Nonlinear Model Predictive Control using Neural ODE Replicas of Dynamic Simulators

Shumpei Kubosawa[1], Takashi Onishi[2], and Yoshimasa Tsuruoka[3]

[1,2,3] *NEC-AIST AI Cooperative Research Laboratory,*
National Institute of Advanced Industrial Science and Technology (AIST), Tokyo, 135-0064 Japan
kubosawa@nec.com
[1,2]*Data Science Laboratories, NEC Corporation, Kanagawa, 216-8666, Japan*
[3]*Department of Information and Communication Engineering, The University of Tokyo, Tokyo, 113-0033, Japan*

Abstract

We propose simulation-based nonlinear model predictive control as a first step towards autonomous decision-making for stable operation of large complex dynamical systems such as chemical plants. The effect of abrupt external disturbances should be quickly eliminated, taking into account such complex dynamic responses, to maintain stable production. In this paper, we propose a control system to eliminate these effects. The system uses engineering models, including dynamic simulators, based on chemical engineering knowledge. Dynamic simulators are generally not differentiable with respect to actions; however, differentiable models are advantageous for fast nonlinear optimization. To take advantage of both reliable dynamic simulators and differentiable models, we introduce neural ordinary differentiable equation models and clone the behaviour of simulators on them. The cloned differentiable neural replica model is then incorporated into a gradient-based nonlinear model predictive control. Evaluation of this method in a real methanol distillation plant confirms that it can significantly remove abrupt heavy rain disturbances compared to existing methods.

1. Introduction

Many automatic controllers, such as proportional-integral-derivative (PID) controllers (Knospe, 2006) and model predictive controllers (MPC) (Qin & Badgwell, 2003), are used in complex process plants, including chemical plants, to ensure stable production.

PID control is limited to maintaining a single process variable (PV) to a single setpoint variable (SV) by adjusting a single manipulated variable (MV); however, they are still the majority of automatic controllers installed in these plants. Fig. 1 shows the process flow of a methanol (MeOH) distillation plant, a standard binary distillation process structure. The dotted lines represent PID loops; this plant has seven PID controllers to continuously separate MeOH and water (H2O) from their mixture liquid.

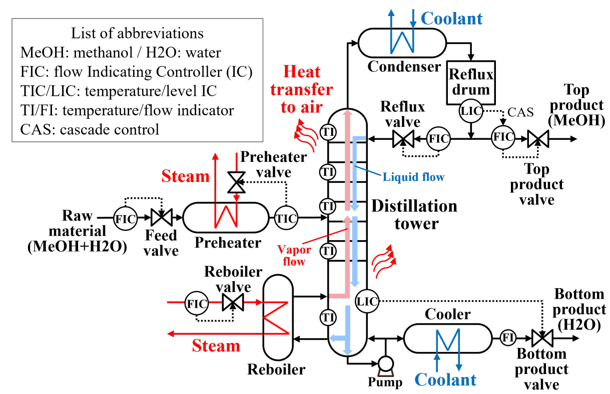

Figure 1. Process flow of a methanol distillation plant.

To maintain multiple PVs by manipulating multiple MVs, MPC, commonly known as a major method of advanced process controls (APC) (Bauer & Craig, 2008), is proposed. MPC repeatedly uses prediction models to predict the complex future responses of the plant, evaluate the sequence of actions that triggered the responses, and improve the actions.

Several chemical plants are equipped with detailed dynamic simulators for training human plant operators (Klatt & Marquardt, 2009). These simulators can accurately reproduce and predict the nonlinear behaviour of the real complex plant and therefore have the potential for use in MPC. For several nonlinear MPC methods, the partial derivatives of the actions are necessary to efficiently compute the optimal actions. However, dynamic simulators, especially the simulator products popular in many industrial applications, are generally not differentiable, making the integration of these existing simulators with MPC difficult. Dynamic simulator vendors have offered the MPC products in conjunction with their simulator products; however, Henson (1998) suggests that open literature on these MPC systems is unlikely to be found because the dynamic model equations are inaccessible to control system engineers other than the vendors.

In this paper, to exploit arbitrary dynamic simulators for MPC, we use the neural ordinary differential equation models (Chen et al., 2018) to construct a differentiable prediction model as a replica of the dynamic simulator and incorporate it with nonlinear MPC. The proposed method only uses the input/output of the dynamic simulators; therefore, it can be applied to dynamic simulators in general. We evaluated our method on the actual MeOH distillation plant and performed that our system significantly rejected abrupt heavy rain disturbance.

2. Related work

2.1. Dynamic simulation

Dynamic simulators can calculate changes in the internal state of a dynamical system over time based on state transition rules. For example, ordinary differential equations,

$$\frac{dx}{dt} = f(x, u), \tag{1}$$

where t, x, u, and f are time, state, action, and state transition function, respectively, are commonly used to model the rules. Then, temporal changes in the situations are calculated by solving the equation using the integration

$$x_t = \int_0^t f(x_\tau, u_\tau) d\tau, \tag{2}$$

where t is the current time step. Numerical integration algorithms, including Runge-Kutta methods, are commonly used to solve them. The models based on scientific and engineering knowledge can accurately reproduce and predict various situations due to environmental and operational changes. Additionally, by calculating various internal states, including the composition and enthalpy of materials flowing in various parts of the plant, the detailed states can be observed.

2.2. State observer

For the simulation calculation in Eq. (2), the initial state x_0 is required. The state x would include unobservable or difficult-to-observe states such as feed composition, heat transfer coefficients to air, other simulation parameters, and the actual environmental situations. A state observer is a standard method to estimate these simulation conditions and the internal states (Soroush, 1997; Kubosawa et al., 2022a). In state observers, the action input to the actual plant is also simultaneously input to the prediction model, and the simulation conditions are adjusted to minimize the residual between the actual and predicted observable response. When the residual is close to zero, the estimated simulation conditions and the predicted internal states would reflect the actual situation; thus, the predicted states including the simulation conditions can be used for the initial state of the simulation.

2.3. Optimal control

Optimal control is a set of problems to achieve the given desired situation by manipulating the target dynamical systems. The preferences of situations are expressed quantitatively as an evaluation function, and the problem is to find the sequence of actions that minimizes the evaluation function. Both reinforcement learning (RL) and MPC are solutions to these problems with different approaches (Görges, 2017).

2.3.1. Reinforcement learning

RL is a machine learning method and is mainly studied in the field of artificial intelligence (Bertsekas, 2019). RL constructs optimal controllers before applying them to actual control; thus, RL can be considered as an off-line optimization-based method. Using real dynamical systems or their simulators, RL tries variable actions in variable situations and collects input/output data and evaluation value (reward) of each situation. RL alternates between estimating the total future reward (value) of each situation using the collected data and improving its policy (controller). Since RL is a data-driven control method, it can be applied to arbitrary dynamical systems where only input/output is accessible. In addition, RL can consider situations far in the future, as long as data can be collected. These advantages led to the success of AI in Go games, which originated in AlphaGo. However, RL consumes large amounts of data, so training takes a long time. If the reward setting is changed, RL again requires long training.

In chemical plant operation, Kubosawa et al. (2022b) introduced an RL-based control framework that formulated state observer, operation planner, and disturbance rejection functionalities as RL tasks and implemented them as RL agents.

2.3.2. Model predictive control

MPCs are control theory methods studied in systems engineering, automatic control and industrial applications (Qin & Badgwell, 2003). An MPC controller calculates the optimal action on the spot in the plant; therefore, MPC can be considered as online optimization-based methods. In MPC, the candidate action sequence is used to predict a fixed period of future responses, evaluated using the prediction with a given cost function, and improved by minimizing the cost with some methods iteratively online. Unlike RL, the future situations considered by MPC are limited to a given and fixed finite period called the horizon. Several methods have been proposed, including linear, nonlinear and stochastic MPCs.

In the early days of MPC, linear models were commonly used to implement the algorithm on computers that were less powerful than modern ones, yet linear models are still a popular choice. To build these models from the data, engineers perform many step response tests using the working plant to collect data and fit the model. Additionally, the plant state or operating condition may change over time or be affected by changes in the external situation; therefore, if the plant state changes to a different state from the data collected situations, the model would hardly be available if the nonlinearity in the state transition is affected. In such cases, the engineer should perform step response tests and all modelling procedures again; however, these tests take a long time and could disturb production. These difficulties can be eliminated if the nonlinear and differentiable plant models for nonlinear MPC can be used without the tests.

3. METHOD

To use dynamic simulator for differentiable prediction model, we introduce neural ordinary differential equation model (Neural ODE or NODE) (Chen et al., 2018) to clone the behaviour of the simulator. We use the NODE model for the nonlinear MPC algorithm. The overall architecture is shown in Fig. 2. The NODE model is first trained on simulation data and then combined with MPC.

In chemical engineering and control engineering in general, ordinary differential equations, a type of continuous-time model, are commonly used to model plant behaviour. We focused on this point and used NODE models, which express the state transition function f of Eq. (1) by a neural network. For the numerical integration algorithm we used conventional RK4. Training data for the NODE model is collected from the dynamic simulator using given operational scenarios and simulation conditions. The NODE model is then trained to minimize the mean squared error between its prediction and the simulation data.

The objective cost function of the MPC is defined as

$$J = \int_{t_0}^{t_f} L(x_\tau, u_\tau) d\tau + \phi(x_{t_f}), \tag{3}$$

where t_0, t_f, L, and ϕ are the initial time, final time, stage cost function, and terminal cost function, respectively. For example, given the target desired state vector $\hat{x}$, the stage cost function can be the form of

$$L(x_\tau, u_\tau) = \sum_i w^{(i)} (x_\tau^{(i)} - \hat{x}_\tau^{(i)})^2, \tag{4}$$

where i and $w^{(i)} \in [0, 1]$ are the vector index denoting the i-th state variable in the state vector and the weight of the i-th state variable defining the severity of the residual in each

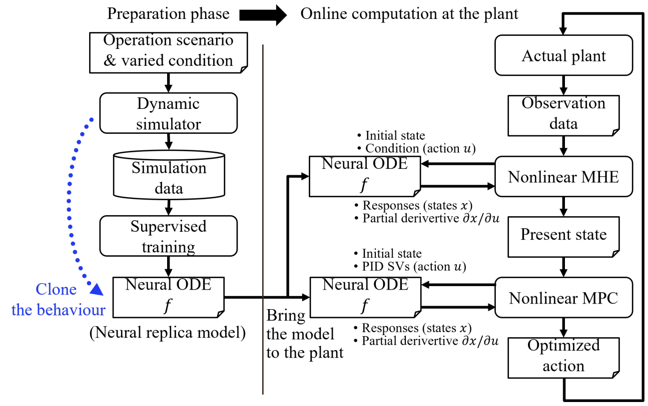

Figure 2. Overall architecture of the proposed method.

state variable, respectively. The terminal cost function ϕ can be defined in the same manner. The stage cost function defines the cost of each time step during the period of action sequence computation, and the terminal cost function defines the total future cost beyond the period. The algorithm finds the optimal sequence of actions $\{u_\tau\}_{\tau=t_0}^{t_f}$ that minimizes J.

For the nonlinear MPC algorithm, we used a gradient-based MPC (Käpernick & Graichen, 2014), which we implemented it independently. Additionally, to update the action input u, we omitted the line search procedure and used the Adam optimization algorithm (Kingma & Ba, 2014) using the partial derivative $\partial H/\partial u$, where

$$H(x, u, \lambda) = L(x, u) + \lambda^\top f(x, u) \tag{5}$$

is the Hamiltonian with co-states λ which have the same dimensionality as the states.

To estimate the initial state x_0 and the simulation conditions, as in state observers, MPC-like algorithms can be used. These methods are referred to as moving horizon estimation (MHE) (Morari & Lee, 1999; Johansen, 2011). In a simple case, the cost can be defined as the sum of the residuals between the actual and predicted observable states of the past finite period up to the present. The predicted last state including the simulation condition can be used as x_0 for MPC.

4. EXPERIMENTS

We conducted two experiments to compare the performance of the existing RL-based method (Kubosawa et al., 2022b) and the proposed method. The experiments are performed on the actual MeOH distillation plant (Fig. 3) and its dynamic simulator implemented with the commercial product. The quality of the top product (MeOH) and bottom product (H2O) is highly dependent on the temperature at each stage of the tower, so the quality of the product is maintained if the operators maintain the temperatures. The tower is heated by the reboiler at the bottom and cooled by the reflux poured from

Figure 3. The MeOH distillation plant used for the evaluation experiment.

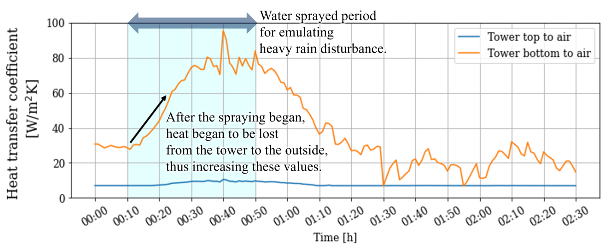

Figure 4. The estimated heat transfer coefficients to air in the experiment using the existing RL and simulation-based method and actual plant data.

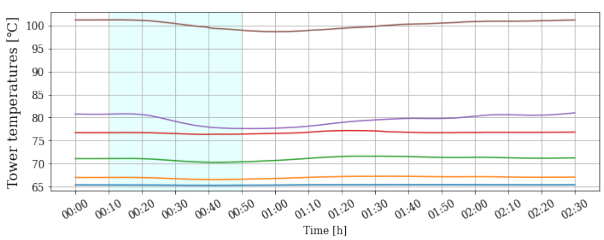

Figure 5. Simulated tower stage temperatures. The bottom stage has the highest temperature (brown) and the top stage has the lowest temperature (blue).

the top, so the reboiler and reflux are the main control points of the process. In the experiments, the manipulation points are the SVs of the two PID controllers (FIC) on the reboiler steam and the reflux flow. The control interval is set to 5 min. The experimental task is to reject abrupt heavy rain disturbances that cause a temperature drop during steady-state operation. The heavy rain disturbance in the real plant is artificially simulated by spraying water from the top of the tower for 40 min. Note that this task requires adjusting the two PID SVs to maintain the PVs of the six temperature sensors on the tower simultaneously, so conventional PID controllers designed for single-input, single-output tasks are unlikely to perform this task.

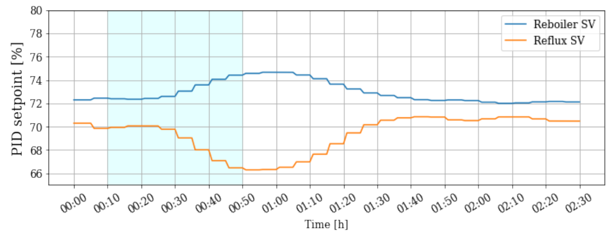

Figure 6. Proposed SVs of reboiler and reflux by the RL agent.

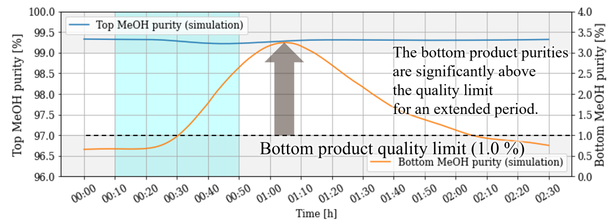

Figure 7. Simulated top and bottom product purity.

4.1. Existing RL-based method

The RL-based disturbance rejection method uses a dynamic simulator as the actual plant for training. The RL agent observes the actual observable states (simulated states for training) and the target states (reference trajectory) and outputs SVs for achieving the target states. The reward function for training is set to minimize the residual between the current actual state and the target state, and to maintain the top and bottom product quality (MeOH purity) on the simulator. The product qualities are not continuously measured in the real plant, so they are unobservable to the agent and are only used for the reward function during training.

Due to the limited opportunities for experiments on the actual plant, we used the simulator for this experiment. On the simulator, the heat transfer coefficients to air at the top and bottom of the tower are abruptly increased to emulate the heavy rain disturbance that occurred in the other experiment with the real plant. The coefficients are unobservable to the agent, which is the same situation as in the real plant case.

In this experiment, the time evolution of the coefficients used was estimated by a state observer consisting of a dynamic simulator and a reinforcement learning agent (Kubosawa et al., 2022a) pre-trained with the simulator, based on the actual data collected from the experiment described on § 4.2. These values are shown in Fig. 4. In the real plant, water is sprayed to emulate a heavy rain disturbance during the blue-shaded period in the figure and others. Shortly after the start of the spraying the lower coefficient value starts to increase and after the spraying stops the value starts to decrease.

The change in tower temperature is shown in Fig. 5. After starting the spray, the temperature of the middle stage, shown as the purple line, starts to decrease, then that of the bottom stage. To maintain the temperature of each stage, the RL agent proposed SVs as shown in Fig. 6. The top and bottom product purity is shown in Fig. 7 and the grey shaded area represents acceptable quality. After starting the spray, the agent increased the reboiler SV and decreased the reflux SV for heating. The agent preferred decreasing reflux to increasing reboiler. The agent eventually recovered the temperatures and maintained the upper product cleanliness; however, the lower product cleanliness was outside the range for a long time due to the slow response to the temperature decrease.

4.2. Proposed MPC-based method

In the experiment with the proposed MPC method, we used the actual plant. The disturbance setting is mentioned in the previous sections. We set the horizon to 10 min for the MHE and 60 min for the MPC. Both procedures are run every minute and the proposed SVs are applied to the plant every 5 minutes. Fig. 8 shows the tower temperature. Compared to the RL case on Fig. 5, the bottom temperature (blown line) maintained a slightly higher value. The proposed SVs by the proposed method are shown on Fig. 9. The proposed method preferred to increase the reboiler than to decrease the reflux, in contrast to the RL agent. However, this difference had a significant effect on the bottom product purity. Fig. 10 shows the actual sampled and measured MeOH purity of the top and bottom products. The bottom purity was immediately higher than the acceptable range; however, the violated amount is small.

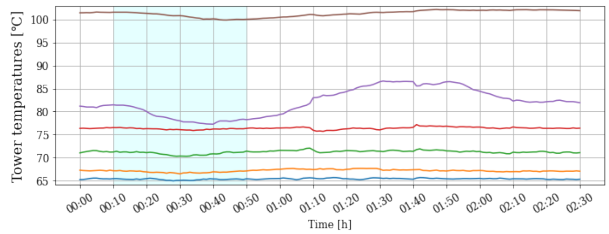

Figure 8. Actual tower stage temperatures.

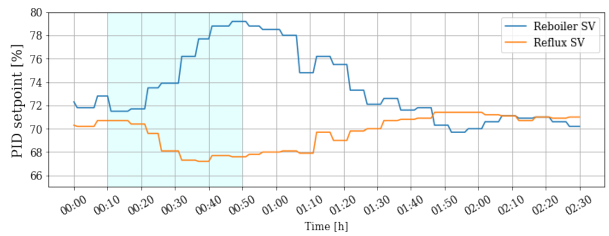

Figure 9. Proposed SVs by the proposed MPC.

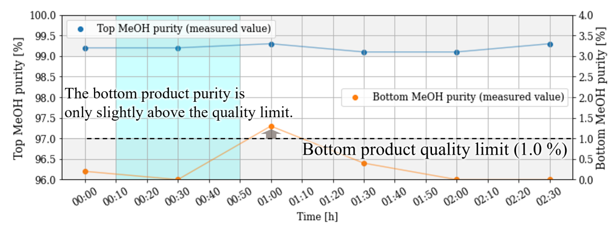

Figure 10. Actually measured top and bottom product purity.

5. Discussion

In the evaluation experiments, both methods eventually recovered the initial temperatures; however, the RL agent responded more slowly than the proposed method and consequently the quality of the bottom product deteriorated significantly. In response to unobservable and unpredictable disturbances, MHE and MPC would be advantageous.

RL can certainly consider situations far in the future, but if the disturbance patterns or plant states differ from the training situations, these considerations, i.e. the optimization (training) results, are of little help, because RL are methods based on training in advance. This is referred to as the "simulation-to-reality (Sim2Real) gap" problem. The existing RL-based methods (Kubosawa et al., 2022b) addressed the issue to some extent, but the fast response to these disturbances still needs to be improved. In such an improvement, adjustments in the reward function are a common approach; however, time-consuming training is required to obtain the changed behaviour of the agent due to the reward change, whereas in MPC the behavioural changes can be obtained instantly, which is a practical advantage for engineering. Online optimization methods with accurate models, such as the proposed neural replica models, would be promising for rapid adaptation to real situations. As NODE is a continuous-time and continuous-state model, discontinuous state transitions involving mode changes are considered inappropriate for modelling; however, in practice its expressiveness would vary depending on the size of the neural networks, so its practical performance in such situations should be evaluated.

As a related recent method to address Sim2Real problems, Jiahao et al. (2023) introduced online training of the NODE dynamics model for MPC. To train the model efficiently, they also integrated NODE with knowledge of the target dynamics. To reduce the Sim2Real gap, they adaptively update the dynamics model with observed data during control of the target actual system.

6. Conclusion

To reject the effect of abrupt disturbance and maintain stable production in chemical plants, we proposed an MPC-based method using dynamic simulators. The proposed method and its related RL-based method are experimentally evaluated, and it is demonstrated that the proposed method significantly rejects the effects of the disturbance in the actual plant. We would improve and implement the method in chemical plants and other fields of control applications.

ACKNOWLEDGMENT

The authors are grateful to Mitsui Chemicals, Inc. and in particular its Plant Operation Technology Training Center for providing the opportunity to conduct experiments on the actual distillation plant and to hear the perspectives of expert operators, and to Omega Simulation Co., Ltd. for providing the license for OmegaLand, the dynamic simulator product.

REFERENCES

Bauer, M., & Craig, I. K. (2008). Economic assessment of advanced process control – a survey and framework. *Journal of Process Control*, *18*(1), 2-18. doi: https://doi.org/10.1016/j.jprocont.2007.05.007

Bertsekas, D. (2019). *Reinforcement learning and optimal control*. Athena Scientific.

Chen, R. T., Rubanova, Y., Bettencourt, J., & Duvenaud, D. K. (2018). Neural ordinary differential equations. *Advances in neural information processing systems*, *31*.

Görges, D. (2017). Relations between model predictive control and reinforcement learning. *IFAC-PapersOnLine*, *50*(1), 4920-4928. (20th IFAC World Congress) doi: https://doi.org/10.1016/j.ifacol.2017.08.747

Henson, M. A. (1998). Nonlinear model predictive control: current status and future directions. *Computers & Chemical Engineering*, *23*(2), 187–202. doi: https://doi.org/10.1016/S0098-1354(98)00260-9

Jiahao, T. Z., Chee, K. Y., & Hsieh, M. A. (2023). Online dynamics learning for predictive control with an application to aerial robots. In K. Liu, D. Kulic, & J. Ichnowski (Eds.), *Proceedings of the 6th conference on robot learning* (Vol. 205, pp. 2251–2261). PMLR.

Johansen, T. A. (2011). Selected topics on constrained and nonlinear control textbook. In (chap. Introduction to Nonlinear Model Predictive Control and Moving Horizon Estimation). STU Bratislava – NTNU Trondheim.

Käpernick, B., & Graichen, K. (2014). The gradient based nonlinear model predictive control software GRAMPC. In *2014 European Control Conference (ECC)* (pp. 1170–1175).

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Klatt, K.-U., & Marquardt, W. (2009). Perspectives for process systems engineering – personal views from academia and industry. *Computers & Chemical Engineering*, *33*(3), 536-550. (Selected Papers from the 17th European Symposium on Computer Aided Process Engineering held in Bucharest, Romania, May 2007) doi: https://doi.org/10.1016/j.compchemeng.2008.09.002

Knospe, C. (2006). PID control. *IEEE Control Systems Magazine*, *26*(1), 30–31.

Kubosawa, S., Onishi, T., & Tsuruoka, Y. (2022a). AI and simulation for soft sensors and process control. *KAGAKU KOGAKU RONBUNSHU*, *48*(4), 141–151 (in Japanese). (The English-translated version is available on arXiv. `https://arxiv.org/abs/2208.04373`) doi: https://doi.org/10.1252/kakoronbunshu.48.141

Kubosawa, S., Onishi, T., & Tsuruoka, Y. (2022b). Sim-to-real transfer in reinforcement learning-based, non-steady-state control for chemical plants. *SICE Journal of Control, Measurement, and System Integration*, *15*(1), 10-23. doi: https://doi.org/10.1080/18824889.2022.2029033

Morari, M., & Lee, J. H. (1999). Model predictive control: past, present and future. *Computers & Chemical Engineering*, *23*(4-5), 667–682.

Qin, S. J., & Badgwell, T. A. (2003). A survey of industrial model predictive control technology. *Control engineering practice*, *11*(7), 733–764.

Soroush, M. (1997). Nonlinear state-observer design with application to reactors. *Chemical Engineering Science*, *52*(3), 387-404. doi: https://doi.org/10.1016/S0009-2509(96)00391-0

BIOGRAPHIES

Shumpei Kubosawa received his BA degree in Social Sciences from Hitotsubashi University in 2011, and ME from Tokyo Institute of Technology in 2013. Currently, he is a Principal Researcher at Data Science Laboratories in NEC Corporation, and a Specialist for Specific Subject (Research) at NEC-AIST AI Cooperative Research Laboratory, AIST. His current research interests include artificial intelligence, natural language processing, and reinforcement learning. He is a member of SICE, SCEJ, and ORSJ.

Takashi Onishi received the B.E. and M.E. degrees from the University of Tokyo in 2004 and 2006, respectively. Currently, he is a Research Manager at Data Science Laboratories in NEC Corporation, and a Project Manager at NEC-AIST AI Cooperative Research Laboratory in AIST. His research interests include applications of reinforcement learning.

Yoshimasa Tsuruoka received his Doctor of Engineering degree from the University of Tokyo in 2002. He worked as a research associate at the University of Tokyo and at the University of Manchester. He was an associate professor at the Japan Advanced Institute of Science and Technology from 2009 to 2011. Since 2011, he has been working at the University of Tokyo. He is now a professor at the university. His research interests include natural language processing, artificial intelligence for games, and reinforcement learning.