Controlling the Production Load of a Vinyl Acetate Monomer Plant by Deep Reinforcement Learning

Taisei Hashimoto[1,2†], Takashi Onishi[1,3], Takuya Hiraoka[1,3] and Yoshimasa Tsuruoka[1,2]

[1] NEC-AIST AI Cooperative Research Laboratory, National Institute of Advanced Industrial Science and Technology, Tokyo, Japan

[2] Department of Information and Communication Engineering, The University of Tokyo, Tokyo, Japan

[3] Data Science Research Laboratories, NEC Corporation, Kanagawa, Japan

Abstract: We report an application of deep reinforcement learning (RL) to the task of controlling a complex large-scale chemical process. RL is a flexible machine learning framework that can be used for control, and deep RL incorporates deep learning into RL as a powerful tool for function approximation. As a case study, a benchmark simulator of a vinyl acetate monomer (VAM) plant is used to train RL agents to control the production load to target values. Considering practical application, 1) rain disturbance and 2) policy smoothing are introduced to learn a robust and stable policy. Experimental results demonstrate that RL is a promising approach for the control of large-scale plants under disturbances.

Keywords: Chemical Process Control, Deep Reinforcement Learning

1. INTRODUCTION

Chemical plants are complex and dynamic systems consisting of many components for manipulation and sensing. Therefore, the operation of chemical plants is not straightforward and requires skilled operators. However, the number of such operators is limited. In order to assist operators, we have developed a system to automatically control the plant. .

To obtain the control policy of our system, we adopt a deep reinforcement learning (RL) approach. Deep RL is a combination of deep learning and RL, and it has recently been demonstrated its effectiveness in various control tasks [1], [2]. In addition, deep RL does not require accurate environment models that are in general difficult to be hand-crafted in complex tasks. This is a strong advantage over control methods such as a model predictive control [3].

To validate the effectiveness of deep RL in a large commercial-level plant, an attempt has been made to learn the policy to control the production load of a vinyl acetate monomer (VAM) plant simulator. VAM plant simulators are widely used as a benchmark model for chemical process control, and their complexity and scale are considered comparable to large commercial plants [4], [5]. In this paper, a policy for the production load-down operation, which is a typical operation in VAM plants [4], is learned via deep RL.

In addition, towards practical application, we particularly consider guaranteeing the robustness and smoothness of the learned policy.

Robustness: the learned policy ought to be robust against disturbances because the real plants are subject to various kinds of disturbances, such as rain and changes in the temperature and the feed composition. In this paper, we focus on learning policies that are robust against rain disturbance.

Smoothness: the learned policy ought to be smooth enough to be feasible in the real world. A control policy which suddenly changes the control signal (e.g., the bang-bang-control policy) might work in simulation, but it can make the real plant unstable. Additionally, considering the situation where our agent is used to guide human operators to control the plant, it is hard for the operators to follow the guidance produced by an un-smoothed policy. We address these issues by regularizing the policy to be sufficiently smooth.

There have been several studies on the application of RL to VAM plant control. Kubosawa et al. [6] developed a system that can suggest appropriate operation procedures for handling a malfunction. Zhu et al. [7] learned a policy that achieved comparative performance to model-based control. Mori et al. [8] improved the gross profit over model-based control. In contrast to these studies, our aim is to learn a robust and smooth policy to control the VAM product load under disturbances.

2. PRELIMINARIES

2.1. Vinyl Acetate Monomer Plant

VAM is produced by the reaction of ethylene (C_2H_4), oxygen (O_2), and acetic acid (AcOH), while water (H_2O) is also generated as a by-product. The whole process flow consists of the following seven sections:

1. **Raw material feed.** C2H4 and O2 are fed in the gas phase, and AcOH is fed in the liquid phase and vaporized with superheated steam in a vaporizer.
2. **Reactor.** The three raw materials are mixed and fed to a reactor and then VAM and H_2O are generated and some unreacted AcOH remains.
3. **Separator and compressor.** The VAM, unreacted AcOH, and H_2O are condensed as liquid VAM crude at a separator.
4. **Absorber.** The separated and compressed gas is absorbed by cold AcOH, fed from the top of an absorber, and the mixture of VAM and AcOH is discharged from the bottom of the absorber.

† Taisei Hashimoto is the presenter of this paper.

5. **Buffer tank and distillation column feed.** The VAM crude coming from the separator is stored at an intermediate buffer tank and fed to the following distillation column.
6. **Distillation column and condenser.** The VAM crude is distilled in a distillation column, and VAM-H_2O mixture is discharged from the top of the column and then condensed at a condenser.
7. **Decanter.** The VAM-H_2O mixture is separated at a decanter. The VAM forms the organic phase and the H_2O constitutes the aqueous phase.

The chemical processes in the VAM production contain sub-processes that commonly appear in a variety of chemical production processes. Hence, its plant model is often implemented as a benchmark model in plant simulators [4], [5].

Visual Modeler [5] is used for our experiments. In Visual Modeler, the modeled VAM plant is equipped with 26 PID controllers and 109 sensors to control the aforementioned processes. The users and RL agents can manipulate the desired value of each PID controller. The simulator also implements several disturbance models including rain disturbance, which was used in our experiments. Under rain disturbance, heat dissipation at all heaters increases due to cooling by rain.

2.2. Reinforcement Learning

In RL, problems are modeled as Markov Decision Process (MDP), which is represented by the tuple: (S, A, R, T, ρ_0). S is a state space, A is an action space, $R : S \times A \times S \to (-\infty, \infty)$ is a reward function, $T : S \times A \times S \to [0, \infty)$ is a transition probability, and ρ_0 is an initial state distribution.

The objective of RL is to find the optimal policy $\pi^* : S \times A \to [0, \infty)$ which maximizes the discounted return:

$$\pi^* = \underset{\pi}{\operatorname{argmax}}\, \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t) \right],$$

where $\gamma \in [0, 1)$ is a discount factor.

Deep RL is a type of RL in which deep learning techniques are used to achieve strong function approximation.

2.3. Soft Actor-Critic

Soft actor-critic (SAC) [9], [10] is a state-of-the-art model-free off-policy Deep RL algorithm. SAC maximizes the policy entropy in addition to the discounted returns to improve exploration and robustness [9], [11].

2.4. Hindsight Experience Replay

In practice, it is difficult to learn a policy under a sparse reward setting. This is because the reward signal is not informative. Sparse rewards can be avoided by using a dense reward function. However, manually designing it is in general complicated and costly.

Hindsight Experience Replay (HER) [12] addresses this problem by relabeling the original goal with the achieved goal. Since the goal does not affect the environment dynamics, the overwritten data can be used to learn a policy. Intuitively, it augments the agent with the ability to learn from mistakes. HER is shown to be effective for goal-reaching tasks, where appropriate reward engineering is difficult.

Our VAM plant control task is a goal-reaching task, in which only sparse rewards are given, and thus HER is introduced for our policy learning.

2.5. Conditioning for Action Policy Smoothness

Deep RL typically does not consider action smoothness and a learned policy produces oscillatory control signals. However, this is especially problematic in real-world applications, because, as discussed in the introduction, it makes a controlling plant unstable and hard to be interpreted by human operators.

Conditioning for Action Policy Smoothness (CAPS) [13] regularizes agents to learn smoother control policies. In the study [13], CAPS successfully learned a smooth policy and reduced motor usage in a quadrotor drone control task.

In CAPS, the objective function for policy learning is augmented with an additional regularization term:

$$J_{\pi_\theta}^{\mathrm{CAPS}} = J_{\pi_\theta} - \lambda_T L_T$$
$$L_T = \mathbb{E}_{\pi_\theta} \left[||\mu_\theta(s_t) - \mu_\theta(s_{t+1})||^2 \right],$$

where π_θ is a policy with a parameter θ, J_{π_θ} is the original objective function of a base RL algorithm, $\lambda_T (> 0)$ is a hyperparameter, and $\mu_\theta(s)$ is the mean vector of the distribution $\pi_\theta(\cdot|s)$. L_T penalizes policies when actions taken at consecutive states are significantly different.

3. EXPERIMENTS

3.1. Task Specification

Our task is to control the production load of the VAM (FC560.PV). In this section, we describe our task specification to apply RL.

Reward:

$$r(p_c, p_t) = \begin{cases} 1 & (\frac{|p_c - p_t|}{p_s} < \varepsilon) \\ 0 & (\text{otherwise}), \end{cases}$$

where p_c is the current production load, p_t is the target production load, and p_s is the production load under steady states [1]. ε is set as $\varepsilon = 0.01$ for the experiments.

State: The state space consists of 109 sensor readings including temperature, flow, quality, pressure, and liquid level. **Action:** Using all of the PID controllers available in the simulator was not a reasonable choice because many of them are irrelevant to the task, and because large action spaces generally impaired learning stability. Hence, on the basis of a control guide for the VAM plant, PID controllers were chosen for interference by the RL agent. More specifically, two groups of PID controllers were used. When there was no disturbance, a pressure

[1] We also tried using dense rewards, but the resulting performance was worse than that of the sparse rewards case. This result demonstrates the difficulty of reward engineering [12].

controller of a steam drum (PC210.SVM) and a quality controller of oxygen feed (QC170.SVM) is used. When the rain disturbance was activated, a flow controller of steam to a reboiler (FC501.SVM) was added to the former controller group.

Weather change was modeled as Markov processes. For instance, if it was currently raining, there was a 96 percent chance of rain at the next time step. If it was not raining, there was a 4 percent chance of rain at the next time step. The intensity of rain, which is expressed as the value of a heat transfer coefficient in the VAM plant simulator, was sampled uniformly from 1 to 50 $\mathrm{W}/(\mathrm{m}^2\mathrm{K})$.

Every episode started from a unique steady state. The interval between the occurrence of each action was thirty minutes and each episode lasted for up to thirty hours. Note that this was measured by virtual time on the simulator, and not by wall-clock time. When any of the safety criteria defined in the simulator [4] was violated, the episode was immediately terminated.

The RL agent was based on SAC [10] with two major modifications. First, HER was applied to efficiently learn a policy under the sparse-reward setting. Second, CAPS was applied to learn a smooth policy. The Q network and the policy network had two hidden layers with 128 ReLU units.

3.2. Results

The case without rain disturbance: Fig. 1 shows the results over five trials of training with different random seeds. The agent achieved a high score in the early stage of the training, but it soon deteriorated significantly. This was probably due to the lack of diversity in the initial states. Namely, the agent was overfitted to a small amount of training data, which was collected at the early stage of training. This might be mitigated by using diverse initial states for training. Example trajectories of the learned policies are shown in Fig. 2. The VAM production load was successfully adjusted to the target value. Finally, Fig. 3 shows that the use of CAPS contributes to obtaining a smoother (stable and interpretable) policy compared to the one without CAPS.

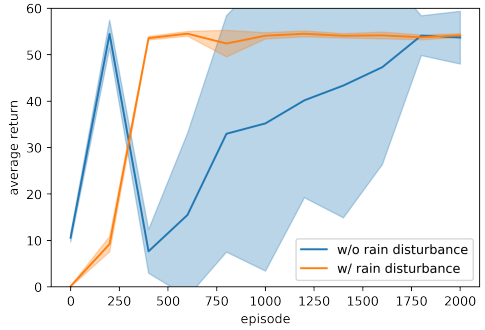

Fig. 1 Learning curves with and without rain disturbance. The solid lines correspond to the mean scores and the shaded areas to the 95% confidence intervals.

The case with rain disturbance: Fig. 1 shows the results over five trials of training with different random seeds. The agent stably achieved a high score despite the disturbance. This stable learning could be attributed to the diversity of the visited space caused by the disturbance. Some examples of the learned policies are shown in Fig. 4. When it began raining, the agent increases the steam flow in the reboiler and mitigated the drop of the production load. Interestingly, it was the same as the proper measures to rainfall that is described in the instruction manual of the plant simulator [4].

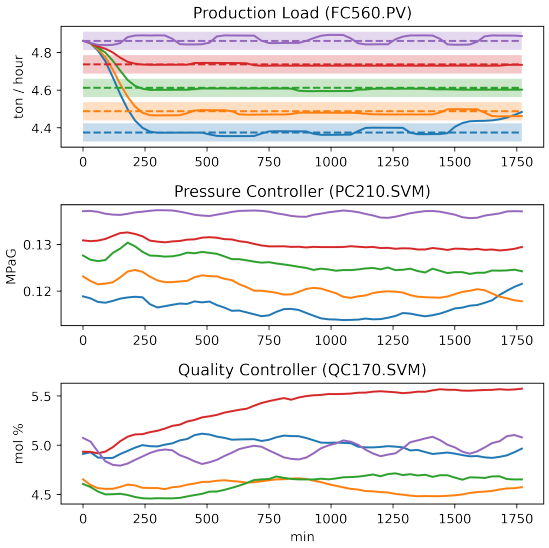

Fig. 2 Examples of a learned control policy. The results in the same trial are highlighted in the same color. The solid lines and dashed lines in the production figure correspond to the achieved production and the target production, respectively. The shaded areas show $\pm\varepsilon$ range used for reward calculation.

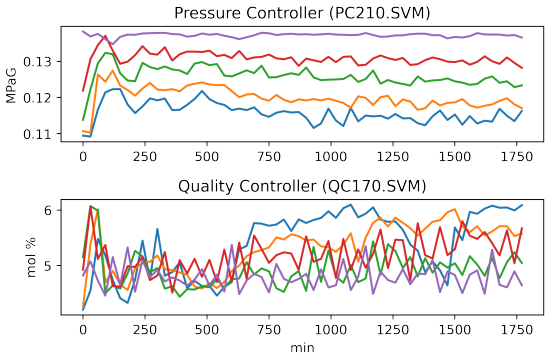

Fig. 3 Examples of a control policy learned without CAPS.

4. CONCLUSION

In this paper, deep RL was applied to the VAM production control task. The trained agent successfully adjusted the production to the target value. It also learned how to deal with rain disturbances, and produced stable and interpretable countermeasures.

In future work, more difficult tasks would be ad-

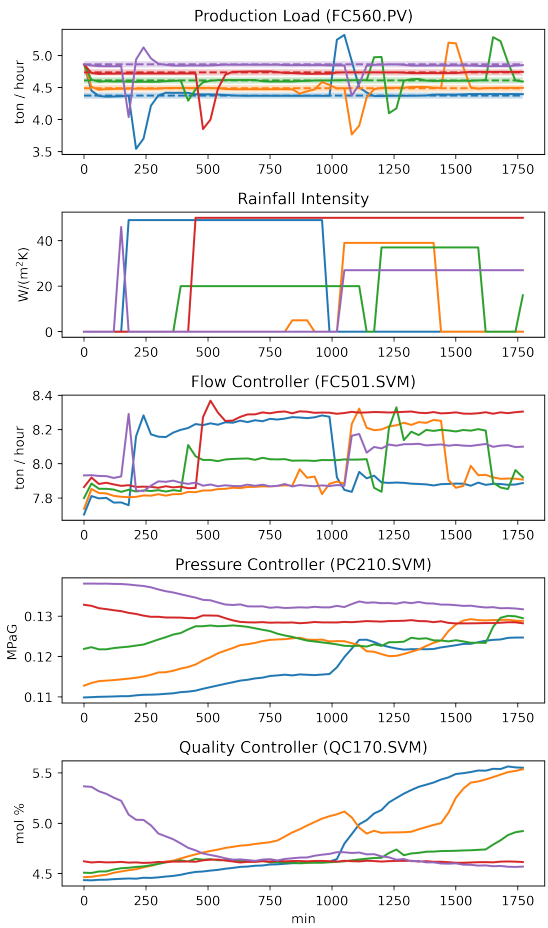

Fig. 4 Example trajectories of a learned control policy under rain disturbance.

dressed, such as a start-up operation. We also plan to train a more general policy dealing with various tasks instead of individually learning policies for each task.

It is also important to deal with various disturbances, such as a sudden change in pressure and a sensoring trouble. In this work, the focus is on learning a robust policy under the influence of disturbances that are experienced in training. However, the ability to perform safe operations even under unseen disturbances is required in practice. This could be much harder than the current condition, but it is important in practice.

REFERENCES

[1] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, "Human-level Control through Deep Reinforcement Learning," *Nature*, vol. 518, pp. 529–533, 2015.

[2] T. Haarnoja, S. Ha, A. Zhou, J. Tan, G. Tucker, and S. Levine, "Learning to Walk via Deep Reinforcement Learning," *Robotics: Science and Systems*, 2019.

[3] T. A. Badgwell, J. H. Lee, and K.-H. Liu, "Reinforcement Learning – Overview of Recent Progress and Implications for Process Control," in *proceedings of 13th International Symposium on Process Systems Engineering*, M. R. Eden, M. G. Ierapetritou, and G. P. Towler, Eds., ser. Computer Aided Chemical Engineering, vol. 44, 2018, pp. 71–85.

[4] Y. Machida, S. Ootakara, H. Seki, Y. Hashimoto, M. Kano, Y. Miyake, N. Anzai, M. Sawai, T. Katsuno, and T. Omata, "Vinyl Acetate Monomer (VAM) Plant Model: A New Benchmark Problem for Control and Operation Study," *Dynamics and Control of Process Systems, including Biosystems*, vol. 49, pp. 533–538, 2016.

[5] *Omega Simulation Co., Ltd. Website*, http://www.omegasim.co.jp/.

[6] S. Kubosawa, T. Onishi, and Y. Tsuruoka, "Synthesizing Chemical Plant Operation Procedures using Knowledge, Dynamic Simulation and Deep Reinforcement Learning," in *proceedings of the SICE Annual Conference*, 2018.

[7] L. Zhu, Y. Cui, G. Takami, H. Kanokogi, and T. Matsubara, "Scalable reinforcement learning for Plant-wide Control of Vinyl Acetate Monomer Process," *Control Engineering Practice*, vol. 97, p. 104 331, 2020.

[8] T. Mori, S. Kubosawa, T. Onishi, and Y. Tsuruoka, "Improving the Gross Profit of a Vinyl Acetate Monomer Plant by Deep Reinforcement Learning," in *proceedings of the SICE Annual Conference*, 2020.

[9] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor," in *proceedings of International Conference on Machine Learning*, 2018, pp. 1861–1870.

[10] T. Haarnoja, A. Zhou, K. Hartikainen, G. Tucker, S. Ha, J. Tan, V. Kumar, H. Zhu, A. Gupta, P. Abbeel, *et al.*, "Soft Actor-Critic Algorithms and Applications," *arXiv preprint arXiv:1812.05905*, 2018.

[11] B. Eysenbach and S. Levine, "Maximum Entropy RL (Provably) Solves Some Robust RL Problems," *arXiv preprint arXiv:2103.06257*, 2021.

[12] M. Andrychowicz, F. Wolski, A. Ray, J. Schneider, R. Fong, P. Welinder, B. McGrew, J. Tobin, O. Pieter Abbeel, and W. Zaremba, "Hindsight Experience Replay," in *proceedings of Advances in Neural Information Processing Systems*, vol. 30, Curran Associates, Inc., 2017.

[13] S. Mysore, B. Mabsout, R. Mancuso, and K. Saenko, "Regularizing Action Policies for Smooth Control with Reinforcement Learning," in *proceedings of IEEE International Confernece on Robotics and Automation*, 2021.