Advances in Complex Systems
Vol. 16, Nos. 2 & 3 (2013) 1350015 (38 pages)

DOI: 10.1142/S021952591350015X

World Scientific
www.worldscientific.com

COMBINING CORRELATION-BASED AND REWARD-BASED LEARNING IN NEURAL CONTROL FOR POLICY IMPROVEMENT

PORAMATE MANOONPONG[*,†,‡], CHRISTOPH KOLODZIEJSKI[*,§], FLORENTIN WÖRGÖTTER[*,¶] and JUN MORIMOTO[*,†,‖]

[*]*Bernstein Center for Computational Neuroscience, The Third Institute of Physics, University of Göttingen, Göttingen 37077, Germany*

[†]*ATR Computational Neuroscience Laboratories, 2-2-2 Hikaridai Seika-cho, Soraku-gun, Kyoto 619-0288, Japan*

[‡]*poramate@physik3.gwdg.de*
[§]*kolo@physik3.gwdg.de*
[¶]*worgott@physik3.gwdg.de*
[‖]*xmorimo@atr.jp*

Received 21 February 2012
Revised 6 February 2013
Accepted 20 February 2013
Published 24 April 2013

Classical conditioning (conventionally modeled as correlation-based learning) and operant conditioning (conventionally modeled as reinforcement learning or reward-based learning) have been found in biological systems. Evidence shows that these two mechanisms strongly involve learning about associations. Based on these biological findings, we propose a new learning model to achieve successful control policies for artificial systems. This model combines correlation-based learning using input correlation learning (ICO learning) and reward-based learning using continuous actor–critic reinforcement learning (RL), thereby working as a dual learner system. The model performance is evaluated by simulations of a cart-pole system as a dynamic motion control problem and a mobile robot system as a goal-directed behavior control problem. Results show that the model can strongly improve pole balancing control policy, i.e., it allows the controller to learn stabilizing the pole in the largest domain of initial conditions compared to the results obtained when using a single learning mechanism. This model can also find a successful control policy for goal-directed behavior, i.e., the robot can effectively learn to approach a given goal compared to its individual components. Thus, the study pursued here sharpens our understanding of how two different learning mechanisms can be combined and complement each other for solving complex tasks.

Keywords: Classical conditioning; operant conditioning; associative learning; reinforcement learning; pole balancing; goal-directed behavior.

1. Introduction

In biological systems, two classes of conditioning for associative learning are known [5]. One is classical conditioning [50] involving presentations of a conditional stimulus (CS) along with a significant or unconditional stimulus (US). The US generally drives an unconditional response (UCR), usually a reflex (e.g., salivation in dogs when they encounter food). Once the US and CS become associated, animals begin to perform a behavioral response to the CS rather than the US where this response is called a conditional response (CR). This modification basically happens only if the CS is a predictor for the US [56]. Thus, the CS normally precedes the US ([50, 70], but see Ref. [5] for detailed clarification). Another type of conditioning is operant or instrumental conditioning [59, 63]. It mainly involves a reinforcer (i.e., a US) associated with behavior modification instead of another stimulus. The probability of a specific behavior is increased or decreased through positive or negative reinforcement at each time that the reinforcement is generated.

Although these conditioning or learning mechanisms are different from each other, a number of studies on animal leaning suggest that they may act in combination [5, 15, 35, 47, 55], rather than separately or alternatively, to obtain an appropriate behavior. Experiments that have supported this idea were presented in, e.g., Refs. [11, 39, 68]. Williams and Williams [68] observed a pigeon pecking at an illuminated key in a Skinner box. The results suggest that the desired key-pecking behavior CR may be shaped by not only operant conditioning[a] but also by classical conditioning; since imposing an omission schedule on the key-light, key-peck association did little to revoke the conditional pecking response. Hence, it seems that the existing occasional pairing of the key-light CS with the food US was adequate to drive the pecking behavior (CR), which thus emerged from classical conditioning. Lovibond [39] performed experiments in rabbits by providing separately trained conditional stimuli during reinforced operant responding. His results showed that the strength of an operant response can be influenced by simultaneously presenting a classically CS. Brembs and Heisenberg [11] conducted experiments in the fruit flies (Drosophila). Their results showed that there is a situation where both operant and classical predictors play their roles at the same time, such that the situation can be more easily learned than in the separate case.

In animal training, evidence also reveals that many animals including rodents, dogs, pigeons, dolphins, seals, and whales, can effectively learn to do some sophisticated tasks when they are trained using a combination of these mechanisms [25]. For instance, marine animal trainers use a whistle as predictive information to "tell" their animals that a reward (e.g., food) is forthcoming. Thus, marine animals learn to associate the sound of whistle and food (i.e., learning via classical conditioning). When the animals perform a desired behavior (e.g., come, jump, flip, etc.), they

[a]In this situation, the animal was induced to respond to the key in association with a reward (i.e., food). This procedure is also called autoshaping.

first hear the sound indicating that they have performed appropriately and then they receive food (i.e., learning via operant conditioning). After several repetitions, the animals will perform a certain behavior as soon as they hear the sound where they expect to receive food afterwards.

Classical conditioning is often modeled as a form of correlation-based (differential Hebbian) learning [32, 37, 52, 70] in computational neuroscience. This approach uses the correlations between external stimuli (i.e., the US and CS) for synaptic plasticity leading to an anticipatory action (see Sec. 2 for more details). Operant conditioning is often modeled as reward-based learning or reinforcement learning (RL, e.g., temporal difference (TD)-learning [7, 62, 67]) in computer science. This approach uses predefined rewards and punishments in the environment as evaluation allowing an agent to maximize or optimize its own expected cumulative future reward (or expected return). As a consequence, this leads to a corresponding behavior (see Sec. 3 for more details).

These two conditioning concepts or learning mechanisms have been widely applied to artificial agents (robots) for solving various tasks including the generation of self-organizing behavior and autonomous systems [8]. Generally, much research has *separately* used them to enable agents to learn solving their tasks [7, 10, 20, 38, 40, 42, 51, 53, 62]. In this study, we point out that these two learning frameworks can complement each other leading to policy improvement. Correlation-based learning can quickly find a correlation between a state and an unwanted condition (i.e., reflex or failure recognized by an immediate reflex signal), but cannot evaluate whether a given state or weight change predicts something "good" or "bad" which will happen many steps away in the future. Consequently, it cannot properly learn solving some difficult tasks (e.g., delayed reward tasks) and cannot explicitly derive a goal-directed policy. On the other hand, reward-based learning can derive a policy according to the (delayed) reward signal but using it without any prior knowledge (predefined control parameters), environment or system models, or appropriate guidance generally takes many learning trials to improve the control performance. Therefore, we combine correlation-based learning (using input correlation learning (ICO learning) [52]) and RL (using continuous actor–critic RL [19]) in parallel to let ICO learning extract important features directly used to guide the learning strategy of continuous actor–critic RL. If we can extract important or proper features for the task, a model of the policy can be simple and the policy can be easily improved.

To investigate this hypothesis, to show how these two biologically-inspired learning mechanisms can be combined as a neural learning system, and to present its performance, we chose pole balancing and goal-directed behavior control problems as two different case studies or testbeds. Generally, we are not interested in solving these two tasks *per se*. Instead, we would like to show that the proposed combination can solve model-free control problems and is not limited to a specific task. Additionally, we would like to suggest that this combination can be an advantageous but simple way (i.e., only combining them in parallel without modifying their

learning mechanisms) to solve (dynamic) sensorimotor control problems with continuous signals. Through the performed experiments we hope that this model may help to better understand interactions between the two learning mechanisms. To a certain extent, the model might be related to neural learning in biological systems and it may provide a computationally oriented perspective on animal learning. Finally, we would like to emphasize that this combinatorial learning framework suggests how a prior knowledge can be provided to RL and how RL can be guided and shaped for policy improvement. To our knowledge, this kind of combinatorial learning method (which is simple, partially related to neural learning mechanisms in the brain — see the Discussion and Conclusion section below — and leads to policy improve) has not been investigated and presented so far.

This paper is organized as follows. In Sec. 2 we present the neural circuit of ICO learning while the neural circuit of continuous actor–critic RL is given in Sec. 3. In Sec. 4, we introduce our learning model which combines both learning mechanisms inspired by biological findings. This model will lead to policy improvement. In Sec. 5, we demonstrate its performance using the pole balancing and goal-directed behavior control problems and provide a comparison of different learning mechanisms. This paper finishes in Sec. 6 with discussion and conclusions.

2. Correlation-Based Learning

For correlation-based learning, we used ICO rule [52] (see Fig. 1) since this learning rule allows implementation of fast and stable learning and it has been also successfully applied to real robots for obtaining adaptive behavior [40, 42, 53].

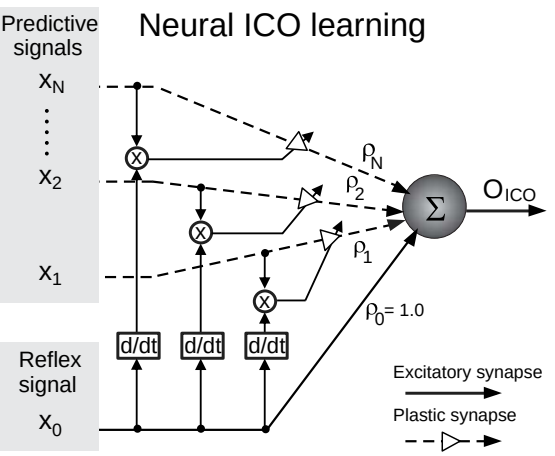


Fig. 1. Neural circuit of ICO learning for the multiple plastic synapses of predictive inputs. The learning rule is derived from differential Hebbian learning. Here, the output neuron (or learner neuron) was modeled as a simple linear neuron (see text for details). It generates a continuous signal for controlling a system.

ICO learning is a form of online unsupervised learning where its rule for synaptic adaptation is based on the cross-correlation of the two types of input signals: Multiple predictive signals (here considered as CS) which are earlier occurring stimuli and a single reflex signal (here considered as a US) which arrives later with certain delays and drives an unwanted response (or reflex). The learning goal of ICO learning is to use a predictive signal ((observable) state of the system) to predict the occurrence of a reflex signal (some exogenous immediate feedback, e.g., reaching a failure state), thereby allowing an agent to react earlier. In other words, this learning mechanism enables the agent to learn to perform an anticipatory action to avoid the reflex. For example, heat radiation (predictive signal) precedes a pain signal (reflex signal) when touching a hot surface. Thus, we learn an anticipatory action to avoid the late unwanted stimulus (i.e., avoiding to touch the hot surface).

Normally, the synaptic adaptation of ICO learning changes through heterosynaptic interactions [27] as a consequence of the order of the arriving inputs. If the predictive inputs are followed by the reflex input, the plastic synapses of the predictive inputs get strengthened but they get weakened if the order is reversed. Hence, this form of plasticity depends on the timing of correlated neural signals. Formally, we have

$$O_{\mathrm{ICO}}(t) = \rho_0 x_0(t) + \sum_{k=1}^{N} \rho_k(t) x_k(t) \tag{1}$$

as the output neuron (O_{ICO}) driven by a linear combination of the reflex input (x_0) and the multiple predictive inputs (x_k). N denotes the number of predictive inputs. ρ_0 is the synaptic strength of the reflex input. This synaptic strength is set to a positive value, e.g., 1.0, and remains unchanged, like an innate reflex. During learning, the plastic synapses (ρ_k) get changed by differential Hebbian learning [32, 37] using the cross-correlation between both inputs (i.e., x_0 and x_k). This is expressed as:

$$\frac{d\rho_k(t)}{dt} = \mu x_k(t) \frac{dx_0(t)}{dt}, \quad k = 1, \ldots, N. \tag{2}$$

μ is the learning rate which defines how fast a system can learn. It is generally set to a value smaller than 1.0. This learning mechanism leads to weight stabilization as soon as $x_0 = 0$ [52], meaning that the reflex has been successfully avoided. As a result, we obtain behavioral and synaptic stability at the same time without any additional weight-control mechanisms.

Due to the learning rule, ICO learning can be considered as a model-free method since its does not require a system or environment model. However, one should note that ICO learning requires the proper design of a reflex into the system from the beginning. This means that we have to set up a feedback system which has a desired state and an error signal ($x_0 \rightarrow 0$) which drives learning. If the tasks become more complex where a reflex cannot be properly designed or no correlation between a reflex signal and a predictive signal exists at all, ICO learning will fail.

3. Reward-Based Learning

For reward-based learning, we used continuous actor–critic RL [19] (see Fig. 2) since it is capable of generating (multidimensional) continuous actions thus providing a smooth control performance. It is also practical for continuous state-action problems [19], like dynamic motion control [20, 46]. In addition, it is based on a biological learning model [70] where its learning rule for synaptic adaptation considers an association between stimuli and/or actions with the reinforcement that an agent receives. Formally, the reinforcement is a reward or a punishment which is "evaluative feedback" defined by the designers of a system. Thus, this kind of

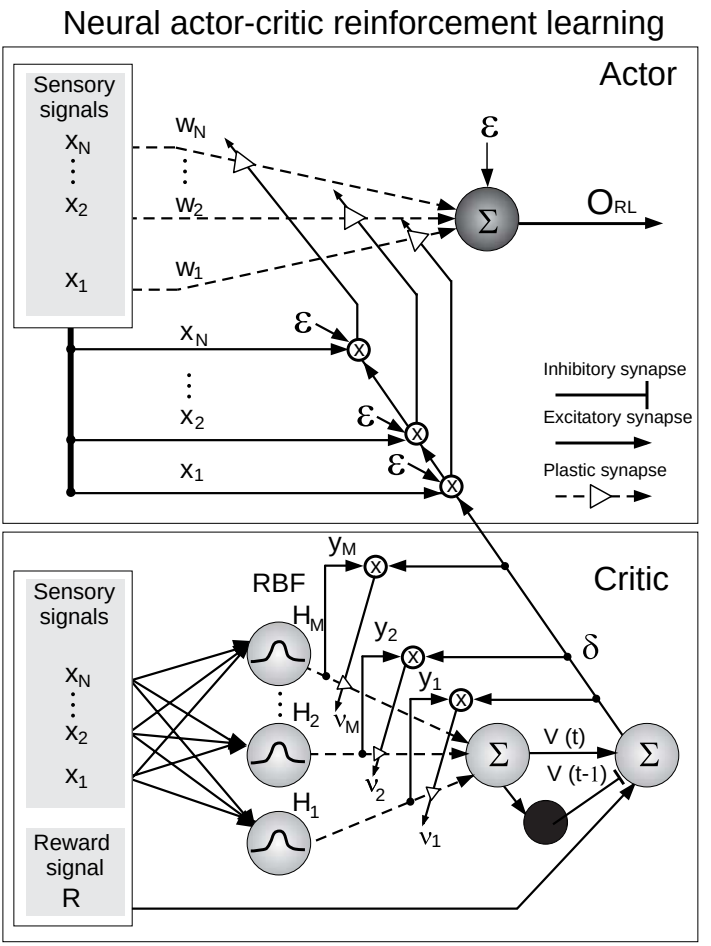


Fig. 2. Neural circuit of continuous actor–critic RL. The learning mechanisms of the actor and critic are based on TD learning. The actor component was modeled as a stochastic neural network while the critic unit was modeled as a radial basis function (RBF) neural network (see text for details). Note that in this framework, the actor provides a continuous output signal for controlling a system.

learning mechanism is minimally supervised because an agent is not told explicitly what actions to take in a particular situation. Rather it must work this out for itself on the basis of the reinforcement.

Continuous actor–critic RL is divided into two sub-mechanisms: The learning of an action function (actor) and the learning of an evaluation function (critic). The action part is the controller of an agent. In this study, it was designed as a stochastic unit proposed in Ref. [24]. If we consider one-dimensional output, its output (O_{RL}) is specified by:

$$O_{\mathrm{RL}}(t) = \varepsilon(t) + \sum_{k=1}^{N} w_k(t) x_k(t), \tag{3}$$

where N denotes the number of sensory inputs (x_k) which, here, are comparable to the predictive inputs of ICO learning. ε is an exploration term. According to Ref. [19], it is varied based on a modulation scheme[b] given by:

$$\varepsilon(t) = \xi\sigma(t) \cdot \min\left[1, \max\left[0, \frac{V_{\max} - V(\mathbf{x}(t))}{V_{\max} - V_{\min}}\right]\right]. \tag{4}$$

σ is the Gaussian distributed noise with zero mean and standard deviation of one. V is a value function (see its equation below) that estimates the expected cumulative future reward or the expected return where the reward is used to estimate how good it is for an agent to be in a given state. $V_{\max}$ and $V_{\min}$ are the maximal and minimal values of V. This way, the exploration is large if V is close to $V_{\min}$. On the other hand, the exploration is small (close to zero) if it is close to $V_{\max}$ meaning that learning shows good prediction or the performance is improved. ξ is an additional scale factor. It is introduced in order to be able to amplify the exploration level.

The stochastic unit is related to two biological learning concepts, called behavior oscillation [26] and successive approximation [59] (see also Ref. [24] for more details). During learning, the synaptic weights (w_k) of the actor change over time. They are basically changed by a stochastic RL algorithm [24]. Instead of using the error of a direct reward, which is one of the learning parameters and originally used in the stochastic RL algorithm, here we used the TD error [7, 19] (i.e., the error of an internal reward [7]). By doing so, delayed reward control problems can also be solved [7, 19]. The equation of the weight adaptation is described by:

$$\frac{dw_k(t)}{dt} = \alpha\delta(t)x_k(t)\varepsilon(t), \quad k = 1, \ldots, N, \tag{5}$$

where α is the learning rate and generally set to a value smaller than 1.0. δ is an approximation to the TD error in continuous time described as an internal reinforcement signal provided by the critic (see below).

[b]The scheme follows the intuition that an agent should explore a lot if its expected cumulative future reward V is small (close to $V_{\min}$). This means that it has a poor control policy. On the other hand, it should exploit or follow the control policy if V is close to $V_{\max}$. However, this normally works if $V_{\min}$ and $V_{\max}$ could be estimated.

For the critic network, according to Ref. [45] we used a radial basis function (RBF) neural network as a function approximator which attempts to construct the approximation of the value function V. It is governed by:

$$V(\mathbf{x}(t)) = \sum_{j=1}^{M} v_j(t) y_j(\mathbf{x}(t)), \tag{6}$$

where y_j are the outputs from the normalized Gaussian basis functions given by:

$$y_j(\mathbf{x}(t)) = \frac{a_j(\mathbf{x}(t))}{\sum_{l=1}^{M} a_l(\mathbf{x}(t))}, \quad a_j(\mathbf{x}(t)) = e^{-\|\mathbf{s}_j^T(\mathbf{x}(t)-\mathbf{c}_j)\|^2}. \tag{7}$$

The vectors $\mathbf{c}_j$ and $\mathbf{x}$ define the center and the input feature, respectively. $\mathbf{s}_j$ is the diagonal matrix of the inverse covariance of the RBF neural network. M is the number of hidden neurons. According to Ref. [19], v_j are synaptic weights which are updated by:

$$\frac{dv_j(t)}{dt} = \lambda \delta(t) y_j(\mathbf{x}(t)), \quad j = 1, \ldots, M, \tag{8}$$

where λ is the learning rate. It is generally set to a value smaller than 1.0. According to Ref. [19], the TD-error δ is basically computed from the prediction as follows:

$$\delta(t) = R(t) - \frac{1}{\tau} V(\mathbf{x}(t)) + \dot{V}(\mathbf{x}(t)), \tag{9}$$

where R is an external reinforcement signal provided by designers. τ is the time constant of a discount factor. V is the value function [see Eq. (6)] and $\dot{V}$ is its derivative with respect to time. Note that using the Euler discretization, the TD error in continuous time is compatible to the conventional TD error [62]: $\delta(t) = R(t) + \gamma V(\mathbf{x}(t)) - V(\mathbf{x}(t-1))$ where $\gamma = 1 - \frac{\Delta t}{\tau}$ is the discount factor and Δt is the time step of the Euler differentiation.

4. Combining Correlation-Based and Reward-Based Learning

In the previous sections we have presented ICO learning and continuous actor–critic RL. It is known that ICO learning can quickly learn a correlation between a failure state recognized by an immediate reflex signal and a failure avoidance behavior (or also called reflex avoidance behavior) controlled by predictive signals [52]. However, it cannot evaluate whether a given state or weight change predicts something "good" or "bad" which will happen many steps away in the future. As a consequence, this makes it difficult for the controller to properly learn solving some difficult tasks (see Sec. 5). On the other hand, continuous actor–critic RL can make predictions through its evaluation process such that it can solve the tasks but in general it is slower than ICO learning (see Sec. 5). In addition, due to its stochastic process it requires several learning repetitions to ensure that a successful control policy has been achieved.

Thus, in this section, we introduce a combinatorial learning model. It makes use of the advantage of each learning mechanism, resulting in an appropriate acquisition

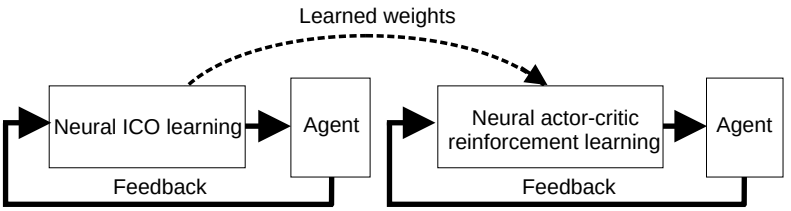


Fig. 3. Sequential combination model. ICO learning first learns to find a solution controlling a system without any prior knowledge. Afterwards the learned weights from ICO learning are provided to continuous actor–critic RL for initialization. Finally, continuous actor–critic RL serves as an add-on learning process to enhance the performance of a controller (see Ref. [43] for more details).

of the control policy that outperforms either ICO learning or continuous actor–critic RL alone (see Sec. 5). Basically, there are two ways of combining ICO learning and continuous actor–critic RL: sequential or parallel.

Sequential combination (see Fig. 3), which we investigated previously [43], is achieved by initially using ICO learning to extract reward-related features for continuous actor–critic RL. Afterwards continuous actor–critic RL uses the extracted features as priors (i.e., initial control parameters) to improve the control policy of the system. However, the drawback of this learning scheme is that it is technically inconvenient since we need to let the ICO learning mechanism learn the whole feature space first such that reward-related features are properly extracted.

In contrast, the parallel combination, proposed in this study and later called here combinatorial learning (see Fig. 4), is technically more convenient since it allows these two learning mechanisms to simultaneously learn, thereby working as a dual learner system. By doing so, they receive sensory feedback from the agent in parallel and adapt their weights accordingly. Their output signal contributes equally to the control of the agent. Thus, the final output (O_{COM}) is described as:

$$O_{\mathrm{COM}}(t) = \zeta \cdot (O_{\mathrm{ICO}}(t) + O_{\mathrm{RL}}(t)), \tag{10}$$

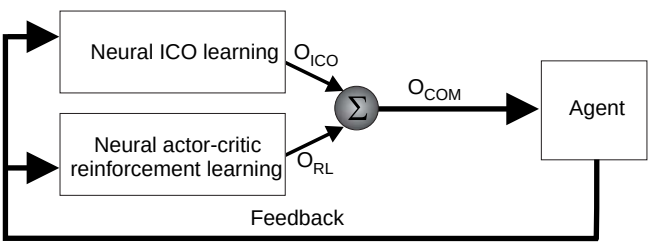


Fig. 4. Combinatorial learning model. It combines ICO learning and continuous actor–critic RL in a parallel manner for controlling an agent. In this learning scheme, each learning mechanism develops its weights independently, but they are coupled by sensory feedback. This way, they basically coadapt the control parameters (i.e., weights) leading to the improvement of the control policy.

where O_{ICO} and O_{RL} are the outputs of ICO learning and continuous actor–critic RL, respectively. ζ is a scale factor which is introduced to ensure that the sum is a valid control signal. The complete algorithm of combinatorial leaning with pseudocode is shown in Table 1.

In this learning scheme, ICO learning and continuous actor–critic RL can complement each other due to their learning principles. ICO learning [see Eq. (2)] relies on the predefined reflex signal, while continuous actor–critic RL [see Eq. (5)] depends on the TD error (δ) based on the estimated value function and the reward. As a consequence of the reflex avoidance learning principle, weight adaptation of ICO learning is initially more relevant than that from continuous actor–critic RL (until the value function is properly estimated). Thus, in some situations, like a pole balancing task (see Sec. 5.1), ICO learning quickly updates weights (i.e., control

Table 1. Combinatorial learning algorithm.

Initialize ρ_k, w_k, and v_j to 0.0; ε = Gaussian random number
Repeat:
At time step t
(1) observe reflex signal x_0 and sensory signals x_k which are the state $\mathbf{x}$
(2) compute control output

$$O_{\mathrm{ICO}} \leftarrow \rho_0 x_0 + \sum_{k=1}^{N} \rho_k x_k$$

$$O_{\mathrm{RL}} \leftarrow \varepsilon + \sum_{k=1}^{N} w_k x_k$$

$$O_{\mathrm{COM}} \leftarrow \zeta \cdot (O_{\mathrm{ICO}} + O_{\mathrm{RL}})$$

(3) perform action
(4) observe reward R, new state $\mathbf{x}'$ and new reflex signal x'_0
(5) obtain value function by computing

$$a_j \leftarrow e^{-\|\mathbf{s}_j^T(\mathbf{x}-\mathbf{c}_j)\|^2}$$

$$y_j \leftarrow \frac{a_j}{\sum_{l=1}^{M} a_l}$$

$$V \leftarrow \sum_{j=1}^{M} v_j y_j$$

(6) compute $\varepsilon \leftarrow \xi\sigma \cdot \min\left[1, \max\left[0, \frac{V_{\max}-V(\mathbf{x})}{V_{\max}-V_{\min}}\right]\right]$
(7) compute $\delta \leftarrow R + \gamma V(\mathbf{x}') - V(\mathbf{x})$
(8) update control parameters

$$\rho_k \leftarrow \rho_k + \mu x_k (x'_0 - x_0)$$

$$w_k \leftarrow w_k + \alpha\delta x_k \varepsilon$$

$$v_j \leftarrow v_j + \lambda\delta y_j$$

Until: Successful control policy is found or the maximum number of trials is reached.

parameters) to enhance or guide the entire learning process that includes continuous actor–critic RL. At the same time, ICO learning also utilizes the exploration strategy of continuous actor–critic RL to indirectly adapt its weights. In other situations, like a goal-directed behavior task (see Sec. 5.2), ICO learning plays roles on guiding continuous actor–critic RL to receive a reward and shaping a control policy. However, if reflex signal and TD error disagree, ICO learning may interfere with continuous actor–critic RL (see also Sec. 6 for more discussion on this).

Beside this, one important property of our approach is that we directly use sensory inputs as the state of a system (i.e., continuous state) without resorting to the explicit discretization of states and actions. Thus, this approach is capable of generating a continuous action leading to smooth control performance. It is also practical for continuous state-action problems (e.g., pole balancing and goal-directed behavior shown below) in particular in the domain of model-free control problems because of the learning rules (i.e., correlation-based learning and reward-based learning) which do not require a system or environment model.

5. Experiments and Results

We tested the performance of our combinatorial learning in two different tasks: A dynamic motion control task using a simulated cart-pole system and a goal-directed behavior control task using a simulated mobile robot system. In each of them we compared the performance of three control schemes: ICO learning, continuous actor–critic RL, and combinatorial learning. In addition, we also investigated interactions between ICO learning and continuous actor–critic RL by observing learning curves in order to understand their roles in combinatorial learning. It is important to note that the aim of this study is not to claim that the combination outperforms other/older methods for solving the tasks or model-free optimal control problems. Thus, comparing our combinatorial learning with other baseline methods (like, dynamic programming) will go beyond the scope of this work. Instead, we emphasize here that a combination is better than its individual components by utilizing the learning properties of ICO learning and continuous actor–critic RL.

5.1. *Dynamic motion control*

In this section, we demonstrate the performance of combinatorial learning (see Fig. 4) applied to a pole balancing problem [7] (see Fig. 5). The task was to balance an inverted pendulum, which is mounted on a cart moving freely in a one-dimensional interval, and to simultaneously avoid the interval boundaries. This cart-pole system was simulated on a desktop PC and updated by using a fourth-order Runge–Kutta method with a time step of 0.01 s. It provides four state variables: The angle of the pole with the vertical (θ), the pole angular velocity ($\dot{\theta}$), the position of the cart on the track (x), and the cart velocity ($\dot{x}$). Similar to Ref. [7], the cart was bound to move in the interval $-2.4 \leq x \leq 2.4\,[m]$ and the angle was

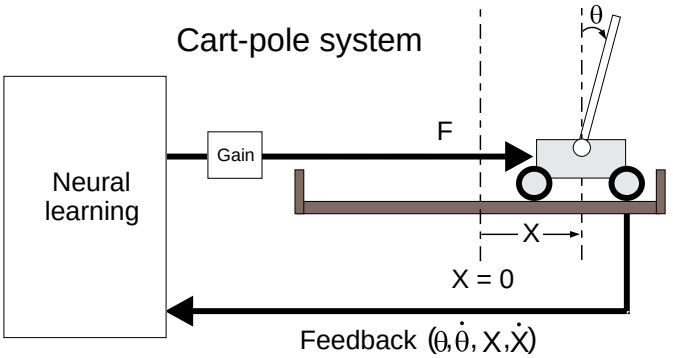


Fig. 5. Cart-pole system for a dynamic motion control task (see text for details).

allowed to vary in the interval $-12 \leq \theta \leq 12\,[^\circ]$. The dynamics of the cart-pole system is modeled by:

$$\ddot{\theta} = \frac{g \sin\theta + \cos\theta \left(\dfrac{-F - ml\dot{\theta}^2 \sin\theta + \mu_c \mathrm{sgn}(\dot{x})}{M+m} \right) - \dfrac{\mu_p \dot{\theta}}{ml}}{l\left(\dfrac{4}{3} - \dfrac{m\cos^2\theta}{M+m} \right)}, \tag{11}$$

$$\ddot{x} = \frac{F + ml(\dot{\theta}^2 \sin\theta - \ddot{\theta}\cos\theta) - \mu_c \mathrm{sgn}(\dot{x})}{M+m}, \tag{12}$$

where $g = 9.8\,\mathrm{m/s^2}$ denotes gravitational acceleration, $M = 1.0\,\mathrm{kg}$ and $m = 0.1\,\mathrm{kg}$ are mass of the cart and pole, respectively. $l = 0.5\,\mathrm{m}$ is half of the pole length. $\mu_c = 5.0 \times 10^{-4}$ and $\mu_p = 2.0 \times 10^{-6}$ are friction coefficient of the cart and pole, respectively. F is a continuous force applied to the cart which is directly derived from the output of learning mechanisms with an amplifier gain of 10.0. Note that all these parameters and the cart-pole equations are generally used [7, 49].

In fact, this task is difficult in its own right due to the limited boundaries of the pole angle and in particular the cart position. The boundaries are used as a standard benchmark setup in most control studies [7, 49]. In addition, its vertical upright equilibrium point to be balanced is inherently unstable (i.e., as any small disturbance may cause the pole to fall on the either side). From this setup, balancing the pole at critical initial conditions (e.g., $\theta = 11\,\mathrm{deg}$, $x = 2.1\,\mathrm{m}$) close to the boundaries is already difficult to find successful control policies by using a simple reward function.

In this setup, the four state variables (x, $\dot{x}$, θ, $\dot{\theta}$) of the system were used as sensory feedback ($x_{1,2,3,4}$) to ICO learning [see Eq. (1)] and continuous actor–critic RL [see Eqs. (3) and (7)]. For ICO learning, these state variables were scaled onto the interval $[-1, 1]$ similar to Ref. [49] and the reflex signal [x_0, see Eq. (1)] was given just before the system failed. The signal shows a positive activation (+1.0) if $x < -2.35\,\mathrm{m}$ or $\theta > 11.5^\circ$, a negative activation (−1.0) if $x > 2.35\,\mathrm{m}$ or $\theta < -11.5^\circ$,

and 0 otherwise. Here, we set the learning rate [μ, see Eq. (2)] of ICO learning to 0.1. Note that the weights [$\rho_{1,2,3,4}$, see Eq. (2)] are changed only for the positive derivatives of the reflex signal, otherwise they remain unchanged. This is to avoid negative correlations resulting in poor performance.

For continuous actor–critic RL, we allocated three bases for x, three for $\dot{x}$, six for θ, and three for $\dot{\theta}$ according to a boxes approach (see Ref. [7] for more details). This leads to $3 \times 3 \times 6 \times 3 = 162$ bases employed as the centers of the critic network. Thus, the network has in total 162 hidden neurons [$M = 162$, see Eqs. (6) and (7)] which cover the state space of the system. The size or width of the Gaussian basis functions was simply set to twice the distance between its center and the center of its nearest neighbor. The reward signal [R, see Eq. (9)] was set to -1 at failure (i.e., cart hits the boundaries or pole falls to $\pm$ 12 deg) and 0 otherwise [7]. Here, $V_{\max}$ and $V_{\min}$ of the modulation scheme controlling the level of the exploration [ε, see Eq. (4)] were set to 0 and -1, respectively. In this setup, we set the scale factor (ξ) of the exploration to 5.0. This is to obtain a better performance. Thus, large changes of the weights of continuous actor–critic can occur. The control parameters α [see Eq. (5)], λ [see Eq. (8)], τ [see Eq. (9)], and ζ [see Eq. (10)] were set to 0.5, 0.5, 0.2, and 0.5, respectively.

We let the combinatorial learning mechanism learn to balance the pole on 25×49 initial conditions (θ, x) while $\dot{\theta}$ and $\dot{x}$ were initially set to small random values using a Gaussian distribution with zero mean and a standard deviation of 0.1% of signal ranges which represents the system noise. Note that the control parameters (i.e., synaptic weights) of ICO learning and continuous actor–critic RL were initially set to 0.0. During a run each trial started with a given initial state and ended either in "success" (which occurs when the pole is kept in balance for at least 5×10^4 time steps or 500 s) or "failure" (which occurs when the pole falls 12 deg to either side or the cart moves 2.4 m to either side). Runs at each initial condition were terminated on failure or when a successful trial was achieved or the maximum number of trials was reached (here 1000 trials). The system was reset to the same initial state at failure. We repeated this for 25 experiments at each initial condition.

The performance of combinatorial learning is shown in Fig. 6(a). It can be seen that this learning mechanism was able to find successful control policies which can balance the pole and avoid the ends of the interval in a very large (x, θ)–domain of initial conditions [see Fig. 6(a)]. The system was successfully stabilized for $\approx 96\%$ of all initial conditions. This is because, on one hand, ICO learning utilizes the exploration strategy of continuous actor–critic RL to explore its parameter space such that a proper weight combination is obtained. At the same time continuous actor–critic RL is also guided by the built-in reflex of ICO learning to adapt its weights in a proper way. The remaining part (black areas), at which learning failed, is because of physical limitation. For example, if the system stands close to the right wall and the pole falls to the right, the cart momentum cannot be high enough to support the pole. Thus, it crashes into the wall before. The results we obtained here are comparable to the ones shown in Ref. [49] where this work employed

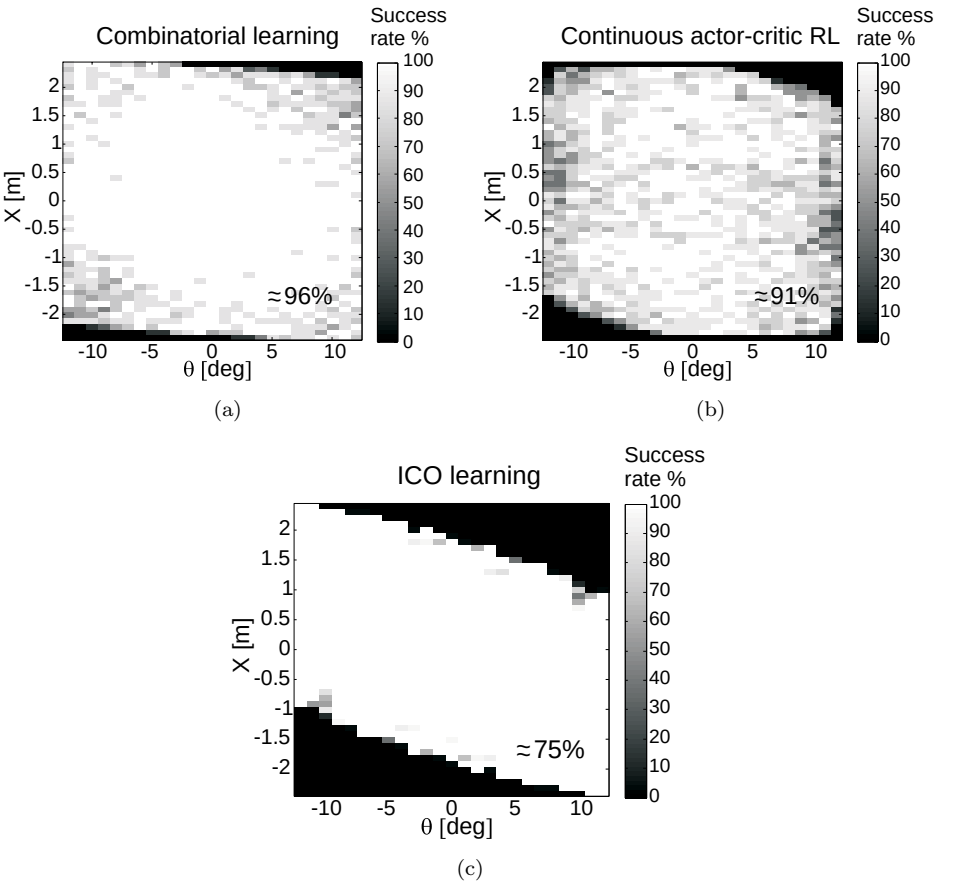


Fig. 6. Performance of three learning mechanisms for the pole balancing problem. (a) Successful control area (≈ 96%) of combinatorial learning on benchmark initial conditions. (b) Successful control area (≈ 91%) of continuous actor–critic RL. (c) Successful control area (≈ 75%) of ICO learning. Black areas represent a domain in which learning failed to solve the problem (i.e., it cannot learn to stabilize the system). A gray scale bar presents the success rate, i.e., the percentage of success from 25 experiments. Recall that "success" means the pole is kept in balance for at least 500 s.

similar linear control with four inputs but used an evolutionary algorithm for weight adaptation.

When only ICO learning or continuous actor–critic RL alone was applied, stabilizing the system was accomplished in smaller domains. For ICO learning alone, the system was balanced for ≈ 75% of all initial conditions [see Fig. 6(c)]. This is because ICO learning cannot explore the entire parameter space due to the lack of an exploration mechanism and it even cannot predict many steps into the future. It basically develops its weights (i.e., control parameters) with respect to an immediate correlation between predictive and reflex signals. In this setup, the built-in reflex occurs only at the last moment that the pole falls or the cart hits the wall.

Therefore, at initial conditions in which the system fails, the reflex signal cannot produce a strong cart momentum to turn the pole into an upright position or keep it balance for a certain period of time (i.e., avoiding the reflex). Thus, ICO learning cannot obtain a proper correlation between the predictive and reflex signals to achieve a proper weight combination. For continuous actor–critic RL alone, due to the lack of a prior knowledge, the system was balanced for $\approx 91\%$ of all initial conditions [see Fig. 6(b)]. This experimental result shows that, among the three learning mechanisms, combinatorial learning, which combines ICO learning and continuous actor–critic RL, was the best approach with respect to the success rate.

Note that, due mainly to the stochastic process of continuous actor–critic RL and partly to the introduced system noise which can easily destabilize the system, combinatorial learning and continuous actor–critic RL alone sometimes had difficulty or failed to find successful control policies in a given number of trials at, e.g., $x = -2.0\,\text{m}$, $\theta = 12\,\text{deg}$ and around $x = 0.0\,\text{m}$, $\theta = 0\,\text{deg}$, respectively. In contrast, ICO learning alone was almost 100% success at these initial conditions and even showed the very clear boundary between white and black areas [see Fig. 6(c)] since it is deterministic control where no exploration is involved.

To compare the learning speed of these three learning mechanisms in general cases, we observed their performance at a noncritical initial condition (e.g., $x = 1.0\,\text{m}$, $\theta = -1\,\text{deg}$), where they all can find successful control policies, and at a critical initial condition (e.g., $x = -1.8\,\text{m}, \theta = -5\,\text{deg}$) where combinatorial learning and continuous actor–critic RL can find the policies but ICO learning cannot. The result is shown in Fig. 7.

At the noncritical initial condition ICO learning was fastest, combinatorial learning was slower, and continuous actor–critic RL was the slowest [see Fig. 7(a)]. At the critical initial condition ICO learning failed while continuous actor–critic RL succeeded but required more learning trials compared to combinatorial learning. This experiment suggests that the fast convergence property of combinatorial learning is generally derived from ICO learning which can quickly learn to find a solution for a task but cannot properly learn solving a difficult task (i.e., here, stabilizing the system at a critical initial condition). Furthermore, the capability for solving a difficult task is basically obtained from continuous actor–critic RL which learns the task but usually takes many learning trials.

To better understand why the combined mechanism outperforms either ICO learning or continuous actor–critic RL alone, we also observed learning curves at a critical initial condition (e.g., $x = 2.1\,\text{m}$, $\theta = 11\,\text{deg}$) at which its individual components failed. Figure 8 shows that control parameters (i.e., synaptic weights) converged to fixed values when combinatorial learning was used [see thick lines in Figs. 8(a) and 8(b)]. This is because ICO learning and continuous actor–critic RL tried to find a proper weight combination such that a proper force is generated to push the cart for balancing the pole. This proper combination can be seen when the weight for one input in ICO learning increased and that in continuous actor–critic RL network decreased (e.g., ρ_x, w_x in Fig. 8).

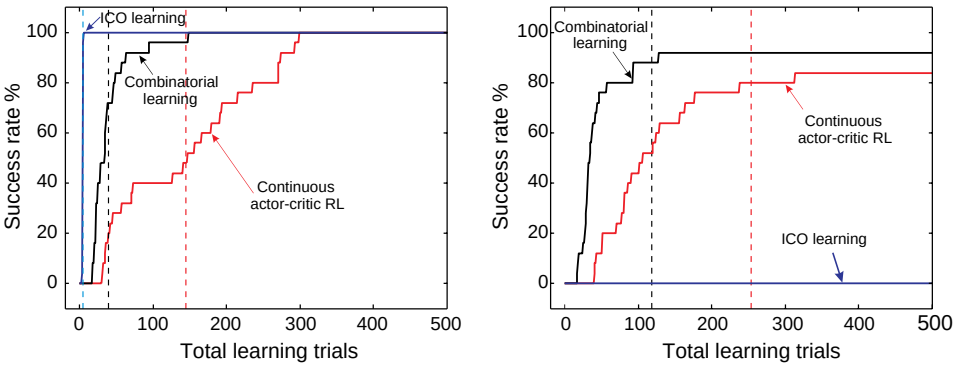
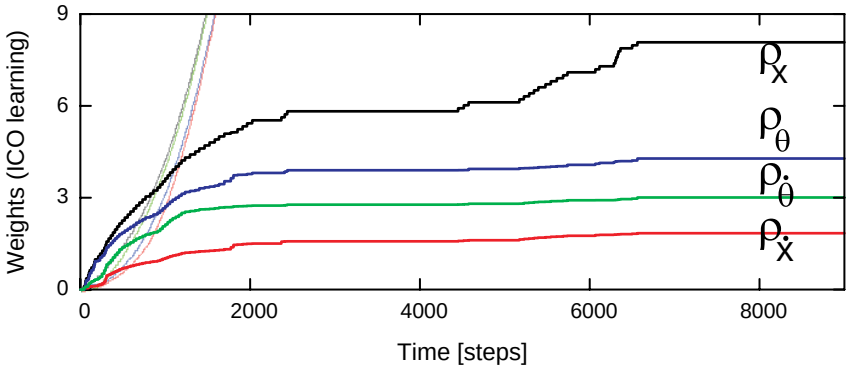


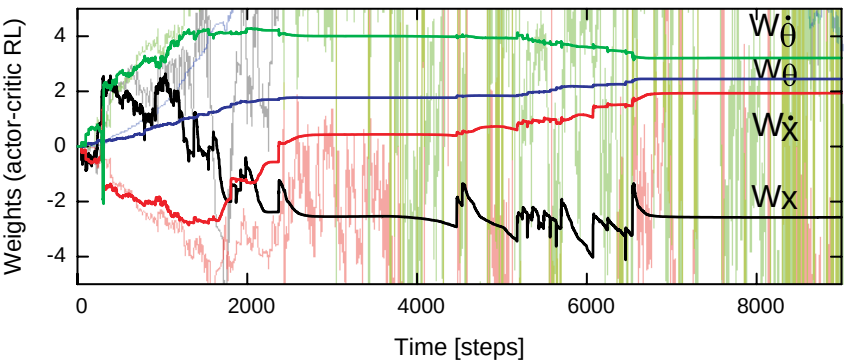
Fig. 7. (Color online) Comparison of the performance of three learning mechanisms. (a) Success rate according to total learning trials at the noncritical initial condition ($x = 1.0$ m, $\theta = -1$ deg). (b) Success rate according to total learning trials at the critical initial condition ($x = -1.8$ m, $\theta = -5$ deg). At this critical initial condition, ICO learning failed because during learning its weights grew more and more. Thus, the output of ICO learning, applied to the cart-pole system, also strongly increased. As a result, the output disturbed the system rather than balancing it. Note that we did not limit the output. Recall that success rate is calculated from the percentage of success in the total 25 experiments after a certain number of trials where "success" means the pole is kept in balance for at least 500 s. Dashed lines indicate the average of the total learning trials at success.

The behavior of the system controlled by all converged weights is shown in Fig. 9. Due to the proper weight combination, at the beginning a proper positive force was generated to push the cart to the right such that the pole could swing to the left to obtain an upward position. Afterwards, a negative force was generated to balance the pole and push the cart to the center. Finally, the system was stabilized at the center where all inputs were converged to zero values, thereby no force was generated. As a result, the pole was successfully balanced. If the converged weights of the ICO learning module were only used to control the system while the weights of the continuous actor–critic RL module were set to 0.0, the controller produced a very strong positive force to the cart at an early state. As a consequence, the pole fell to -12 deg (see Fig. 10). On the other hand, if the converged weights of the continuous actor–critic RL module were only used to control the system while the weights of the ICO learning module were set to 0.0, the controller produced a very strong negative force at an early state, thereby making the pole quickly fall to 12 deg (see Fig. 11). For these two cases, the system could not be stabilized since the forces were not properly generated.

When ICO learning alone [see transparent lines in Fig. 8(a)] was used to learn to balance the pole at the critical initial condition ($x = 2.1$ m, $\theta = 11$ deg), its control parameters diverged since the reflex signal cannot be avoided (i.e., the pole always fell). Although ICO learning is designed to learn to avoid a reflex signal, in this pole balancing setup the built-in reflex occurs only at the last moment that the pole falls



(a)



(b)

Fig. 8. (Color online) Learning curves at a critical initial condition ($x = 2.1\,\text{m}$, $\theta = 11\,\text{deg}$). (a) Weight changes in ICO learning. (b) Weight changes in continuous actor–critic RL. Thick lines present the weight changes in each learning mechanism in the combinatorial learning framework while transparent lines show the weight changes when only ICO learning or continuous actor–critic RL was used. Using combinatorial learning, the weights became stable after around 6500 time steps (or 70 trials) meaning that the system was successfully stabilized. In contrast, the weights diverged when only ICO learning [see (a)] was used while they changed a lot in the case of continuous actor–critic RL alone [see (b)]. Note that sudden change in w_x occurred (e.g., around 4400 steps) because there was a high correlation between the TD error and the input (x) while there were low correlations between the other inputs and the TD error. Here, $\rho_x = \rho_1$, $\rho_{\dot{x}} = \rho_2$, $\rho_\theta = \rho_3$, $\rho_{\dot{\theta}} = \rho_4$, $w_x = w_1$, $w_{\dot{x}} = w_2$, $w_\theta = w_3$, $w_{\dot{\theta}} = w_4$.

or the cart hits the wall. Therefore, in this difficult situation, a reflex signal cannot produce a strong cart momentum to turn the pole into an upright position or keep it balance for a certain period. Thus, ICO learning cannot obtain a proper correlation between the predictive and reflex signals; thereby its weights just increased more and more to try to find any proper weight combination. However, in this situation such a combination leading to a proper force cannot be achieved. While the weights were increasing, the output of ICO learning, applied to the cart-pole system, also

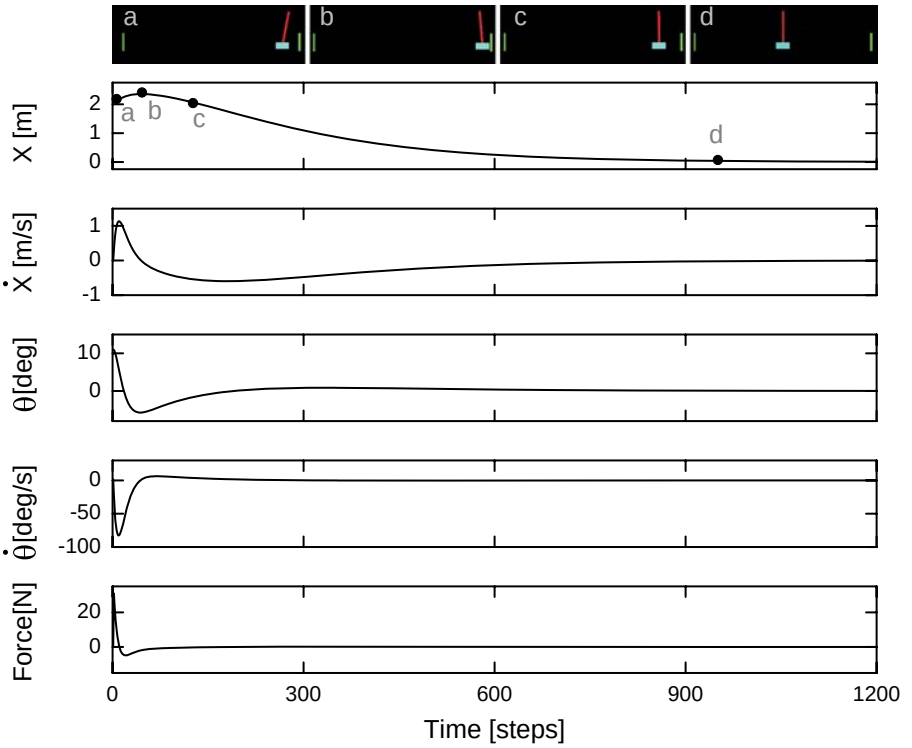


Fig. 9. States of the cart-pole system (x, $\dot{x}$, θ, $\dot{\theta}$) and the force under control of the learned weights ($\rho_x \approx 8.0774$, $\rho_{\dot{x}} \approx 1.8395$, $\rho_\theta \approx 4.2815$, $\rho_{\dot{\theta}} \approx 3.0069$, $w_x \approx -2.5790$, $w_{\dot{x}} \approx 1.9225$, $w_\theta \approx 2.4527$, $w_{\dot{\theta}} \approx 3.2216$, see e.g., Fig. 8) for the critical initial condition $x = 2.1$ m, $\theta = 11$ deg. A series of photos visualizing the cart-pole behavior at particular points is shown above.

increased. As a result, at some point the output disturbed the system rather than balancing it. In the case of continuous actor–critic RL alone [see transparent lines in Fig. 8(b)], the control parameters changed a lot due to the stochastic process employed which tried to search for a successful control policy. Note that at the early state of learning the weights of the ICO learning module in combinatorial learning became larger than the weights of ICO learning alone due to the stochastic process of the continuous actor–critic RL module in combinatorial learning. It can easily destabilize the system. Thus, the pole can often fall at the early state. This leads to the triggering of a reflex signal. On the other hand, in the case of ICO learning alone the pole fell less often at the early state such that the weights grew slower.

Finally, we investigated interactions between these two learning mechanisms. We first started one learning mechanism and then after a number of learning trials (e.g., 100 trials) we activated the other one (see Fig. 12). This is to observe two effects: (i) Can a later activated learning mechanism assist an earlier activated learning mechanism for policy improvement? and (ii) Can the earlier one provide

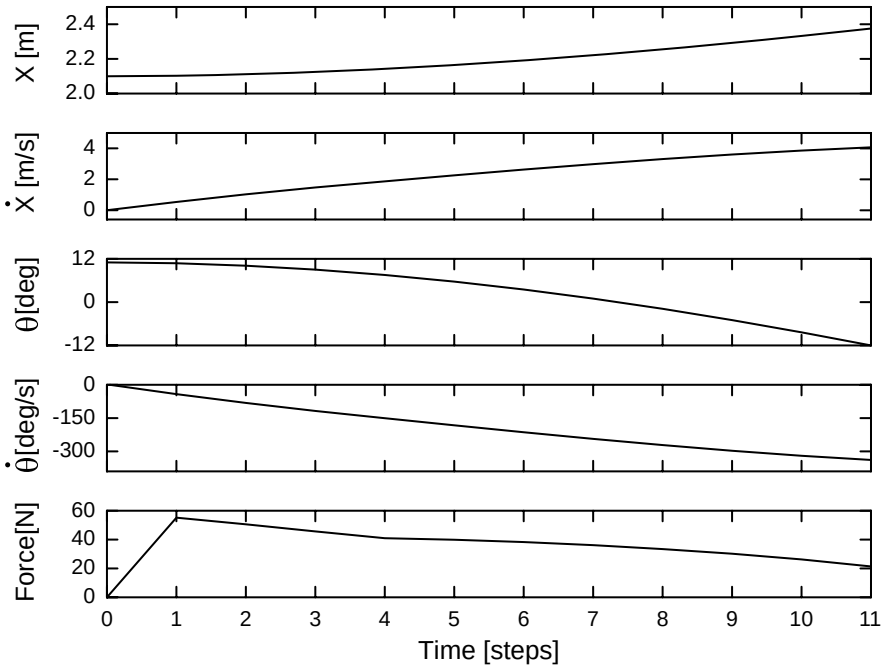


Fig. 10. States of the cart-pole system (x, $\dot{x}$, θ, $\dot{\theta}$) and the force under control of the learned weights ($\rho_x \approx 8.0774$, $\rho_{\dot{x}} \approx 1.8395$, $\rho_\theta \approx 4.2815$, $\rho_{\dot{\theta}} \approx 3.0069$, [see Fig. 8(a)] for the critical initial condition $x = 2.1$ m, $\theta = 11$ deg. We set w_x, $w_{\dot{x}}$, w_θ, and $w_{\dot{\theta}}$ to 0.0.

an appropriate developed control policy to the later one such that a successful control policy can still be achieved at the end?

Figures 12(a) and 12(b) show learning curves when continuous actor–critic RL was first started and followed by ICO learning after 100 trials [see dashed line in Fig. 12(a)]. It can be observed that after around 5500 time steps (or 130 trials), where ICO learning was already activated, the weight (w_x) of continuous actor–critic RL started to gradually change its growing direction into a different way [see e.g., thick line in Fig. 8(b)]. A similar effect also appears for the weight ($w_{\dot{x}}$) after around 9000 time steps (or 190 trials). This is because ICO learning can quickly find a correlation between a state and an unwanted condition (i.e., pole falls) and additionally generates the proper action when the pole falls through its built-in reflex. Thus, it can extract important features[c] serving to guide the learning strategy of continuous actor–critic RL. As a result, the weights of continuous actor–critic RL (e.g., w_x, $w_{\dot{x}}$) gradually changed to their appropriate directions but they did not change considerable compared to continuous actor–critic RL alone [see transparent lines in Fig. 8(b)].

[c]By feature we mean the combination between the weights and input signals.

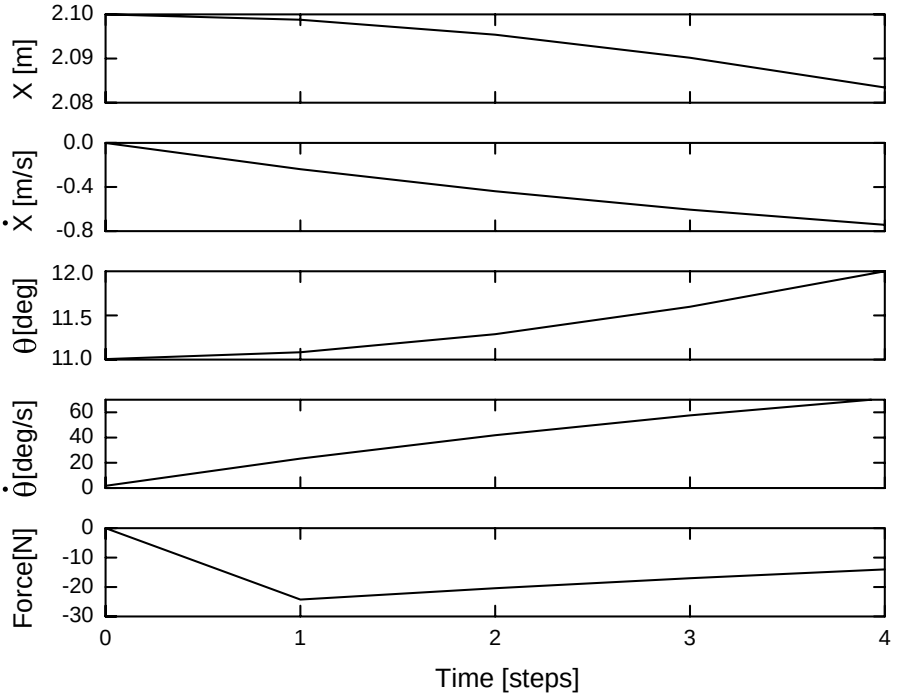


Fig. 11. States of the cart-pole system (x, $\dot{x}$, θ, $\dot{\theta}$) and the force under control of the learned weights ($w_x \approx -2.5790$, $w_{\dot{x}} \approx 1.9225$, $w_\theta \approx 2.4527$, $w_{\dot{\theta}} \approx 3.2216$, [see Fig. 8(b)]) for the critical initial condition $x = 2.1$ m, $\theta = 11$ deg. We set ρ_x, $\rho_{\dot{x}}$, ρ_θ, and $\rho_{\dot{\theta}}$ to 0.0.

Another interesting effect of the interaction is shown in Figs. 12(c) and 12(d) where ICO learning was first started and followed by continuous actor–critic RL after 100 trials [see dashed line in Fig. 12(d)]. After 115 trials (or around 2500 time steps), the pole did not fall anymore leading to reflex avoidance. As a consequence, the weights of ICO learning converged. However, the weights of continuous actor–critic RL still slightly changed due to the TD error. They finally converged (i.e., TD error ≈ 0) after around 17,600 time steps. This experiment shows that, on the one hand, continuous actor–critic RL seems to highly influence ICO learning such that the weights of ICO learning became stable shortly after continuous actor–critic RL was activated. On the other hand, ICO learning seems to provide an adequate control policy or an important feature to continuous actor–critic RL such that it can quickly adapt its weights to appropriate directions leading to convergence.

5.2. *Goal-directed behavior control*

Next, we present the performance of combinatorial learning (see Fig. 4) on a different task. Here, we employed it to a goal-directed behavior control problem. The task was to steer a wheeled mobile robot to move toward and finally approach a desired object (i.e., its goal) in a given time. In this scenario, we put the robot in

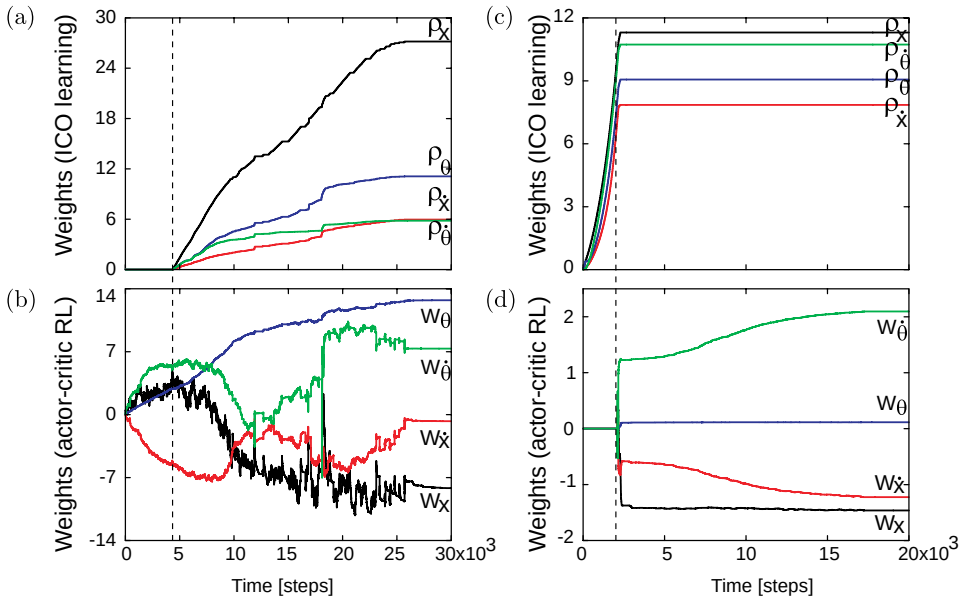


Fig. 12. (Color online) Learning curves at a critical initial condition ($x = 2.1\,\text{m}$, $\theta = 11\,\text{deg}$) when ICO learning and continuous actor–critic RL were not activated at the same time. (a), (b) Weight changes of ICO learning and continuous actor–critic RL in the combinatorial learning framework. In this experiment, ICO learning was activated after 100 trials (dashed line). (c), (d) Weight changes of ICO learning and continuous actor–critic RL where here continuous actor–critic RL was activated after 100 trials (dashed line).

a square area where one desired green object and one undesired blue object were provided. We used the physics simulator LPZROBOTS[d] to simulate the robot and its environment (see Fig. 13). The simulator was implemented on a desktop PC with an update time step of 0.01 s.

The mobile robot system provides four state variables, which are two relative orientations ($\phi_{G,B}$) and two relative positions ($D_{G,B}$) of the robot to the locations of the green (G) and blue (B) objects, and additional eight state variables of infrared (IR) sensors for boundary detection (see Fig. 13). $\phi_{G,B}$ provide information of how much the robot's direction deviates from the objects. They vary in the interval $[-180°, 180°]$ [see Fig. 13(b)] and show continuous values. If the objects are directly in front of the robot, $\phi_{G,B}$ show 0. If they are to the left-hand side of the robot, $\phi_{G,B}$ show negative values. If they are to the right-hand side of the robot, $\phi_{G,B}$ have positive values [see e.g., Fig. 13(b)]. $D_{G,B}$ provide information of how close the robot is to the objects. They are mapped onto the interval $[0, 1]$, with 0 representing near, and $+1$ representing far. If the robot comes close to an object in a certain range [i.e., $D_{G,B} > 0.7$, see dashed areas in Fig. 13(c)], a reward is given for continuous

[d]It is based on the open dynamics engine (ODE) for more details of the LPZROBOTS simulator see http://robot.informatik.uni-leipzig.de/software/.

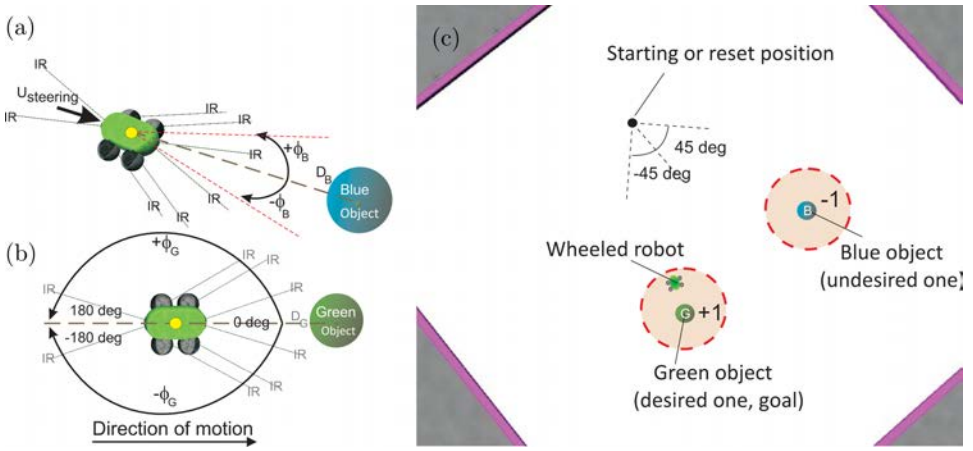


Fig. 13. (Color online) Simulated mobile robot system for a goal-directed behavior control task. (a) The mobile robot with different types of sensors (i.e., relative orientation ϕ and position D sensors and infrared IR sensors). (b) The variation of the relative orientation ϕ_G of the robot to the green object. (c) The environmental setup of the robot. The black dot represents the starting or reset position where the robot was initially started or reset after it hit a boundary or reached one of the objects. Dashed circles show areas where a positive (+1) or negative reward (−1) was given for continuous actor–critic RL and reflex signals were triggered for ICO learning (see text for details).

actor–critic RL and a reflex signal is triggered for ICO learning. The IR sensory signals are mapped onto the interval $[-1, +1]$, with -1 representing no boundary detection, and $+1$ representing hitting a boundary. This IR information was only used to reset the robot position on hitting a boundary.

It is important to note that in this setup, only $\phi_{G,B}$ were used as inputs (i.e., the state) to the control policy while $D_{G,B}$ were used only to generate reward and reflex signals for learning and to reset the robot position when approaching an object (i.e., $D_{G,B} > 0.95$). Thus, the robot has insufficient sensor data for reliably identifying its state in the environment. Furthermore, $\phi_{G,B}$ overlap with each other, i.e., the robot simultaneously senses its relative orientation to the locations of both objects in the whole area. Thus, both sensor signals try to steer the robot toward the corresponding objects once their synaptic weights have been developed.

Here, for ICO learning, $\phi_{G,B}$ were used as predictive signals [$x_{1,2}$, see Eq. (1)]. Two independent reflex signals were configured: One was for the green object [x_{0_G}, see Eq. (1)] and the other for the blue one [x_{0_B}, see Eq. (1)]. They depend on the orientations ($\phi_{G,B}$) and the positions ($D_{G,B}$) of the robot to the objects. The reflex signals are triggered as soon as the robot comes close to the objects [i.e., entering areas inside the dashed circles as shown in Fig. 13(c)], and 0 otherwise. In fact, the reflex signals elicit a turn which is proportional to the deviations defined by $\phi_{G,B}$, i.e., the larger the deviations, the sharper the turn. Thereby, they turn the robot toward the objects. In other words, ICO learning tries to control the heading direction of the robot to align with an object. This way, it can implicitly optimize

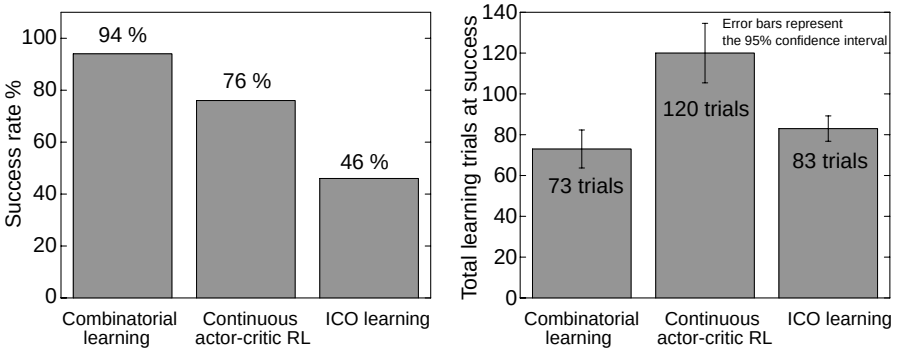
the behavior over the entire path. Since the green and blue objects are far from each other, the reflex areas do not overlap. Thus, the two reflex signals cannot be triggered at the same time. We set the learning rate [μ, see Eq. (2)] of ICO learning to 0.005. The weights [$\rho_{1,2}$, see Eq. (2)] were initially set to 0.0. They change only if the positive derivatives of the reflex signals are higher than a threshold, otherwise they remain unchanged. For example, when the robot comes close to the green object and the reflex signal is triggered, the weight (ρ_1) of the orientation signal with respect to the green object increases while the weight (ρ_2) of the blue one remains unaffected and vice versa when the robot comes close to the blue object.

For continuous actor–critic RL, we allocated four bases for ϕ_G and ϕ_B each. This leads to $4 \times 4 = 16$ bases employed as the centers of the critic network. Thus, the network has in total 16 hidden neurons [$M = 16$, see Eqs. (6) and (7)] which cover the state space of the system. The size or width of the Gaussian basis functions was simply set to twice the distance between its center and the center of its nearest neighbor. The reward signal [R, see Eq. (9)] was set to $+1$ when the robot came close to the green object (desired object or goal) and -1 to the blue object (undesired object). In order to promote exploration, we used low-pass filtered noise for low-frequency probing which was appropriate for the robot. We also used the modulation scheme for controlling the exploration level where ξ, $V_{\max}$ and $V_{\min}$ were here set to 5.0, 50 and 0, respectively. In addition to this scheme, the exploration term was exponentially reduced as soon as the performance improved (i.e., the robot frequently approached the goal). The control parameters α [see Eq. (5)], λ [see Eq. (8)], τ [see Eq. (9)], and ζ [see Eq. (10)] were set to 0.001, 0.7, 0.2, and 0.5, respectively. The weights [$w_{1,2}$, see Eq. (3)] were initially set to 0.0 and changed by Eq. (5).

We let the combinatorial learning mechanism learn to steer the robot to approach the desired goal (i.e., the green object). Without control, the robot randomly moved around. During a run in each trial, the robot started at a specific location [i.e., the black dot shown in Fig. 13(c)]. A run was terminated when the robot approached one of the objects or hit a boundary as well as when simulation time was above 15 s. After termination, the robot was reset to the same starting location with a random orientation in the interval $[-45°, 45°]$. We repeated this 50 experiments where each experiment was terminated after 200 trials. The performance of combinatorial learning compared to ICO learning and continuous actor–critic alone is shown in Fig. 14.

As can be seen, combinatorial learning had the highest success rate, continuous actor–critic RL a lower one, and ICO learning the lowest. With respect to the number of learning trials, combinatorial learning and ICO learning were not significantly different. However, they were substantially faster than continuous actor–critic RL. Among these learning mechanisms, combinatorial learning was the best approach, showing highest success rate with the lowest number of learning trials.

To better understand why the combinatorial learning mechanism outperforms its individual components in this task, we also plotted learning curves. Figure 15



(a) (b)

Fig. 14. Comparison of the performance of three learning mechanisms. (a) Success rate in a total of 50 experiments. Here, "success" means that the robot can approach the green object from the starting position with different random orientations in the interval $[-45°, 45°]$ [see e.g., Fig. 13(c)]. (b) Average of the total learning trials at success.

exemplifies the learning curves showing the changes of the control parameters (i.e., synaptic weights) of ICO learning and continuous actor–critic RL in combinatorial learning. The weights converged to fixed values [see thick lines in Figs. 15(a) and 15(b)] resulting in a goal-directed behavior. The input and output signals during this learning experiment and the behavior of the system after the weights converged are shown in Fig. 16.

When only ICO learning was used, the weights sometimes converged to other fixed values [see transparent lines in Fig. 15(a)] producing an undesired behavior; i.e., the robot moved toward the undesired blue object instead of the desired green object. When only continuous actor–critic RL was used, the weights sometimes changed a lot to negative values [see transparent lines in Fig. 15(b)]. As a consequence, the robot moved away from the objects. However, they will finally converge but this will require a lot of learning trials, e.g., > 600 trials.

In principle, ICO learning can recognize a correlation only between its inputs (i.e., predictive and reflex signals, see e.g., Fig. 1) without recognizing a goal (i.e., reward or punishment). Thus, for this task it can only generate an anticipatory reaction towards objects, rather than a goal-directed behavior. On the other hand, continuous actor–critic RL can achieve this in most cases but requires more learning trials than ICO learning. By contrast, combinatorial learning allows ICO learning and continuous actor–critic RL to complement each other leading to control policy improvement (high success rate and fast convergence [see Fig. 14]). This is because continuous actor–critic RL tries to drive a robot toward a goal with a certain degree of exploration. At the same time, ICO learning tries to limit the exploration area (i.e., guiding) since it tries to drive the robot toward the point of interest (green or blue object) defined by a prior knowledge. Without ICO learning, due to the exploration, the robot sometimes has difficulties to go back to the goal or it requires

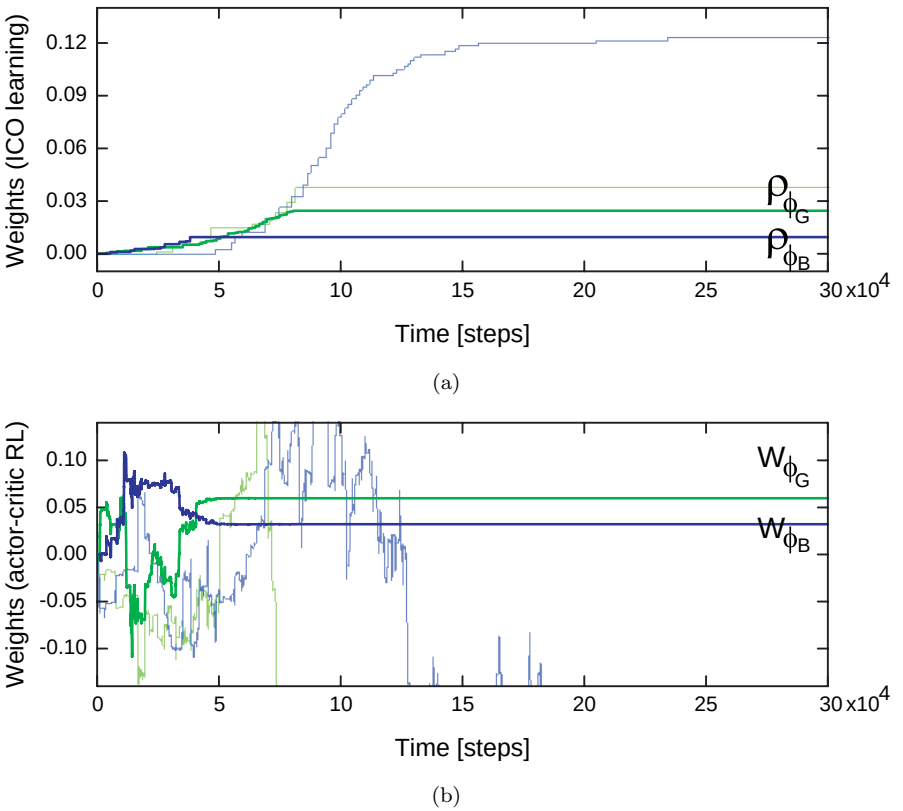
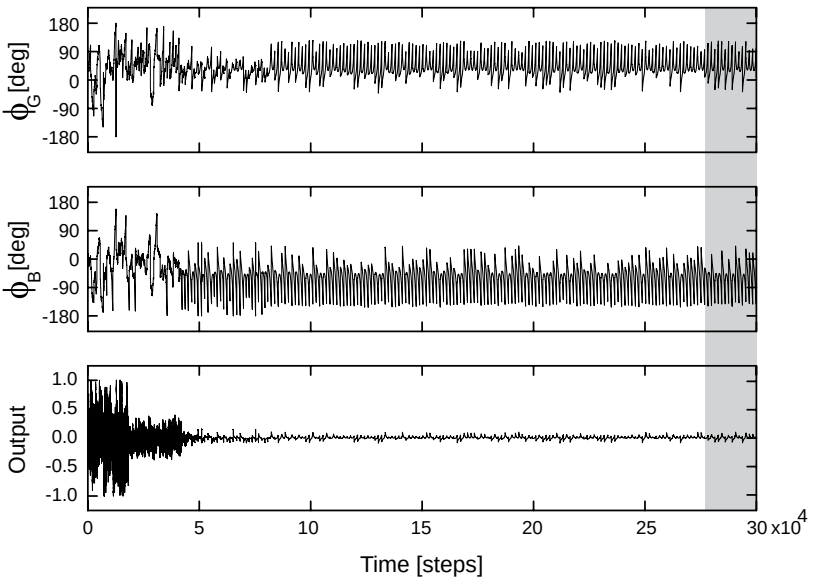


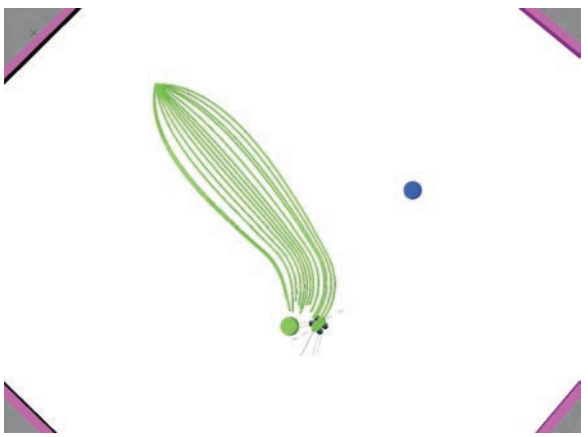
Fig. 15. (Color online) Learning curves of a goal-directed behavior. (a) Weight changes in ICO learning. (b) Weight changes in continuous actor–critic RL. Thick lines present the weight changes in each learning mechanism in the combinatorial learning framework while transparent lines show the weight changes when only ICO learning or continuous actor–critic RL was used. Using combinatorial learning, the weights became finally stable after around 75,000 time steps (or 50 trials) meaning that the robot can successfully approach the goal. In contrast, the weights sometimes converged to other fixed values in the case of ICO learning alone [see (a)] producing an undesired behavior (i.e., the robot went to a blue object) while they sometimes changed a lot in the case of continuous actor–critic RL alone [see (b)]. Note that here $\rho_{\phi_G} = \rho_1$, $\rho_{\phi_B} = \rho_2$, $w_{\phi_G} = w_1$, $w_{\phi_B} = w_2$.

more trials. In addition ICO learning can also shape the learning process such that the robot can approach the goal on a short path (see below).

To see the guiding and shaping effects from ICO learning, we took the developed weights of ICO learning and continuous actor–critic RL before convergence occurred to control the robot and observed its behavior. Then, we compared the behavior to the one controlled by only the developed weights of continuous actor–critic RL, i.e., we set the weights of ICO learning to 0.0 while the weights of continuous actor–critic RL remained unchanged. Interestingly, we found three different behaviors (I, II, III, see Figs. 17–19). Recall that in these goal-directed behavior experiments, the two relative orientations ($\phi_{G,B}$) overlap with each other, i.e.,



(a)



(b)

Fig. 16. (Color online) (a) States of the mobile robot system ($\phi_{G,B}$) and the output (O_{COM}) during learning. Learning curves belonging to these signals are shown in Fig. 15 (see thick lines). (b) Robot trajectories observed from around 28×10^4 to 30×10^4 time steps [see gray area in (a)]. Positive and negative values of the output means turning right and left, respectively. At the beginning the robot explored a lot (i.e., large amplitude of the output signal). After learning converged, the robot did not turn much (i.e., small amplitude of the output signal). It only turned if it deviated from the goal. As a result, it always approached the goal (green object). In other words, the learned control policy drove the robot toward the goal and kept it away from the blue object; thereby, ϕ_B shows most of the time negative values above 90 deg (i.e., heading away from the blue object) while ϕ_G shows most of the time positive values around 90 deg (i.e., turning towards the goal).

the robot simultaneously sensed its relative orientations to the locations of both objects in the whole area. In addition, for continuous actor–critic RL a positive reward (+1) was given when the robot got into the circle around the green object while a negative reward (−1) was given when the robot got into the circle around the blue object [see Fig. 13(c)]. Thus, during learning as long as the exploration term and the TD error existed, the weights of both signals simultaneously changed, no matter where the robot was.

Figure 17 shows the first behavior I where we took the developed weights at around 3×10^4 time steps [see dashed line in Figs. 17(a) and 17(b)], slightly before the weights of continuous actor–critic RL became stable, to test the robot. It can be seen that the robot always moved toward the desired green object [see Fig. 17(c)] when the developed weights of ICO learning and continuous actor–critic RL were used. On the other hand, it sometimes moved to the undesired blue object or went straight when only the developed weights of continuous actor–critic RL were used [see Fig. 17(d)] because of exploration as well as a large weight w_{ϕ_B}. Note that ρ_{ϕ_B}

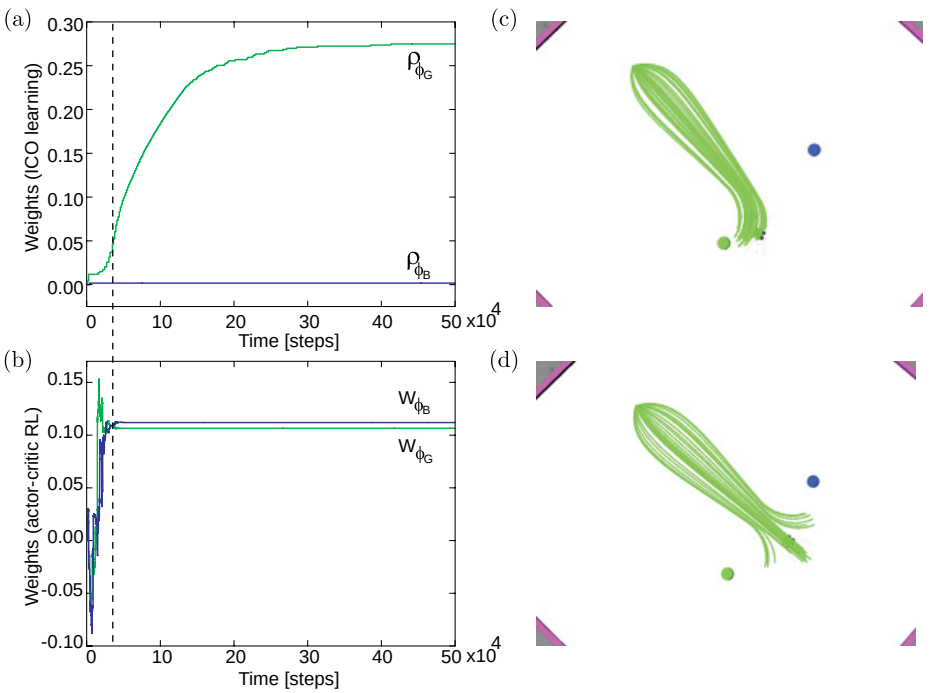


Fig. 17. (Color online) Learning curves and robot behaviors I. (a), (b) Weight changes of ICO learning and continuous actor–critic RL in the combinatorial learning framework. Dashed line shows the point (i.e., around 3×10^4 time steps or 300 s) where the weights of ICO leaning and continuous actor–critic RL were used to test the robot. (c) Trajectories of the robot from the starting position with different random orientations in the interval $[-45°, 45°]$. (d) The trajectories when ICO learning control policy was switched off, i.e., we set the weights of ICO learning to 0.0 while the weights of continuous actor–critic RL remained unchanged. Note that during the test we removed the exploration term from the controller in order to clearly see the trajectories.

did not become large since when the robot approached the blue object it did not deviate much from the object. Thus, the positive derivative of the reflex signal was smaller than threshold, thereby ρ_{ϕ_B} remained unchanged. This experimental result shows that ICO learning complemented continuous actor–critic RL leading to goal-directed behavior. In other words, ICO learning guided a learning strategy enabling continuous actor–critic RL to exploit more the positive reward. As a consequence, convergence finally occurred.

Figure 18 shows the second behavior II where we took the developed weights at around 20×10^4 time steps [see dashed line in Figs. 18(a) and 18(b)], slightly before the weights of continuous actor–critic RL reversed their growing directions, to test the robot. At this point, it can be seen that when the developed weights of ICO learning and continuous actor–critic RL were used the robot always approached the undesired blue object [see Fig. 18(c)] where the negative reward (-1) was given. Thus continuous actor–critic RL could use this reward signal to correct its current control policy. This effect can be observed from the weights of continuous

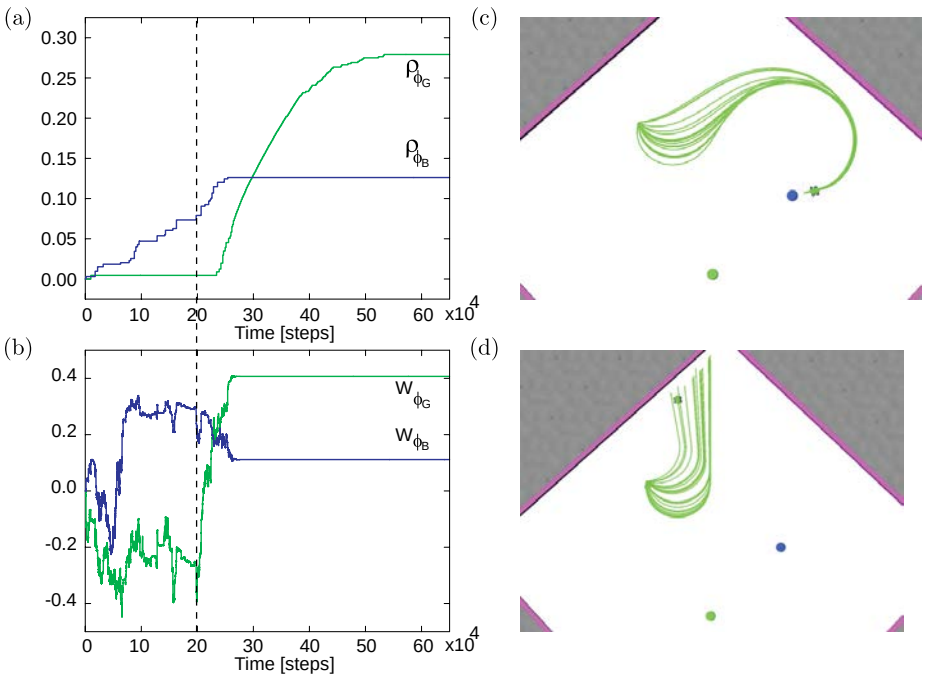


Fig. 18. (Color online) Learning curves and robot behaviors II. (a), (b) Weight changes of ICO learning and continuous actor–critic RL in the combinatorial learning framework. Dashed line shows the point (i.e., around 20×10^4 time steps or 2000 s) where the weights of ICO leaning and continuous actor–critic RL were used to test the robot. (c) Trajectories of the robot from the starting position with different random orientations in the interval $[-45°, 45°]$. (d) The trajectories when the ICO learning control policy was switched off, i.e., we set the weights of ICO learning to 0.0 while the weights of continuous actor–critic RL remained unchanged. Note that during the test we removed the exploration term from the controller in order to clearly see the trajectories.

actor–critic RL which significantly reversed their growing directions at around 20×10^4 time steps or 2000 s. On the other hand, when only the developed weights of continuous actor–critic RL were used, the robot always moved away from the objects [see Fig. 18(d)]. Therefore, in this situation continuous actor–critic RL had difficulty to obtain any reward signal to correct its current control policy. Due to the stochastic process employed, which tried to search for a successful control policy, the weights might change to a large degree [see transparent lines in Fig. 15(b)]. As a result, continuous actor–critic RL might fail to solve the task in a given number of trials (here, maximal 200 trials). This result suggests that ICO learning shaped or guided the learning strategy of continuous actor–critic RL such that it can receive a reward (i.e., here a negative one). Then it used this reward to correct the current control policy. As a consequence, convergence finally occurred.

Figure 19 shows the third behavior *III* where we took the developed weights at two states to test the robot. The early state was around 75×10^3 time steps where only the weights of continuous actor–critic RL became stable [see dashed line *State I* in Figs. 19(a) and 19(b)]. They were stable since the exploration term

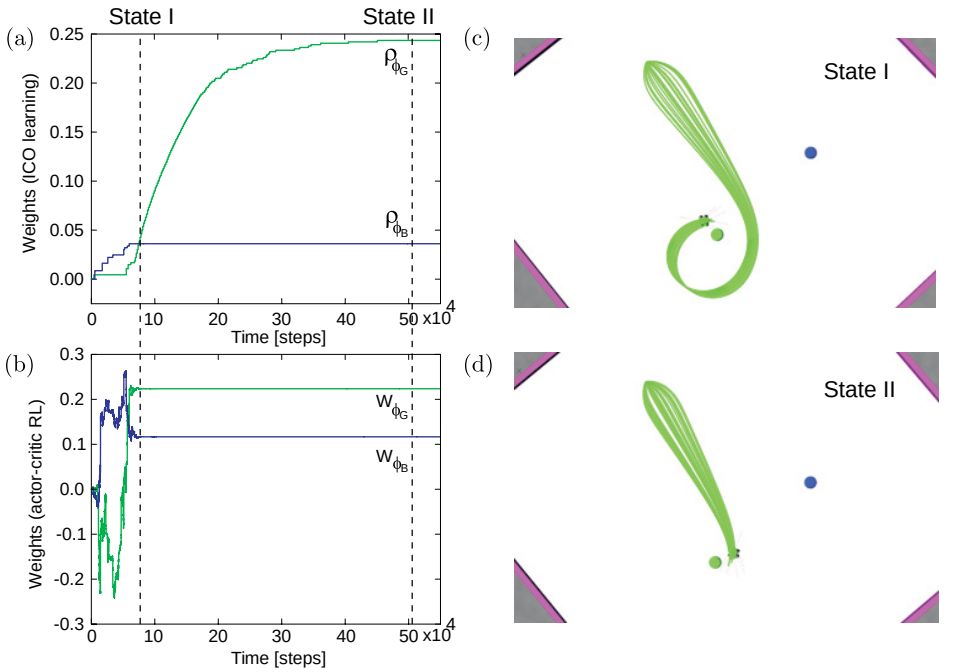
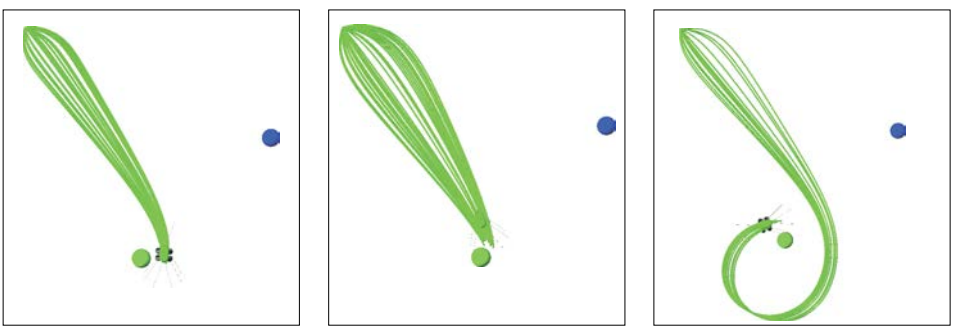


Fig. 19. (Color online) Learning curves and robot behaviors *III*. (a), (b) Weight changes of ICO learning and continuous actor–critic RL in the combinatorial learning framework. Dashed lines show two states where the weights of ICO leaning and continuous actor–critic RL were used to test the robot. State *I* was around 75×10^3 time steps or 750 s and state *II* was around 50×10^4 or 5000 s. (c) Trajectories of the robot from the starting position with different random orientations in the interval $[-45°, 45°]$ at state *I*. (d) The trajectories at state *II*. Note that during the test we removed the exploration term from the controller in order to clearly see the trajectories.

was zero. Recall that the exploration term was exponentially reduced as soon as the performance improved (i.e., the robot frequently approached the goal). The later state was around 50×10^4 time steps where the weights of ICO leaning became also stable [see dashed line *State II* in Figs. 19(a) and 19(b)]. It can be seen that the robot moved toward the goal in long trajectories [see Fig. 19(c)] when the control policy at the early state was used. In contrast, it moved on shorter trajectories when the control policy at the later state was used [see Fig. 19(d)]. This suggests that although continuous actor–critic RL was stopped due to the inhibition of its exploration term, ICO learning still shaped the control policy. This is because the reflex signal was not completely avoided since the robot still had large deviations to the goal when it came close to it. As a consequence, ICO learning improved robot performance by making it head directly to the goal, thereby leading to shorter trajectories.

It is important to note that although the resulting weights of the experiments shown in Figs. 15 and 17–19 converged to different values, they generally converged to almost the same weight ratio ($\frac{\rho_{\phi_G} + w_{\phi_G}}{\rho_{\phi_B} + w_{\phi_B}}$) of 2.9 ± 0.6. This shows that in combinatorial learning the combination of the weights of these two modules is necessary to successfully solve the task. Using all learned weights even yields a better result (i.e., the robot moved on the shortest trajectories) compared to using only the learned weights of either the ICO learning module or the continuous actor–critic RL module (see Fig. 20). In addition, the weight ratio also suggests that the positive reward attracts the system approximately three times larger than the negative reward repulses it.



(a) (b) (c)

Fig. 20. (Color online) Robot behaviors at three different control parameter setups of combinatorial learning. For our investigation here, we used the learned weights from the experiment shown in Fig. 19, i.e., the weights at State *II*. (a) All learned weights were used, i.e., $\rho_{\phi_G} \approx 0.245$, $\rho_{\phi_B} \approx 0.036$, $w_{\phi_G} \approx 0.224$, $w_{\phi_B} \approx 0.117$. (b) Only learned weights of the ICO learning module were used while the weights of the continuous actor–critic RL modules were set to 0.0. (c) Only learned weights of the continuous actor–critic RL module were used while the weights of the ICO learning module were set to 0.0. Note that in each test the robot started from the starting position with different random orientations in the interval $[-45°, 45°]$ and we removed the exploration term from the controller in order to clearly see the trajectories.

6. Conclusion

In the following, we will discuss some remaining issues while other relevant discussion points have been treated alongside the experimental section above.

In this study, we introduced a neural combinatorial learning model for policy improvement. The learning model combines ICO learning and continuous actor–critic RL in a parallel manner where the ICO learning output and the continuous actor–critic RL output are equally weighted to control the agent. The equal contribution used here is a simple and straightforward strategy for combining them [see Eq. (10)]. In general, ICO learning alone can quickly learn to solve tasks but has limitations for more difficult tasks (i.e., here, balancing the pole in critical initial conditions as well as goal-directed behavior). On the other hand, pure continuous actor–critic RL can often solve the tasks but learns slowly.

Mainly we found that the performance of the controller can be strongly improved when the combinatorial learning model was applied. To make this model work properly we need to design a proper reflex for ICO learning as well as an appropriate correlation between predictive and reflex signals. For example, in the pole balancing task we configured ICO learning such that the weights were changed only for the positive derivatives of the reflex signal, otherwise they remained unchanged. This is to avoid a negative correlation resulting in poor learning performance or even failure. Another condition which would make the current form of the model problematic is a strong conflict between the reward function and the reflex. Some problematic cases would be:

(i) The first case is if continuous actor–critic RL moves the agent toward a target due to the reward function, while ICO learning tries to move it away from it due to the built-in reflex.
(ii) Another case is robot navigation in an environment with obstacles when ICO learning is used to generate a negative tropism behavior (e.g., avoiding obstacles) while continuous actor–critic RL is used to generate a positive tropism behavior (e.g., approaching a goal). In this scenario, a conflict will occur when the goal is behind an obstacle or directly close to it.
(iii) The last case is a dynamic motion control task like balancing a humanoid robot (many degrees of freedom system) against an external disturbance (pushing) where ICO learning controls the robot to avoid pushing (e.g., leaning action) while continuous actor–critic RL wants to keep it balance (e.g., upright position).

However, these complex tasks might also be solved but this will require a modification of the model or an improvement by:

(i) Properly designing the reward function of continuous actor–critic RL and the reflex of ICO learning,
(ii) Using more appropriate sensory signals or predictive signals,

(iii) Transforming the low-dimensional input of the actor part into a higher-dimensional one by using nonlinear functions (e.g., an RBF network [46]) or using a decoder [7],
(iv) Using an adaptive critic network or another type of a critic network (e.g., a self-adaptive reservoir computing network [14] having high-dimensional nonlinear dynamics and internal memory) for a better approximation of the value function. These issues (i–iv) are still under investigation and go beyond the scope of this paper.

Besides our approach, there are a number of investigations on combining conventional RL with other (learning) mechanisms or applying other adaptive methods to it in order to enhance learning capability, reduce learning time, or counteract the curse of dimensionality. For example, Price and Boutilier [54] developed an imitation model called "implicit imitation" and integrated it into RL. It basically combines its own experience with its observations of the behavior of an expert mentor for learning. Doya [18] proposed hybrid RL based on the "actor-tutor" framework, which uses a model of the system dynamics as the tutor part. There, the actor (or controller) is trained by supervised learning to minimize the difference between its output and the tutor's output (desired output). This framework, basically resembling "feedback error learning" [23], was applied to nonlinear control tasks. Centina [12] introduced a supervised reinforcement learning (SRL) architecture for robot control problems with high-dimensional state spaces. There, a behavior model learned from examples is used to dynamically reduce the set of actions available from each state during the early RL process. In addition to these, other efforts have been made by developing advanced RL techniques [21, 31, 58] like hierarchical RL [3, 6, 9, 17, 66], by employing adaptive state representation [28, 34, 57], by using different exploration/exploitation techniques [4, 44, 60, 64], and by introducing algorithms for shaping rewards [2, 16, 48]. While all these advanced methods can successfully solve several (robot) tasks and are effective in their own right, they are quite difficult to match to biological neural learning and conditioning paradigms.

Only a few works have developed different types of learning models for robot control where the models mimic principles of these learning or conditioning mechanisms of animal leaning [1, 13, 65]. Alonso *et al.* [1] introduced the associative learning based approach (called the Pavlovian and Instrumental Q-learning framework) to deal with generalization in Q-learning. This approach improves RL (i.e., Q-learning) by applying the Rescorla-Wagner model [56] as a part of the control scheme for stimulus-stimulus associations. Its performance was tested in a grid simulator where agents have to approach or avoid appetitive and aversive stimuli. Chang and Gaudiano [13] presented a neural network based on operant and classical conditioning. It was tested on mobile robots. As a consequence, it allows the robots to simultaneously learn to approach light sources and avoid obstacles. Touretzky and Saksida [65] developed a model of operant conditioning that incorporates aspects of chaining in which behavioral routines are built up from smaller action segments.

The model was implemented on a mobile robot for solving the delay match to sample task (i.e., the task that involves behavioral sequences). Although all these learning models [1, 13, 65] employed learning and conditioning aspects of animal learning, they did not show or provided an understanding of how different mechanisms interact or complement each other, resulting in successful control policies. Instead, they were developed for improving conventional RL models or solving particular robotic tasks (i.e., goal-directed behavior control). Therefore, it is still unclear whether these models can also deal with qualitatively different tasks, like dynamic motion control.

Compared to many of these approaches just summarized, our combinatorial learning model applied the principles of classical and operant conditioning of animal learning. It was developed using ICO learning (a simplified model of classical conditioning) and continuous actor–critic RL (a simplified model of operant conditioning). They were implemented based on artificial neural networks; thereby, making them conceptually closer to biological systems compared to any other solution. Furthermore, ICO learning and continuous actor–critic RL are partially related to neural learning mechanisms in the brain. Specifically, ICO learning implements plain heterosynaptic plasticity associated with modulatory processes found in the brain [27] and continuous actor–critic RL uses TD learning which is related to dopaminergic responses in the brain. Especially, some cells in the substantia nigra and ventral tegmental area (VTA) show a behavior similar to representing the error of TD-learning [61], which we use for weight adaptation in actor and critic networks. We demonstrated the capability of this model in solving two different tasks: Pole balancing and goal-directed behavior control. This shows that the learning model is not limited to a specific task.

In addition to this, our model can be considered as a model-free method since its learning rules do not require a system or environment model. Instead, ICO learning requires only a built-in reflex as a self-supervised mechanism to quickly find the correlations between a state and an unwanted condition (i.e., reflex action), while continuous actor–critic RL uses its prediction mechanism including its own experiences and some exploration to obtain a good control policy. Although our models use a fixed state representation in the critic, one could also extend the critic to adaptive state partitioning [45] since the actor and critic are independently constructed. Our work also shares a connection to Kolter and Ng [33] where they presented a policy gradient method called the "Signed Derivative" approximation. The general concept of this approach is similar to our model in the sense that it is a model-free method which uses intuition to guess the direction where control inputs affect future state variables. This intuition is used to construct the signed derivative approximation which is directly applied to the update rule of RL. Generally speaking, the signed derivative approximation can be viewed as an instance of the built-in reflex of ICO learning. However, in our model the built-in reflex indirectly affects RL through ICO learning since it is used to guide and shape learning.

In summary, the study pursued here sharpens our understanding of how different learning mechanisms (i.e., correlation-based learning and RL) can be appropriately combined and how they complement each other leading to policy improvement. This study also suggests that correlation-based learning, on the one hand, can be used to speed up the learning process of RL. On the other hand, it can shape and correctly guide RL for searching an optimal policy. While the proposed combination of these two learning mechanisms can improve the performance of the systems, they are still combined in a simple way [see Eq. (10)]. Thus, for future work, we will investigate adaptive combinations. One possible option is to employ a learning mechanism based on a correlation between a direct reward signal and the outputs of ICO learning and continuous actor–critic RL for adapting their output weights. This way, an active output will have a high correlation with the reward signal, thereby strengthening its weight. The output weights will finally determine the behavior of the agent. Another option is to use a hierarchical RL framework [46] to find an optimal combination. Furthermore, we will also apply the combinatorial learning mechanism to more complex tasks, like the double-pendulum scenario [22, 29, 30], including ones with high dimensional states and actions (e.g., helicopter control [36] and octopus arm problems [69]). We also aim to use it as online learning for real robotic tasks, e.g., adaptive walking of hexapod robots [41], dynamic motion control of biped robots, and real robot navigation in complex environments. However, solving such tasks may require a modification of some components, e.g., using a nonlinear actor and/or an adaptive critic network, which could be easily done due to the modularity of the framework.

Acknowledgments

This research was supported by Emmy Noether grant MA4464/3-1 of the Deutsche Forschungsgemeinschaft (DFG), Bernstein Center for Computational Neuroscience II Göttingen (BCCN grant 01GQ1005A, project D1), Japan Society for the Promotion of Science (JSPS), European Communitys Seventh Framework Programme FP7/2007-2013 (Theme 3, Information and Communication Technologies) under grant agreement 270273, Xperience, and a part of this research was supported by "Brain Machine Interface Development" SRPBS, MEXT, MEXT KAKENHI 23120004 and Strategic International Cooperative Program, JST. P.M. would like to thank NICT for its support within the JAPAN TRUST International Research Cooperation Program. We thank Tomas Kulvicius for critical discussions and Martin Biehl and Frank Hesse for technical advice.

References

[1] Alonso, E., Mondragon, E. and Kjäll-Ohlsson, N., Pavlovian and instrumental Q-learning: A Rescorla and Wagner-based approach to generalization in Q-learning, in *Proc. Adaptation in Artificial and Biological Systems* (2006), pp. 23–29.

[2] Asmuth, J., Littman, M. L. and Zinkov, R., Potential-based shaping in model-based reinforcement learning, in *Proc. 23rd National Conf. Artificial Intelligence* (*AAAI'08*) (2008), pp. 604–609.
[3] Bakker, B. and Schmidhuber, J., Hierarchical reinforcement learning based on subgoal discovery and subpolicy specialization, in *Proc. 8th Conf. Intelligent Autonomous Systems* (2004), pp. 438–445.
[4] Banerjee, B. and Kraemer, L., Action discovery for single and multi-agent reinforcement learning, *Adv. Complex Syst.* **14** (2011) 279–305.
[5] Barnard, C., *Animal Behavior*: *Mechanism*, *Development*, *Function*, *and Evolution* (Pearson Education, 2004).
[6] Barto, A. G. and Mahadevan, S., Recent advances in hierarchical reinforcement learning, *Discrete Event Dyn. Syst.* **13** (2003) 41–77.
[7] Barto, A. G., Sutton, R. S. and Anderson, C. W., Neuron-like adaptive elements that can solve difficult learning control problems, *IEEE Trans. Syst. Man, Cybern.* **13** (1983) 834–846.
[8] Bekey, G., *Autonomous Robots From Biological Inspiration to Implementation and Control* (MIT Press, Cambridge, 2005).
[9] Botvinick, M. M., Niv, Y. and Barto, A. C., Hierarchically organized behavior and its neural foundations: A reinforcement learning perspective, *Cognition* **113** (2009) 262–280.
[10] Bovet, S., Robots with self-developing brains, Ph.D. thesis, University of Zurich (2007).
[11] Brembs, B. and Heisenberg, M., The operant and the classical in conditioned orientation in drosophila melanogaster at the flight simulator, *Learn. Memory* **7** (2000) 104–115.
[12] Cetina, V. U., Supervised reinforcement learning using behavior models, in *Proc. Sixth Int. Conf. Machine Learning and Applications* (*ICMLA 2007*) (2007), pp. 336–341.
[13] Chang, C. and Gaudiano, P., Application of biological learning theories to mobile robot avoidance and approach behaviors, *Adv. Complex Syst.* **1** (1998) 79–114.
[14] Dasgupta, S., Wörgötter, F. and Manoonpong, P., Information theoretic self-organised adaptation in reservoirs for temporal memory tasks, in *Proc. 13th Int. Conf. Engineering Applications of Neural Networks* (*EANN 2012*) (2012), pp. 31–40.
[15] Dayan, P. and Balleine, B., Reward, motivation, and reinforcement learning, *Neuron* **36** (2002) 285–298.
[16] Devlin, S., Kudenko, D. and Grzes, M., An empirical study of potential-based reward shaping and advice in complex, multi-agent systems, *Adv. Complex Syst.* **14** (2011) 251–278.
[17] Dietterich, T. G., Hierarchical reinforcement learning with the MAXQ value function decomposition, *J. Artif. Intell. Res.* **13** (2000) 227–303.
[18] Doya, K., Efficient nonlinear control with actor-tutor architecture, in *Advances in Neural Information Processing Systems* (1997), pp. 1012–1018.
[19] Doya, K., Reinforcement learning in continuous time and space, *Neural Comput.* **12** (2000) 219–245.
[20] Endo, G., Morimoto, J., Matsubara, T., Nakanish, J. and Cheng, G., Learning CPG-based biped locomotion with a policy gradient method: Application to a humanoid robot, *Int. J. Robot. Res.* **27** (2008) 213–228.
[21] Fischer, J., *A Modulatory Learning Rule for Neural Learning and Metalearning in Real World Robots with Many Degees of Freedom* (Shaker Verlag GmbH, 2003).

[22] Gomez, F., Schmidhuber, J. and Miikkulainen., R., Accelerated neural evolution through cooperatively coevolved synapses, *J. Mach. Lear. Res.* **9** (2008) 937–965.
[23] Gomi, H. and Kawato, M., Neural network control for a closed-loop system using feedback-error-learning, *Neural Netw.* **6** (1993) 933–946.
[24] Gullapalli, V., A stochastic reinforcement learning algorithm for learning real-valued functions, *Neural Netw.* **3** (1990) 671–692.
[25] Howery, L. D., Why do animals behave the way they do?, *Backyards and Beyond*: *Rural Living in Arizona* **3** (2007) 17–18.
[26] Hull, C. L., *A Behavior System*: *An Introduction to Behavior Theory Concerning the Individual Organism* (Yale University Press, New Haven, CT, 1952).
[27] Humeau, Y., Shaban, H., Bissiere, S. and Lüthi, A., Presynaptic induction of heterosynaptic associative plasticity in the mammalian brain, *Nature* **426** (2003) 841–845.
[28] Iida, S., Kuwayama, K., Kanoh, M., Kato, S. and Itoh, H., A dynamic allocation method of basis functions in reinforcement learning, in *Proc. 17th Australian Joint Conf. Artificial Intelligence* (2004), pp. 71–73.
[29] Kassahun, Y., de Gea, J., Edgington, M., Metzen, J. H. and Kirchner, F., Accelerating neuroevolutionary methods using a kalman filter, in *Proc. 10th Genetic and Evolutionary Computation Conf.* (*GECCO-2008*) (2008), pp. 1397–1404.
[30] Kassahun, Y., Wöhrle, H., Fabisch, A. and Tabie, M., Learning parameters of linear models in compressed parameter space, in *Proc. Artificial Neural Networks and Machine Learning* (*ICANN2012*) (2012), pp. 108–115.
[31] Kawarai, N. and Kobayashi, Y., Learning of whole arm manipulation with constraint of contact mode maintaining, *J. Robot. Mechatron.* **22** (2010) 542–550.
[32] Klopf, A. H., A neuronal model of classical conditioning, *Psychobiology* **16** (1988) 85–123.
[33] Kolter, J. and Ng, A., Policy search via the signed derivative, in *Proc. Robotics*: *Science and Systems* (*RSS*) (2009), pp. 27, Online.
[34] Kondo, T. and Ito, K., A reinforcement learning with adaptive state space recruitment strategy for real autonomous mobile robots, in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems* (2002), pp. 897–902.
[35] Konorski, J., *Integrative Activity of the Brain* (University of Chicago Press, Chicago, 1967).
[36] Koppejan, R. and Whiteson, S., Neuroevolutionary reinforcement learning for generalized helicopter control, in *GECCO 2009*: *Proc. Genetic and Evolutionary Computation Conf.* (2009), pp. 145–152.
[37] Kosco, B., Differential hebbian learning, in *Proc. Neural Networks for Computing*: *AIP*, Vol. 151 (1986), pp. 277–282.
[38] Lee, H., Shen, Y., Yu, C., Singh, G. and Ng, A., Quadruped robot obstacle negotiation via reinforcement learning, in *Proc. IEEE Int. Conf. Robotics and Automation* (2006), pp. 3003–3010.
[39] Lovibond, P. F., Facilitation of instrumental behavior by a pavlovian appetitive conditioned stimulus, *J. Exp. Psychol. Anim. B* **9** (1983) 225–247.
[40] Manoonpong, P., Geng, T., Kulvicius, T., Porr, B. and Wörgötter, F., Adaptive, fast walking in a biped robot under neuronal control and learning, *PLoS Comput. Biol.* **3** (2007) e134.
[41] Manoonpong, P., Parlitz, U. and Wörgötter, F., Neural control and adaptive neural forward models for insect-like, energy-efficient, and adaptable locomotion of walking machines, *Front. Neural Circuits* **7** (2013). Doi: 10.3389/fncir.2013.00012.

[42] Manoonpong, P. and Wörgötter, F., Adaptive sensor-driven neural control for learning in walking machines, in *Neural Information Processing*, *LNCS* (2009), pp. 47–55.
[43] Manoonpong, P., Wörgötter, F. and Morimoto, J., Extraction of reward-related feature space using correlation-based and reward-based learning methods, in *Neural Information Processing*, *LNCS* (2010), pp. 414–421.
[44] Morihiro, K., Matsui, N., and Nishimura, H., Effects of chaotic exploration on reinforcement maze learning, in *Knowledge Based Intelligent Information and Engineering Systems* (2004), pp. 833–839.
[45] Morimoto, J. and Doya, K., Reinforcement learning of dynamic motor sequence: Learning to stand up, in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems* (1998), pp. 1721–1726.
[46] Morimoto, J. and Doya, K., Acquisition of stand-up behavior by a real robot using hierarchical reinforcement learning, *Robot. Auton. Syst.* **36** (2001) 37–51.
[47] Mowrer, O., *Learning Theory and Behavior* (New York, Wiley, 1960).
[48] Ng, A. Y., Harada, D. and Russell, S. J., Policy invariance under reward transformations: Theory and application to reward shaping, in *Proc. 16th Int. Conf. Machine Learning* (1999), pp. 278–287.
[49] Pasemann, F., Evolving neurocontrollers for balancing an inverted pendulum, *Network-Comp. Neural* **9** (1998) 495–511.
[50] Pavlov, I., *Conditioned Reflexes* (Oxford University Press, Oxford, UK, 1927).
[51] Phon-Amnuaisuk, S., Learning cooperative behaviours in multiagent reinforcement learning, in *Neural Information Processing*, *LNCS* (2009), pp. 570–579.
[52] Porr, B. and Wörgötter, F., Strongly improved stability and faster convergence of temporal sequence learning by using input correlations only, *Neural Comput.* **18** (2006) 1380–1412.
[53] Porr, B. and Wörgötter, F., Fast heterosynaptic learning in a robot food retrieval task inspired by the limbic system, *Biosystems* **89** (2007) 294–299.
[54] Price, B. and Boutilier, C., Accelerating reinforcement learning through implicit imitation, *J. Artif. Intell. Res.* **19** (2003) 569–629.
[55] Rescorla, R. and Solomon, R., Two process learning theory: Relationship between pavlovian conditioning and instrumental learning, *Psychol. Rev.* **88** (1967) 151–182.
[56] Rescorla, R. and Wagner, A., A theory of pavlovian conditioning: Variations in the effectiveness of reinforcement and nonreinforcement, in *Classical Conditioning II*: *Current Research and Theory* (1972) 64–99.
[57] Sherstov, A. and Stone, P., Function approximation via tile coding: Automating parameter choice, in *Proc. Symp. Abstraction, Reformulation, and Approximation* (*SARA-05*) (2005), pp. 194-205.
[58] Shibata, K., Nishino, T. and Okabe, Y., Active perception and recoginition learning system based on Actor-Q architecture, *Syst. Comput. Jpn.* **33** (2002) 12–22.
[59] Skinner, B., *The Behavior of Organisms*: *An Experimental Analysis* (Appleton Century Croft, New York, 1938).
[60] Smith, S. C. and Herrmann, J. M., Homeokinetic reinforcement learning, in *First IAPR TC3 Workshop*, *PSL 2011*, *LNCS* (2012), pp. 82–91.
[61] Suri, R. E., Bargas, J. and Arbib, M. A., Modeling functions of striatal dopamine modulation in learning and planning, *Neuroscience* **103** (2001) 65–85.
[62] Sutton, R. and Barto, A., *Reinforcement Learning*: *An Introduction* (MIT Press, Cambridge, 1998).
[63] Thorndike, E., Animal intelligence: An experimental study of the associative process in animals, *Psychol. Rev. Monogr. Suppl.* **8** (1898) 68–72.

[64] Tokic, M., Adaptive ϵ-greedy exploration in reinforcement learning based on value differences, in *Proc. KI 2010*: *Advances in Artificial Intelligence* (2010), pp. 203–210.
[65] Touretzky, D. and Saksida, L., Operant conditioning in skinnerbots, *Adapt. Behav.* **5** (1997) 219–247.
[66] van Dijk, S. G. and Polani, D., Grounding subgoals in information transitions, in *Proc. IEEE Symp. Adaptive Dynamic Programming and Reinforcement Learning* (Paris, France, 2011), pp. 105–111.
[67] Watkins, C. J. C. H., Learning from delayed rewards, Ph.D. thesis, University of Cambridge (1989).
[68] Williams, D. and Williams, H., Auto — maintenance in the pigeon: Sustained pecking despite contingent non-reinforcement, *J. Exp. Anal. Behav.* **12** (1969) 511–520.
[69] Woolley, B. G. and Stanley, K. O., Evolving a single scalable controller for an octopus arm with a variable number of segments, in *Parallel Problem Solving from Nature*, *Lecture Notes in Computer Science* (2010), pp. 270–279.
[70] Wörgötter, F. and Porr, B., Temporal sequence learning, prediction and control — A review of different models and their relation to biological mechanisms, *Neural Comp.* **17** (2005) 245–319.