# Interaction dynamics of two reinforcement learners

Walter J. Gutjahr[1]

[1]Dept. of Statistics and Decision Support Systems, University of Vienna
Universitaetsstrasse 5/9, A-1010 Wien, Austria
e-mail: walter.gutjahr@univie.ac.at

## Abstract

The paper investigates a stochastic model where two agents (persons, companies, institutions, states, software agents or other) learn interactive behavior in a series of alternating moves. Each agent is assumed to perform "stimulus–response–consequence" learning, as studied in psychology. In the presented model, the response of one agent to the other agent's move is both the stimulus for the other agent's next move and part of the consequence for the other agent's previous move. After deriving general properties of the model, especially concerning convergence to limit cycles, we concentrate on an asymptotic case where the learning rate tends to zero ("slow learning"). In this case, the dynamics can be described by a system of deterministic differential equations. For reward structures derived from $[2 \times 2]$ bimatrix games, fixed points are determined, and for the special case of the prisoner's dilemma, the dynamics is analyzed in more detail on the assumptions that both agents start with the same or with different reaction probabilities.

**Key words:** Dynamic systems, interaction dynamics, multiagent systems, prisoner's dilemma, reinforcement learning

## 1 Introduction

Learning has been studied in psychology, economic theory, cognitive science and artificial intelligence for a long time. Most of these investigations consider the learning process as an interaction between a single learning agent and an environment that is not learning itself. In the last decade, however, both in the "Learning in Games" approaches in game theory (see, e.g., Roth and Erev [27], Wedekind and Milinsky [33], Posch [25], Fudenberg and Levine [13], Hofbauer and Sigmund [17], Posch, Pichler and Sigmund [26], Greenwald, Friedman and Shenker [14], Laslier, Topol and Walliser [20], Hopkins [18]) and in multiagent reinforcement learning (MARL) theories (see, e.g., Littman [21], Sandholm and Crites [29], Claus and Boutilier [8], Hu and Wellman [19], Banerjee, Mukherjee and Sen [1]), the perspective has changed to a consideration of the dynamics of a system of "co-learning" agents, each of them being part of the environment of the other agent(s).

Such a framework shows more complex interaction patterns than one where the environment is assumed as stationary.

Our investigation in this paper differs both from the mainstream of the "Learning in Games" articles and from those in the literature on MARL. Whereas the "Learning in Games" publications are focused on *iterated games*, we study a series of interactions between two agents that cannot be conceived anymore as a sequence of separate "stage games", since the payoffs of the agents resulting from their moves are inter-linked in a certain way. Moreover, we assume *alternating moves* instead of simultaneous action decisions. In a game-theoretical description, this results in a single, non-decomposable extensive-form game.

Iterated-game models are often easier to analyze than alternating-moves models, but their application range has limitations. In an iterated-game model, in each stage, decisions have to be made simultaneously and independently from the decision(s) of the other agent(s). In many or even most real-life situations, however, there is no "synchronization clock" providing a structure of repeated mutually independent decisions. Rather than to be bound to make a decision simultaneously with another person, in everyday life, a person A can typically observe what the last action of a partner or competitor B has been, and reacts to it only after that. Then, person B is able to observe the (re)action of A before being forced to make the next own move. In total, this leads to an overlapping chain of actions/reactions.

Let us give a few examples of situations where the alternating-moves perspective, which fits to the *dialogue* character of human communication, is more appropriate than a simultaneous-moves point of view.

*1. Negotiations.* Wherever negotiations between two persons or organizations take place, they typically follow an interactive pattern with alternating-move structure. This is true for simple bargaining in street markets or bazars as well as for complex negotiation processes between firms or political institutions. Also the classical game-theoretical bargaining models (e.g., the Rubinstein Model [28]) mimic this sequential, alternating structure.

*2. Couple interactions.* In marital therapy, interactions between couples and possible change by learning effects have been thoroughly studied (see, e.g., [30]). Usually, couple interactions are described in the form of alternating actions, irrespectively of whether the level of behavior under consideration is verbal or non-verbal. The psychological literature addresses also the difficulties and conflicts arising from the fact that "punctuations" of interactions, i.e., ways of grouping them to action–reaction sequences, are not unique (cf. [4]). In our terms, this issue is a consequence of the non-decomposability of the overall interaction game into stage games.

*3. International conflicts.* Situations of bilateral international conflicts follow a path of escalation or de-escalation characterized by diplomatic, economic, legislative, military or other actions of both countries which coincide in time only in exceptional cases; the process can better be represented by an alternating sequence of measures. In the literature on conflict research,

game-theoretical models for, e.g., the process of arms race are available (see [32] as an example), but usually they resort to the simpler assumption of an iterated game.

*4. Project cooperation.* If two partners or partner institutions collaborate in a joint project, the development of long-term cooperation or defection can be modelled by traditional game-theoretic approaches. Also in this context, however, one can hardly imagine a mechanism that synchronizes the project collaboration in such a way that the partners have to make (or even are able to make) their decisions always simultaneously and independently from each other. Rather than that, there will be an alternating sequence of actions each of which, after having been interpreted as cooperative or non-cooperative, influences the other partner's next decision.

Alternating-moves models have been investigated in the literature, particularly in the special context of the prisoner's dilemma (see, e.g., Nowak and Sigmund [24] or Neill [23]), but usually they have been developed within a replicator dynamics framework, i.e., in the perspective of the genetic evolution of a population, rather than in a context of learning. In contrast to that, the present article addresses the dynamics of alternating moves governed by a (mutual) reinforcement learning process. Moreover, in [24], the payoff for a move is assumed to be independent of previous moves. As mentioned, in our model, the payoffs can be inter-linked.

Now let us consider the difference of our own approach to the MARL approaches. It concerns another point: The MARL publications are motivated by the development of efficient learning algorithms for software agents in multiagent systems, e.g., in an internet environment. Sophisticated algorithms of this type, mainly based on the Q-learning concept from Markov decision processes, have been presented, tested experimentally and analyzed theoretically (for examples, see [8] or [19]). Nevertheless, because of their mathematical complexity, it is unlikely that *humans* apply such learning algorithms in their economic or social behavior. In order to be able to develop a theory that is not only applicable to software agents but also to the analysis of interactions between humans, we have decided to study a simple, basic learning mechanism that is closely related to the concepts of psychological learning theories. A similar, slightly less general model has been presented in Eder, Gutjahr and Neuwirth [10] and Gutjahr and Eder [15]; some additional observations are provided in Eder and Gutjahr [11]. In the present paper, a modified, extended version of the mentioned model is analyzed in much more depth, mainly on the "slow learning" assumption described further below.

The key idea of the model is that the basic stimulus-response-consequence (S-R-C) scheme of *discriminant learning* is applied to a dyadic situation of two learning agents. Each of them learns according to the S-R-C scheme: The stimulus is given by the last action of the other agent, the response is the own action of the considered agent, and the consequence results from the relation of the *next* action of the other agent to stimulus and response. Learning takes place via a *memory matrix* (called "transition probability

matrix" in [10]) containing the probabilities of reacting to a certain stimulus by a certain response. A favorable consequence increases the probability of the previous stimulus-reaction sequence, a process that is usually called *reinforcement*. On the other hand, the probability of a not reinforced stimulus-reaction sequence decreases.

The amount of reinforcement is governed by a *learning rate* $\lambda$. Contrary to the cited literature on MARL, we keep $\lambda$ fixed over time. A high value of $\lambda$ introduces a large amount of randomness into the development of the system. To get a clearer picture of the system dynamics, it is convenient to consider low values of $\lambda$ ("slow learners"). We do this in this paper by studying an asymptotic case where $\lambda \to 0$, compensated by a suitable scaling of the time axis. In this way, deterministic differential equations for the elements of the memory matrix are obtained. In its flavor, this approach is related to the deterministic approximation of stochastic games by Börgers and Sarin [5] or by Benaïm and Weibull [2], but differs from these articles insofar as the iterated game structure used there is replaced by the S-R-C structure explained above.

The organization of the paper is the following: In Section 2, the model is presented in formal terms, and a general result on the limiting behavior of the system is derived. Section 3 contains the basic results on "slow learners", and Section 4 applies these results to the special case of learning processes derived from $[2 \times 2]$ bimatrix games. In particular, fixed points of the dynamics are determined, and the process in the case of identical and of different initial memory matrices for both agents is investigated. Section 5 contains concluding remarks.

# 2 The Model and its Basic Properties

## 2.1 Formal Description

Two agents, denoted by 1 and 2, are considered. To each agent $k$ ($k = 1, 2$), a finite *action set* is assigned. We can identify the possible actions of an agent with their indices in the action set, i.e., $\{1, \ldots, N_1\}$ is the action set of agent 1, and $\{1, \ldots, N_2\}$ is the action set of agent 2, where $N_1$ and $N_2$ are positive integers.

The two agents choose their actions in *alternating moves*. In round 0, agent 1 starts by choosing action $i_0$. Agent 2 observes $i_0$ and chooses then its action $j_0$. Round 0 is finished.[1] In round 1, agent 1, which has observed action $j_0$, chooses action $i_1$. After observing $i_1$, agent 2 chooses its action $j_1$. Round 1 is finished. Etc.

To agent 1 and 2, *reward arrays* $A$ resp. $B$ are assigned. $A$ and $B$ are

[1] We start counting the rounds with index 0 instead of 1, since the initial round 0 forms an exception in the sense that no learning takes place in it yet, as will be explained below.

3-dimensional arrays with nonnegative elements

$$a(j', i, j) \quad (j' = 1, \ldots, N_2;\ i = 1, \ldots, N_1;\ j = 1, \ldots, N_2)$$

resp.

$$b(i, j, i') \quad (i = 1, \ldots, N_1;\ j = 1, \ldots, N_2;\ i' = 1, \ldots, N_1).$$

The number $a(j', i, j)$ denotes the reward agent 1 obtains as a consequence of its action $i$ in the current round, if the action of agent 2 in the previous round has been $j'$, and the action of agent 2 in the current round (following agent 1's action) is $j$. The number $b(i, j, i')$ denotes the reward agent 2 obtains as a consequence of its action $j$ in the current round, if the action of agent 1 in the current round has been $i$, and the action of agent 1 in the next round is $i'$. Schematically, this is shown in Table 1.

| round | agent 1 | agent 2 | reward for 1 | reward for 2 |
|---|---|---|---|---|
| $n-1$ | $i_{n-1}$ | | | $b(i_{n-2}, j_{n-2}, i_{n-1})$ |
| | | $j_{n-1}$ | $a(j_{n-2}, i_{n-1}, j_{n-1})$ | |
| $n$ | $i_n$ | | | $b(i_{n-1}, j_{n-1}, i_n)$ |
| | | $j_n$ | $a(j_{n-1}, i_n, j_n)$ | |
| $n+1$ | $i_{n+1}$ | | | $b(i_n, j_n, i_{n+1})$ |
| | | $j_{n+1}$ | $a(j_n, i_{n+1}, j_{n+1})$ | |

*Table 1.* A sequence of moves and rewards.

The reward arrays $A$ resp. $B$ can also be represented by the following $N_2$ resp. $N_1$ matrices

$$A^{(j')} = (a(j', i, j))_{i=1,\ldots,N_1;\, j=1,\ldots,N_2} \quad (j' = 1, \ldots, N_2),$$

$$B^{(i)} = (b(i, j, i'))_{j=1,\ldots,N_2;\, i'=1,\ldots,N_1} \quad (i = 1, \ldots, N_1). \tag{1}$$

The purpose of the described reward determination rule is to mimic the S-R-C scheme (see Section 1) for each agent: For agent 1 in round $n$, the previous action $j_{n-1}$ of agent 2 has been the stimulus (S), and its own action $i$ is the response (R) to that stimulus. The consequence (C) basically depends on the subsequent action $j_n$ of agent 2, however (as we shall see below), the model gains flexibility if the consequence is also allowed to depend on $i_n$ and even $j_{n-1}$. (In the model of [10], dependence on $j_{n-1}$ is not allowed.) For agent 2 in round $n$, the roles are interchanged: The previous action $i_n$ of agent 1 is the stimulus, $j_n$ is the response, and the consequence depends on agent 1's next action $i_{n+1}$, but can also be influenced by $j_n$ and $i_n$.

In round 0, for the first action $i_0$ of agent 1, no previous action of the other agent exists, so in cases where $a(j', i, j)$ actually depends on $j'$, we

make the reward rule well-defined by deciding that action $i_0$ gets no reward: Setting $j_{-1} = 0$ (the "zero action"), let $a(0, i_0, j_0) = 0$. This modification is not necessary in cases where $a(j', i, j)$ is actually independent of $j'$.

The model above can be extended to a *game* by defining total payoffs for the two agents. Of course, adding the rewards would yield infinite sums in general. As usual in iterated games, however, we can consider *discounted rewards* as the total payoffs. An alternative way to define total payoffs is to restrict the strategy sets of both agents to classes of (mixed) strategies guaranteeing that the action sequence converges to a steady state, and to take the average rewards in this steady state as the total payoff.

As stated in the Introduction, we assume that both agents adapt their response behavior in a *learning process*: Agents 1 and 2 use *memory matrices* $X_n$ and $Y_n$, respectively, where, for round $n = 0, 1, \ldots$, memory matrix $X_n$ contains elements $x_n(j, i)$ $(j = 1, \ldots, N_2;\ i = 1, \ldots, N_1)$, and memory matrix $Y_n$ contains elements $y_n(i, j)$ $(i = 1, \ldots, N_1; j = 1, \ldots, N_2)$. Element $x_n(j, i)$ denotes the probability that agent 1 reacts in round $n$ to action (stimulus) $j$ of agent 2 (chosen in round $n - 1$) by action (response) $i$. Element $y_n(i, j)$ denotes the probability that agent 2 reacts in round $n$ on action (stimulus) $i$ of agent 1 by action (response) $j$. We always assume $x_n(j, i) \geq 0$ and $y_n(i, j) \geq 0$ for all $i, j, n$, and

$$\sum_{i=1}^{N_1} x_n(j, i) = 1 \ (j = 1, \ldots, N_2), \quad \sum_{j=1}^{N_2} y_n(i, j) = 1 \ (i = 1, \ldots, N_1) \tag{2}$$

for all $n$. As the subscript $n$ indicates, the memory matrices depend on the current round. The initial matrices $X_0$ and $Y_0$ are arbitrary, provided that conditions above are satisfied. The update is performed according to the following *reinforcement rules:*

$$x_{n+1}(j_{n-1}, i_n) \ = \ x_n(j_{n-1}, i_n) + \lambda a(j_{n-1}, i_n, j_n)\,(1 - x_n(j_{n-1}, i_n)), \tag{3}$$

$$x_{n+1}(j_{n-1}, i) \ = \ x_n(j_{n-1}, i)\,(1 - \lambda a(j_{n-1}, i_n, j_n)) \quad (i \neq i_n), \tag{4}$$

$$x_{n+1}(j, i) \ = \ x_n(j, i) \quad (j \neq j_{n-1},\ i = 1, \ldots, N_1), \tag{5}$$

$$y_{n+1}(i_n, j_n) \ = \ y_n(i_n, j_n) + \lambda b(i_n, j_n, i_{n+1})\,(1 - y_n(i_n, j_n)), \tag{6}$$

$$y_{n+1}(i_n, j) \ = \ y_n(i_n, j)\,(1 - \lambda b(i_n, j_n, i_{n+1})) \quad (j \neq j_n), \tag{7}$$

$$y_{n+1}(i, j) \ = \ y_n(i, j) \quad (i \neq i_n,\ j = 1, \ldots, N_2). \tag{8}$$

Therein, $\lambda > 0$ is the *learning rate*. We always choose $\lambda$ smaller than all reciprocals $1/a(j', i, j)$ and $1/b(i, j, i')$ of nonzero elements in the reward arrays.

Let us give some historical remarks concerning the formulas above. Basically, the applied model goes back to the well-known *Bush-Mosteller linear reinforcement scheme* (Bush and Mosteller [7]), the oldest and simplest model for reinforcement learning. However, we use a more recent modification of this scheme which has been developed during the last decades

in several steps. In their model, Bush and Mosteller do not distinguish between different amounts of reward; they only consider the two possible cases of success and failure. Cross [9] extended the Bush-Mosteller scheme to the case where rewards can be of a different size, effecting reinforcements of different strengths. As well as in [7], also in [9], the reinforcement process is not described in a game-theoretical setting, but in a context where a single decision maker is faced with a stochastic, but stationary environment. Börgers and Sarin [5] take a game-theoretic viewpoint by investigating co-learning of two agents in an iterated stage game, each of the agents being equipped with a reinforcement mechanism defined by the Cross scheme. As the older cited models, the Börgers-Sarin model has the property that reinforcement is based exclusively on the results of the last round; rewards of previous rounds do not enter into the reinforcement computation. This is contrary to *melioration models* (see Herrnstein and Prelec [16]) which take the *average* reward in a fixed number of rounds preceding the current round as the basis for the computation of the reinforcement. In [6], Brenner compares the Bush-Mosteller type models with the melioration models and shows that both types tend to equivalent behavior as time goes to infinity. Therefore, we do not use the melioration approach here.

The main difference of our reinforcement scheme eqs. (3) – (8) to that in [5] or [6] is that by applying the S-R-C discriminant learning model, instead of the probability $x_n(i)$ of choosing action $i$, the probability $x_n(j,i)$ of reacting to action $j$ by action $i$ is stored and updated. This also necessitates the addition of formulas (5) and (7) which state that for actions that did not occur in the previous round, response probabilities remain unchanged. A further extension is that in our model, the reward is allowed to depend not only on the events in the current round, but, again following the three-step S-R-C scheme, also (partially) on the previous round.[2]

Let us add a short comment on the application of *discriminant* learning. If $x_n(i)$ would be used instead of $x_n(j,i)$, such that the probability of an action $i$ were not allowed to depend on the previous action $j$ of the other agent, it would be impossible to correctly represent some frequently occurring situations of real life. This holds already in a context where persons are confronted with the environment instead of other persons. Humans are able to learn a quite different behavior for the case where the traffic light has just switched to red than for the case where it has just switched to green. Representing the currently learned tendency of crossing a certain street by only *one* probability value would not be appropriate; the dependency of the behavior on the *stimulus* has to be taken into account. Similarly, in order not to over-simplify interpersonal learning effects, modelling the ability of making one's reaction dependent on the previous action of the other person seems indispensable. To give a trivial example, for most people, being complimented and being criticized certainly triggers different reactions.

In game-theoretic terms, the use of the probabilities $x_n(j,i)$ and $y_n(i,j)$

[2] In [10], [15] and [11], a different, but for $\lambda \to 0$ asymptotically equivalent reinforcement scheme has been used, although with less flexibility in the definition of the reward.

(reflecting discriminant learning) amounts to the introduction of *conditional strategies* which are behavioral rules that take at least the actions in the previous round, perhaps also those in former rounds, into consideration. Such strategies have already been studied in the literature, especially in the context of the prisoner's dilemma; examples are [24] or [23]. What distinguishes our model from these publications is (i) the reinforcement learning approach which replaces the replicator dynamics approach chosen there, and (ii) the fact that we do not only allow *strategic* dependencies between successive rounds by considering $x_n(j,i)$ and $y_n(i,j)$, but also three-step *reward* dependencies by considering $a(j_{n-1},i_n,j_n)$ and $b(i_n,j_n,i_{n+1})$. The motivation for the last-mentioned extension is that it allows us to generalize the alternating-moves consideration in a natural way from special forms of the prisoner's dilemma to bimatrix games with arbitrary structure (see Subsection 2.2 below).

We are interested in the following stochastic dynamic process: In round $n$ ($n = 1, 2, \ldots$),

- agent 1 chooses action $i_n$ randomly according to the probability vector
$$(x_n(j_{n-1},1),\ldots,x_n(j_{n-1},N_1)),$$
- based on $i_{n-1}$, $j_{n-1}$ and $i_n$, agent 2 updates $Y_{n-1}$ to $Y_n$, using (6)–(8),
- agent 2 chooses action $j_n$ randomly according to the probability vector
$$(y_n(i_n,1),\ldots,y_n(i_n,N_2)),$$
- based on $j_{n-1}$, $i_n$ and $j_n$, agent 1 updates $X_n$ to $X_{n+1}$, using (3)–(5).

The initial round 0 needs a specific protocol (in particular, learning does not yet take place in this round):

- Agent 1 selects a random line $j$ of matrix $X_0$, where each $j = 1,\ldots,N_2$ has the same probability, and chooses action $i_0$ according to the probability vector
$$(x_0(j,1),\ldots,x_0(j,N_1))$$
(note that $j_{n-1}$ is not specified for $n = 0$, therefore it is chosen randomly),
- agent 2 chooses action $j_0$ randomly according to the probability vector
$$(y_0(i_0,1),\ldots,y_0(i_0,N_2)).$$

Throughout the paper, we restrict ourselves to *nonnegative* rewards $a(j',i,j)$ and $b(i,j,i')$. In principle, the model equations could be extended to possibly negative rewards. This extension is not considered here. Readers interested in models for possibly negative rewards are referred to Bereby-Meyer and Erev [3]. For the effect of adding a constant to all rewards, cf. Erev, Bereby-Meyer and Roth [12].

As already stated in Section 1, the game underlying our model does not fit into the framework of *iterated games*: subsequent rounds (stages) are not independent from each other. However, let us remark without proof that the considered game (with discounted payoffs) can be represented as a special case within the broader framework of *Markov games* (see [21] or [19]), a generalization of Markov decision processes to more than one agent, where *states* can be used to propagate information between rounds.

**Example 2.1.** Two firms 1 and 2 decide between cooperative (action 1) and competing (action 2) behavior. If firm 1 has cooperated, it judges a cooperative response of firm 2 as a fortunate event (accepted offer), and a competing response of firm 2 as a *very* unfortunate event (exploitation of good-will). If firm 1 has competed, it judges a cooperative response of firm 2 as a *very* fortunate event (positive surprise), and a competing response of firm 2 as unfortunate only to a slighter degree (expected reciprocal competition). Analogously for firm 2. This yields the following reward arrays (formulated by the matrices of eq. (1)):

$$A^{(1)} = A^{(2)} = B^{(1)} = B^{(2)} = \begin{pmatrix} 2 & 0 \\ 3 & 1 \end{pmatrix}. \tag{9}$$

Note that in this case, $a(j', i, j)$ is actually independent of the previous action $j'$ of the other agent, and also $b(i, j, i')$ is independent of $i$. Let us set $\lambda = 0.1$, and assume that both firms start with equal probabilities for cooperation and competition:

$$X_0 = Y_0 = \begin{pmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{pmatrix}. \tag{10}$$

In round 0, firm 1 chooses action $i_0$ with equal probabilities from the set $\{1, 2\}$. Say the result is $i_0 = 2$ (competition). Next, firm 2 chooses action $j_0$ with equal probabilities from $\{1, 2\}$, which yields, say, $j_0 = 1$ (cooperation). At the beginning of round 1, the memory matrices are still unchanged, so firm 1 chooses a random action from $\{1, 2\}$, again with equal probabilities. Suppose this yields $i_1 = 1$ (cooperation). Now, firm 2 gets a reward $b(2, 1, 1) = b^{(2)}(1, 1) = 2$ which reinforces the reaction to stimulus 2 by response 1: According to (6)–(8),

$$y_1(2, 1) = 0.5 + 0.1 \cdot 2 \cdot 0.5 = 0.6, \quad y_1(2, 2) = 0.5 \cdot (1 - 0.1 \cdot 2) = 0.4,$$

whereas $y_1(1, 1) = y_0(1, 1) = 0.5$ and $y_1(1, 2) = y_0(1, 2) = 0.5$. Firm 2 has "learned" to react rather cooperatively to competition, but reacts to cooperation still with the initial probabilities, so its next action $j_1$ (the response to action $i_1 = 1$) is chosen with equal probabilities from $\{1, 2\}$. Say the result is $j_1 = 1$ (cooperation). This yields a reward of $a(1, 1, 1) = a^{(1)}(1, 1) = 2$ for firm 1, such that, analogously as before, $x_2(1, 1) = 0.6$ and $x_2(1, 2) = 0.4$, whereas $x_2(2, 1) = x_2(2, 2)$ remain equal to 0.5. Firm 1 has "learned" to respond to cooperation rather by cooperation. Therefore, in round 2, firm 1 chooses action 1 with the increased probability $x_2(1, 1) = 0.6$, etc.

## 2.2 Derivation of reward arrays from payoff bimatrices

For mainly psychologically determined situations, values in the reward arrays can often be guessed directly by estimating emotional reactions of the two involved persons or parties (cf. Example 2.1). For economic applications, however, where rationality plays a more important (though usually not exclusive) role, it is convenient to derive reward arrays from the well-established game-theoretical modelling concept of *payoff (bi-)matrices*. Reward arrays derived from payoff matrices, however, are only a special case of general reward arrays.

Let us consider a bimatrix game with payoff matrix $(p^{(k)}(i,j))$ $(i = 1,\dots,N_1;\ j = 1,\dots,N_2)$ for agent $k$ $(k = 1,2)$. The game is played in alternating moves: in round $n$, agent 1 chooses a line $i_n$, then agent 2 chooses a column $j_n$ $(n = 0,1,\dots)$. At each time, the *current state* of the interaction process depends on the last decisions of the two agents: During the first part of round $n$ (before agent 2 has acted) this state is $(i_n, j_{n-1})$; during the second part, after agent 2 has updated its decision to $j_n$, the state is $(i_n, j_n)$.

Now, let us assume that while being in a certain state, both agents obtain per time unit the payoffs associated with this state, i.e., agent $k$ gets a payoff of $p^{(k)}(i_n, j_{n-1})$ per time unit in the first part of round $n$ and a payoff of $p^{(k)}(i_n, j_n)$ in the second part of round $n$ $(k = 1,2)$. In the simplest case, we can assume that the two parts of round $n$ take the same time, such that the payoffs $p^{(k)}(i_n, j_{n-1})$ and $p^{(k)}(i_n, j_n)$ have equal weights; however, it is also possible to model situations where the durations are different or even depend on the chosen decisions (e.g., an agent may decide to react quickly or by an action with a preceding delay). In the case of equal weights — say: one unit for each of the two parts of the round —, the total reward agent 1 obtains after its action $i_n$ in round $n$ and before its next action $i_{n+1}$ in round $n+1$ is

$$a(j_{n-1}, i_n, j_n) = p^{(1)}(i_n, j_{n-1}) + p^{(1)}(i_n, j_n), \tag{11}$$

and the total reward agent 2 obtains after its action $j_n$ in round $n$ and before its next action $j_{n+1}$ in round $n+1$ is

$$b(i_n, j_n, i_{n+1}) = p^{(2)}(i_n, j_n) + p^{(2)}(i_{n+1}, j_n). \tag{12}$$

**Example 2.2.** Consider a *prisoner's dilemma* with payoff bimatrix

$$P = ((p^{(1)}(i,j),\, p^{(2)}(i,j)) = \begin{pmatrix} (4,4) & (0,5) \\ (5,0) & (1,1) \end{pmatrix}, \tag{13}$$

where agent 1 is the line player, agent 2 is the column player, the first action is "cooperate" and the second action is "defect". Suppose the sequence of actions $i_0 = 1,\ j_0 = 1,\ i_1 = 2,\ j_1 = 2,\ i_2 = 1,\ j_2 = 2,\dots$ has been chosen. In round 0, state $(1,1)$ (both agents cooperate) is established. In round 1, agent 1 changes this state by its action $i_1 = 2$ to $(2,1)$ (agent 1 now defects,

but agent 2 still cooperates), which gives agent 1 a reward of 5 units during the first part of round 1. However, agent 2 then decides to react by defecting ($j_1 = 2$), thus shifting the state to $(2, 2)$ (both agents defect), such that in the second part of round 1, agent 1 only obtains a reward of 1 unit. So, the overall reward of agent 1 following its decision in round 1 (preceding its next decision in round 2) is $a(1, 2, 2) = 5 + 1 = 6$. Similarly for agent 2: its overall reward following its decision in round 1 (preceding its next decision in round 2) is $b(2, 2, 1) = 1 + 5 = 6$, etc. In total, using (11) and the matrix representation (1), we obtain the reward array for agent 1 as

$$A^{(1)} = \begin{pmatrix} 8 & 4 \\ 10 & 6 \end{pmatrix}, \qquad A^{(2)} = \begin{pmatrix} 4 & 0 \\ 6 & 2 \end{pmatrix},$$

and $B^{(1)} = A^{(1)}$, $B^{(2)} = A^{(2)}$ because of symmetry. The main difference to Example 2.1 is that now $A^{(1)} \neq A^{(2)}$, i.e., the reward depends on the previous action of the other agent: if it has cooperated before, the reward is higher.

## 2.3 Basic Properties of the Process

For the slightly different learning model treated in [10], [15] and [11], it has been observed in the last-mentioned article that the corresponding stochastic dynamical process can be interpreted in a Markov process framework, and that on certain additional assumptions, it always converges to some *limit cycle*. Here, we show that these observations are also valid for the model considered in this paper.

**Proposition 2.1.** The two stochastic processes with the quadruples

$$(j_{n-1}, X_n, i_n, Y_n) \quad \text{resp.} \quad (i_{n-1}, Y_{n-1}, j_{n-1}, X_n)$$

as states are Markov processes in discrete time. (Proof: see Appendix.)

Now let us turn to the limiting behavior of the process as $n \to \infty$, which is characterized by a steady-state distribution of the first or second Markov process described above. In view of Proposition 2.1, such a steady-state distribution can be of two types:

*Type (a):* In the steady-state situation, $X_n$ and $Y_n$ have point mass distributions, i.e., $x_n(j, i)$ and $y_n(i, j)$ are, for each pair $(i, j)$, fixed values independent of $n$.

*Type (b):* The distribution of $X_n$ or that of $Y_n$ (or both) is not a point mass.

In computational experiments, only steady-state distributions of type (a) have been observed.[3] For this type, the following assertion can be made:

**Proposition 2.2.** Let $a(j', i, j) > 0$ and $b(i, j, i') > 0$ for all indices $j'$, $i$, $j$, $i'$. Then a steady-state distribution of type (a) has the property that there

[3] We conjecture that only type (a) steady-state distributions exist, but cannot prove this at the moment.

are index sets $I \subseteq \{1, \ldots, N_1\}$ and $J \subseteq \{1, \ldots, N_2\}$ and bijective functions $\varphi_1 : J \to I$ and $\varphi_2 : I \to J$, such that agent 1 resp. 2 chooses actions from $I$ resp. $J$ only, $\varphi_1(j)$ is the fixed response (chosen with probability 1) of agent 1 to action $j \in J$ of agent 2, and $\varphi_2(i)$ is the fixed response (chosen with probability 1) of agent 2 to action $i \in I$ of agent 1. (Proof: see Appendix.)

Given its conditions are satisfied, Proposition 2.2. predicts a deterministic *limit cycle*

$$i^{(1)} \ \to_{\varphi_2} \ j^{(1)} \ \to_{\varphi_1} \ i^{(2)} \ \to_{\varphi_2} \ j^{(2)} \ \to_{\varphi_1} \ \ldots \ \to_{\varphi_2} \ j^{(t)} \ \to_{\varphi_1} \ i^{(1)}$$

as the finally resulting interaction sequence: After some time, the agents run through the same loop of alternating moves again and again. Interestingly, stochasticity has vanished in the limiting behavior. Note, however, that this does not mean that the probabilities $x_n(j, i)$ and $y_n(i, j)$ need to degenerate to values 0 or 1 for *all* index pairs $(j, i)$ resp. $(i, j)$. Only those that are actually "played" in the limiting case must degenerate.

# 3 Slow Learning: General Properties

As our computer experiments in [10], [11] and [11] with a closely related model show, the outcome of learning at a medium-sized rate $\lambda$ is qualitatively not very different from that of slow learning (small $\lambda$), but simply biased by random noise. So it seems most interesting to study the asymptotic case where $\lambda$ tends to zero, compensated by a growing frequency of interactions per time unit. (For a similar consideration, see [5].) As we shall see, randomness disappears in this asymptotic case in some sense: The dynamical process of the matrices $X_n$ and $Y_n$ (albeit not including the actions $i_n$ and $j_n$) approaches a deterministic process that only depends on $X_0$ and $Y_0$ and, of course, on the reward arrays.

Now let us outline the indicated asymptotics in detail. We shall define a second, related process obtained from the original process by four successive approximation steps, each of which is justified by the mentioned asymptotic consideration. First, we compose each sequence of $M$ rounds to a *period:* Period 1 contains rounds 0 to $M-1$, period 2 rounds $M$ to $2M-1$, etc. Let $\lambda = \epsilon/M$, where $\epsilon \ll 1$. Then, the sum of probability changes $x_{n+1}(j, i) - x_n(j, i)$ according to (3) is of order $O(M \cdot \lambda) = O(\epsilon)$, hence small compared with unity (the sum of the probabilities $x_n(j, i)$ in a line $j$ of the memory matrix). Therefore, in a first-order approximation, $x_{n+1}(j, i) \approx x_n(j, i)$ during the period, i.e., the memory matrix $X_n$ does not essentially change. The same holds for $Y_n$. In approximation step 1, we consider a process where $X_n$ and $Y_n$ are held *exactly* constant for the $M$ rounds $(p-1)M, \ldots, pM-1$ of period $p$ by resetting $X_{n+1} = X_n$ after the increment $\Delta X_n = X_{n+1} - X_n$ has been computed from eqs. (3) – (5), and analogously for $Y_n$. The increments $\Delta X_n$ and $\Delta Y_n$ are cumulated and added to $X_n$ resp. $Y_n$ only after period $p$ is over.

Next, the time axis is scaled in such a way that a period is assumed to take $\epsilon$ time, i.e., $K = 1/\epsilon = 1/(M\lambda)$ periods are executed during one time unit. If we prefer to consider the number $K$ of periods per time unit as the control parameter and to derive from it the learning rate $\lambda$, we can also say that $\lambda$ is set to the value $1/(MK)$. Thus, the more rounds are performed during a time unit, the lower has the learning rate $\lambda$ to be chosen in order to keep the overall change during one time unit always of an "observable" size. Obviously, also the parameter $M$ can be chosen arbitrarily large in this consideration, larger values of $M$ leading to even smaller values of $\lambda$. Let $x(j,i)(t)$ and $y(i,j)(t)$ denote the values of the variables $x_n(j,i)$ resp. $y_n(i,j)$ in the period corresponding to time $t$ in the time scaling described above. For the sake of brevity, we shall simply write $x(j,i)$ resp. $y(i,j)$ in the sequel. The numbers $x(j,i)$ and $y(i,j)$ represent the transition probabilities of action $j$ of agent 2 to action $i$ of agent 1, resp. of of action $i$ of agent 1 to action $j$ of agent 2, in the period corresponding to time $t$.

In the following, we let $M \to \infty$. This has two consequences: One the one hand, the actions $i_n$ and $j_n$, which form a homogeneous Markov chain during period $p$ by approximation step 1, are performed sufficiently often such that the Markov chain not only reaches its steady state[4], but even remains in this steady state (or, more precisely, in a distribution close to the steady state) for "almost all" time of period $p$. Therefore, it is justified to consider a second (additional) approximation step consisting in the assumption that in each period $p$, the distribution of the actions $i_n$ and $j_n$ is *exactly* their steady-state distribution throughout the entire period.

The other consequence of assuming $M$ as a large number is that the contributions of the increments $\Delta X_n$ and $\Delta Y_n$, which are independent during period $p$ by approximation step 1, follow the Law of Large Numbers. Therefore, in our third approximation step, we can assume that what is added to $X_n$ resp. $Y_n$ after period $p$ (representing the cumulative effect of the considered increments) is the *expected value* of the sum of the increments $\Delta X_n$ and $\Delta Y_n$ instead of the sum itself. Approximation steps 1 to 3 transform the original process into a discrete *deterministic* process.

Finally, in the fourth approximation step, we let $K \to \infty$, such that the duration $dt = \epsilon$ of a period tends to zero, which approximates the mentioned discrete deterministic process by its continuous counterpart. The resulting process will be called the *counterpart process* to the original process. It can be interpreted as a deterministic, time-continuous approximation to the original stochastic process for the case of many rounds per time unit and low learning rate in each round.

The expressions

$$\dot{x}(j,i) = \frac{dx(j,i)}{dt}, \quad \dot{y}(i,j) = \frac{dy(i,j)}{dt}$$

denote the differential quotients of the variables $x(j,i)$ resp. $y(i,j)$ of the

[4]Existence and uniqueness of the steady state are not trivial here, but require some additional technical arguments, see the proof of Proposition 3.1 in the Appendix.

counterpart process with respect to time $t$. Furthermore, let us introduce

$$\alpha(j', i) \;=\; \sum_{j=1}^{N_2} y(i,j)\, a(j', i, j) \tag{14}$$

denoting the expected reward achieved by agent 1 for reaction $i$ to $j'$, and

$$\beta(i,j) = \sum_{i'=1}^{N_1} x(j,i')\, b(i,j,i') \tag{15}$$

denoting the expected reward achieved by agent 2 for reaction $j$ to $i$. It should be observed that the values $\alpha(\cdot,\cdot)$ and $\beta(\cdot,\cdot)$ implicitly contain the probabilities $y(\cdot,\cdot)$ resp. $x(\cdot,\cdot)$.

**Proposition 3.1.** For nonzero starting probabilities $x_0(j,i)$ and $y_0(i,j)$, the counterpart process is characterized by the following system of ordinary differential equations:

$$\dot{x}(j,i) = \rho(j)\, x(j,i) \left[\alpha(j,i) - \sum_{m=1}^{N_1} x(j,m)\, \alpha(j,m)\right], \tag{16}$$

$$\dot{y}(i,j) = \pi(i)\, y(i,j) \left[\beta(i,j) - \sum_{m=1}^{N_2} y(i,m)\, \beta(i,m)\right], \tag{17}$$

where $i = 1,\ldots,N_1$, $j = 1,\ldots,N_2$, $\pi(i)$ resp. $\rho(j)$ are the steady-state probabilities of action $i$ by agent 1 resp. action $j$ by agent 2 under given transition probabilities $x(j,i)$ and $y(i,j)$, and $\alpha(j,i)$ resp. $\beta(i,j)$ are the expected rewards (14) resp. (15) for the reactions of the two agents under given $x(j,i)$ and $y(i,j)$. (Proof: see Appendix.)

The steady-state probabilities $\pi(i)$ and $\rho(j)$ can be expressed as functions of the variables $x(j,i)$ and $y(i,j)$ by solving a system of linear equations: In steady state, for an action $i$ of agent 1, the balance equation

$$\rho(1)\cdot x(1,i) + \ldots + \rho(N_2)\cdot x(N_2,i) \;=\; \pi(i) \quad (i = 1,\ldots,N_1) \tag{18}$$

must hold. Similarly, for an action $j$ of agent 2, the balance equation

$$\pi(1)\cdot y(1,j) + \ldots + \pi(N_1)\cdot y(N_1,j) \;=\; \rho(j) \quad (j = 1\ldots,N_2) \tag{19}$$

must hold. These $N_1+N_2$ equations are not linearly independent. Summing up all equations in (18), we get $1 = 1$, and the same for (19). So we can omit both the last equation of (18) and that of (19), and set

$$\pi(N_1) = 1 - \sum_{i=1}^{N_1-1} \pi(i), \quad \rho(N_2) = 1 - \sum_{j=1}^{N_2-1} \rho(j).$$

This yields the following linear system of $N_1+N_2-2$ equations in the same number of variables $\rho(1),\ldots,\rho(N_2-1)$, $\pi(1),\ldots,\pi(N_1-1)$:

$$\begin{aligned}&[x(N_2,i)-x(1,i)]\rho(1)+\ldots+[x(N_2,i)-x(N_2-1,i)]\rho(N_2-1)+\pi(i)=x(N_2,i)\\&[y(N_1,j)-y(1,j)]\pi(1)+\ldots+[y(N_1,j)-y(N_1-1,j)]\pi(N_1-1)+\rho(j)=y(N_1,j)\end{aligned}\tag{20}$$

for $i=1,\ldots,N_1-1$ resp. for $j=1,\ldots,N_2-1$. For given numbers $N_1$ and $N_2$, the desired functional representation of $\pi(i)$ and $\rho(j)$ in the variables $x(j,i)$ and $y(i,j)$ can be computed from (20) by Cramer's rule. (This will be done in Section 4 for the case of $N_1=N_2=2$.) By using Cramer's rule, it can also be shown that the velocity vector $(\ldots,\dot{x}(j,i),\ldots,\dot{y}(i,j),\ldots)^t$ of the process is always proportional to a vector of polynomials in the variables $x(\cdot,\cdot)$ and $y(\cdot,\cdot)$.

It should be emphasized that also in the asymptotic case $\lambda\to 0$, the order in which the two agents make their moves is important, such that the alternating-moves dynamics considered here essentially differs from the well-investigated iterated-game dynamics.

We do not make an assertion on the convergence of the *trajectories* of the learning process defined in Subsection 2.1 to that of the counterpart process as $\lambda\to 0$, neither do we assert that the steady-state properties of Proposition 2.2 converge. Indeed, the last is not true in general, see the Remark at the end of Subsection 4.2. Possibly, trajectory convergence can be shown, at least under certain definitions of stochastic convergence (cf. the convergence result for an iterated-game model in [5], or the notion of "asymptotic trajectories of a semiflow" introduced in [20]), but a mathematical investigation of this question would exceed the scope of this paper. We restrict us here to the statement that "closeness" of the trajectories of the original learning process for small $\lambda$ to that of the counterpart process could be observed in all of our experiments.

# 4 Slow Learning: The Two-Actions Case

In this section, the special case $N_1=N_2=2$ of two possible actions for each agent is investigated in more detail. Throughout the whole section, we use subscripts (e.g., $x_{ji}$) instead of arguments (e.g., $x(j,i)$) to make notation more concise.

## 4.1 Structure of the Process in the Two-Actions Case

For $N_1=N_2=2$, the system (20) of equations takes the special form

$$\begin{array}{rcrcl}\pi_1 & + & (x_{21}-x_{11})\rho_1 & = & x_{21},\\ (y_{21}-y_{11})\pi_1 & + & \rho_1 & = & y_{21},\end{array}\tag{21}$$

with solution

$$\pi_1=\frac{y_{21}x_{11}+(1-y_{21})x_{21}}{\Delta}\quad\text{and}\quad\rho_1=\frac{x_{21}y_{11}+(1-x_{21})y_{21}}{\Delta},\tag{22}$$

where

$$\Delta = 1 - (x_{21} - x_{11})\,(y_{21} - y_{11}). \tag{23}$$

Furthermore, one finds

$$\pi_2 = \frac{1 - x_{11}y_{11} - (1 - x_{11})y_{21}}{\Delta}, \quad \rho_2 = \frac{1 - y_{11}x_{11} - (1 - y_{11})x_{21}}{\Delta}. \tag{24}$$

Straightforward estimations show that $0 \leq \Delta \leq 2$. The special case $x_{11} = y_{11} = 0$ and $x_{21} = y_{21} = 1$, implying $\Delta = 0$, has to be excluded from consideration. In all other cases, $\pi_1$ and $\rho_1$ (and therefore also $\pi_2$ and $\rho_2$ are valid probabilities satisfying $0 \leq \pi_1 \leq 1$, $0 \leq \rho_1 \leq 1$.

Let us now compute the expected rewards $\alpha_{ji}$ resp. $\beta_{ij}$ defined by (14) resp. (15). Since $x_{12} = 1 - x_{11}$ and $y_{12} = 1 - y_{11}$, we obtain

$$\alpha_{ji} = (a_{ji1} - a_{ji2})\,y_{i1} + a_{ji2} \;\text{ and }\; \beta_{ij} = (b_{ij1} - b_{ij2})\,x_{j1} + b_{ij2}$$

for $j = 1, 2$ resp. $i = 1, 2$. Let us define

$$\mathcal{A}_j(y_{11}, y_{21}) = \alpha_{j1} - \alpha_{j2} = (a_{j11} - a_{j12})\,y_{11} + (a_{j22} - a_{j21})\,y_{21} + (a_{j12} - a_{j22}) \tag{25}$$

and

$$\mathcal{B}_i(x_{11}, x_{21}) = \beta_{i1} - \beta_{i2} = (b_{i11} - b_{i12})\,x_{11} + (b_{i22} - b_{i21})\,x_{21} + (b_{i12} - b_{i22}) \tag{26}$$

for $j = 1, 2$ resp. $i = 1, 2$. This enables a specific representation of the system (16)–(17) for the two-actions case given in the following proposition. The reader should notice that equations for $\dot{x}_{12}$ and $\dot{y}_{12}$ are redundant, since $x_{12} = 1 - x_{11}$, $y_{12} = 1 - y_{11}$.

**Proposition 4.1.** For $N_1 = N_2 = 2$, the counterpart process is characterized by the system

$$\begin{pmatrix} \dot{x}_{11} \\ \dot{x}_{21} \\ \dot{y}_{11} \\ \dot{y}_{21} \end{pmatrix} = \frac{1}{\Delta} \begin{pmatrix} (x_{21}y_{11} + (1 - x_{21})y_{21})\; x_{11}\,(1 - x_{11})\,\mathcal{A}_1(y_{11}, y_{21}) \\ (1 - x_{11}y_{11} - (1 - x_{11})y_{21})\; x_{21}\,(1 - x_{21})\,\mathcal{A}_2(y_{11}, y_{21}) \\ (y_{21}x_{11} + (1 - y_{21})x_{21})\; y_{11}\,(1 - y_{11})\,\mathcal{B}_1(x_{11}, x_{21}) \\ (1 - y_{11}x_{11} - (1 - y_{11})x_{21})\; y_{21}\,(1 - y_{21})\,\mathcal{B}_2(x_{11}, x_{21}) \end{pmatrix} \tag{27}$$

of differential equations, where $\Delta$, $\mathcal{A}_1$, $\mathcal{A}_2$, $\mathcal{B}_1$ and $\mathcal{B}_2$ are given by (23), (25) and (26). (Proof: see Appendix.)

In some cases of application, the reward $a_{j'ij}$ depends on $j'$ only in an additive way:

$$a_{j'ij} = \gamma_{j'} + a_{ij} \tag{28}$$

with arbitrary numbers $\gamma_{j'}$ $(j' = 1, 2)$. In these cases, we obtain $\mathcal{A}_1 = \mathcal{A}_2$. Similarly, if $b_{iji'}$ depends on $i$ only in an additive way, $\mathcal{B}_1 = \mathcal{B}_2$ follows.

**Example 4.1** (Prisoner's Dilemma). Let us take the data of Example 2.2. Since in this example, (28) is satisfied, e.g., with $(a_{ij}) = A^{(2)}$, $\gamma_1 = 4$ and

$\gamma_2 = 0$ (the other agent's previous cooperation simply increases the reward by 4), $\mathcal{A}_1 = \mathcal{A}_2$ must hold. Indeed, we find

$$\mathcal{A}_1(y_{11}, y_{21}) = \mathcal{A}_2(y_{11}, y_{21}) = 4y_{11} - 4y_{21} - 2.$$

Because of the symmetry of the game,

$$\mathcal{B}_1(x_{11}, x_{21}) = \mathcal{B}_2(x_{11}, x_{21}) = 4x_{11} - 4x_{21} - 2.$$

## 4.2 Fixed Points

An obvious question is to which fixed points the dynamics (27) can possibly converge: we are looking for quadruples $(x_{11}, x_{21}, y_{11}, y_{21})$ for which the assigned velocity vector $(\dot{x}_{11}, \dot{x}_{21}, \dot{y}_{11}, \dot{y}_{21})^t$ is $\mathbf{0} = (0, \ldots, 0)^t$. There are several reasons why a component on the r.h.s. of (27) can vanish; we may distinguish reward-specific reasons ($\mathcal{A}_1$, $\mathcal{A}_2$, $\mathcal{B}_1$ or $\mathcal{B}_2$ vanish), or reward-independent reasons (one of the other factors vanishes). Because they are of general interest, irrespectively of the specific reward array under consideration, let us start with the determination of those fixed points that are caused exclusively by reward-independent reasons.

The simplest of these reward-independent causes for a component of the velocity vector getting zero is that a variable $x_{ji}$ or $y_{ij}$ *degenerates*, i.e., takes one of its boundary values 0 or 1. To make $x_{11}$ vanish, e.g., it suffices that either $x_{11} = 0$ or $x_{11} = 1$. However, $\dot{x}_{11}$ also vanishes if $x_{21}\, y_{11} + (1 - x_{21})\, y_{21} = 0$, which is the case if and only if

$$(y_{11} = 0 \wedge y_{21} = 0) \vee (y_{11} = 0 \wedge x_{21} = 1) \vee (y_{21} = 0 \wedge x_{21} = 0). \qquad (29)$$

In total, this gives four logical conditions (say, I to IV), which are sufficient for the quadruple to be a fixed-point if they are satisfied simultaneously. By a straightforward examination, we obtain the following classification of the possible reward-independent fixed points into different types. We write a fixed point as a vector $(x_{11}, x_{21}, y_{11}, y_{21})$; a symbol $*$ as a component denotes an arbitrary *non-degenerated* value, i.e., a value from the interval $]0, 1[$. For later reference, the types will be labelled by symbols A1 etc.

*Case 1.* Exactly two degenerated components.

A1: $(0, *, *, 1)$ A2: $(1, *, 1, *)$ A3: $(*, 0, *, 0)$ A4: $(*, 1, 0, *)$

*Case 2.* Exactly three degenerated components.

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| B1: | $(*, 0, 0, 0)$ | B5: | $(0, *, 0, 1)$ | B9: | $(0, 0, *, 0)$ | B13: | $(0, 1, 0, *)$ |
| B2: | $(*, 0, 1, 0)$ | B6: | $(0, *, 1, 1)$ | B10: | $(1, 0, *, 0)$ | B14: | $(1, 1, 0, *)$ |
| B3: | $(*, 1, 0, 0)$ | B7: | $(1, *, 1, 0)$ | B11: | $(0, 0, *, 1)$ | B15: | $(1, 0, 1, *)$ |
| B4: | $(*, 1, 0, 1)$ | B8: | $(1, *, 1, 1)$ | B12: | $(0, 1, *, 1)$ | B16: | $(1, 1, 1, *)$ |

*Case 3.* All four components are degenerated. In this case, every possible vector

$$(x_{11}, x_{21}, y_{11}, y_{21}) \text{ with } x_{ji} \in \{0, 1\},\ y_{ij} \in \{0, 1\}$$

is a fixed point, which yields types C1 to C16 (indexed according to lexicographic order of the quadruples).

In case 1, only limit cycles of length 2 result. E.g., type A1 always leads to the limit cycle $(2,1)$ (agent 1 chooses action 2, agent 2 chooses action 1 in every round). In cases 2 and 3, also longer limit cycles are possible, e.g., the fixed point $(0,1,1,0)$ (type C7) effects the limit cycle $(2,2,1,1)$, where both agents do the same in each round, but with alternating actions in successive rounds.

Reward-specific fixed points can occur by a replacement of any of the conditions I to IV explained above (possibly even more than one) by the corresponding condition $\mathcal{A}_j(y_{11}, y_{21}) = 0$ $(j = 1, 2)$ resp. $\mathcal{B}_i(x_{11}, x_{21}) = 0$ $(i = 1, 2)$ from (27). Let us subsume reward-specific fixed points that are not yet covered by one of the types A1–C16 under "type D".

**Example 4.1** (continued). The reward-independent fixed points in the setting of the prisoner's dilemma example given above are those of the general case. As the reward-specific fixed points are concerned, we distinguish three cases:

(i) $4x_{11} - 4x_{21} - 2 = 0$, but $4y_{11} - 4y_{21} - 2 \neq 0$. Short reflection shows that this leaves the possibilities $(\frac{1}{2}, 0, *, 0)$, which is already covered by type A3, and $(1, \frac{1}{2}, 1, *)$, which is already covered by type A2.

(ii) $4y_{11} - 4y_{21} - 2 = 0$, but $4x_{11} - 4x_{21} - 2 \neq 0$. This leaves the possibilities $(*, 0, \frac{1}{2}, 0)$, which is already covered by type A3, and $(1, *, 1, \frac{1}{2})$, which is already covered by type A2.

(iii) Both $4x_{11} - 4x_{21} - 2 = 0$ and $4y_{11} - 4y_{21} - 2 = 0$. This yields a set (with two degrees of freedom) of type-D fixed points. They do not possess stable, deterministic limit cycles.

**Remark.** The properties of the type-D fixed points in Example 4.1 may seem to contradict Proposition 2.2, since they do not result in deterministic limiting behavior. The reader should be aware, however, that there is no analogue to Proposition 2.2 for the counterpart process. In the original learning process, type-D fixed points of the counterpart process admit fluctuations of the probabilities $x_{11}$ etc. that obstruct the invariance equation (34) in the proof of Proposition 2.2. Nevertheless, these fluctuations get infinitesimal as $\lambda \to 0$, therefore the process stays in the neighborhood of such a point for a long time, such that in the *intermediate* term, also the original process behaves around this point in a similar way as around a fixed point. Related observations have also been made for other approximations of stochastic reinforcement learning dynamics by underlying deterministic dynamics, see, e.g., Skyrms and Pemantle [31].

## 4.3 Dynamics for Identical Start Points

Of course, it is also interesting along which trajectories the fixed points identified in Subsection 4.2 are reached and how their attraction domains

look like. For given reward arrays, trajectories can be computed numerically. Let us restrict ourselves to *symmetric* rewards for both agents, i.e., $A^{(1)} = B^{(1)}$ and $A^{(2)} = B^{(2)}$. Furthermore, let us assume in this subsection that the initial values of the memory matrices are identical for both agents: $X_0 = Y_0$. In other words, both agents start with their reaction probabilities located at the same point $(x_{11}, x_{21}) = (y_{11}, y_{21})$ of the unit square $[0,1]^2$. This models a situation where there is no initial difference between the "characters" of the two agents. In this case, the third resp. fourth component in (27) get copies of the first resp. second component, since $\mathcal{A}_1 = \mathcal{B}_1$, $\mathcal{A}_2 = \mathcal{B}_2$, and the initial equalities $x_{11} = y_{11}$ and $x_{21} = y_{21}$ continue to hold during the process because of the identity of the respective differential equations. This yields the following dynamics:

$$\begin{pmatrix} \dot{x}_{11} \\ \dot{x}_{21} \end{pmatrix} = \frac{1}{\Delta} \begin{pmatrix} (x_{21}x_{11} + (1-x_{21})x_{21})\, x_{11}\, (1-x_{11})\, \mathcal{A}_1(x_{11}, x_{21}) \\ (1 - x_{11}^2 - (1-x_{11})x_{21})\, x_{21}\, (1-x_{21})\, \mathcal{A}_2(x_{11}, x_{21}) \end{pmatrix}$$

$$= K(x_{11}, x_{21}) \cdot \begin{pmatrix} x_{11}\, \mathcal{A}_1(x_{11}, x_{21}) \\ (1-x_{21})\, \mathcal{A}_2(x_{11}, x_{21}) \end{pmatrix} \tag{30}$$

with

$$K(x_{11}, x_{21}) = \frac{1}{\Delta}\, x_{21}\, (1-x_{11})\, (x_{11} - x_{21} + 1).$$

In particular, for points $(x_{11}, x_{21})$ where $K(x_{11}, x_{21}) \neq 0$ and also the product $x_{11}\, \mathcal{A}(x_{11}, x_{21})$ is $\neq 0$, the slope of the function $x_{21} = x_{21}(x_{11})$ obtained from the trajectory $(x_{11}(t), x_{21}(t))$ by elimination of the parameter $t$ is given by

$$\frac{dx_{21}}{dx_{11}} = \frac{\frac{dx_{21}}{dt}}{\frac{dx_{11}}{dt}} = \frac{\dot{x}_{21}}{\dot{x}_{11}} = \frac{(1-x_{21})\, \mathcal{A}_2(x_{11}, x_{21})}{x_{11}\, \mathcal{A}_1(x_{11}, x_{21})}. \tag{31}$$

Let us call the straight lines where $\mathcal{A}_1(x_{11}, x_{21}) = 0$ resp. $\mathcal{A}_2(x_{11}, x_{21}) = 0$ the first resp. second *characteristic line* of the dynamics (30). The characteristic lines are crossed by the trajectory either vertically (first characteristic line) or horizontally (second characteristic line).

The reward-independent fixed points of (30) according to the classification of Subsection 4.2 are those of type C1, C6, C11 and C16 (corners of the unit square in $(x_{11}, x_{21})$-plane), of type A2 (right boundary of the square) and of type A3 (bottom of the square).

For given $\mathcal{A}_1$ and $\mathcal{A}_2$, the differential equation (31) can be solved, which yields a parametrized set $x_{21} = x_{21}(x_{11}, c)$ of curves. Each trajectory follows one of these curves. The orientation of the movement along the curve is given by the signs of the two components in (30). Since $K(x_{11}, x_{21}) \geq 0$, $x_{11} \geq 0$ and $1 - x_{21} \geq 0$ for all $(x_{11}, x_{21}) \in [0,1]^2$, these signs are determined by the signs of $\mathcal{A}_1(x_{11}, x_{21})$ and $\mathcal{A}_2(x_{11}, x_{21})$, i.e., they change at the characteristic lines.

The analysis gets particularly simple in the case where $\mathcal{A}_1 = \mathcal{A}_2$, which is guaranteed, e.g., by (28). In this case, the two characteristic lines coincide

to a line containing fixed points of type D. Eq. (31) simplifies to

$$\frac{dx_{21}}{dx_{11}} = \frac{1 - x_{21}}{x_{11}}. \tag{32}$$

Solving this differential equation, we obtain as the general solution

$$x_{21} = 1 - c/x_{11} \tag{33}$$

with constant $c$ as parameter.

**Example 4.1** (continued). In our prisoner's dilemma context with the numbers of Example 2.2, we have

$$\mathcal{A}_1(x_{11}, x_{21}) = \mathcal{A}_2(x_{11}, x_{21}) = 4x_{11} - 4x_{21} - 2,$$

so the characteristic line (containing type-D fixed points) is the straight line from $(\frac{1}{2}, 0)$ to $(1, \frac{1}{2})$. Fig. 1 shows the resulting dynamics. It can be observed that there are three attraction domains I, II and III:

(i) If the agents start at a point in domain I, a point $(1, x_{21})$ with $x_{21} \in [0, \frac{1}{2}]$ is finally reached, which corresponds to a type-A2 fixed point $(1, *, 1, *)$. At this point, both agents cooperate forever (choose only action 1), and obtain a constant reward of 8 units each.

(ii) If the agents start at a point in domain II, a point $(x_{11}, x_{11} - \frac{1}{2})$ with $x_{11} \in [\frac{1}{2}, \frac{3}{4}]$ on the lower part of the characteristic line is finally reached (straight line between $(\frac{1}{2}, 0)$ and $(\frac{3}{4}, \frac{3}{4})$ in Fig. 1.) In this case, the final behavior is not a limit cycle, but remains stochastic. The average reward for each agent is between 2 and 8, depending on the exact location of the attractor point. As discussed in the Remark after Example 4.1, the trajectories of the *original* learning process cannot have attractors on the indicated characteristic line, but they can fluctuate around it for a long time in the intermediate term of the process.

(iii) If the agents start at a point in domain III, except those where $x_{11} = 1$, a point $(x_{11}, 0)$ with $x_{11} \in [0, \frac{1}{2}]$ is finally reached, which corresponds to a type-A3 fixed point $(*, 0, *, 0)$. At such a point, both agents defect forever (choose only action 2), and obtain a constant reward of 2 units each.

We see that the learning dynamics may lead to the Pareto-efficient cooperative solution of the prisoner's dilemma as well as to the non-cooperative equilibrum in dominant strategies, depending on the initial values: For initial reaction probabilities in a certain area around the "Tit-for-Tat" corner $(1, 0)$ (see below), the process tends to final mutual cooperation; for initial reaction probabilities in the left or upper part of the unit square, on the other hand, it tends to final mutual defection. There is also a rather small area of initial points for which the process tends to a stochastic mix between cooperative and non-cooperative behavior.

The dynamics in Fig. 1 has some similarity to that observed by Hofbauer and Sigmund [17] (Subsection 9.3) for an evolutionary dynamic model of the iterated prisoner's dilemma, although the differential equations are basically different.

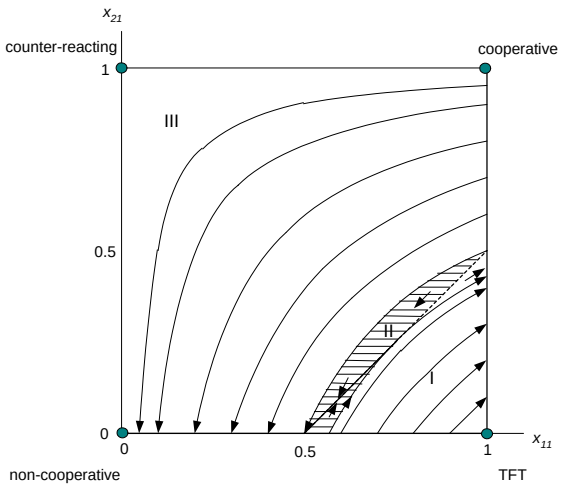


*Fig. 1.* Dynamics in the prisoner's dilemma situation of Example 2.2 with identical start points of the two agents. The domains of attraction are marked: I and III unshaded, II shaded by horizontal lines.

In Fig. 1, we have labelled the point $(1, 0)$ by "TFT" (Tit-for-Tat), because $x_{11} = 0$ and $x_{21} = 1$ prescribe that cooperation is responded by cooperation and defection is responded by defection. To indicate their intuitive meaning, the other corner points of the unit square have been labelled as follows: $(0, 0)$ is "non-cooperative" (always defect), $(1, 1)$ is "cooperative" (always cooperate), and $(0, 1)$ is "counter-reacting" (respond cooperation by defection and vice versa). A somewhat disquieting observation is that even two "good-natured" agents starting near point $(1, 1)$ end up near the disastrous point $(0, 0)$ as a consequence of their co-learning.

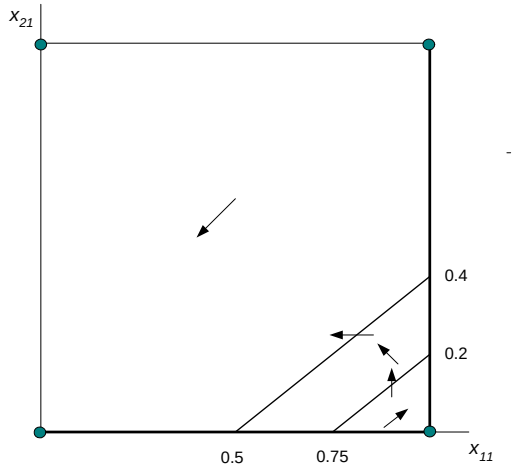


*Fig. 2.* Dynamics in the prisoner's dilemma situation with changed payoff.

The concrete numbers of Example 2.2 have the property that each agent gets a certain reward (2 units) by defecting and gives the other agent a certain reward (4 units) by cooperating. This property makes (28) satisfied (with the consequence that the two characteristic lines coincide), but it is not the general case of a prisoner's dilemma. Let us change the values in (13) to

$$P = \begin{pmatrix} (4,4) & (0,6) \\ (6,0) & (1,1) \end{pmatrix}.$$

The we obtain the reward matrices

$$A^{(1)} = \begin{pmatrix} 8 & 4 \\ 12 & 7 \end{pmatrix}, \qquad A^{(2)} = \begin{pmatrix} 4 & 0 \\ 7 & 2 \end{pmatrix},$$

such that $\mathcal{A}_1(x_{11}, x_{21}) = 4x_{11} - 5x_{21} - 3$ and $\mathcal{A}_2(x_{11}, x_{21}) = 4x_{11} - 5x_{21} - 2$. Now $\mathcal{A}_1 \neq \mathcal{A}_2$. The characteristic lines are parallel.[5] The dynamics is outlined in Fig. 2: The arrays show the orientation of the trajectories. They cut the first (lower) characteristic line vertically, the second (upper) characteristic line horizontally. The result is similar to that depicted in Fig. 1: Only when starting in a certain area around the TFT corner $(1, 0)$, the process tends to stable cooperative behavior. Contrary to area II in Fig. 1, there is no domain leading to stochastic limiting behavior in this case.

## 4.4 Dynamics for Not Identical Start Points

Identical initial probabilities leading to identical trajectories for both players may be seen as a very unlikely situation. Numerical results show that typically, the trajectories do not change too much if the starting point for one of the two players is slightly disturbed, as long as the basin of attraction remains the same. However, if the starting points are chosen in a larger distance from each other, qualitatively new phenomena may occur. In Fig. 3, we have plotted the prisoners's dilemma trajectories for the six combinations of starting points near the four corners of the strategy square where the two agents start in the neighborhood of *different* corners. All these six trajectory pairs end with mutual non-cooperation, which indicates that for a "happy ending" in the prisoner's dilemma case, it is necessary that *both* agents start with a behavior not too far from the tit-for-tat pole. Furthermore, it can be observed that some of these trajectories show sharp *turns*. For example, in the case of the starting point combination where agent 1 starts near the tit-for-tat corner and agent 2 starts near the cooperative corner (second picture of third row in Fig. 3), there are two such turns in the trajectory of agent 1: After a phase where she moves rather straight away towards the non-cooperative corner, increasingly exploiting agent 2, she abruptly changes her path towards a more friendly behavior, presumably caused by

[5] In fact, it can be shown that this is true for all reward arrays derived from symmetric $[2 \times 2]$ bimatrix games.

the fact that agent 2 has begun to play tit-for-tat in the meantime, but after a while she makes a second abrupt turn towards non-cooperation which is presumably due to the increasingly non-cooperative behavior of agent 2.

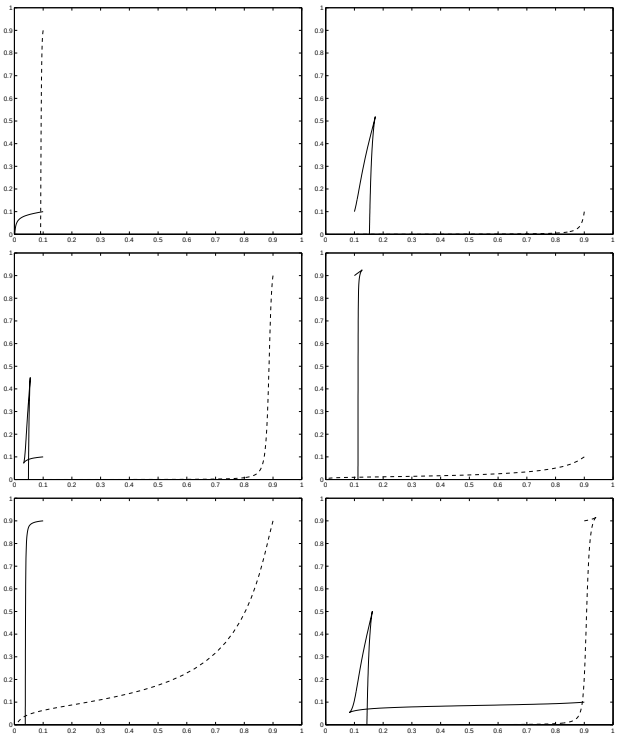

*Fig. 3.* Trajectories in the prisoner's dilemma situation for starting points $((0.1, 0.1), (0.1, 0.9))$ and $((0.1, 0.1), (0.9, 0.1))$ (first row), $((0.1, 0.1), (0.9, 0.9))$ and $((0.1, 0.9), (0.9, 0.1))$ (second row), and $((0.1, 0.9), (0.9, 0.9))$ and $((0.9, 0.1), (0.9, 0.9))$ (third row). Trajectory of agent 1: solid, trajectory of agent 2: dotted.

# 5 Conclusions

We have investigated the dynamics of an interactive system of two economic, social or artificial agents performing "stimulus–response–consequence" learning in a way where the response of one agent is the stimulus for the other agent's next action and essentially contributes to the consequence for the other agent's previous action, and vice versa. The "alternating moves" point of view adopted in this article differs from the "simultaneous moves" consideration in the majority of investigations on learning in games, but it seems more appropriate for many situations of real life where people rather act at different times, observing the prior behavior of others, instead of

performing a series of simultaneous decisions, as it is modelled by the classical iterated-games paradigm. Most part of our investigation deals with the limiting case of a learning rate near zero, i.e., the case of slow learning. This limiting case, which forms also the "backbone" of the process in the more stochastic case of a higher learning rate, is governed by a system of deterministic differential equations that can be solved either analytically or numerically.

Some specific insights into the interactive process for situations of particular interest in economic theory have been outlined, especially situations of prisoner's dilemma type. In the last-mentioned special case, it has turned out that in our interactive "stimulus–response–consequence" model, even two initially almost perfectly cooperative agents can be driven by the systems dynamics to final mutual defection through a long gradual process of growing distrust and increasing frequency of non-cooperative actions. For initially *different* "characters" of the two agents, it can been observed that stable final cooperation is only to be expected if *both* agents start with an adaptive, "tit-for-tat"-like behavior (perhaps with eventual toleration of defection and/or eventual own turn to defection). However, these results should be translated into predictions on social dynamics only with some caution, because our model is restricted to reinforcement learning and does not include other learning mechanisms such as imitation learning, nor does it take human emotions into account.

A lot of questions remain open. Of course, the effect of a higher learning rate should be examined in detail. Alternative reinforcement schemes might be studied, e.g., the propensity-based scheme by Roth and Erev [27] as well as discounted and exponential variants of it (see Nagel and Tang [22]). Moreover, in the concrete examples we referred only to the special case of two possible actions for each agent. For several areas of application, it would be worthwhile to investigate a larger variety of actions. Another relevant question is how the interaction process is influenced by possible misinterpretations (modelled by random noise) of the other agent's actions, as they are frequent in human life.

Finally, for computer science applications, e.g., in multiagent systems technology or e-business, the co-behavior of more elaborate learning strategies than that considered here should be investigated.

**Acknowledgement** — The author wants to express thanks to Anselm Eder for fruitful discussions on the subject of the paper, to Martin Posch for several valuable hints, and to an anonymous referee for very helpful comments on a first version of the article.

# References

[1] Banerjee, B., Mukherjee, R., Sen, S. (2000) Learning mutual trust. Working Notes of AGENTS-00 (Workshop on Deception, Fraud and Trust in Agent Societies), pp. 9–14.

[2] Benaïm, M., Weibull, J. (2001) Deterministic Approximation of stochastic evolution in games. Technical Report, S-WOPEC (Scandinavian Working Papers in Economis).

[3] Bereby-Meyer, Y., Erev, I. (1998) On learning to become a successful loser: a comparison of alternative abstractions of learning processes in the loss domain. *J. of Mathematical Psychology* 42, pp. 266-286.

[4] Bernal, G., Golann, S. (1980) Couple interaction: a study of the punctuation process. *Int. J. of Familiy Therapy* 2, pp. 47–56.

[5] Börgers, T., Sarin, R. (1997) Learning through reinforcement and replicator dynamics. *J. of Economic Theory* 77, pp. 1–14.

[6] Brenner, T. (1999) *Modelling Learning in Economics*. Cheltenham: Edward Elgar Publishing Ltd.

[7] Bush, R.R., Mosteller, F. (1955) *Stochastic Models of Learning*. New York: Wiley.

[8] Claus, C., Boutilier, C. (1998) The dynamics of reinforcement learning in ccoperative multiagent systems. *Proc. Fifteenth National Conf. on Artificial Intelligence*, AAAI Press / MIT Press, pp. 746–752.

[9] Cross, J.G. (1973) A stochastic learning model of economic behavior. *Quarterly J. of Economics* 87, pp. 239–266.

[10] Eder, A., Gutjahr, W.J., Neuwirth, E. (2001) Modelling social interactions by learning Markovian matrices. *Proc. FACET '01* (Int. Conf. on Facet Theory), Prague, pp. 94–106.

[11] Eder, A., Gutjahr, W.J. (2003) Can simulation techniques contribute to microsociological theory? The case of learning matrices. *Developments in Applied Statistics* (Eds.: A. Ferligoj, A. Mrvar), Metodoloski zvseski 19, FDV (Ljubljana), pp. 219–239.

[12] Erev, I., Bereby-Meyer, Y., Roth, A.E. (1999) The effect of adding a constant to all payoffs: experimental investigation, and implications for reinforcement learning models. *J. of Economic Behavior & Organisation* 39, pp. 111–128.

[13] Fudenberg, D., Levine, D. (1998) *Theory of Learning in Games*. Cambridge, MA: MIT-Press.

[14] Greenwald, A., Friedman, E.J., Shenker, S. (2001) Learning in network contexts: experimental results from simulation. *Games and Economic Behavior* 35, pp. 80–123.

[15] Gutjahr, W.J., Eder, A. (2001) A Markov model for dyadic interaction learning. *Proc. EWRL '01* (Fifth European Workshop on Reinforcement Learning), Utrecht, Netherlands, pp. 17–18.

[16] Herrnstein, R.J., Prelec, D. (1991) Melioration: a theory of distributed choice. *J. of Economic Perspectives* 5, pp. 137–156.

[17] Hofbauer, J., Sigmund, K.(1998) *Evolutionary Games and Population Dynamics*. Cambridge UP.

[18] Hopkins, E. (2002) Two competing models of how people learn in games. *Econometrics* 70, pp. 2141–2166.

[19] Hu, J., Wellman, M.P. (1998) Multiagent reinforcement learning: Theoretical framework and an algorithm. *Proc. Fifteenth International Conf. on Machine Learning*, San Francisco, CA, pp. 242–250.

[20] Laslier, J.-F., Topol, R., Walliser, B. (2001) A behavioral learning process in games. *Games and Economic Behavior* 37, pp. 340-366.

[21] Littman, M.L. (1994) Markov games as a framework for multi-agent reinforcement learning. *Proc. Eleventh International Conf. on Machine Learning*, San Mateo, CA, pp. 157–163.

[22] Nagel, R., Tang, F.F. (1998) Experimental results on the centipede game in normal form: an investigation on learning. *J. of Mathematical Psychology* 42, pp. 356-384.

[23] Neill, D.B. (2001) Optimality under noise: higher memory strategies for the alternating prisoner's dilemma. *J. Theor. Biol.* 211, pp. 159–180.

[24] Nowak, M.A., Sigmund, K. (1994) The alternating prisoner's dilemma. *J. Theor. Biol.* 168, pp. 219-226.

[25] Posch, M. (1997) Cycling in a stochastic learning algorithm for normal form games. *J. Evolutionary Econ.* 7, pp. 1993-207.

[26] Posch, M., Pichler, A., Sigmund, K. (1999) The efficiency of adapting aspiration levels. *Proceedings of the Royal Society*, Series B, 266, pp. 1427–1436.

[27] Roth, A.E., Erev, I. (1995) Learning in extensive-form games: experimental data and simple dynamic models in the intermediate term. *Games and Economic Behavior* 8, pp. 164-212.

[28] Rubinstein, A. (1982) Perfect equilibrum in a bargaining model. *Econometrica* 50, pp. 97–109.

[29] Sandholm, T., Crites, R.H. (1995) Multiagent reinforcement learning in the iterated prisoner's dilemma. *Biosystems* 37, pp. 147-166.

[30] Shohan, V., Rohrbaugh, M.J. (2002) Brief strategic couple therapy. *Clinical Handbook of Couple Therapy*, eds.: Gurman, A., Jacobson, N.S., pp. 5–26.

[31] Skyrms, B., Pemantle, R. (2004) Learning to network. Technical Report, University of California – Irvine, http://hypatia.ss.uci.edu/lps/home/facstaff/faculty/skyrms/Skyrmspapers.html

[32] Smith, R., Sola, M., Spagnolo, F. (2000) The prisoner's dilemma and regime-switching in the greek-turkish arms race. *J. of Peace Research* 37, pp. 737-750.

[33] Wedekind, C., Milinski, M. (1996) Human cooperation in the simultaneous and the alternating Prioner's Dilemma: Pavlov versus generous tit-for-tat. *Proc. Nat. Acad. Sci. USA* 93, pp. 2686-2689.

## APPENDIX

**Proof of Proposition 2.1.** We have to show that the distribution of the $(n+1)$th state $(j_n, X_{n+1}, i_{n+1}, Y_{n+1})$ of the first stochastic process is determined by its $n$th state $(j_{n-1}, X_n, i_n, Y_n)$. The integer $j_n$ is chosen randomly according to the probabilities contained in line $i_n$ of $Y_n$. As soon as $j_n$ is fixed, $X_{n+1}$ results deterministically from $X_n$ and the reward $a(j_{n-1}, i_n, j_n)$. The integer $i_{n+1}$ is chosen randomly according to the probabilities contained in line $j_n$ of $X_{n+1}$, and $Y_{n+1}$ results deterministically from $Y_n$ and $b(i_n, j_n, i_{n+1})$. So everything is shown for the first process. The proof for the second process is similar. □

**Proof of Proposition 2.2.** By definition, in a type (a) steady-state distribution, the values $x_n(j,i)$ and $y_n(i,j)$ do not change anymore over time. Let us denote them by $x_\infty(j,i)$ resp. $y_\infty(i,j)$. By (3), we obtain

$$x_\infty(j',i) = x_\infty(j',i) + \lambda a(j',i,j)(1 - x_\infty(j',i)) \tag{34}$$

for each triple $(j',i,j)$ that is eventually played, i.e., for which $(j_{n-1}, i_n, j_n) = (j',i,j)$ with a nonzero probability in the steady-state distribution. Hence $x_\infty(j',i) = 1$, which implies $x_\infty(j',\tilde{i}) = 0$ for all $\tilde{i} \neq i$. We set $\varphi_1(j') = i$. Analogously, one obtains $y_\infty(i,j) = 1$ for each triple $(i,j,i')$ that is eventually played, which implies $y_\infty(i,\tilde{j}) = 0$ for all $\tilde{j} \neq j$. We set $\varphi_2(i) = j$.

Let $I$ be the set of all start indices $i$ in a triple $(i,j,i')$ that is eventually played, and let $J$ be the set of all start indices $j'$ in a triple $(j',i,j)$ that is eventually played. Then the functions $\varphi_1 : J \to I$ and $\varphi_2 : I \to J$ are well-defined. Because $\varphi_1(J) = I$ (since only an action $i$ that occurs in a pair $(j',i)$ with $\varphi_1(j') = i$ can eventually be chosen) and $\varphi_2(I) = J$, both $\varphi_1$ and $\varphi_2$ must be bijective. This completes the proof. □

**Proof of Proposition 3.1.** Within a period $p$ of the process resulting from approximation step 1, the actions $\ldots, i_n, j_n, i_{n+1}, j_{n+1}, \ldots$ of the two agents, chosen based on the fixed matrices $X_{(p-1)M}$ and $Y_{(p-1)M}$, form a Markov chain. This Markov chain is periodic (with period 2). However, we may compose the actions to pairs by defining

$$\xi_n = (i_n, j_n) \quad (n = n(p), \ldots, n(p+1) - 1).$$

Since the starting probabilities $x_0(j,i)$ and $y_0(i,j)$ are assumed to be different from 0, by the special properties of the applied reinforcement scheme, also all $x_n(j,i)$ and $y_n(i,j)$ remain different from 0. It is easy to see that as a consequence, the stochastic process with states $\xi_n$ is an ergodic (i.e., aperiodic and irreducible) homogeneous Markov chain during period $p$. Therefore, the Markov chain $(\xi_n)$ converges to a uniquely defined steady-state distribution. In the process obtained by approximation step 2, the random pairs $\xi_n = (i_n, j_n)$ of actions obey this steady-state distribution during the entire period $p$.

Let $\pi(i,j)$ denote the steady-state probability of state $\xi = (i,j)$. Using the indicator function $I(\text{statement}) = 1$ if statement is true and 0 otherwise, it is easily seen that the increments $\Delta Y_n$ resulting from eqs. (6) - (8) can be represented as follows:

$$\Delta y_n(i,j) \;=\; I(i_n = i) \cdot \lambda \left\{ I(j_n = j) \sum_{i'=1}^{N_1} I(i_{n+1} = i')\, b(i,j,i') \right.$$

$$\left. - \sum_{m=1}^{N_2} I(j_n = m) \sum_{i'=1}^{N_1} I(i_{n+1} = i')\, b(i,m,i')\, y_n(i,j) \right\}.$$

Therein (since the values $y_n(i,j)$ are assumed as constant in period $p$ in the counterpart process), only the indicator functions are random variables. By approximation step 2, in the counterpart process, the expected value of $I(i_n = i)\, I(j_n = m)\, I(i_{n+1} = i')$ is $\pi(i,m)\, x_n(m,i')$. (Note that the sum of transition probabilities from state $(i,m)$ to the states $(i',1), \ldots, (i',N_2)$ is just $x_n(m,i')$). Hence, in the

process obtained by approximation step 3, the increment of $y_n(i,j)$ in period $p$ is

$$M{\cdot}\lambda\left\{\pi(i,j)\,\sum_{i'=1}^{N_1} x_n(j,i')\,b(i,j,i') - \sum_{m=1}^{N_2}\pi(i,m)\,\sum_{i'=1}^{N_1} x_n(m,i')\,b(i,m,i')\,y_n(i,j)\right\}. \tag{35}$$

Since a period takes $dt = \epsilon = M\lambda$ time in our scaling, dividing the increment (35) by $M\lambda$ and using (15) yields:

$$\dot{y}(i,j) \;=\; \pi(i,j)\,\beta(i,j) - y(i,j)\,\sum_{m=1}^{N_2}\pi(i,m)\,\beta(i,m). \tag{36}$$

By a quite similar consideration, we can also compose the actions to pairs

$$\eta_n = (j_{n-1}, i_n) \quad (n = n(p)+1, \ldots, n(p+1)).$$

Let $\rho(j,i)$ denote the steady-state probability of state $\eta = (j,i)$ in the Markov chain $(\eta_n)$. Then an analogous derivation as above yields

$$\dot{x}(j,i) = \rho(j,i)\,\alpha(j,i) - x(j,i)\,\sum_{m=1}^{N_1}\rho(j,m)\,\alpha(j,m). \tag{37}$$

With

$$\pi(i) \;=\; \sum_{j=1}^{N_2}\pi(i,j) \quad \text{and} \quad \rho(j) \;=\; \sum_{i=1}^{N_1}\rho(j,i)$$

denoting the probabilities that in steady state during a fixed period, agent 1 chooses action $i$ resp. agent 2 chooses action $j$, we must have

$$\pi(i,j) = \pi(i)\,y(i,j) \quad \text{and} \quad \rho(j,i) = \rho(j)\,x(j,i).$$

Using the last equations, (37) and (36) can still be slightly re-formulated to obtain the assertion of the Proposition. □

**Proof of Proposition 4.1.** Using (25) – (26), the square brackets in (16) resp. (17) can be re-written as follows:

$$\alpha_{ji} - \sum_{m=1}^{2} x_{jm}\alpha_{jm} = \alpha_{j1} - x_{j1}\alpha_{j1} - (1-x_{j1})\,\alpha_{j2} = (1-x_{j1})\,\mathcal{A}_j(y_{11}, y_{21}),$$

$$\beta_{ij} - \sum_{m=1}^{2} y_{im}\beta_{im} = \beta_{i1} - y_{i1}\beta_{i1} - (1-y_{i1})\,\beta_{i2} = (1-y_{i1})\,\mathcal{B}_i(x_{11}, x_{21}).$$

This gives the system (16)–(17) the following specific form:

$$\dot{x}_{j1} = \rho_j x_{j1}(1-x_{j1})\,\mathcal{A}_j(y_{11}, y_{21}) \quad (j = 1,2), \tag{38}$$

$$\dot{y}_{i1} = \pi_i y_{i1}(1-y_{i1})\,\mathcal{B}_i(x_{11}, x_{21}) \quad (i = 1,2). \tag{39}$$

Inserting (22) and (24) proves the assertion. □