Towards Autonomous Reinforcement Learning for Self-Adaptive Gas Turbine Control

Alexander Hans[1,2] ALEXANDER.HANS.EXT@SIEMENS.COM
Steffen Udluft[1] STEFFEN.UDLUFT@SIEMENS.COM

1- Siemens AG, Corporate Technology, Learning Systems, Otto-Hahn-Ring 6, D-81739 Munich, Germany

2- Ilmenau University of Technology, Neuroinformatics and Cognitive Robotics Lab, P.O.Box 100565, D-98684 Ilmenau, Germany

The Learning Systems department of Siemens CT has been working on the application of reinforcement learning (RL) for control of complex technical systems for several years, especially in the context of gas turbines. One of the long-term goals is to enhance gas turbine control by deploying RL-based methods in series. Our poster will present our general approach, RL methods developed in the past and used in that context, and open problems that are subject of current and future research.

When trying to apply off-the-shelf RL methods for a problem like gas turbine control, one faces several problems. The state space is typically continuous and high-dimensional, moreover it often is non-Markovian. Normally, the action space is continuous as well and spanned by several dimensions, but here a discretization and use of a set of discrete actions usually works well. Many standard RL algorithms require a large amount of interactions with the system to obtain a good policy, which causes another difficulty, as interactions with the system and even just observations are extremely expensive. It is therefore important to be able to deal with RL problems with possibly non-Markovian, high-dimensional state spaces in a data-efficient manner.

In recent years, methods have been developed that are able to deal with high-dimensional state spaces data-efficiently. We mention neural rewards regression (NRR) (Schneegass et al., 2007), which can be regarded as a generalization of neural fitted Q iteration (Riedmiller, 2005), and the recurrent control neural network (RCNN) (Schaefer et al., 2007). NRR transforms the optimization problem stated by the Bellman optimality equation into a regression problem by utilizing a neural network with support for shared weights. The network consists of two parts, both representing the Q-function and sharing the same weights. Typically, only one part of the network is allowed to learn (left part), while the gradient flow to the other part is blocked (right part); the state and action are input to the left sub-network, the successor state is fed into the right network. The output node has the reward as target and sums the output of the left network with the output of the right network weighted by $-\gamma$. The network is trained offline. Once the training is finished, the left network can be extracted and represents the Q-function. The RCNN also consists of two parts. The first part is a recurrent neural network with past and future lags. It is used to approximate the system dynamics. The second part is a control network that builds on the first part. It represents and learns the policy, its target is maximizing the sum of the discounted future rewards. The system identifying characteristic of the RCNN can also be used to construct a compact approximate Markovian state representation from the high-dimensional input state space (Schäfer et al., 2007). Based on this state estimation it is also possible to use other (standard) RL methods. NRR and RCNN have both successfully been used to identify near-optimal policies for a RNN-based gas turbine simulation. Since the simulation is based on actual observations of a real turbine, it was possible to transfer new favorable working points found by RL applied to the simulation to the real turbine.

When moving from a simulation to a real turbine, additional problems have to be considered. It is not advisable to use RL to learn from scratch to control a gas turbine. Instead, an already existing default controller should be enhanced by RL methods. In that context the problem of safe exploration is important, for which first ideas have been developed using a simplified simulation that represents a system that might become unstable when moved beyond a certain point (Hans et al., 2008). Subject of safe exploration is exploring the system without reaching that "point of no return". The proposed approach requires access to an already existing and safely acting controller, referred to as *default policy*. The behavior of that default policy is observed and gradually exploratory actions are taken while learning a *safety function*. That safety function is queried to decide whether a state-action pair is safe to explore. Meanwhile, the approach has also been successfully applied to a RNN-based simulation.

Another important aspect of a real-world application is the consideration of uncertainty. Only when dealing with completely deterministic systems and noise-free observations, one observation suffices to fully describe a transition with no uncertainty left. In all other cases the estimators are affected by uncertainty. Ignoring the uncertainty might lead to false conclusions. An application of uncertainty propagation to the Bellman iteration was proposed in (Schneegaß et al., 2008), which leads to a Q-function together with its uncertainty. The knowledge of uncertainty is then used to obtain so-called *certain-optimal* policies, which are considerably more robust. However, propagating the uncertainty in the proposed way adds a high computational burden. Recent research has tried to weaken that problem by approaching an approximation instead of exact uncertainty propagation. Although policies generated by the approximate algorithm do not perform as well as those produced by the original, exact version, the fraction of policies performing extremely badly can be significantly lowered compared to the uncertainty-ignorant standard approach. Future research will consider using the knowledge of uncertainty to guide exploration.

For truly autonomous RL in the context of gas turbine control more issues have to be considered. We expect a RL-based gas turbine controller to calculate new policies offline and switch to a new policy once a better one has been identified. It is therefore necessary to evaluate a policy without actually executing it to make sure that the policy performs well. An obvious solution seems the use of a simulation, but in that case one has to ensure that the simulation represents the real system sufficiently well. Other issues concern the learning process. Many existing algorithms have the potential to deliver near-optimal policies, but need lots of manual tuning. Parameters are usually not easily transferable from one problem to another, so each problem needs its own tuning. For autonomous RL it is desirable to lessen then need for manual tuning, which will thus be subject of future work.

References

Hans, A., Schneega, D., Schaefer, A. M., & Udluft, S. (2008). Safe Exploration for Reinforcement Learning. *Proceedings of the European Symposium on Artificial Neural Networks* (pp. 413–418).

Riedmiller, M. (2005). Neural Fitted Q Iteration - First Experiences with a Data Efficient Neural Reinforcement Learning Method. *Proc. 16th European Conference on Machine Learning (ECML)* (pp. 317–328).

Schaefer, A. M., Udluft, S., & Zimmermann, H. G. (2007). A recurrent control neural network for data efficient reinforcement learning. *Proceedings of the IEEE International Symposium on Approximate Dynamic Programming and Reinforcement Learning (ADPRL-2007).* Honolulu, HI.

Schäfer, A. M., Schneegaß, D., Sterzing, V., & Udluft, S. (2007). A neural reinforcement learning approach to gas turbine control. *Proc. of the International Joint Conference on Neural Networks (IJCNN).*

Schneegass, D., Udluft, S., & Martinetz, T. (2007). Neural rewards regression for near-optimal policy identification in markovian and partial observable environments. *Proc. European Symposium on Artificial Neural Networks (ESANN)* (pp. 301 – 306).

Schneegaß, D., Udluft, S., & Martinetz, T. (2008). Uncertainty propagation for quality assurance in reinforcement learning. *Proc. of the Int. Joint Conf. on Neural Networks* (pp. 2589–2596).