A Combinatorial Learning Model Using Correlation-Based and Reward-Based Learning for Efficient Control Policy Development: Acquisition of Goal-Directed Behavior

Poramate Manoonpong[1,2], Florentin Wörgötter[1], Jun Morimoto[2]
[1]Bernstein Center for Computational Neuroscience, Georg-August-Universität Göttingen, III. Physics Institute – Biophysics, Friedrich-Hund Platz 1, 37077 Göttingen, Germany
[2]ATR Computational Neuroscience Laboratories, 2-2-2 Hikaridai Seika-cho, Soraku-gun, Kyoto 619-0288, Japan

Correlation-based learning and reward-based learning have been widely applied to artificial agents (robots) for solving various tasks including the generation of self-organizing behaviors. In general such learning mechanisms are *separated* employed. As a consequence, agents sometimes fail to solve difficult tasks requiring evaluative feedback (e.g., goal-directed behavior in a complex environment) when using only correlation-based learning which is fast but cannot easily learn control policies. On the other hand, they might slowly learn to find a solution for the tasks when using only reward-based learning which can obtain a good control policy based on its prediction mechanism including its own experiences and partly exploration. From this point of view, we introduce here a new learning framework that in parallel combines correlation-based learning using input correlation learning (ICO learning, Porr & Wörgötter, 2006) and reward-based learning using continuous actor-critic reinforcement learning (RL) (Doya, 2000). The parallel combination of these learning mechanisms is performed by allowing them to simultaneously learn solving a problem as a multiple learner system (Fig. 1a). In other words, they simultaneously adapt control parameters (i.e., synaptic weights) leading to a good control policy. Interestingly, due to the fast learning property (Fig. 1b) of the ICO rule considering only a correlation between a state (i.e., predictive information) and an unwanted condition (i.e., built-in reflex action), ICO learning indirectly guides the learning strategy of continuous actor-critic RL resulting in speeding up the whole learning process and finally acquiring a good control policy.

To evaluate the performance of the proposed learning paradigm, we use goal-directed behavior as a concrete example. The task is to let a simulated mobile robot learn to turn towards and approach a given goal mainly controlled by continuous actor-critic RL while at the same time it needs to avoid obstacles (i.e., reflex avoidance) mainly controlled by ICO learning. As a result, the robot can effectively learn to solve the task at different starting positions (Fig. 1c) compared to the one using a single learning mechanism (i.e., continuous actor-critic RL). The study pursued here sharpens our understanding of how correlation-based learning (e.g., ICO learning) and RL (e.g., continuous actor-critic RL) can be appropriately combined for solving a complex task like goal-directed behavior. **Acknowledgements:** This research was supported by the JAPAN TRUST Program, the Emmy Noether Program (DFG, MA4464/3-1), and SRBPS, MEXT. We thank Dr. Frank Hesse for technical advice on simulator implementation.

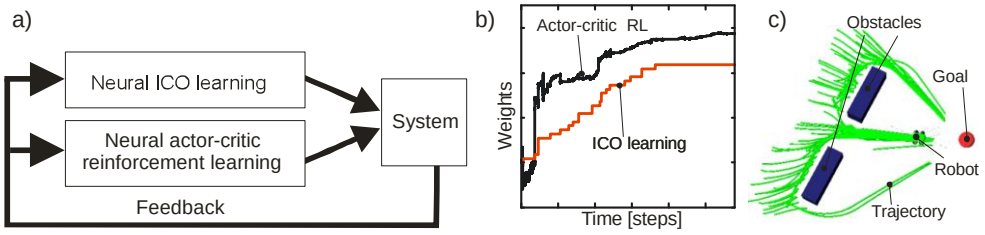

Figure 1: a) Parallel combination model of ICO learning and continuous actor-critic RL. They are implemented as neural control. b) Example of learning process of ICO learning and continuous actor-critic RL presented by weight changes (i.e., control parameters). ICO learning quickly develops weights controlling obstacle avoidance behavior and earlier converges such that it guides actor-critic RL to develop other weights controlling goal-directed behavior. Both learning mechanisms have the same learning rate of 0.01. Here, we show only two developed weights for clarity where the neural controller of this setup has eight weights in total. c) Learned goal-directed behavior. The robot has infrared sensors for detecting obstacles and position and orientation sensors providing its relative position and orientation to the goal location. We simulate the robot and its environment using the LPZROBOTS simulation software.