Multiagent Reinforcement Learning: Spiking and Nonspiking Agents in the Iterated Prisoner's Dilemma

Vassilis Vassiliades, Aristodemos Cleanthous, and Chris Christodoulou

***Abstract*— This paper investigates multiagent reinforcement learning (MARL) in a general-sum game where the payoffs' structure is such that the agents are required to exploit each other in a way that benefits all agents. The contradictory nature of these games makes their study in multiagent systems quite challenging. In particular, we investigate MARL with spiking and nonspiking agents in the Iterated Prisoner's Dilemma by exploring the conditions required to enhance its cooperative outcome. The spiking agents are neural networks with leaky integrate-and-fire neurons trained with two different learning algorithms: 1) reinforcement of stochastic synaptic transmission, or 2) reward-modulated spike-timing-dependent plasticity with eligibility trace. The nonspiking agents use a tabular representation and are trained with Q- and SARSA learning algorithms, with a novel reward transformation process also being applied to the Q-learning agents. According to the results, the cooperative outcome is enhanced by: 1) transformed internal reinforcement signals and a combination of a high learning rate and a low discount factor with an appropriate exploration schedule in the case of non-spiking agents, and 2) having longer eligibility trace time constant in the case of spiking agents. Moreover, it is shown that spiking and nonspiking agents have similar behavior and therefore they can equally well be used in a multiagent interaction setting. For training the spiking agents in the case where more than one output neuron competes for reinforcement, a novel and necessary modification that enhances competition is applied to the two learning algorithms utilized, in order to avoid a possible synaptic saturation. This is done by administering to the networks additional global reinforcement signals for every spike of the output neurons that were not "responsible" for the preceding decision.**

***Index Terms*— Multiagent reinforcement learning, Prisoner's Dilemma, reward transformation, spiking neural networks.**

I. Introduction

MULTIAGENT reinforcement learning (MARL) has recently attracted an influx of scientific work. Its problem lies in the dynamic environment created by the presence of more than one learning agent. Such an environment is affected by the actions of all agents, thus, for a system to perform well, the agents need to base their decisions on a history of joint past actions and on how they wish to influence future ones. In MARL, there could be different kinds of situations: fully competitive or adversarial (which could be modeled with zero-sum games); fully cooperative or coordinative (which could be modeled with team games); and a mixture of both (which could be modeled with general-sum games). Since different issues arise in each situation, many algorithms were proposed to address them.

Some of these algorithms are the following: 1) minimax-Q [1], which replaces the maximum value function with a function that calculates each player's best response and can be applied to two-player zero-sum games; 2) Nash-Q [2], which is an extension of minimax-Q to general-sum games; 3) Joint Action Learners (JAL) [3], which is an approach investigated in cooperative settings in which the agents maintain some beliefs about the strategies of the other agents and therefore learn joint action values; 4) Friend-or-Foe Q (FoF-Q) [4], which can be interpreted as consisting of two algorithms, Friend-Q, which is suited for coordination games, and Foe-Q for zero-sum games and is equivalent to minimax-Q; 5) Win or Learn Fast-Policy Hill Climbing (WoLF-PHC) [5], which uses an extension of Q-learning [6] to play mixed strategies based on the WoLF principle that uses a higher learning rate when the agent is losing and a lower one when it is winning; 6) WoLF-Infinitesimal Gradient Ascent (WoLF-IGA) [7], [8], which combines gradient ascent with an infinitesimal step size (IGA) [9] with the WoLF method; 7) Correlated Equilibria Q (CE-Q) [10], which learns correlated equilibrium policies and can be thought of as a generalization of Nash-Q and FoF-Q; 8) Frequency Maximum Q (FMQ) [11], which is a heuristic method proposed for coordination in heterogeneous environments and is based on the frequency with which actions yielded the maximum corresponding rewards in the past; 9) Generalized IGA-WoLF (GIGA-WoLF) [12], which achieves convergence and no regret (i.e., the algorithm performs as good as the best static strategy); 10) Adapt When Everybody is Stationary Otherwise Move to Equilibrium (AWESOME) [13], which is an algorithm that is guaranteed to converge to a Nash equilibrium in self-play and learns to play optimally against stationary opponents in games with an arbitrary number of players and actions; 11) Weighted Policy Learning (WPL) [14], which assumes that an agent neither knows the underlying game nor observes other agents, and achieves convergence in benchmark two-player-two-action games; and 12) Max or MiniMax Q (M-Qubed) [15], which is a robust algorithm that was shown to achieve high degrees of

Manuscript received January 19, 2010; revised July 12, 2010 and November 22, 2010; accepted January 16, 2011. Date of publication March 18, 2011; date of current version April 6, 2011. This work was supported in part by the Cyprus Research Promotion Foundation and the European Union Structural Funds under Grant PENEK/ENISX/0308/82, and the University of Cyprus under an Internal Research Project Grant.

The authors are with the Department of Computer Science, University of Cyprus, Nicosia 1678, Cyprus (e-mail: v.vassiliades@cs.ucy.ac.cy; aris@cs.ucy.ac.cy; cchrist@cs.ucy.ac.cy).

Digital Object Identifier 10.1109/TNN.2011.2111384

coordination and cooperation in several two-player repeated general-sum games in homogeneous and heterogeneous settings. For a comprehensive coverage of MARL algorithms, see [16] and references therein.

Much work is focused in deriving theoretical guarantees, based on different sorts of criteria such as rationality and convergence [5], targeted optimality, safety and auto-compatibility [17], or security, coordination, and cooperation against a wide range of agents [15]. Theoretical work has also been done in analyzing the dynamics of multiagent learning. For instance, Tuyls *et al.* [18] investigated MARL from an evolutionary dynamical perspective, while Iwata *et al.* [19] approached the field from a statistical and an information-theoretical perspective. Since the problem is not very well defined, Shoham *et al.* [20] attempted to classify the existing work by identifying five distinct research agendas. They argued that when researchers design algorithms, they need to place their work under one of these categories which are: 1) computational, which is aimed in designing learning algorithms that iteratively compute properties of games; 2) descriptive, which is to determine how natural agents (such as humans, animals, or populations) learn in the context of other learners and make decisions; 3) normative, in which the learning algorithms give a means to determine which sets of learning rules are in equilibrium with one another; 4) prescriptive cooperative, which is how to design learning algorithms in order to achieve distributed control; and 5) prescriptive noncooperative, which is how to design effective algorithms, or how agents should act to obtain high rewards, for a given environment, i.e., in the presence of other (intelligent) agents. Subsequently, some work did focus on specific agendas (e.g., [21]), but more agendas were proposed [22]. In addition, the original agendas [20] have been criticized as being not distinct, since they may complement each other [23], [24]. Stone [25] extended the criticism by arguing that the game-theoretic approach is not appropriate in complex multiagent problems. Despite these criticisms, our study lies in the original prescriptive noncooperative agenda [20]. For training the nonspiking agents, we use two simple algorithms from single-agent reinforcement learning (RL), namely Q-learning [6] and State-Action-Reward-State-Action (SARSA) [26].

Reinforcement learning [27] has successfully been applied to spiking neural networks (NNs) in recent years. These schemes achieve learning by utilizing various biological properties of neurons, whether they are neurotransmitter release [28], spike timing [29], or firing irregularity [30]. Their degree of experimental justification varies and they need to be further assessed, nevertheless, all these methods are biologically plausible and constitute the basis of successful RL applications on biologically realistic neural models. A popular implementation of RL on spiking NNs is achieved by modulating spike-timing-dependent synaptic plasticity (STDP) with a reward signal [29], [31]– [33]. STDP is the change in synaptic efficacy that occurs according to the relative timing of pre- and postsynaptic spikes and has been experimentally observed [34]– [36]. Other examples of RL on spiking NNs include Seung's reinforcement of stochastic synaptic transmission [28] as well as reinforcement of irregular spiking [30], where the learning rules perform stochastic gradient ascent on the expected reward by correlating the neurotransmitter release probability and the fluctuations in irregular spiking, respectively, with a reward signal. Other policy gradient methods, such as the latter, that are applied to spiking NNs resulting in spike-based formulation of reward-based learning include [37]– [39]. Moreover, in another study, a spiking NN implements an actor-critic TD learning agent [40], while in [41] a spiking NN architecture with local Hebbian spike-timing learning and RL schemes is implemented by a field-programmable gate array chip for cardiac resynchronization therapy. In addition, in [42], classical conditioning [43] is combined with STDP and applied to navigation control. Some of these algorithms were shown to be able to solve simple tasks like the XOR problem [28], [29]. In addition, reward-modulated STDP could learn arbitrary spike patterns [31] or precise spike patterns [33] as well as temporal pattern discrimination [33], and could be used in simple credit assignment tasks [32]. For training the spiking NNs in this paper, we use reinforcement of stochastic synaptic transmission [28] as well as reward-modulated STDP with eligibility trace [29]. To the best of our knowledge, we are the first group employing spiking NNs with biologically plausible learning schemes in a challenging multiagent game-theoretical situation.

It has to be pointed out that the NN agents in our multiagent system are represented by entire NNs and not by single neurons as in the multiagent modeling in [44], where the NN is interpreted as a model of interaction between a large number of decision makers (neurons representing the real market agents' buying/selling decisions) and serves as a model of the market process.

This paper investigates cooperation between self-seeking reward agents in a noncooperative setting. This situation can be modeled with the Iterated Prisoner's Dilemma (IPD) which is a general-sum game. Although the cooperative (CC) outcome is a valid equilibrium of the IPD, our study does not aim to assess the strength of the learning algorithms to attain equilibria of the game or best responses to any given strategy. Instead, we focus on mutual cooperation and investigate whether it can be achieved by spiking and simple nonspiking agents trained with RL and attempt to compare them. It is very interesting and beneficial to understand how and when cooperation is achieved in the IPD's competitive and contradictive environment, as it would then become possible to prescribe optimality in real-life interactions through cooperation, analogous to the IPD. In its standard one-shot version, the Prisoner's Dilemma (PD) [45] is a game summarized by the payoff matrix of Table I. There are two players, Row and Column. Each player has the choice of either to "Cooperate" (C) or "Defect" (D). For each pair of choices, the payoffs are displayed in the respective cell of the payoff matrix of Table I. In game-theoretical terms, where rational players are assumed, DD is the only Nash equilibrium outcome [46] (i.e., a state in which no player can benefit by changing his/her strategy while the other players keep theirs unchanged), whereas the cooperative (CC) outcome satisfies Pareto optimality [47] (i.e., a state in which it is impossible to increase the gains of one player without increasing the losses of other players). The "dilemma" faced by the players

TABLE I

PAYOFF MATRIX OF THE PRISONER'S DILEMMA GAME WITH THE VALUES USED IN OUR EXPERIMENTS. PAYOFF FOR THE ROW PLAYER IS SHOWN FIRST. R IS THE "REWARD" FOR MUTUAL COOPERATION. P IS THE "PUNISHMENT" FOR MUTUAL DEFECTION. T IS THE "TEMPTATION" FOR UNILATERAL DEFECTION AND S IS THE "SUCKER'S" PAYOFF FOR UNILATERAL COOPERATION. THE ONLY CONDITION IMPOSED TO THE PAYOFFS IS THAT THEY SHOULD BE ORDERED SUCH THAT T>R>P>S

	Cooperate (C)	Defect (D)
Cooperate (C)	R(= 4), R(= 4)	S(= –3), T(= 5)
Defect (D)	T(= 5), S(= −3)	P(= −2), P(= 2)

in any valid payoff structure is that, whatever the other does, each one of them is better off by defecting than cooperating. The outcome obtained when both players defect, however, is worse for each one of them than the outcome they would have obtained if both had cooperated. In the IPD, an extra rule (2R>T+S) guarantees that the players are not collectively better off by having each player alternate between C and D, thus keeping the CC outcome Pareto-optimal. Moreover, contrary to the one shot game, CC can be a Nash equilibrium in the infinite version of the IPD.

As pointed out in [48], "perfectly predicting the environment is not enough to guarantee good performance," because the performance depends partly on properties of the environment. In our case, we believe that the property of the environment which plays a significant role in the CC outcome is the reward function (i.e., the payoff matrix), since it specifies the type and strength of the reinforcement the agents receive. Agents that rely only on predesigned reward functions in order to be trained might not truly be called adaptive and autonomous, because they can only cope with environment types to which these functions apply. Snel and Hayes [49] investigated the evolution of valence systems (i.e., systems that evaluate positive and negative nature of events) in an environment largely based on the artificial life world by Ackley and Littman [50]. They compared the performance of motivational systems that are based on internal drive levels versus systems that are based purely on external sensory input and showed that the performance of the former is significantly better than that of the latter.

Moreover, an elaborated view of the agent–environment interaction [51], [52] splits the environment into the external environment and an internal one, the latter being considered to be part of the agent. The external environment provides the sensations to the agent and receives its actions, while the internal environment provides the states and rewards to the agent, since it contains the critic and receives the agent's decisions. Some other line of work shows that by discriminating external rewards from "utilities" (i.e., internal stimuli elicited by rewards) and using utilities for learning in the IPD, cooperation can be facilitated [53].

Inspired by an extension of the WoLF-PHC algorithm that manipulates its own payoffs showing that it is possible to transform the game so that higher payoffs can be accumulated [54], we made a preliminary experimentation showing that for the nonspiking agents it is beneficial to mix positive and negative values [55], instead of using the most commonly studied payoff values (see [56], and all chapters in the edited book by Kendall *et al.* [57]). The payoff values used in both spiking and nonspiking simulations are shown in Table I.

Based on the above, in a separate investigation [58], we adopted the idea of splitting the rewards coming from an external environment from the utilities [51], [53] that are a transformation of the rewards generated by the internal environment of the agent. More specifically, we investigated (in [58]) how the internal environment could transform the external values coming from the payoff matrix to more appropriate rewards and punishments that motivate the agents to cooperate. In addition, in the same work [58] we describe how an evolutionary algorithm could be used to find these internal reward values that still satisfy the constraints of the IPD, so that simple Q-learners can rapidly reach the CC outcome in self-play. In this paper (see Sec. II.A), we apply a reward transformation process to the nonspiking Q-learning agents, so we use these internal reward values we found in [58]). Moreover, a transformation of the external payoff values to internal ones has been employed by our spiking agents as well (see Section II-B).

The remainder of this paper is organized as follows. Section II describes our methodology for both spiking and nonspiking simulations. The results are presented and analyzed in Section III, while the conclusions are given in the last section. A very preliminary version of this paper has been presented in a conference [55].

II. METHODOLOGY

A. Nonspiking Agents

Generally, there are two approaches to RL: 1) value-function methods (e.g., [6], [26], [59]), which build long-term utilities of decisions that induce a control policy, and 2) policy-search methods (e.g., [60]– [64]), which directly optimize the parameters of the policy with respect to the long-term cumulative reward. Our nonspiking agents implement simple value-function reinforcement learning algorithms and, more specifically, Q-learning [6] and SARSA [26].

The value function is stored in a lookup table since there is no need for function approximation in our simple scenario, where the state-action space is small. Moreover, as Sandholm and Crites [65] point out, lookup tables yield better results and are faster than simple recurrent NNs in the IPD.

In all simulations, the agents are provided only with incomplete information: they only receive the state of the environment, i.e., the actions of both the agent and the opponent in the previous round, but not the payoffs associated with their opponent's action. A Boltzmann exploration schedule is utilized, as it gives a good balance between exploration and exploitation. More specifically, an action a_i is selected from state s with probability $p(a_i)$ given by

$$p(a_i) = \frac{e^{Q(s,a_i)/t}}{\sum\limits_{a \in \{C,D\}} e^{Q(s,a)/t}} \tag{1}$$

where $Q(s, a_i)$ is the value of state s and action a_i, and the temperature t is given by $t = 1 + 10 \times 0.99^n$, with n being the number of games played so far. The constants 1, 10, and 0.99 are chosen empirically.

Two types of simulations were performed: 1) between a learning and a nonlearning agent, and 2) between two learning agents.

1) Learning Agents Against Nonlearning Opponents: These initial simulations compare three learning agents against opponents that do not use any learning algorithm (single-agent RL). Our learning agents are the following: a) "Q" (for Q-learning); b) "SARSA"; and c) "RTQ" (for reward transformation Q-learning), i.e., an agent that transforms the external payoff values to different internal rewards. These internal rewards were found by evolutionary optimization with the purpose of increasing mutual cooperation between two Q agents, while still satisfying the constraints of the IPD [58]. Their values are the following: $T' = 3.81$, $R' = 1.76$, $P' = -44.82$, and $S' = -46.32$ (details of the method used to determine these values can be found in [58]). The nonlearning opponents are named after the strategies they use: "Only-Cooperate" selects only the Cooperate (C) action; "Only-Defect" selects only the Defect (D) action; "Random(p)" selects a random action with a specified probability of cooperation p; "Tit-for-Tat" (TFT) [66] starts with action C and afterwards repeats its opponent's previous action; and "Pavlov" (also known as "Win-Stay, Lose-Shift") [67] changes its actions only if the two lowest payoffs, i.e., S and P, were received.

2) Learning Agents Against Learning Opponents: The latter simulations compare two learning agents (i.e., Q, SARSA, and RTQ) against each other (MARL). The multiagent system can either be homogeneous or heterogeneous where: a) a homogeneous system is when both agents employ the same learning algorithm, with the same or similar parameters, and b) a heterogeneous system is when both agents employ the same algorithm with dissimilar parameters, or different algorithms. The parameters are the step size, α, and the discount factor γ.

B. Spiking Agents

The game simulation is repeated with the two players implemented by two spiking NNs. The networks' architecture is depicted in Fig. 1. Each network has a hidden layer of 60 leaky integrate-and-fire (LIF) neurons [68], [69] and an output layer of 2 LIF neurons. Choosing the right spiking neuron model when building a spiking NN is extremely important [70]. In our case, given the complexity of our spiking NN system, the LIF neuron model was chosen as the basic node of each spiking NN because of its simplicity and computational effectiveness compared to the more biologically detailed conductance-based models like the Hodgkin and Huxley model [71] or even spiking neuron models of intermediate complexity such as the Izhikevich model [72] (used in a spiking network model by Arena *et al.* [42]), the model proposed by Christodoulou *et al.* [73], or the McGregor model [74], [75] (used in a network of spiking neurons by Lin *et al.* [76] and by Swiercz *et al.* [77]).

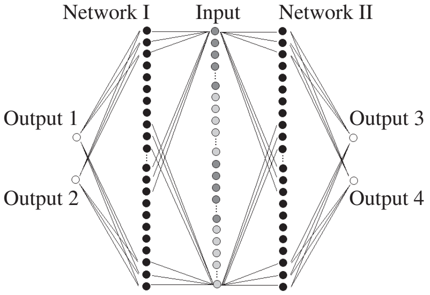

Fig. 1. Two spiking NNs competing in the IPD. Two individual networks with multilayer perceptron-type architecture receive a common input of 60 neurons, depicted in the middle of the figure. Each network (left and right) has two layers that make full feedforward connections between three layers of neurons; the 60 shared input neurons, 60 LIF hidden neurons, and 2 LIF output neurons. The networks have full connectivity, though only some connections are shown for clarity. Neurons are randomly chosen to be either excitatory or inhibitory. The two networks simulate the corresponding two players of the game.

Learning is implemented through reinforcement of stochastic synaptic transmission [28] as well as through reward-modulated STDP with eligibility trace [29]. Although these optimization algorithms are not experimentally proven, both synaptic facilitation (which can be enabled by synaptic transmission) and STDP are biological processes that were shown to be related with prediction tasks [78] and thus could be used by the brain for the purposes of optimization. Two different sets of simulations were carried out, one for each learning scheme. The equation and values of the parameters used for modeling the LIF neurons, for each simulation set, as well as the learning equations, are the same as the ones used in [28] and [29] (given also in the Appendix), apart from the value of the mean weight of the conductance used for the excitatory synapses in reinforcement of stochastic synaptic transmission [28], which is now set to 14 nS.

In reinforcement of stochastic synaptic transmission Seung [28] makes the hypothesis that microscopic randomness is harnessed by the brain for the purposes of learning (see Appendix A for details). The model of the hedonistic synapse is developed along this hypothesis. Briefly, within the framework of the model, each synapse acts as an agent who pursues reward maximization through the actions of releasing or not a neurotransmitter. The algorithm includes a dynamical variable of eligibility trace [79], which signifies when a synapse is eligible for reinforcement by keeping a record of the synapse's recent actions with respect to neurotransmitter release. Synapses effectively learn by computing a stochastic approximation to the gradient of average reward. Moreover, if each synapse behaves hedonistically, then the network as a whole behaves hedonistically, pursuing reward maximization. The learning rate is set equal to 0.1.

In reward-modulated STDP with eligibility trace [29] the modulation of standard antisymmetric STDP with a reward signal leads to RL (see Appendix B for details). The synaptic efficacies exhibit Hebbian STDP when the network is rewarded and anti-Hebbian when punished, allowing the network to associate an output to a given input only when accompanied by a positive reward and disassociate one when accompanied by a punishment, permitting thus the exploration of better strategies. Moreover (as in Seung's [28] algorithm) it involves a biologically plausible variable, the eligibility trace [79], that serves as a decaying memory of the relation between recent pre- and postsynaptic spike pairs. The eligibility trace signifies when a pair of spikes is eligible for reinforcement. The learning rate used in the respective simulations is equal to 0.7×10^{-4}, while a single game of IPD has 200 rounds and the results recorded are averaged over 10 games.

Both algorithms are derived as an application of the online partially observable Markov decision process reinforcement learning algorithm [62] and also keep a record of the agents' recent actions through the eligibility trace. In reinforcement of stochastic synaptic transmission, the synaptic connection strengths are constant, the agent is regarded to be the synapse itself that acts by releasing a neurotransmitter vesicle, and the parameter that is optimized is one that regulates the release of the vesicle. On the other hand, in reward-modulated STDP, the agent is regarded to be the neuron that acts by spiking and the parameter that is optimized is its synaptic connection strengths.

The networks learn simultaneously but separately where each network seeks to maximize its own accumulated reward. The game is simulated through an iterative procedure which starts with a decision by the artificial agents, continues by feeding this information to the agents, during which learning takes place, and ends by a new decision. The agents take their first decision randomly. During each learning round, the input to the system is presented for 500 ms and encodes the decisions the two networks had at the previous round. This means that after round k, the outcome of the game (at round k) is fed into the system for 500 ms and the synapses are changed according to it. Each network's decision is encoded in the input, by the firing rate of two groups of Poisson spike trains. The first group will fire at 40 Hz if the network cooperated and at 0 Hz otherwise. The second group will fire at 40 Hz if the network defected and at 0 Hz otherwise. Consequently, four groups of Poisson spike trains provide the system's input, with two groups always being active, preserving thus a balance at the output neurons' firing rates at the beginning of learning. Any significant difference in the output neurons' firing rate at any time should only be induced by learning and not by the differences of the driving input firing rates. At the end of each learning round, the networks decide whether to cooperate or defect for the game's next round, according to the value each network assigns to the two actions. These values are reflected by the output neurons' firing rates at the end of each learning round. The cooperation value for network I and II is taken to be proportional to the firing rate of output neurons 1 and 3, respectively. Similarly, the defection value for network I and II is taken to be proportional to the firing rate of output neurons 2 and 4, respectively. At the end of each learning round, the firing rates of the competing output neurons are compared, for each network separately, and the decisions are drawn. When the two networks decide their play for the next round of the IPD, they each receive a distinct payoff given their actions and according to the game's payoff matrix (see Table I). This same payoff is also the global reinforcement signal (scaled down) that will train each network during the next learning round and thus guide the networks to their next decisions. The payoffs are scaled down when administered as reinforcements to the networks in order to incorporate the distinction between signals given by the environment and how these signals are internally processed (as in [51]). The modeling of this process makes our spiking agents similar to our nonspiking RTQ agents (see Section II-A). The scaled-down payoffs combined with a small learning rate ensure in addition that changes on the variables controlled by learning are made in a smooth and gradual way. For example, if the outcome of the agents was a CD, then according to the payoff matrix network I should receive a payoff of -3 for cooperating and network II a payoff of $+5$ for defecting. As stated, the reinforcement signal is specified according to the aggregate activation of the output units at the end of a learning round since the decision of the agents whether to cooperate or defect depends on the aggregate relative activation of each network's output units. This reinforcement is constant in value during the next 500 ms of learning (and is different from 0) and is applied in the time step following the spikes of the output neurons, as prescribed by the original learning algorithms [28], [29]. In addition, each network is reinforced for every spike of their output neuron that was "responsible" for the decision at the last round and therefore for the payoff received. Hence in the CD case, network I would receive a constant penalty of -3 (scaled down to -1.3) for every spike of output neuron 1 (remember that the firing rate of output neuron 1 reflects the value that network I has for the action of cooperation) and network II would receive a constant reward of $+5$ (scaled down to 1.5) for every spike of output neuron 4 (remember that the firing rate of output neuron 4 reflects the value that network II has for the action of defection). Since the learning algorithms work with positive and negative reinforcements that are directly applied to the synapses and are extracted from the payoff matrix, it is then necessary that the payoff matrix contains both positive and negative values. The networks thus learn through global reinforcement signals which strengthen the value of an action that elicited a reward and weaken the value of an action that resulted in a penalty.

III. Results and Discussion

A. Nonspiking Agents

The games were run for 50 trials with 1000 rounds per trial. All combinations of agents were tested with α and γ taking the values 0.1 or 0.9 (i.e., slow/fast learning, weak/strong discounting).

1) Learning Agents Against Nonlearning Opponents: As described in Section II-A, the first simulations are between three learning agents (i.e., Q, SARSA, and RTQ) and opponents that do not use any learning algorithms (i.e., OnlyCooperate, OnlyDefect, Random, TFT, and Pavlov).

Table II shows the best results of the simulations between the three agents and the nonlearning opponents. In terms of performance, they are ranked based on the percentage of: 1) DC in the case of OnlyCooperate opponent, since the agent needs to learn to play D in order to exploit its opponent's weakness and accumulate more reward; 2) DD in the case of OnlyDefect, as the agent needs to learn to play D so that it does not get exploited by the opponent; 3) CC in the cases of TFT and Pavlov, since CC can be attained against such reactive

TABLE II

RESULTS WITH NONSPIKING LEARNING AGENTS AGAINST NONLEARNING OPPONENTS. THE HIGHEST RANKING PERFORMANCE FOR EACH LEARNING AGENT IS SHOWN IN BOLD (SEE TEXT FOR EXPLANATION OF THE BASIS OF THE RANKINGS). HIGHEST CC IS ACHIEVED WHEN AN RTQ AGENT COMPETES AGAINST A TFT OR A PAVLOV OPPONENT

Learning agent			**Nonlearning**	**Outcome (%)**			
	α	γ	**Agent**	CC	CD	DC	DD
RTQ	0.1	0.9	OnlyCooperate	6	0	**94**	0
SARSA	0.1	0.9		15	0	85	0
Q	0.1	0.9		20	0	80	0
RTQ	0.9	0.1	OnlyDefect	0	25	0	**75**
SARSA	0.9	0.9		0	25	0	**75**
Q	0.9	0.1		0	32	0	68
RTQ	0.9	0.9	TitForTat	**99**	0.5	0.5	0
SARSA	0.9	0.9		95	2	2	1
Q	0.9	0.9		93	3	3	1
RTQ	0.9	0.9	Pavlov	**99**	0	0.5	0.5
SARSA	0.9	0.9		93	1	3	3
Q	0.9	0.9		91	1	4	4
SARSA	0.9	0.9	Random($p = 0.25$)	5	14	20	**61**
Q	0.9	0.9		6	19	19	56
RTQ	0.9	0.1		6	20	18	56
Q	0.1	0.9	Random($p = 0.50$)	8	8	**42**	**42**
SARSA	0.1	0.9		8	9	42	41
RTQ	0.9	0.1		14	13	37	36
Q	0.1	0.9	Random($p = 0.75$)	14	4	**62**	20
SARSA	0.1	0.9		14	5	61	20
RTQ	0.9	0.1		26	9	49	16

opponents; 4) DD in the case of Random($p = 0.25$), since this agent is similar to the OnlyDefect agent with the difference being that it cooperates with a small probability; 5) DC and DD in the case of Random($p = 0.50$), as this agent cooperates half of the time; and 6) DC in the case of Random($p = 0.75$), since this agent is similar to the OnlyCooperate agent with the difference that it defects with a small probability.

The results show that the RTQ agent achieves the DC outcome faster than SARSA and Q, when playing against OnlyCooperate, which is most probably due to the fact that the rewards it uses (T' and R') are smaller than the corresponding payoff values. The difference between SARSA and Q is negligible. When playing against OnlyDefect, RTQ and SARSA achieve 75% DD and Q achieves only 68%. As this is a simple deterministic bandit problem (where the agents are facing the problem of selecting C or D and receiving the rewards S or P), we would have expected the agents to play D more. The results shown in Table II were obtained with the exploration method described in Section II-A, which was chosen to illustrate rapid convergence to the CC outcome in the MARL case and more specifically, for RTQ in self-play, as shown in Section III-A.2. Simulations with ϵ-greedy exploration (with $\epsilon = 0.1$) (not shown) demonstrated that against the OnlyDefect opponent, all three agents reach the DD outcome nearly 95% of the time with $\alpha = 0.9$ and $\gamma = 0.1$, for the same number of trials and rounds, the remaining 5% can be attributed to exploration. When the opponent was a TFT agent, RTQ achieves CC 99% of the time, followed by SARSA and then Q, where CC attained 95% and 93%, respectively. Similar results are observed when the opponent is Pavlov, where RTQ manages to achieve CC 99% of the time, but SARSA 93% and Q 91%. It is worth noting that for both TFT and Pavlov, when the discount factor of the learning agents was low (i.e., $\gamma = 0.1$), CC percentage was 70−76% for RTQ and 18−25% for both Q and SARSA (results not shown). This indicates that a high discount factor is required to reach and maintain mutual cooperation. Against the Random($p = 0.25$) opponent (i.e., the agent that chooses C with probability 0.25), SARSA achieves DD 61% of the time, while Q and RTQ 56%. As with the case of the OnlyDefect opponent, results with ϵ-greedy exploration (with $\epsilon = 0.1$) (not shown) indicate that the learning agents play D more often than in the case of softmax exploration, since the DD outcome occurs 69% of the time with Q and SARSA and 67% of the time with RTQ ($\alpha = 0.9$ and $\gamma = 0.1$ for all agents, and the number of trials and rounds was kept the same as in all simulations). As the Random agent becomes more cooperative ($p = 0.50$, $p = 0.75$), the DD outcome diminishes, while the DC outcome rises, for all learning agents.

2) Learning Agents Against Learning Opponents: In the previous section, the problem was effectively single-agent RL, as the opponents did not use any learning algorithms. This section deals with the MARL problem. More specifically, the three learning agents compete against each other and themselves under different parameter settings. Fig. 2 depicts the results of the RTQ agent against itself, ranked by the percentage of CC. The system is homogeneous (i.e., both agents use the same parameters) only in the 5th, 7th, 8th, and 10th cases. Interestingly, the results illustrate that CC is highest (i.e., 97% in the 10th case) when both agents learn fast ($\alpha = 0.9$) and use weak discounting ($\gamma = 0.1$). It has to be noted that at the 200th round (note that the spiking simulations run only for 200 rounds, see Section III-B) the percentage of CC was already very high, i.e., 89%, and the accumulated reward of the system, i.e., the sum of rewards of the agents, was 1401.52, while the theoretical maximum that corresponds to playing CC all the time (thus receiving $R + R = 4 + 4 = 8$ for each round) is 1600 (200×8).

Although not shown here, convergence to the CC outcome was observed only for the best four cases (cases 7−10 of Fig. 2), and was measured with two metrics. More specifically, for every round we calculate: a) the deviation of the reward the system receives (i.e., the sum of the agents' rewards) from the "desired" one that corresponds to the CC behavior (and is equal to 8 according to Table I), and b) the rate of change of the Q-values, whose calculation involves normalization to accommodate for the fact that the scale of rewards of the RTQ agent is different from the one of the Q and SARSA agents.

Slower learning, by at least one of the agents, seems to decrease the percentage of the CC outcome because the agents might need more time to converge. This is clearly illustrated in Fig. 2, since the best three results (cases 8–10) are obtained when both agents learn fast. The only exception is observed between the 6th and 7th cases: when both agents use smaller learning rates and discount factors, the performance of the system is better (i.e., 89% in the 7th case) than when the first agent switches to a higher learning rate (i.e., 86% in

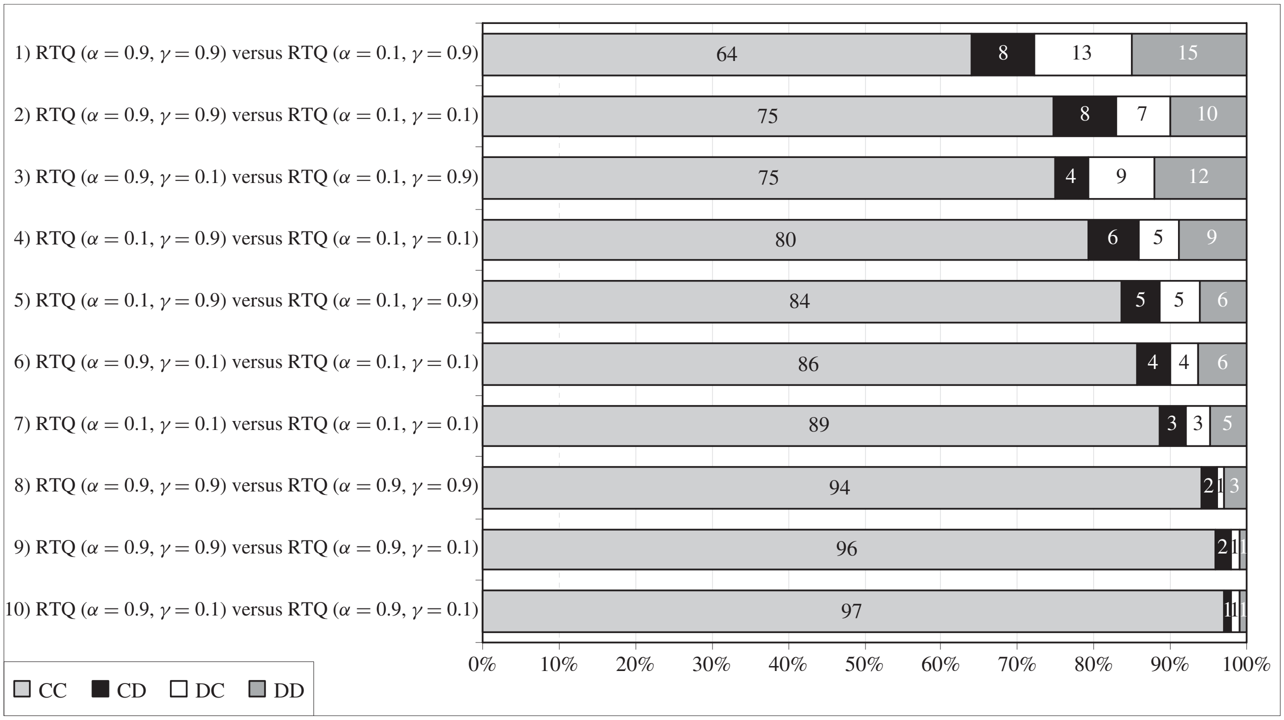

Fig. 2. Performance of RTQ agent against itself with different parameter settings ranked according to the percentage of mutual cooperation (CC). Highest CC is achieved when both agents learn fast ($\alpha = 0.9$) and have weak discounting ($\gamma = 0.1$).

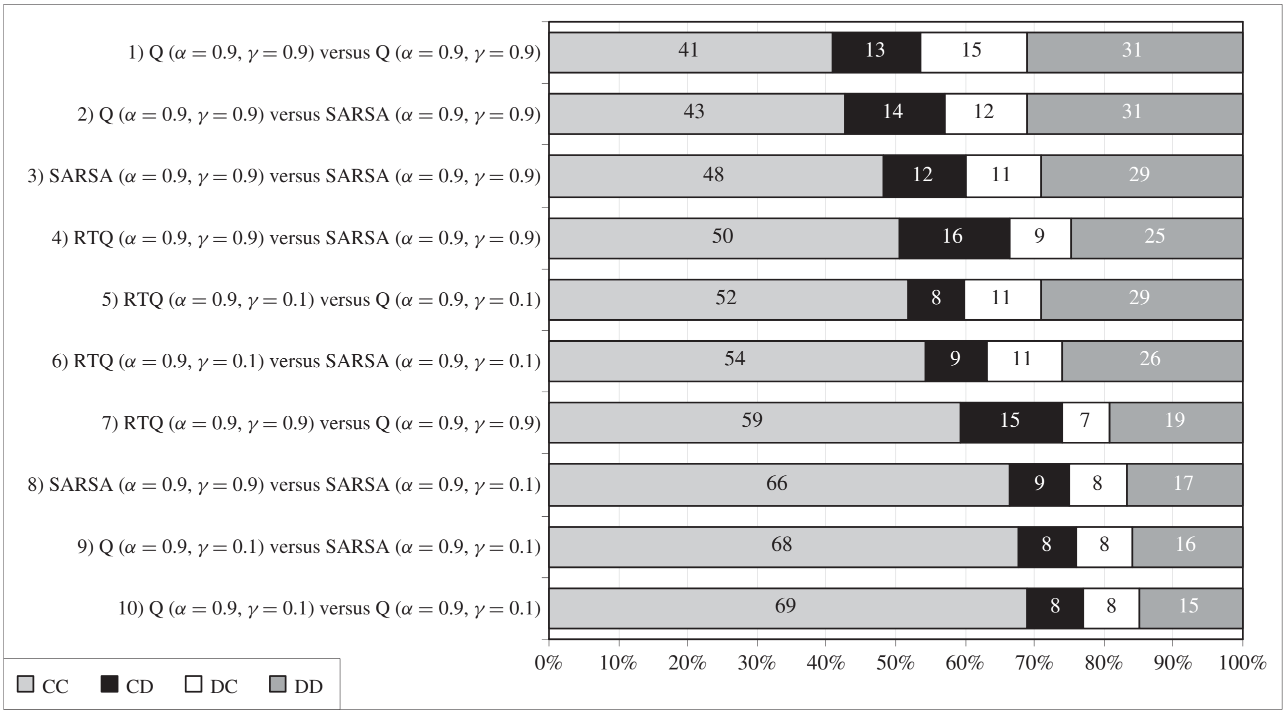

Fig. 3. Performance of Q-learning, SARSA, and RTQ agents against each other in homogeneous and heterogeneous configurations ranked based on the percentage of mutual cooperation (CC). Highest CC is achieved for Q-learning with $\alpha = 0.9$ and $\gamma = 0.1$ in self-play.

the 6th case). This might suggest that the system should be homogeneous (as in the 7th case).

On the other hand, weak discounting, at least by one of the agents, seems to increase performance, as the CC outcome occurs more frequently. For example, consider the 1st and 2nd cases: the first agent uses a high learning rate and discount factor, whereas the second agent uses low ones (2nd case). When the second agent switches to a strong discount factor (1st case), the CC performance drops from 75 to 64%. This is more clearly illustrated in the 8th, 9th, and 10th cases (i.e., the best three configurations). Starting from a high discount factor for both agents, the performance increases as one agent switches to a low discount factor (i.e., from 94 to 96%) and increases even more as the other agent switches to a low discount factor as well (97%). It is worth noting that this observation is in contrast with the results obtained when all learning agents competed against the reactive opponents TFT and Pavlov in Section III-A.1, as a higher discount factor is required there in order to reach the CC outcome. The exception is found between the 4th and 5th cases: both agents use a smaller learning rate and the first agent uses a high discount factor, whereas the second agent uses a low one (4th case). When the second agent changes to a high discount factor, the performance increases from 80 to 84% (from 4th to 5th case). This might suggest that the system should be homogeneous (as in the 5th case).

Fig. 3 shows the best results from the comparison of the other learning agents, ranked on the basis of the CC percentage. Here the system is homogeneous in the 1st, 3rd, 8th, and 10th cases. Comparing the 1st and 10th cases, when both Q-agents switch from a high discount factor (1st case) to a low one (10th case), CC occurs more frequently, i.e., from 41 to 69% of the time. Similar results are obtained with SARSA agents (3rd and 8th cases), where the CC percentage changed from 48 to 66%. The configurations "Q versus SARSA" (2nd and 9th cases) have similar results as well, since the performance changed from 43 to 68%. When RTQ competes with SARSA, the results follow the same pattern (i.e., highest CC percentage when both agents are myopic). This does not happen, however, when RTQ competes with Q. More specifically, when switching from farsighted to myopic agents, in the former case the percentage of CC increases from 50 to 54%, whereas in the latter case CC decreases from 59 to 52%. It is worth noting that none of the configurations converged.

Although not shown in Fig. 3, when the learning rate is low, the DD outcome occurs more frequently. The parameters that were found to increase DD are $\alpha = 0.1$ and $\gamma = 0.9$ for both agents, except for the case of "RTQ versus SARSA," where both agents used a low discount factor and RTQ used a high learning rate while SARSA a low one. For all these configurations, the range of mutual defection was $41 - 46\%$, but the systems did not converge.

By observing the Q-values, we noticed that in the best three configurations of Fig. 2 (cases 8–10, where two RTQ agents with $\alpha = 0.9$ competed), at the end of the simulation the Q-value corresponding to state CC and action C converged to a positive number, whereas all the other Q-values converged to negative numbers, for both agents. In the best three configurations of Fig. 3 (cases $8-10$), we observed a separation of the Q-values to positive, for the ones that correspond to action C from any state, and negative, for the ones that correspond to action D from any state, for both agents, but without reaching a plateau. This observation for cases 8-10 of Fig. 3 was different from what we observed in cases 8-10 of Fig. 2, and in addition, it was not observed in the worst case of Fig. 3 (case 1), where the discount factor for both Q-agents was set to 0.9, as some values that were positive for the first agent were negative for the other agent and vice versa.

In order to show the effect the evolved internal rewards have on the performance of the system, in terms of the accumulated payoff over time, we need to compare two configurations that have the same parameters but change only the type of agents from Q to RTQ (since RTQ agents use the Q-learning algorithm with different reinforcement signals). This effect is clearly illustrated in Fig. 4, where the best configuration (10th case) of Fig. 2 is compared with the best configuration (10th case) of Fig. 3. The system manages to engage in mutual cooperation from very early in the game when RTQ agents are used, and thus accumulates an average reward of 1401.52 (thick solid line) in 200 rounds, which is 87.6% of the best possible performance (1401.52/1600), whereas when Q agents are used, the system accumulates an average reward of 372.64 (dashed line) in 200 rounds, which is only 23.3% of the best possible performance (372.64/1600). As the exploration schedule was selected with the purpose of running the simulation for 1000 rounds, it has to be noted that the performance of the system at 1000 rounds when RTQ agents are used, slightly increases to 96.5% of the best possible performance (7715.92/8000). When Q agents are used, however, the performance increases more drastically to 65.6% (5246.24/8000). This shows that at 200 rounds the Q agents did not learn to cooperate as frequently as they do in 1000 rounds.

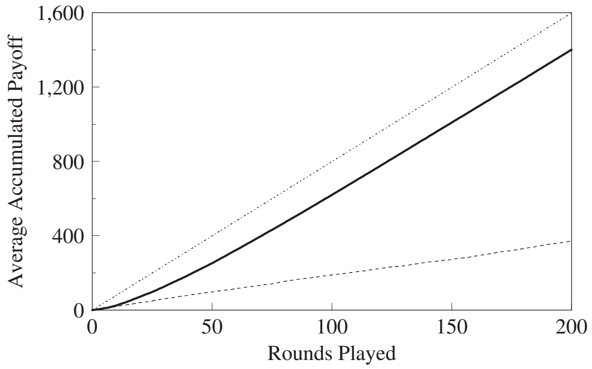

Fig. 4. Performance of nonspiking Q-agents: effect of the evolved internal rewards on the accumulated payoff. The performance of the system when both agents use internal reinforcement signals (thick solid line) and when the source of reinforcement is the payoff matrix (dashed line). The performance increased significantly when the internal reinforcement signals were used. The agents managed to engage in mutual cooperation. The theoretically best performance is shown for comparison (dot-dashed line).

B. Spiking Neural Network Agents

For the system configuration described in Section II-B, a single game of the IPD consists of 200 rounds during which the two networks seek to maximize their individual accumulated payoff by cooperating or defecting at every round of the game. The simulations aim to investigate the capability of the spiking NNs to cooperate in the IPD. It has to be noted that when the spiking agents played with nonlearning opponents, we had similar results to the nonspiking agents (not shown).

The following simulation involves an implementation of the game where the spiking agents learn through reinforcement of stochastic synaptic transmission [28]. When we directly applied the algorithm as originally proposed by Seung [28], using a single global reinforcement signal (see Section II-B),

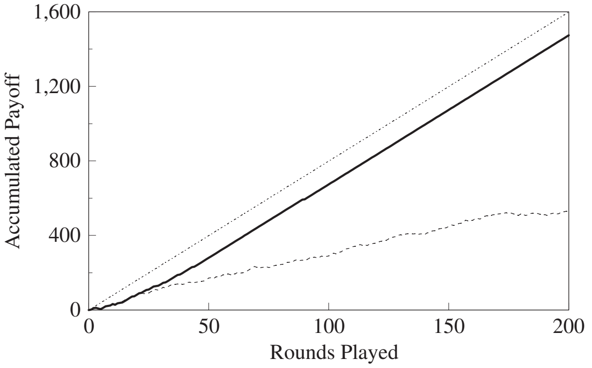

Fig. 5. Spiking NNs learning through reinforcement of stochastic synaptic transmission. The system's performance during the IPD with (thick solid line) and without (dashed line) the extra reinforcement administration. The performance increased dramatically when extra global signals were given as a feedback to the agents. The agents managed to engage in mutual cooperation. The theoretically best performance is shown for comparison (dot-dashed line).

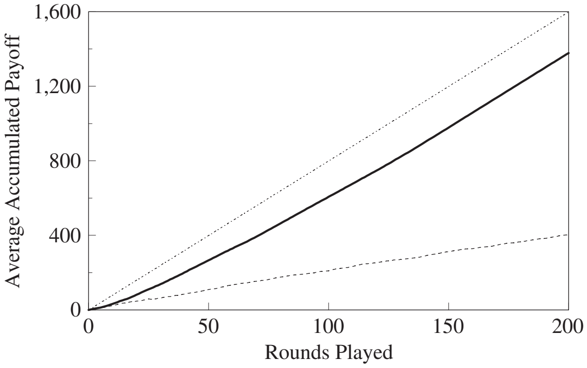

Fig. 6. Spiking NNs learning through reward-modulated STDP with eligibility trace: the effect of the extra reinforcement administration. The system performed much better when extra global reinforcement signals were given as a feedback to the agents (thick solid line). In contrast, it accumulated a very small total payoff when no additional signals were given (dashed line). The theoretically best performance is shown for comparison (dot-dashed line).

the agents did not show the capacity needed in order to learn how to cooperate according to the results shown in Fig. 5 (dashed line).

The accumulated payoff is calculated by adding together the payoff each network received according to the payoff matrix of Table I. For example, if at a given round the outcome was CC then a total $4 + 4 = 8$ will be added on the accumulated payoff. For the DC and CD outcome the total added payoff is 2 and for DD is -4. Given this, the system could achieve a maximum of 1600 (200 rounds × 8) if the two networks cooperated all the time. Results show that the system accumulated a total reward of less than 550 because of a low cooperative outcome. The CC outcome occurred only 31% of the time, which is a little more than if it had occurred by chance (25%). This is because the agents did not learn how to cooperate in order to maximize their long-term reward and the system performed suboptimally. A closer examination revealed that at the end of each learning round both output neurons of each network resulted with approximately the same firing rate. This effect was due to positive feedback which increases synaptic strength without bounds, leading to saturation of the synaptic connection and thus preventing further learning from taking place (like the limitation of classical Hebbian learning).

The problem was tackled by enhancing the competition between the output neurons through introducing additional global reinforcement signals that were administered alongside the original ones. These signals were administered to the networks for every spike of the output neurons that was not "responsible" for the decision at the last round. In the CD case, an additional reward of $+1.15$ (scaled down value corresponding to $+1.5$, see Section II-B) is provided to network I for every spike of output neuron 2 and an additional penalty of -1.15 (scaled down value corresponding to -1.5, see Section II-B) is provided to network II for every spike of output neuron 3. The value of 1.15 applies to all outcomes and is chosen to be small enough such that: 1) any changes to the values of the players' actions are primarily induced by the reinforcement signals provided by the payoff matrix, and 2) it does not cause any violation of the IPD rules. In effect, these opposite-in-sign signals update the value of the action that was not chosen by each network and can be justified as an additional feedback to the agents for their performance in the previous round. Overall, during a learning round, each network receives global, constant in value, and opposite-in-sign reinforcements that are applied in the time step following the spikes of both of its output neurons. One of the two signals is due to the payoff matrix of the game and its purpose is to "encourage" or "discourage" the action that elicited reward or penalty, and the other signal is complementary and its purpose is to "encourage" or "discourage" the action that could have elicited reward or penalty if had been chosen in the previous round of the game.

Fig. 5 (thick solid line) shows the system's performance when the additional reinforcement signals were incorporated into the learning algorithm. The simulation was identical to the previous one apart from the enhanced reinforcement administration scheme. The difference in performance is evident. The networks accumulated a total payoff of almost 1500 by cooperating 91% of the time. The results reveal that the agents learned to maximize long-term reward through cooperative behavior. It has to be noted that the CC outcome not only persisted during the final rounds of the simulations, but it also did not change after a point due to the system's dynamics that were evolved by that point in time in such a way to produce CC consistently.

The following simulation implements the IPD where the agents learn through reward-modulated STDP with eligibility trace [29]. As shown in the previous simulations (where the agents learned through reinforcement of stochastic synaptic transmission [28]), the administration of additional, opposite-in-sign, global reinforcement signals proved to be vital for the successful training of the competing agent that attained the cooperative outcome. We therefore tested the importance of this additional reinforcement administration, for the performance of the system when trained with reward-modulated STDP with eligibility trace. Fig. 6 shows that the implementation of the game was successful when the additional reinforcement signal was administered.

The CC outcome was attained after a relatively short learning period, which enhanced the accumulation of reward by

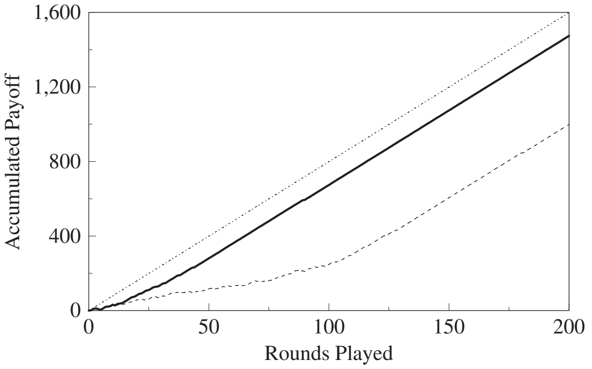

Fig. 7. Eligibility trace time constant effect (with extra reinforcement) when the spiking NNs learn with reinforcement of stochastic synaptic transmission. The system collected a much higher total reward when the eligibility trace time constant of both networks was equal to 20 ms (thick solid line) compared to 2 ms (dashed line). The theoretically best performance is shown for comparison (dot-dashed line).

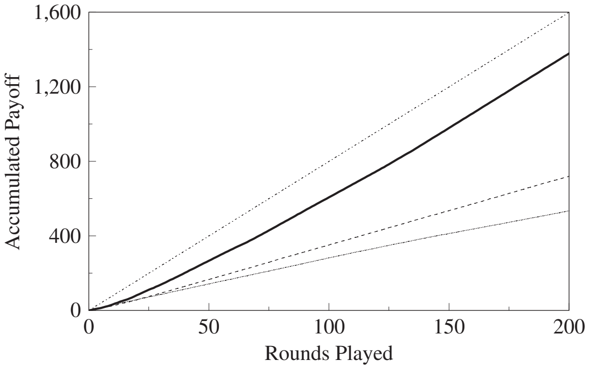

Fig. 8. Eligibility trace time constant effect (with extra reinforcement) when the spiking NNs learn with reward-modulated STDP. The system collected a much higher total reward when the eligibility trace time constant of both networks was equal to 25 ms (thick solid line) compared to 2 ms (thin dotted line). The system performed in between when one network was configured with 25 ms and the other with 2 ms (dashed line). The theoretically best performance is shown for comparison (dot-dashed line).

the system. This reveals that after a certain point the networks successfully learned to resist the temptation payoff provided by defection in order to maximize their long-term reward through cooperation, enabling thus reward maximization by the system as well. However, the system performed badly when no extra reinforcement was given. The agents cooperated 88% of the time when the extra reinforcement was added. The performance deteriorated significantly when no additional reinforcement signals were administered to the networks since the cooperation level fell 60 percentage points (from 88 to 28%) and the defection level increased 21.5 percentage points (from 6 to 27.5%). The results with the current learning scheme are in line with our previous results (see Fig. 5) with regard to the effectiveness of the additional reinforcement in the attainment of a cooperative behavior. The administration of extra reinforcement is thus vital for a high payoff accumulation by the spiking NN agents and therefore all the subsequent simulations are carried out with extra reinforcement administration.

As explained, the eligibility trace is a dynamical variable used to integrate time-related events and is utilized in Seung's algorithm [28] as a memory for each synapse's past actions with respect to releasing a neurotransmitter (see Section II-B). The eligibility trace time constant regulates the decay of the variable and signifies for how long these time-related events are in effect integrated. In other words, a synapse with longer eligibility trace time constant has a stronger memory than a synapse with shorter eligibility trace time constant. The following simulations are carried out in order to investigate the effect memory has on attaining cooperative behavior. Two simulations were performed with the synapses of the two networks having different eligibility trace time constants. The values for both networks were set to 20 and 2 ms for the two simulations, respectively. Therefore, during the first simulation the networks have a "strong memory," whereas in the second they have a "weak memory." The results of both simulations are shown in Fig. 7. The difference in the system's performance is obvious. When the system was configured with 20 ms eligibility trace time constants, the accumulated payoff is much higher than the one with 2 ms, which is a result of the difference in the CC outcome. With the eligibility trace time constants set at 20 ms, the two networks learned quickly to cooperate in order to maximize their long-term reward and achieved the CC outcome 182 out of the 200 times. On the contrary, when the system was configured with "weak memory," learning took effect much later during the game (after the 100th round) and thus the system exhibited much less cooperation (120 out of 200). However, the system with both configurations eventually managed to learn to cooperate. Results show that agents' memory influences the cooperative outcome of the game in the sense that it delays it to a great extent. However, a weak memory does not destroy learning as the networks eventually learned to cooperate.

In reward-modulated STDP with eligibility trace [29], the latter serves as a decaying memory of the relation between recent pre- and postsynaptic spike pairs. Three simulations were performed with the neurons of the two networks having different eligibility trace time constants. The values for both networks were set to 25 and 2 ms, respectively, for the two simulations, whereas during the third one, one network was configured with 25 ms and the other with 2 ms. Therefore, during the first simulation the agents had a strong memory, in the second they had a weak memory, and in the third simulation one agent had strong memory and the other had weak. The results of all simulations are shown in Fig. 8. The difference in the system's performance is evident. When the system was configured with 25 ms eligibility trace time constants for both agents, the accumulated payoff is much higher than in the case when the system was configured with 2 ms eligibility trace time constants. During the former simulation, the agents engaged in a behavior of mutual cooperation, whereas in the latter they primarily defected. With the eligibility trace time constants set at 25 ms, the two networks learned quickly to cooperate in order to maximize their long-term reward and achieved a total payoff of 1379 with the CC outcome chosen 88% of the time. On the contrary, when the system was configured with weak memory (2 ms eligibility trace time constant for both agents), learning was sometimes totally destroyed and thus the system resulted in exhibiting much less average cooperation level (50%) and a total payoff of 534. The system performed slightly better in the final simulation where one agent had a strong memory and the other had a weak one. It accumulated a total payoff of 720 compared to the 534 of the "memory-less" agents. However, the cooperation remained at the same low level (49%).

The difference in the total payoff occurs due to the difference in the DD outcome rather than in the CC outcome. In the case where both agents had a weak memory, they both aimed for the temptation payoff and thus engaged in a behavior of increased mutual defection (DD outcome was 38%), accumulating thus the smallest total payoff. Only the system with the strong memory configuration managed to exhibit high cooperation levels.

Overall, the results show that spiking NNs can successfully implement artificial agents in a demanding MARL task. The best results were obtained with high values of eligibility trace time constants whereas the additional, opposite-in-sign, global reinforcement signals proved to be vital for successful implementation of the game.

IV. Conclusion

The first part of this paper examined MARL with nonspiking agents that use simple algorithms from the single-agent RL literature and, more specifically, Q-learning and SARSA. A comparison was done with another Q-learning agent that uses different internal reinforcement signals than the external payoff values, which we called the RTQ agent. We found these signals by evolutionary optimization with the purpose of rapidly increasing the CC outcome [58].

We initially tested the performance of the nonspiking learning agents against opponents that do not use any learning algorithm and observed that cooperation was established when the opponent used a TFT or a Pavlov strategy. The performance was maximized when the learning agents were farsighted and used a high learning rate. When the learning agents played against other nonlearning opponents, the results were as expected, i.e., the learning agent tried to get as much reward as possible.

The experiments with the nonspiking agents were then extended to the MARL case, i.e., when both agents use a learning algorithm. According to these results, the only agent that was able to rapidly converge to the CC outcome was the RTQ agent. All other agents did not converge with the given exploration schedule, however, it has to be noted that some configurations did manage to frequently choose to cooperate during the final rounds. In addition, the RTQ agents converged faster when both agents use high learning rates and low discount factors. This fast convergence of the RTQ agents might be due to the combination of high learning rates and low discount factors with the empirically chosen exploration schedule. In particular, this exploration schedule was chosen so as to achieve convergence to CC in less than 1000 rounds with the RTQ agents. Other homogeneous configurations (i.e., with Q and SARSA agents in self-play) benefit from high learning rates and low discount factors as well, since mutual cooperation is increased with these settings.

While many algorithms from the MARL literature were designed in order to fulfill certain criteria, in this paper we showed that, in the case of nonspiking agents, the evolved internal reinforcement signals could make the "naive" Q-learning behave efficiently in repeated general-sum games. Our approach [58] did not evolve the payoff values while the agents learn. More specifically, the algorithm we used [58] searched for fixed internal rewards for the agents without changing their goal, since the evolved solutions have a valid payoff structure. This is in line with the way biological evolution hard-wires primary rewards in animals due to their reproductive success. While the distinction between internal rewards and external sensations was taken into account [51], for simplicity we ignored the mapping between them. The difference in performance between normal Q agents and our newly created RTQ agents might indicate that the evolved internal reinforcement signals create agents that are motivated to cooperate since they implicitly contain a sense of reward and penalty, thereby "pointing at" the goal which in our case is mutual cooperation.

The work in this paper applies also spiking neural agents combined with biologically plausible reinforcement learning schemes in a demanding multiagent task. In particular, it evaluates the effectiveness of reinforcement of stochastic synaptic transmission [28] and reward-modulated STDP [29] in the general-sum game of the IPD. Results showed that both investigated learning algorithms managed to exhibit "sophisticated intelligence" in a nontrivial task. The spiking agents showed a capacity for playing the game along the lines of game theory in a way that resembles the behavior of real players. During most of the simulations, the networks managed to adapt to the challenges of the game and make decisions according to the other player's decisions in order to maximize their accumulated payoff. Most importantly, they "displayed intelligence" because when the game flow allowed for the Pareto-optimum solution to be reached they "took advantage of the possibility" and settled to the solution by choosing cooperation for the rest of the game.

In addition, this paper extended the reinforcement learning algorithms for spiking NNs with additional, opposite-in-sign and global reinforcement signals that were concurrently administered along with the signals specified by the payoff matrix of the game. The extended reinforcement administration scheme applied a positive global reinforcement to one output and a negative global reinforcement to the other output. The administration of additional global reinforcement signals, which increased competition at the neuronal and synaptic level, proved both novel and necessary for the successful performance of the learning algorithms, which enabled the agents to learn to engage in high mutual cooperation. More specifically, the administration of additional global reinforcement signals was essential, so as to avoid a positive feedback effect which would have increased the synaptic strength without bounds, leading to saturation of the synaptic connection and thus preventing further learning from taking place (like the limitation of classical Hebbian learning). Therefore, one could conclude that, in cases where more than one output neuron competes for reinforcement in a spiking NN, the global evaluation signal (in both Seung's reinforcement of stochastic synaptic transmission [28] and Florian's reward-modulated STDP [29]) should consist of global reward and penalty accordingly, for avoidance of possible synaptic saturation.

In the case of spiking NNs, the successful application of both learning algorithms [28], [29] to the IPD required high values of eligibility trace time constants for both networks. It follows that the extent to which the reinforcement applies in changes that happened before determines the success of the learning algorithms. Results showed that reinforcement should apply to changes over a longer period as agents with a "stronger memory" configuration achieved the best CC result, indicating the importance of memory in effective MARL.

Taking into consideration our results with both spiking and nonspiking agents, we can see that in both cases the system accumulates higher CC reward when both agents have: 1) high learning rate and low discount factor in the case of nonspiking agents, or 2) "strong" memory (achieved with long eligibility trace time constant), in the case of spiking agents. In order to make a more direct comparison between the spiking and nonspiking systems, we could employ temporal difference learning in our spiking NNs as in [40] and use NN nonspiking agents instead of tabular ones. In addition, it is also desirable for the payoff matrix of the nonspiking agents to mix positive and negative values (as in [80]), which, if viewed as another technique of introducing competition into the system, could explain the enhancement of the CC outcome. As mentioned in Section II-B, this mixture is necessary for the spiking agents. Moreover, as in [51], both spiking and nonspiking RTQ agents incorporate the notion of the separation of the environment into an external and an internal one, where the external environment is effectively the payoff matrix and the internal one is part of the agent and provides the transformed rewards. The modeling of this process proved to be beneficial for spiking and nonspiking RTQ agents.

Potential applications for MARL with nonspiking agents arise in negotiations and conflict resolution, with notable examples being the Cyprus problem [81], [82] and the Greek–Turkish arms race [83]. Spiking neural agents can be used when more biological realism is required. For example, they have already been used in our ongoing study that aims to investigate how and when internal conflict can be resolved through self-control behavior [84], [85]. Brain imaging results on internal conflict reveal two distinct brain systems competing for control of the organism in the form of relative activations [86]. In addition, according to [87], such conflicts might be resolved as if they were a strategic interaction between rational subagents of the brain. This particular interaction can be modeled by the IPD, where the players map to the brain's subagents which have conflicting and distinct value systems [87]. Moreover, the CC outcome corresponds to the agents compromising and the organism exhibiting self-control. In our study [84], [85], a spiking NN maps to each of these neural systems implementing a subagent of the brain. Through this particular interaction, we investigated the neuronal and psychological variables that enable the conflict to be resolved through self-control behavior. Certainly, a spiking agent system is more computationally expensive and should only be used when the task in question demands more biologically realistic models, as in our modeling of the high level behavior of self-control [84], [85], or in our recent finding where we proved that high firing irregularity enhances learning [88].

In general, as it can be seen from the results, the behavior of spiking and nonspiking agents is in effect similar. We could therefore argue that spiking agents could equally well be used in multiagent interactions as nonspiking agents.

Appendix
Equations for Spiking Neural Network Agents

A. Reinforcement of Stochastic Synaptic Transmission

Briefly, within the framework of the model [28], each synapse acts as an agent that pursues reward maximization. Upon arrival of a presynaptic spike, a synapse can take two possible actions with complementary probabilities: release a neurotransmitter with probability p or fail to release with probability $1 - p$. The release parameter q is monotonically related to p by the sigmoidal function given by

$$p = \frac{1}{1 + e^{-q}}. \tag{2}$$

Each synapse keeps a record of its recent actions through a dynamical variable, the eligibility trace ($\bar{e}$) [79]. It increases by $1 - p$ with every release and decreases by $-p$ with every failure. Otherwise, it decays exponentially with a given time constant. When a global reinforcement signal (h) is given to the network, it is subsequently communicated to each synapse which modifies its release probability according to the nature of the signal (reward or penalty) and its recent releases and failures. Learning is driven by modifying q according to the rule given by

$$\Delta q = \eta \times h \times \bar{e} \tag{3}$$

where η is the learning rate.

Each network has a hidden layer of 60 neurons and an output layer of 2 neurons, all modeled with the LIF neuron given by

$$C\frac{dV_i}{dt} = -g_L(V_i - V_L) - \sum_j G_{ij}(V_i - E_{ij}) \tag{4}$$

where $V_L = -74$ mV, $g_L = 25$ nS, and $C = 500$ pF, giving a membrane time constant of $\tau = 20$ ms. The differential equations are integrated using an exponential Euler update with a 0.5 ms time step. When the membrane potential V_i reaches the threshold value of -54 mV, it is reset to -60 mV (values as in the numerical simulations by Seung [28]). The reversal potential E_{ij} of the synapse from neuron j to neuron i is set to either 0 or -70 mV, depending on whether the synapse is excitatory or inhibitory. The synaptic conductances are updated via $G_{ij} = W_{ij} r_{ij}$, where r_{ij} is the neurotransmitter release variable that takes the value of 1 with probability equal to the probability that the synapse from neuron j to i releases a neurotransmitter (when j spikes) and 0 otherwise [28]. In the absence of presynaptic spikes, G_{ij} decays exponentially with time constant $\tau_s = 5$ ms. W_{ij} are the "weights" which do not change over time and are chosen randomly from an exponential distribution with mean 14 nS for excitatory synapses and 45 nS for inhibitory synapses.

B. Reward-Modulated STDP with Eligibility Trace

In reward-modulated STDP with eligibility trace [29], the efficacy of the synapse from neuron j to i is changed according to

$$w_{ij}(t+\delta t) = w_{ij}(t) + \gamma \ \delta t \ r(t+\delta t) \ z_{ij}(t+\delta t) \tag{5}$$

where γ is the learning rate, δt is the duration of a time step, r is the global reward signal, and z is the eligibility trace which is modified according to

$$z_{ij}(t+\delta t) = \beta \ z_{ij}(t) + \zeta_{ij}(t)/\tau_z. \tag{6}$$

β is a discount factor between 0 and 1, ζ is a notation for the change of z resulting from the activity in the last time step, and τ_z is the time constant for the exponential decay of z. At time t, ζ is computed by the following set of equations

$$\zeta_{ij}(t) = P_{ij}^{+}(t) \ f_i(t) + P_{ij}^{-}(t) \ f_j(t) \tag{7}$$

$$P_{ij}^{+}(t) = P_{ij}^{+}(t-\delta t)\exp(-\delta t/\tau_{+}) + A_{+} \ f_j(t) \tag{8}$$

$$P_{ij}^{-}(t) = P_{ij}^{-}(t-\delta t)\exp(-\delta t/\tau_{-}) + A_{-} \ f_i(t) \tag{9}$$

where the variable P_{ij}^{+} tracks the influences of presynaptic spikes and the varable P_{ij}^{-} tracks the influence of postsynaptic spikes. The time constants τ_{+} and τ_{-} determine the ranges of interspike intervals over which synaptic changes occur and according to the standard antisymmetric STDP model, while A_{+} and A_{-} are positive and negative constant parameters respectively. Finally, $f_i(t)$ is 1 if neuron i has fired at time step t or 0 otherwise.

The hidden and output layers of the networks are composed of integrate-and-fire neurons with resting potential $u_r = -70$ mV, firing threshold $\theta = -54$ mV, reset potential equal to the resting potential, and decay time constant $\tau =$ 20 ms. These are the same values as used in the simulations by Florian [29]. We also used the same dynamics for the neurons' membrane potential given by

$$u_i(t) = u_r + [u_i(t-\delta t) - u_r]\exp(-\delta t/\tau) + \sum_j w_{ij} f_j(t-\delta t). \tag{10}$$

The membrane potential was reset to u_r when surpassed θ. We used $\tau_{+} = \tau_{-} = 20$ ms, $A_{+} = 1$, $A_{-} = -1$, $\delta t = 1$ ms, $\gamma = 0.7 \times 10^{-4}$, and, unless specified, $\tau_z = 25$ ms.

Acknowledgment

The authors would like to thank the three anonymous reviewers for their useful and constructive comments.

References

[1] M. L. Littman, "Markov games as a framework for multiagent reinforcement learning," in *Proc. 11th Int. Conf. Mach. Learn.*, 1994, pp. 157–163.

[2] J. Hu and M. P. Wellman, "Nash Q-learning for general-sum stochastic games," *J. Mach. Learn. Res.*, vol. 4, pp. 1039–1069, Nov. 2003.

[3] C. Claus and C. Boutilier, "The dynamics of reinforcement learning in cooperative multiagent systems," in *Proc. 15th Nat. Conf. Artif. Intell. 10th Innovat. Appl. Artif. Intell. Conf.*, 1998, pp. 746–752.

[4] M. L. Littman, "Friend-or-foe Q-learning in general-sum games," in *Proc. 18th Int. Conf. Mach. Learn.*, 2001, pp. 322–328.

[5] M. H. Bowling and M. M. Veloso, "Rational and convergent learning in stochastic games," in *Proc. 17th Int. Joint Conf. Artif. Intell.*, 2001, pp. 1021–1026.

[6] C. J. C. H. Watkins, "Learning from delayed rewards," Ph.D. dissertation, Dept. Psychol., Cambridge Univ., Cambridge, U.K., May 1989.

[7] M. Bowling and M. Veloso, "Multiagent learning using a variable learning rate," *Artif. Intell.*, vol. 136, no. 2, pp. 215–250, Apr. 2002.

[8] B. Banerjee and J. Peng, "Convergent gradient ascent in general-sum games," in *Machine Learning: ECML* (Lecture Notes in Computer Science), vol. 2430, T. Elomaa, H. Mannila, and H. Toivonen, Eds. Berlin, Germany: Springer-Verlag, 2002, pp. 1–9.

[9] S. P. Singh, M. J. Kearns, and Y. Mansour, "Nash convergence of gradient dynamics in general-sum games," in *Proc. 16th Conf. Uncert. Artif. Intell.*, 2000, pp. 541–548.

[10] A. R. Greenwald and K. Hall, "Correlated Q-learning," in *Proc. 20th Int. Conf. Mach. Learn.*, 2003, pp. 242–249.

[11] S. Kapetanakis and D. Kudenko, "Reinforcement learning of coordination in heterogeneous cooperative multiagent systems," in *Adaptive Agents and Multiagent Systems* (Lecture Notes in Computer Science), vol. 3394, D. Kudenko, D. Kazakov, and E. Alonso, Eds. Berlin, Germany: Springer-Verlag, 2005, pp. 119–131.

[12] M. Bowling, "Convergence and no-regret in multiagent learning," in *Advances in Neural Information Processing Systems 17*, L. K. Saul, Y. Weiss, and L. Bottou, Eds. Cambridge, MA: MIT Press, 2005, pp. 209–216.

[13] V. Conitzer and T. Sandholm, "AWESOME: A general multiagent learning algorithm that converges in self-play and learns a best response against stationary opponents," *Mach. Learn.*, vol. 67, nos. 1–2, pp. 23–43, 2007.

[14] S. Abdallah and V. Lesser, "A multiagent reinforcement learning algorithm with non-linear dynamics," *J. Artif. Intell. Res.*, vol. 33, no. 1, pp. 521–549, Sep. 2008.

[15] J. W. Crandall and M. A. Goodrich, "Learning to compete, coordinate, and cooperate in repeated games using reinforcement learning," *Mach. Learn.*, 2011, to be published.

[16] L. Buşoniu, R. Babuška, and B. De Schutter, "A comprehensive survey of multiagent reinforcement learning," *IEEE Trans. Syst., Man, Cybern., Part C: Appl. Rev.*, vol. 38, no. 2, pp. 156–172, Mar. 2008.

[17] R. Powers, Y. Shoham, and T. Vu, "A general criterion and an algorithmic framework for learning in multiagent systems," *Mach. Learn.*, vol. 67, nos. 1–2, pp. 45–76, May 2007.

[18] K. Tuyls, P. J. Hoen, and B. Vanschoenwinkel, "An evolutionary dynamical analysis of multiagent learning in iterated games," *Auton. Agents Multiagent Syst.*, vol. 12, no. 1, pp. 115–153, 2006.

[19] K. Iwata, K. Ikeda, and H. Sakai, "A statistical property of multiagent learning based on Markov decision process," *IEEE Trans. Neural Netw.*, vol. 17, no. 4, pp. 829–842, Jul. 2006.

[20] Y. Shoham, R. Powers, and T. Grenager, "If multiagent learning is the answer, what is the question?" *Artif. Intell.*, vol. 171, no. 7, pp. 365–377, May 2007.

[21] I. Erev and A. E. Roth, "Multiagent learning and the descriptive value of simple models," *Artif. Intell.*, vol. 171, no. 7, pp. 423–428, May 2007.

[22] G. J. Gordon, "Agendas for multiagent learning," *Artif. Intell.*, vol. 171, no. 7, pp. 392–401, May 2007.

[23] D. Fudenberg and D. K. Levine, "An economist's perspective on multiagent learning," *Artif. Intell.*, vol. 171, no. 7, pp. 378–381, May 2007.

[24] K. Tuyls and S. Parsons, "What evolutionary game theory tells us about multiagent learning," *Artif. Intell.*, vol. 171, no. 7, pp. 406–416, May 2007.

[25] P. Stone, "Multiagent learning is not the answer. It is the question," *Artif. Intell.*, vol. 171, no. 7, pp. 402–405, May 2007.

[26] G. A. Rummery and M. Niranjan, "On-line Q-learning using connectionist systems," Dept. Eng., Cambridge Univ., Cambridge, U.K., Tech. Rep. CUED/F-INFENG/TR 166, Sep. 1994.

[27] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA: MIT Press, 1998.

[28] H. S. Seung, "Learning in spiking neural networks by reinforcement of stochastic synaptic transmission," *Neuron*, vol. 40, no. 6, pp. 1063–1073, Dec. 2003.

[29] R. V. Florian, "Reinforcement learning through modulation of spike-timing-dependent synaptic plasticity," *Neural Comput.*, vol. 19, no. 6, pp. 1468–1502, Jun. 2007.

[30] X. Xie and H. S. Seung, "Learning in neural networks by reinforcement of irregular spiking," *Phys. Rev. E*, vol. 69, no. 4, pp. 041909-1–041909-10, Apr. 2004.

[31] M. A. Faries and A. L. Fairhall, "Reinforcement learning with modulated spike timing-dependent synaptic plasticity," *J. Neurophysiol.*, vol. 98, no. 6, pp. 3648–3665, Dec. 2007.

[32] E. M. Izhikevich, "Solving the distal reward problem through linkage of STDP and dopamine signaling," *Cereb. Cortex*, vol. 17, no. 10, pp. 2443–2452, Jan. 2007.

[33] R. Legenstein, D. Pecevski, and W. Maass, "A learning theory for reward-modulated spike-timing-dependent plasticity with application to biofeedback," *PLoS Comput. Biol.*, vol. 4, no. 10, p. e1000180, 2008.

[34] H. Markram, J. Lübke, M. Frotscher, and B. Sakmann, "Regulation of synaptic efficacy by coincidence of postsynaptic APs and EPSPs," *Science*, vol. 275, no. 5297, pp. 213–215, Jan. 1997.

[35] G.-Q. Bi and M.-M. Poo, "Synaptic modifications in cultured hippocampal neurons: Dependence on spike timing, synaptic strength, and postsynaptic cell type," *J. Neurosci.*, vol. 18, no. 24, pp. 10464–10472, Dec. 1998.

[36] Y. Dan and M.-M. Poo, "Spike timing-dependent plasticity of neural circuits," *Neuron*, vol. 44, no. 1, pp. 23–30, Sep. 2004.

[37] J.-P. Pfister, T. Toyoizumi, D. Barber, and W. Gerstner, "Optimal spike-timing-dependent plasticity for precise action potential firing in supervised learning," *Neural Comput.*, vol. 18, no. 6, pp. 1318–1348, Jun. 2006.

[38] D. Baras and R. Meir, "Reinforcement learning, spike-time-dependent plasticity, and the BCM rule," *Neural Comput.*, vol. 19, no. 8, pp. 2245–2279, Aug. 2007.

[39] E. Vasilaki, N. Frémaux, R. Urbanczik, W. Senn, and W. Gerstner, "Spike-based reinforcement learning in continuous state and action space: When policy gradient methods fail," *PLoS Comput. Biol.*, vol. 5, no. 12, p. e1000586, 2009.

[40] W. Potjans, A. Morrison, and M. Diesmann, "A spiking neural network model of an actor-critic learning agent," *Neural Comput.*, vol. 21, no. 2, pp. 301–339, Feb. 2009.

[41] R. Rom, J. Erel, M. Glikson, R. A. Lieberman, K. Rosenblum, O. Binah, R. Ginosar, and D. L. Hayes, "Adaptive cardiac resynchronization therapy device based on spiking neurons architecture and reinforcement learning scheme," *IEEE Trans. Neural Netw.*, vol. 18, no. 2, pp. 542–550, Mar. 2007.

[42] P. Arena, L. Fortuna, M. Frasca, and L. Patane, "Learning anticipation via spiking networks: Application to navigation control," *IEEE Trans. Neural Netw.*, vol. 20, no. 2, pp. 202–216, Feb. 2009.

[43] I. P. Pavlov, *Conditioned Reflexes: An Investigation of the Physiological Activity of the Cerebral Cortex* (Transl. by G. V. Anrep, Ed.). London, U.K.: Oxford Univ. Press, 1927.

[44] H. G. Zimmermann, R. Neuneier, and R. Grothmann, "Multiagent modeling of multiple FX-markets by neural networks," *IEEE Trans. Neural Netw.*, vol. 12, no. 4, pp. 735–743, Jul. 2001.

[45] A. Rappoport and A. M. Chammah, *Prisoner's Dilemma: A Study in Conflict and Cooperation*. Ann Arbor, MI: Univ. Michigan Press, 1965.

[46] J. F. Nash, "Equilibrium points in N-person games," *Proc. Nat. Acad. Sci. Unit. Stat. Amer.*, vol. 36, no. 1, pp. 48–49, 1950.

[47] V. Pareto, *Manuale di Economia Politica*. Milan, Italy: Societa Editrice, 1906.

[48] M. Zinkevich, A. Greenwald, and M. L. Littman, "A hierarchy of prescriptive goals for multiagent learning," *Artif. Intell.*, vol. 171, no. 7, pp. 440–447, May 2007.

[49] M. Snel and G. M. Hayes, "Evolution of valence systems in an unstable environment," in *From Animals to Animats 10* (Lecture Notes in Computer Science), vol. 5040, M. Asada, J. C. T. Hallam, J.-A. Meyer, and J. Tani, Eds. Berlin, Germany: Springer-Verlag, 2008, pp. 12–21.

[50] D. E. Ackley and M. L. Littman, "Interactions between learning and evolution," in *Proc. 2nd Conf. Artif. Life*, 1991, pp. 487–509.

[51] S. Singh, A. G. Barto, and N. Chentanez, "Intrinsically motivated reinforcement learning," in *Advances in Neural Information Processing Systems 17*, L. K. Saul, Y. Weiss, and L. Bottou, Eds. Cambridge, MA: MIT Press, 2005, pp. 1281–1288.

[52] S. Singh, R. L. Lewis, A. G. Barto, and J. Sorg, "Intrinsically motivated reinforcement learning: An evolutionary perspective," *IEEE Trans. Auton. Mental Develop.*, vol. 2, no. 2, pp. 70–82, Jun. 2010.

[53] K. Moriyama, "Utility based Q-learning to facilitate cooperation in Prisoner's Dilemma games," *Web Intell. Agent Syst.*, vol. 7, no. 3, pp. 233–242, Aug. 2009.

[54] R. Aras, A. Dutech, and F. Charpillet, "Efficient learning in games," in *Proc. Conf. Fran. sur l'Apprentissage Automatique - CAP*, Trégastel, France, Sep. 2006.

[55] V. Vassiliades, A. Cleanthous, and C. Christodoulou, "Multiagent reinforcement learning with spiking and non-spiking agents in the Iterated Prisoner's Dilemma," in *Artificial Neural Networks – ICANN* (Lecture Notes in Computer Science), vol. 5768, C. Alippi, M. M. Polycarpou, C. Panayiotou, and G. Ellinas, Eds. Berlin, Germany: Springer-Verlag, 2009, pp. 737–746.

[56] R. M. Axelrod, *The Evolution of Cooperation*. New York, NY: Basic Books, 1984.

[57] G. Kendall, X. Yao, and S. Y. Chong, "The Iterated Prisoner's Dilemma: 20 years on," in *Advances in Natural Computation*, vol. 4. Singapore: World Scientific, 2007.

[58] V. Vassiliades and C. Christodoulou, "Multiagent reinforcement learning in the Iterated Prisoner's Dilemma: Fast cooperation through evolved payoffs," in *Proc. Int. Joint Conf. Neural Netw.*, Barcelona, Spain, Jul. 2010, pp. 2828–2835.

[59] R. S. Sutton, "Learning to predict by the methods of temporal differences," *Mach. Learn.*, vol. 3, no. 1, pp. 9–44, 1988.

[60] R. J. Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning," *Mach. Learn.*, vol. 8, nos. 3–4, pp. 229–256, 1992.

[61] J. Baxter and P. L. Bartlett, "Infinite-horizon policy-gradient estimation," *J. Artif. Intell. Res.*, vol. 15, no. 4, pp. 319–350, Nov. 2001.

[62] J. Baxter, P. L. Bartlett, and L. Weaver, "Experiments with infinite-horizon, policy-gradient estimation," *J. Artif. Intell. Res.*, vol. 15, no. 1, pp. 351–381, Nov. 2001.

[63] J. Moody and M. Saffell, "Learning to trade via direct reinforcement," *IEEE Trans. Neural Netw.*, vol. 12, no. 4, pp. 875–889, Jul. 2001.

[64] X. Ma and K. K. Likharev, "Global reinforcement learning in neural networks," *IEEE Trans. Neural Netw.*, vol. 18, no. 2, pp. 573–577, Mar. 2007.

[65] T. W. Sandholm and R. H. Crites, "Multiagent reinforcement learning in the Iterated Prisoner's Dilemma," *Biosyst.*, vol. 37, nos. 1–2, pp. 147–166, 1996.

[66] R. Axelrod and W. D. Hamilton, "The evolution of cooperation," *Science*, vol. 211, no. 4489, pp. 1390–1396, Mar. 1981.

[67] M. Nowak and K. Sigmund, "A strategy of win-stay, lose-shift that outperforms tit-for-tat in the Prisoner's Dilemma game," *Nature*, vol. 364, no. 6432, pp. 56–58, Jul. 1993.

[68] L. Lapicque, "Recherches quantitatives sur l'excitation électrique des nerfs traitée comme une polarization," *J. Physiol. Pathol. Gen.*, vol. 9, pp. 620–635, 1907.

[69] R. B. Stein, "Some models of neuronal variability," *Biophys. J.*, vol. 7, no. 1, pp. 37–68, Jan. 1967.

[70] E. M. Izhikevich, "Which model to use for cortical spiking neurons?" *IEEE Trans. Neural Netw.*, vol. 15, no. 5, pp. 1063–1070, Sep. 2004.

[71] A. L. Hodgkin and A. F. Huxley, "A quantitative description of membrane current and its application to conduction and excitation in nerve," *J. Physiol.*, vol. 117, no. 4, pp. 500–544, Aug. 1952.

[72] E. M. Izhikevich, "Simple model of spiking neurons," *IEEE Trans. Neural Netw.*, vol. 14, no. 6, pp. 1569–1572, Nov. 2003.

[73] C. Christodoulou, G. Bugmann, and T. G. Clarkson, "A spiking neuron model: Applications and learning," *Neural Netw.*, vol. 15, no. 7, pp. 891–908, Sep. 2002.

[74] R. J. MacGregor and R. M. Oliver, "A model for repetitive firing in neurons," *Biol. Cybern.*, vol. 16, no. 1, pp. 53–64, 1974.

[75] R. J. MacGregor, *Neural and Brain Modeling*. San Diego, CA: Academic, 1987.

[76] J. K. Lin, K. Pawelzik, U. Ernst, and T. J. Sejnowski, "Irregular synchronous activity in stochastically-coupled networks of integrate-and-fire neurons," *Netw.: Comput. Neural Syst.*, vol. 9, no. 3, pp. 333–344, 1998.

[77] W. Swiercz, K. J. Cios, K. Staley, L. Kurgan, F. Accurso, and S. Sagel, "A new synaptic plasticity rule for networks of spiking neurons," *IEEE Trans. Neural Netw.*, vol. 17, no. 1, pp. 94–105, Jan. 2006.

[78] H. Lim and Y. Choe, "Extrapolative delay compensation through facilitating synapses and its relation to the flash-lag effect," *IEEE Trans. Neural Netw.*, vol. 19, no. 10, pp. 1678–1688, Oct. 2008.

[79] A. H. Klopf, *The Hedonistic Neuron: A Theory of Memory, Learning, and Intelligence*. Bristol, PA: Hemisphere, 1982.

[80] D. Kraines and V. Kraines, "The threshold of cooperation among adaptive agents: Pavlov and the stag hunt," in *Intelligent Agents III Agent Theories, Architectures, and Languages* (Lecture Notes in Computer Science), vol. 1193, J. P. Müller, M. Wooldridge, and N. R. Jennings, Eds. Berlin, Germany: Springer-Verlag, 1997, pp. 219–231.

[81] M. Lumsden, "The Cyprus conflict as a Prisoner's Dilemma game," *J. Confl. Resol.*, vol. 17, no. 1, pp. 7–32, Mar. 1973.

[82] B. A. Yesilada and A. Sozen, "Negotiating a resolution to the Cyprus problem: Is potential EU membership a blessing or a curse?" *J. Int. Negot.*, vol. 7, no. 2, pp. 261–285, 2002.

[83] R. Smith, M. Sola, and F. Spagnolo, "The Prisoner's Dilemma and regime-switching in the greek-turkish arms race," *J. Peace Res.*, vol. 37, no. 6, pp. 737–750, Nov. 2000.

[84] C. Christodoulou, G. Banfield, and A. Cleanthous, "Self-control with spiking and non-spiking neural networks playing games," *J. Physiol.-Paris*, vol. 104, nos. 3–4, pp. 108–117, May–Sep. 2010.

[85] A. Cleanthous and C. Christodoulou, "Is self-control a learned strategy employed by a reward maximizing brain?" *BMC Neurosci.*, vol. 10, no. 1, p. P14, 2009.

[86] S. M. McClure, D. I. Laibson, G. Loewenstein, and J. D. Cohen, "Separate neural systems value immediate and delayed monetary rewards," *Science*, vol. 306, no. 5695, pp. 503–507, Oct. 2004.

[87] G. S. Kavka, "Is individual choice less problematic than collective choice?" *Econ. Philos.*, vol. 7, no. 2, pp. 143–165, Oct. 1991.

[88] C. Christodoulou and A. Cleanthous, "Does high firing irregularity enhance learning?" *Neural Comput.*, vol. 23, no. 3, pp. 656–663, Mar. 2011.

Vassilis Vassiliades received the B.Sc. degree in computer science from the University of Cyprus, Nicosia, Cyprus, in 2007, and the M.Sc. degree (with distinction) in intelligent systems engineering, from the University of Birmingham, Birmingham, U.K., in 2008. He is currently pursuing the Ph.D. degree at the Department of Computer Science, University of Cyprus.

His current research interests include reinforcement learning, neuroevolution, multiagent systems, and computational intelligence.

Aristodemos Cleanthous received the B.Sc. degree in mathematics and economics from the London School of Economics and Political Science, London, U.K., in 2002, and the M.Sc. degree (with distinction) in computer science from the University College London, London, in 2005. In 2011, he received the Ph.D. degree in computational neuroscience from the University of Cyprus, Nicosia, Cyprus, studying the problem of self-control through computational modeling of internal conflict.

His research has been funded by Cyprus Research Promotion Foundation, Nicosia, and the University of Cyprus. His current research interests include spiking neural networks and reinforcement learning with special interest in multiagent reinforcement learning with spiking agents in game theoretical situations.

Chris Christodoulou received the B.Eng. degree in electronic engineering from Queen Mary and Westfield College, University of London, London, U.K., and the Ph.D. degree in neural networks/computational neuroscience from King's College, University of London. He also received the B.A. degree in German from Birkbeck College, University of London.

He was a Postgraduate Research Assistant from 1991 to 1995 and a Post-Doctoral Research Associate from 1995 to 1997 at the Center for Neural Networks, King's College, University of London. He joined Birkbeck College as a Lecturer in 1997, where he worked until 2005, and was also a Visiting Research Fellow at King's College from 1997 to 2001, and a Visiting Assistant Professor at the University of Cyprus, Nicosia, Cyprus, for one semester in 2001. In 2005, he joined the University of Cyprus as an Assistant Professor and became an Associate Professor in 2010. Since 2005, he is also a Visiting Research Fellow at Birkbeck College. His current research interests include computational and cognitive neuroscience, and neural networks.