Control of 1-D and 2-D Coupled Map Lattices through Reinforcement Learning

Sabino Gadaleta and Gerhard Dangelmayr
Colorado State University
Department of Mathematics
Engineering E121,
Ft. Collins, CO 80521, USA
{sabino, gerhard}@math.colostate.edu

Abstract

In this paper we show that coupled 1-D and 2-D logistic map lattices can be controlled to different types of behaviour through a recently introduced control algorithm (S. Gadaleta and G. Dangelmayr, Chaos **9**, 775-788 (1999)) which is based on reinforcement learning. The control policy is established through information from the local neighborhood of a subsytem and does not require any explicit knowledge on system dynamics or location of desired patterns. The algorithm is applicable in noisy and nonstationary environments. We demonstrate that the control policies established from interaction with small systems can be applied to larger systems and combined to trigger the formation of new states.

1 Introduction

Starting with the pioneering work of Ott *et al.* [1], much research has been devoted during the last decade to the control of chaotic dynamical systems (see [2] for a review). The control of low-dimensional chaotic systems is now well understood and finds more and more applications. In contrast, the control of the large variety of complex patterns observed in spatiotemporal systems is far less understood and provides a major challenge for researchers working in the field of nonlinear dynamics. A popular model for spatiotemporal behaviour is the logistic coupled map lattice (LCML). Feedback methods [3, 4] as well as self-adaptive parameter control methods [5] are the most popular control methods for such systems. These methods are in general easy to implement but very sensitive to the correct choice of the feedback (or stiffness) parameter. It is often unclear how the correct choice for this parameter is established. Furthermore these methods require exact knowledge of the target state. If no explicit knowledge on system dynamics is available, this state must be extracted from observation.

Algorithms which do not require analytical knowledge on system dynamics or the target state must learn the correct behaviour from interaction with the controlled system. Neural networks have been suggested for the control of low-dimensional chaotic systems [6] but since they apply supervised learning, they also require analytical knowledge on system dynamics. Learning algorithms which do not require any analytical knowledge can be based on reinforcement learning. The use of reinforcement learning to control chaotic systems was first suggested by Der and Herrmann [7] who applied it to the logisitic map. In [8] we generalized the method and applied it to the control of several discrete and continous low-dimensional chaotic and hyperchaotic systems. Lin and Jou [9] proposed a reinforcement learning neural network for the control of chaos and applied it to the logistic and the Hénon map. Especially in the context of smart matter applications [10] reinforcement learning might proove very useful. In smart matter applications microscopic sensor and control units are distributed over a physical system in order to allow the system to respond to the environment in a controlled way. A key difficulty in the potential of these applications is developing the appropriate control programs [10]. We propose to base these programs on reinforcement learning algorithms.

To our knowledge, the control of spatiotemporal chaotic systems through artificial intelligence learning, in particular reinforcement learning, has not been demonstrated. In this work we present a control method for spatiotemporal systems which relies on the recently introduced chaos control method [8] and uses reinforcement learning to establish control of a coupled map system in an optimal manner with respect to a certain optimality criterion. The control policy is established through interaction with a local neighborhood of a controlling unit and is able to find a global control as will be demonstrated through stabilization of different unstable patterns in a 1-D and 2-D LCML.

2 The control algorithm

The symmetric diffuse 2-D LCML can be written in the form

$$\begin{aligned}x_{n+1}^{i,j} =&(1-\epsilon) f\left(x_n^{i,j}\right)+\frac{\epsilon}{2}\left[f\left(x_n^{i-1,j}\right)+f\left(x_n^{i+1,j}\right)\right] \\ &+\frac{\epsilon}{2}\left[f\left(x_n^{i,j-1}\right)+f\left(x_n^{i,j+1}\right)\right],\end{aligned} \tag{1}$$

where the integers i, j mark discrete spatial locations and $f(x) = ax(1-x)$. We assume periodic boundary conditions with basic domain $1 \leq i \leq i_d$, $1 \leq j \leq j_d$. In the one-dimensional (1-D) case ($j_d = 1$) the last term in (1) has to be omitted. For notational convenience we describe our control algorithm only for the 1-D lattice, the extension to the 2-D case is obvious.

In [8] we showed how a single logistic map can be controlled through a method based on reinforcement learning. Here we extend this method to the control of 1-D and 2-D LCML. Our approach to control the system is based on small discrete state dependent pertubations to either the coupling strength ϵ or the system parameter a. In this work we apply control to every site, such that the equation for the controlled 1-D lattice can be written as

$$\begin{aligned}x_{n+1}^{i} =& \left(1-\epsilon\left(x_n^i\right)\right) f\left(x_n^i\right) \\ &+\frac{\epsilon\left(x_n^i\right)}{2}\left[f\left(x_n^{i-1}\right)+f\left(x_n^{i+1}\right)\right],\end{aligned} \tag{2}$$

where now $\epsilon(x) = \epsilon_0 + \epsilon_u u_\epsilon(x)$ and $f(x) = (a_0 + a_u u_a(x))x(1-x)$. Here, $u_a(x)$ or $u_\epsilon(x)$ represent the discrete control perturbations. Distributed control would be obtained by applying control signals only to a subset of sites. We restrict the set of allowed values for the control to $u_\epsilon \in U_\epsilon = \{0, \Delta\epsilon, -\Delta\epsilon\}$ and $u_a \in U_a = \{0, \Delta a, -\Delta a\}$, where $\Delta\epsilon$ and Δa are the "amplitudes" of the control signal and ϵ_u and a_u determine which parameter is controlled: $(\epsilon_u, a_u) = (0, 1)$ or $(1, 0)$ (we could also choose to perturb both paramters). Although we consider only two small, nonzero perturbations of one of the parameters, the algorithm leads to a successful control performance. Which value for u ($= u_\epsilon$ or u_a) is chosen depends on the current state x^i and is determined from x^i according to a certain control policy $\hat{\Pi}(x^i)$, which associates a control u to x^i.

In principle each control unit could establish its own "optimal" control policy independently. In this work however we use the same policy for all cells. The policy is established by monitoring the local neighborhood of one "master" unit which "slaves" other units in that it updates the globally used control policy. To establish this "master" policy, we first discretize the state space $X = [0, 1]$ of a single logistic map. In general, this discretization can be accomplished by means of a vectorquantization, but in the case of the one-dimensional state space of a logistic map a uniform distribution of X is sufficient. We chose a set of 100 reference points w distributed uniformly over $[0, 1]$ which form the reduced state space W. To every $x_n \in X$ we define $w_n := w(x_n) \in W$ to be the reference point in W which is closest to x_n. The optimal control policy $\hat{\Pi}(x)$ is then constructed in the discretized state space W as $\hat{\Pi}(x) = \Pi(w(x))$. To compute $\Pi(w)$, we associate to each possible state-action pair (w_μ, u_ν) a state-action value $Q_{\mu\nu} = Q(w_\mu, u_\nu)$. Given an optimal state-action value function $Q^*(w, u)$, an optimal policy $\Pi(w)$ is obtained by choosing controls corresponding to maximum value. If no explicit knowledge on system dynamics is available, temporal-difference learning [11] can be used to approximate Q^*. In particular the update rule

$$\begin{aligned}\Delta Q_{n+1}(w_n, u_n) = \beta_n \Big[& r_{n+1} \\ &+ \gamma \max_{u \in U} Q_n(w_{n+1}, u) - Q_n(w_n, u_n)\Big],\end{aligned} \tag{3}$$

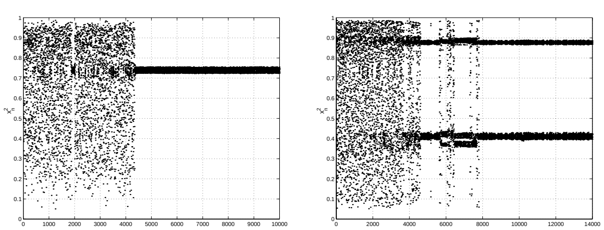
known as Q-learning [12], can be used to establish Q^* from immediate reinforcement signals r_{n+1} representing an immediate reward (or punishment) when performing control u_n in state w_n. In this work we punish unsuccessful controls: $r_{n+1} = -1$ if at iteration $n+1$ a control goal criterion is not met. If the control goal is met at iteration $n+1$ we set $\Delta Q_{n+1}(w_n, u_n) = 0$. (Alternatively we could use $r_{n+1} = 0$ which, however, resulted in slower performance.) This choice of reward measures the quality of a policy by λ, the number of iterations it takes to achieve the goal when starting from a random initial condition (an episode). For convergence to the optimal policy, β_n has to be frozen slowly to zero [13]. Moreover, to ensure exploration of the whole state-action space, controls have to be chosen from Q through a stochastic policy which is slowly frozen into a deterministic one [11] during learning in many episodes (offline control [8]). In practice, the global optimal policy can only rarely be found due to time constraints and one usually sets $\beta_n = \beta$ [11]. For chaotic systems, exploration of the whole state-space is a property of the system itself, thus we can restrict learning to a deterministic policy and few epsiodes (online control [8]).

The presented algorithm is very versatile and can be used under noisy and nonstationary conditions as was demonstrated in [8] for simple systems. No previous knowledge about system dynamics or of location of the desired state is necessary. The control goal represents only the general characteristica of the goal state. For example, if stabilization of a period 1 state is desired the criterion $w_n^i = w_{n+1}^i$ can be used. A period 2 state is characterized by $w_n^i = w_{n+2}^i$ and $w_n^i \neq w_{n+1}^i$. For the coupled system, synchronous and asynchronous period 2 states can appear. If the synchronous state is desired we have to require $w_n^i = w_n^{i-1}$ in addition.

3 Learning control of LCML

In all simulations described below, we set $a_0 = 3.8$ and pick ϵ such that the pattern to be controlled is unstable (we discuss only weak couplings here). Concerning the parameters of the control algorithm, we set $\beta = 1$ and $\gamma = 0.5$ (the particular values do not have a strong influence on the results) and r_{n+1} and W are as stated before. Initially the Q values are all set to zero.

First we established a successful control policy by controlling a small 1-D LCML with $i_d = 3$ and $j_d = 1$ through an online approach. We established control of three different patterns: a period 1 pattern, a synchronous period 2 pattern and an asynchronous period 2 pattern. The resulting action-value functions are denoted Q_1, Q_{s2} and Q_{a2} respectively. In Figure 1 a) we show the dynamics for the "master" map, x_n^2. The system was started in a random initial condition. After approximately 4,500 iterations a successful policy Q_1 is established. For the 1-D pattern we set $\epsilon = 0.1$, $a_u = 1$, $\Delta a = 0.15$ and $\epsilon_u = 0$. Figure 1 b) shows x_n^2 for online control of an asynchronous period 2 pattern. For the period 2 pattern we set $\epsilon = 0.05$ (for larger ϵ the period 2 patten becomes attractive [14]). To stabilize the period 2 pattern we chose $a_u = 1$, $\Delta a = 0.15$ and $\epsilon_u = 0$. The asynchronous period 2 pattern could also be stabilized by perturbing the coupling strengths between units with $\epsilon_u = 1$, $\Delta\epsilon = 0.025$ and $a_u = 0$.



a) b)

Figure 1: Online control of a) period 1 pattern, and b) period 2 pattern with parameters as stated in the text.

In a similar fashion, a larger 1-D LCML ($i_d = 100$) could be controlled to the different patterns. The map with $i = 50$ was chosen as the master system. Figure 2 shows online control of the LCML with the period one pattern as target. We see that the number of iterations until control is reached is about the same as for control of the small LCML.

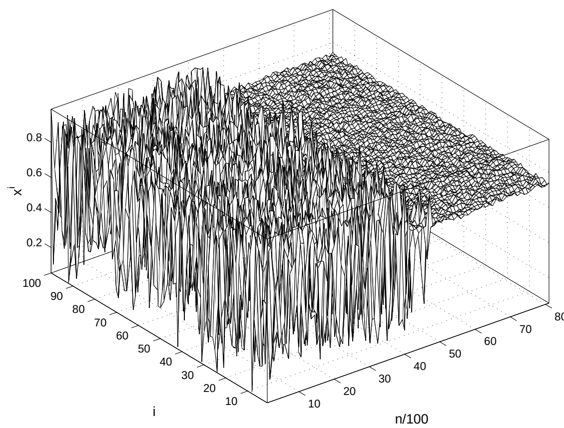


Figure 2: Control of a 1-D LCML with $N = 100$ elements.

In the context of hierarchical coupled systems it is interesting to see if a control policy established from interaction with a small system can succesfully control a larger system. To see how the control policies Q_1, Q_{s2} and Q_{a2}, established from the system with $i_d = 3, j_d = 1$, act on a larger LCML we used these policies to control a 1-D lattice with $i_d = 100$ and a 2-D lattice with $i_d = 20$ and $j_d = 20$. During control, the policies were not updated further. Control onto a period 1 pattern is shown in Figure 3. Starting from a random initial condition, the system is stabilized into a global fixed pattern after approximately 200 iterations. It is apparent that once a successful policy is established, control is achieved quickly. Similiar results were obtained for the synchronous and asynchronous period 2 patterns in both the 1-D and 2-D lattices.

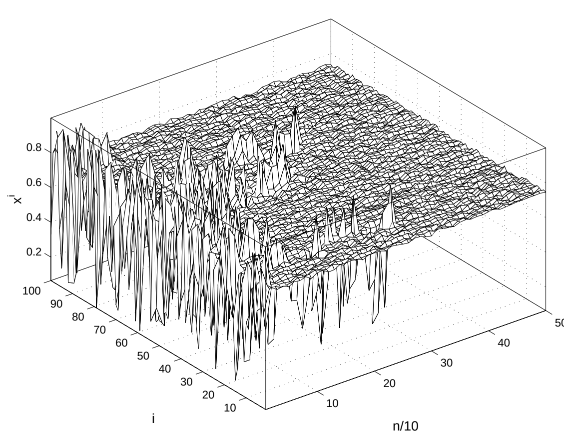


Figure 3: Control of a 1-D LCML with $i_d = 100$ elements using a policy established through control of a LCML with $i_d = 3$.

Various modifications of our control strategy are possible. For example, different subsystems may use different policies to establish control of a variety of patterns. To illustrate this, we show in Figure 4 the control of a 2-D lattice into a combination of a period 1 and period 2 pattern for $\epsilon = 0.05$. The subsystems with $i \leq 10$ were controlled according to policy Q_1, while the subsystems with $i > 10$ were controlled according to policy Q_{s2}. We see that after 1,400 iterations the system has reached a combination of a period 1 and a period 2 pattern.

To check performance under noisy conditions we added uniformly distributed noise $\delta_n \in \{-\Delta, \Delta\}$ to x^i at each iteration step and found that control could be established up to noise levels of $\Delta \approx 0.03$ if a was perturbed. With ϵ as control parameter we found successful control up to $\Delta \approx 0.01$.

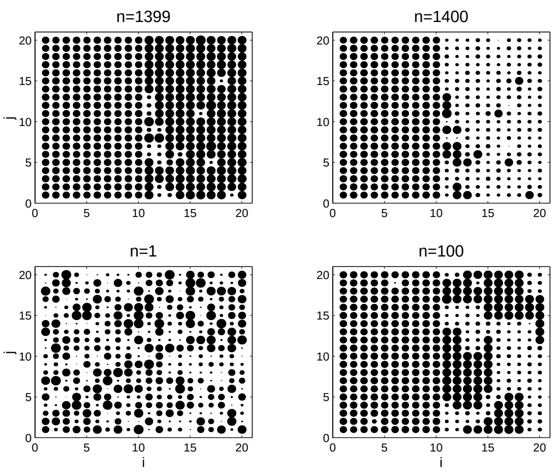


Figure 4: Control of a 2-D LCML with 20×20 elements into a combination of period 1 and period 2 patterns.

4 Conclusion

In this paper we demonstrated the control of 1-D and 2-D LCML using a method based on reinforcement learning. We were able to successfully stabilize different period 1 and period 2 patterns for small coupling strengths for which the desired pattern is unstable. The control was fed into the system by choosing an optimal state dependent perturbation of a control parameter in a discrete set of a priori fixed small values. A successful control policy could be established from information from a small neighborhood of a chosen "master" unit which is responsible for updating the globally used control policy. We have shown that a policy established from interaction with a small system can succesfully control a larger system and that the combination of different policies in different spatial regions can trigger the formation of new patterns. These findings might lead to interesting perspectives for information processing, in particular for biological systems, and open promising directions for *smart matter applications* (see [10]). To investigate this further, future research will focus on the application of the proposed method to continous coupled systems such as coupled oscillators and consider distributed control.

References

[1] E. Ott, C. Grebogi, and J.A. Yorke. Controlling chaos. *Physical Review Letters*, **64**:1196–1199, 1990.

[2] T. Kapitaniak. *Controlling Chaos*. Academic Press, San Diego, CA 92101, 1996.

[3] H. Gang and Q. Zhilin. Controlling spatiotemporal chaos in coupled map lattice systems. *Physical Review Letters*, **72**:68–71, 1994.

[4] K. Konishi and H. Kokame. Decentralized delayed-feedback control of a one-way coupled ring map lattice. *Physica D*, **127**:1–12, 1999.

[5] N. Parekh, R. Kumar, and D. Kulkarni. Synchronization and control of spatiotemporal chaos using time-series data from local regions. *Chaos*, **8**:300–305, 1998.

[6] P. Alsing, A. Gavrielides, and V. Kovanis. Using neural networks for controlling chaos. *Physica Review E*, **49**:1225–1231, 1994.

[7] R. Der and M. Herrmann. Q-learning chaos controller. *1994 IEEE International Conference on Neural Networks*, **4**:2472–2475, 1994.

[8] S. Gadaleta and G. Dangelmayr. Optimal chaos control through reinforcement learning. *Chaos*, **9**:775–788, 1999.

[9] C. Lin and C. Jou. Controlling chaos by GA-based reinforcement learning neural network. *IEEE Transactions on Neural Networks*, **10**:846–859, 1999.

[10] O. Guenther, T. Hogg, and B.A. Huberman. Learning in multiagent control of smart matter. *American Association for Artificial Intelligence Workshop on Multiagent Learning*, pages 1–5, 1997.

[11] R. Sutton and A. Barto. *Reinforcement Learning: An Introduction.* Bradford, MIT press, Cambridge, Massachusetts, 1998.

[12] C. Watkins. *Learning from delayed rewards.* PhD thesis, University of Cambridge, England, 1989.

[13] H. Robbins and S. Monro. A stochastic approximation method. *Ann. Math. Stat.*, **22**:400–407, 1951.

[14] F. Xie and G. Hu. Spatiotemporal periodic and chaotic patterns in a two-dimensional coupled map lattice system. *Physical Review E*, **55**:79–86, 1997.