in section 3. They use a bidirectional target network to stabilize the residual policy learning and evaluate the performance on the DeepMind Control Suite benchmark. They show that the residual algorithm also solves the problem of the distribution mismatch, even with a weaker model assumption.

2.1. *Literature Gap*

Several hybrid structures of MPC with RL were proposed Lee and Wong (2010). The first method is a hierarchical structure where MPC determines the state regions to focus on for RL. The second includes a learning value function for states to capture the uncertainties within the system model and incorporate them within the MPC formulation. The third approach uses switching between MPC and RL, where MPC is used instead of RL

when a new state is observed. Another example is the dual MPC methodology introduced by Morinelly and Ydstie (2016), where RL is used to incorporate the predicted information within the model.

Most of the hybrid DRL-based conventional controller either uses RL to predict uncertainties within the environment and then incorporate such information within the mathematical modeling or uses RL independently with the conventional controllers with a switching probability. However, we propose to use DRL alongside the process controllers and to act as a correcting agent that feeds in the information of the current state and action proposed by the conventional control and outputting a correction factor added to the output of the conventional control same as RPL. This method can help correct the control signal during the process disturbances and abnormalities where the probability of occurrence of multiple alarms is high, and it can help minimize or mitigate such alarm scenarios. Therefore, the main contribution of this paper is to use the method of RPL within the context of process control to help optimize the process in the presence of stochastic events.

3. Proposed Methodology

RPL uses the approach of learning on top of the baseline policy. We present the mathematical concept for it in the next subsection.

3.1. *Preliminaries*

We propose a Deep Residual Policy Reinforcement Learning (DRPRL) as a corrective term in process control to reduce process alarms and deviations when the system observes some uncertain and stochastic events or when the system requires recalibration. PID or conventional controllers tend to lose their effectiveness in such scenarios and these cases, the process observes an alarm flood.

3.1.1. *Residual Policy Learning (RPL)*

RPL is a strategy to enhance non-differentiable baseline policies through model-free DRL. The goal is to improve the baseline policy in cases where manual improvement and retuning is not viable option. It can also be categorized into the paradigm of imitation learning and learning from demonstrations, where the initial guidance is provided to the machine learning algorithm for a more directed learning approach. RPL works within a framework of Partially Observable Markov Decision processes (POMDPs), where the state of the system is not fully observable and depends on some hidden unobservable factors. This makes it ideally suited to be used in the process industry which normally is viewed as POMDP. Given an initial policy π_θ, it learns a residual policy f_θ over the initial baseline policy π, at a given state s as shown in eq. (1). Further, eq. (2a) represents a PID control law, where K_p, K_i, and K_d are the tunable gains of the control, and $e(t)$ is the error between the actual process output and the desired setpoint. eq. (2b) is the control action provided by the DRL agent, where Q is the state-action value. RPL is data-efficient and practical in the case of process industries, where the DRL agent is not allowed to explore to avoid catastrophic failures than learning from scratch (i.e. learning without any knowledge of the world/environment). There are two ways in which an RPL can be used. The first one is for the case when the initial baseline policy is nearly perfect and only requires recalibration or retuning when anomalous behaviour occurs, therefore, in these cases, the residual policy is used as a corrective term. The second case is when the initial policy is far from ideal and therefore, the initial policy provided by the conventional controller just provides the guidance to the reinforcement learning for exploration. We use the first case in the context of the process industry.

$$\pi_\theta(s) = u(t) + f_\theta(s) \tag{1}$$

$$u(t) = K_{\mathrm{p}} e(t) + K_{\mathrm{i}} \int_0^t e(t)\mathrm{d}t + K_{\mathrm{d}} \frac{\mathrm{d}e(t)}{\mathrm{d}t} \tag{2a}$$

$$f_\theta(s) = \max_\theta \mathop{\mathrm{E}}_{s\sim\mathcal{D}} \left[Q_\phi \left(s, \mu_\theta(s) \right) \right] \tag{2b}$$

3.2. *Deep Residual Policy Reinforcement Learning (DRPRL)*

Our proposed methodology is to add the integrated DRL agent to the industrial process. The agent continuously observes the state of the system and at each time step provides a corrected signal to be added to the output of the PID/MPC controller, which is then fed as the control signal to the plant. We propose two different architectures in terms of the representative state of the RL as shown in fig. 1 and fig. 2. The objective function of the agent is to minimize the deviation from the setpoint during the disturbance phase, where the PID/MPC controller fails to mitigate the error and reduce the number of alarms.

The first architecture shown in fig. 1 represents the state as a function of the industrial process (plant) output concatenated with the output signal from the PID controller. In the figure, the term Y_{sp} denotes the setpoint or a reference set by the control operator at the beginning of the plant operation. The difference operator after the reference is used to calculate the difference between the actual plant output at the current time y_t and the reference signal Y_{sp}. This error term is then fed to a controller, which provides the control signal to the plant to mitigate the error as smoothly and as quickly as possible. The proposed methodology is to add a corrective term to this controller output before feeding it to the plant.

The modified architecture as shown in fig. 2 represents the state as the function of output from the plant, output signal from the PID controller, deviation of process variables from the setpoint, and the previous action proposed by the DRL agent. The difference between the two architectures is in terms of the input fed to the DRL agent. The advantage of having a more informative state representation for the DRL makes the state fully observable; however, on the other hand, with more features in the state space, it becomes difficult for the agent to explore the whole state space efficiently to find the optimal policy.

3.3. *Reward Formulation*

Three different reward functions can be used based on the error signal received from the environment ($Y_{sp} - y_t$): (i) the norm L1 as shown in eq. (3), (ii) the norm L2 as shown in eq. (4), and (iii) the polar reward as shown in eq. (5), as represented in Spielberg et al. (2019). In these equations n_y represents the number of process outputs (sensor readings), y_t represents the current measurements and y_sp represents the setpoint configured by the operator at the start of the process. The first two reward functions will enable an agent to learn faster but will likely result in more oscillation in the control signals than the third. The third reward function stops penalizing the agent once it observes the improvement in terms of the current reward compared with the reward at the previous time step.

$$r\left(s_t, a_t, s_{t+1}\right) = -\sum_{i=1}^{n_y} \left|y_{i,t} - y_{i,sp}\right| \quad (3)$$

$$r\left(s_t, a_t, s_{t+1}\right) = -\sum_{i=1}^{n_y} \left|y_{i,t} - y_{i,sp}\right|^2 \quad (4)$$

$$r\left(s_t, a_t, s_{t+1}\right) = \begin{cases} 0 & \text{if } \left|y_{i,t} - y_{i,sp}\right| > \left|y_{i,t+1} - y_{i,sp}\right| \\ -1 & \text{otherwise} \end{cases} \quad (5)$$

4. Conclusion and Future Work

In this paper, we proposed a hybrid architecture of conventional process control and DRL (RPL) and its potential applications in the case of alarm reduction and mitigation. It aims to help the operators in an abnormal situation where handling multiple alarms simultaneously becomes difficult, which obscures the root cause failure of the system. In the future, we aim to use this methodology in a real-world case study with historical data or with the help of a simulator and compare the performance of such hybrid architecture over conventional control or the replacement of conventional control by DRL. The state and reward architectures presented in the simplified and modified methodology will be compared and evaluated against the benchmark of an average of the total number of alarms generated compared to the conventional control.

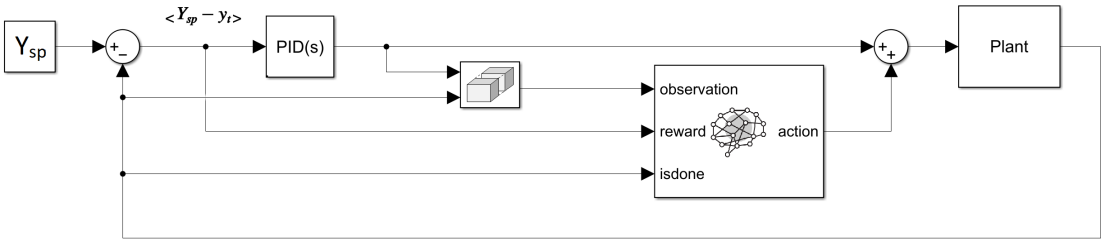

Fig. 1.: Simplified DRL-RA methodology.

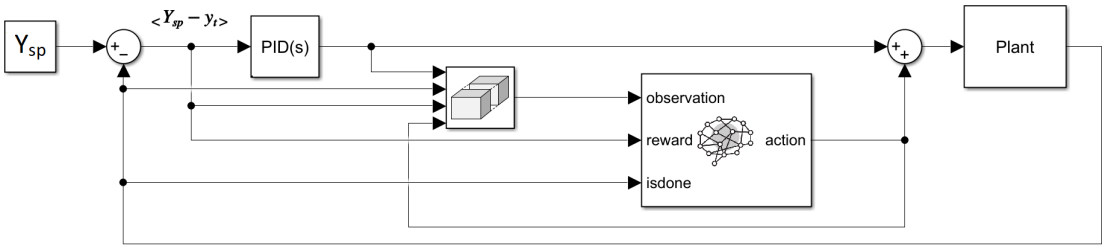

Fig. 2.: Enriched-state-modified DRL-RA methodology (ES-DRL-RA).

Acknowledgement

This research is supported by the Collaborative Intelligence for Safety-Critical systems (CISC) project; the CISC project has received funding from the European Union's Horizon 2020 Research and Innovation Program under the Marie Skłodowska-Curie grant agreement no. 955901. The work of Kelleher is also partly funded by the ADAPT Centre which is funded under the Science Foundation Ireland (SFI) Research Centres Program (Grant No. 13/RC/2106_P2).

References

Conradie, A. and C. Aldrich (2001). Plant-wide neuro-control of the tennessee eastman challenge process using evolutionary reinforcement learning. In *In 3rd international conference in intelligent processing and manufacturing of materials*.

Efheij, H., A. Albagul, and N. Ammar Albraiki (2019). Comparison of model predictive control and pid controller in real time process control system. In *2019 19th International Conference on Sciences and Techniques of Automatic Control and Computer Engineering (STA)*, pp. 64–69.

He, Z., K.-P. Tran, S. Thomassey, X. Zeng, J. Xu, and C. Yi (2021). A deep reinforcement learning based multi-criteria decision support system for optimizing textile chemical process. *Computers in Industry 125*, 103373.

Hernández-del Olmo, F., E. Gaudioso, R. Dormido, and N. Duro (2018). Tackling the start-up of a reinforcement learning agent for the control of wastewater treatment plants. *Knowledge-Based Systems 144*, 9–15.

Johannink, T., S. Bahl, A. Nair, J. Luo, A. Kumar, M. Loskyll, J. A. Ojea, E. Solowjow, and S. Levine (2019). Residual reinforcement learning for robot control. In *2019 International Conference on Robotics and Automation (ICRA)*, pp. 6023–6029. IEEE.

Konda, V. and J. Tsitsiklis (1999). Actor-critic algorithms. *Advances in neural information processing systems 12*.

Kulkarni, P., J. Kober, R. Babuška, and C. Della Santina (2022). Learning assembly tasks in a few minutes by combining impedance control and residual recurrent reinforcement learning. *Advanced Intelligent Systems 4*(1), 2100095.

Lee, J. H. and W. Wong (2010). Approximate dynamic programming approach for process control. *Journal of Process Control 20*(9), 1038–1048.

Lillicrap, T. P., J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra (2015). Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*.

Liu, Y. T., E. Price, M. J. Black, and A. Ahmad (2022). Deep residual reinforcement learning based autonomous blimp control. *arXiv preprint arXiv:2203.05360*.

Mageli, E. R. (2019). Reinforcement learning in pro-

cess control. Master's thesis, NTNU.

McClement, D. G., N. P. Lawrence, P. D. Loewen, M. G. Forbes, J. U. Backström, and R. B. Gopaluni (2021). A meta-reinforcement learning approach to process control. *IFAC-PapersOnLine 54*(3), 685–692.

Morinelly, J. E. and B. E. Ydstie (2016). Dual mpc with reinforcement learning. *IFAC-PapersOnLine 49*(7), 266–271.

Nian, R., J. Liu, and B. Huang (2020). A review on reinforcement learning: Introduction and applications in industrial process control. *Computers & Chemical Engineering 139*, 106886.

Noel, M. M. and B. J. Pandian (2014). Control of a nonlinear liquid level system using a new artificial neural network based reinforcement learning approach. *Applied Soft Computing 23*, 444–451.

Panzer, M. and B. Bender (2021). Deep reinforcement learning in production systems: a systematic literature review. *International Journal of Production Research*, 1–26.

Shin, J., T. A. Badgwell, K.-H. Liu, and J. H. Lee (2019). Reinforcement learning–overview of recent progress and implications for process control. *Computers & Chemical Engineering 127*, 282–294.

Silver, T., K. Allen, J. Tenenbaum, and L. Kaelbling (2018). Residual policy learning. *arXiv preprint arXiv:1812.06298*.

Spielberg, S., R. Gopaluni, and P. Loewen (2017). Deep reinforcement learning approaches for process control. In *2017 6th international symposium on advanced control of industrial processes (AdCONIP)*, pp. 201–206. IEEE.

Spielberg, S., A. Tulsyan, N. P. Lawrence, P. D. Loewen, and R. Bhushan Gopaluni (2019). Toward self-driving processes: A deep reinforcement learning approach to control. *AIChE journal 65*(10), e16689.

Sutton, R. S. and A. G. Barto (2018). *Reinforcement learning: An introduction*. MIT press.

Yu, J. and P. Guo (2020). Run-to-run control of chemical mechanical polishing process based on deep reinforcement learning. *IEEE Transactions on Semiconductor Manufacturing 33*(3), 454–465.

Zhang, S., W. Boehmer, and S. Whiteson (2019). Deep residual reinforcement learning. *arXiv preprint arXiv:1905.01072*.