Learning to control a complex multistable system

Sabino Gadaleta* and Gerhard Dangelmayr†
Department of Mathematics, Colorado State University, Weber Building, Fort Collins, Colorado 80523

(Received 25 June 2000; revised manuscript received 19 September 2000; published 27 February 2001)

In this paper the control of a periodically kicked mechanical rotor without gravity in the presence of noise is investigated. In recent work it was demonstrated that this system possesses many competing attracting states and thus shows the characteristics of a complex multistable system. We demonstrate that it is possible to stabilize the system at a desired attracting state even in the presence of high noise level. The control method is based on a recently developed algorithm [S. Gadaleta and G. Dangelmayr, Chaos **9**, 775 (1999)] for the control of chaotic systems and applies reinforcement learning to find a global optimal control policy directing the system from any initial state towards the desired state in a minimum number of iterations. Being data-based, the method does not require any information about governing dynamical equations.

DOI: 10.1103/PhysRevE.63.036217 PACS number(s): 05.45.−a, 05.40.Ca, 02.30.Xx

I. INTRODUCTION

The long-term behavior of nonlinear dynamical systems is generally classified as either stationary, periodic, quasiperiodic, or chaotic. These types of behaviors and their control are well studied and understood if the available states are well separated and their dimensions rather low. In recent years the attention has shifted to systems exhibiting more complex behaviors such as many coexisting attracting states. In general the term ‘‘complexity’’ has been coined to denote systems that have both elements of order and elements of randomness [1]. Such systems typically, but not necessarily, have many degrees of freedom, are composed of many complicated inter-related parts, and possess competing attracting sets. Minor perturbations induced, for example, by noise, can cause the system to undergo random transitions between different attracting states. Furthermore, due to the nontrivial relationship between the coexisting states and their basins of attraction, a final state depends crucially on the initial conditions [2]. This behavior is called *multistability* and was first studied experimentally in [3] and since then was observed in a variety of systems from different areas such as physics [4–6], chemistry [7], and neuroscience [8]. Adding noise to a multistable system will generate complex behavior and induce competition between the attractiveness towards regular motion in the neighborhood of an attracting state and the jumping between basins of attractions induced by noise [2]. The dynamics is then characterized by a large number of periodic attractors ‘‘embedded’’ in a sea of transient chaos [1]. The time the system spends in an attracting state corresponds to its ‘‘ordered’’ phase, and the transient time to its ‘‘random’’ phase. Added noise can prevent the system from settling down into an ordered phase.

Besides their importance for specific applications, a further motivation to study the dynamics and control of such complex systems is their possible role as information processing devices in neural information processing [9]. Complex systems are characterized by a very large number of coexisting states and, with adequate noise, the system can rapidly access these ordered states. The control of such systems, in particular under noisy conditions, would then offer the opportunity to utilize this multistable behavior for the processing and storage of information, i.e., different ordered states are identified with different ‘‘memorized’’ pieces of information. External input can be thought of as triggering a certain control mechanism that stabilizes a selected ordered state that would be associated with the given input.

The simplest prototype of a complex multistable system is provided by the model equations of a periodically kicked mechanical rotor [10,11,2] whose quantum mechanical counterpart plays an important role in the study of quantum chaos [12]. Until now, control was achieved for low noise levels through a simple feedback mechanism [11] that perturbs directly all system variables and requires computation of the Jacobian of the map close to the desired state. Moreover, this control technique is only local, i.e. the control is usually switched on only if the system is close to the desired state. In [11] the Jacobian was computed from the model equations. In many real-world applications, this information will not be available and specifically in the context of neural information processing it is unrealistic to base control methods on the basis of analytical knowledge of governing system equations. In some cases, the Jacobian can be estimated from observed data as suggested in [13]. In the presence of noise, however, this estimation can become very difficult.

Learning algorithms that do not require any analytical knowledge can be based on reinforcement learning and in recent works [14,15] reinforcement learning was shown to play an important role in neural information processing. It is therefore interesting to investigate the control of complex systems through reinforcement learning.

Related to the control of complex systems is the control of chaos. The use of reinforcement learning to control chaotic systems was first suggested by Der and Herrmann [16] who applied it to the logistic map. In [17] we generalized the method and applied it to the control of several discrete and continuous low-dimensional chaotic and hyperchaotic systems and recently to coupled logistic map lattices [18]. Lin and Jou [19] proposed a reinforcement learning neural network for the control of chaos and applied it to the logistic and the Hénon map. To control chaotic systems, unstable states embedded in the chaotic attractor are stabilized. To

*Email address: sabino@math.colostate.edu
†Email address: gerhard@math.colostate.edu

control multistable systems, the desired state is typically chosen to be one of the many existing attracting states. These states are *metastable* (stable only for a finite time) above a certain noise level and the control must stabilize dynamics against noise.

Although the characteristics of the stabilized state in a chaotic system differ from the desired state in a multistable system, we will show in this paper for the case of a periodically kicked rotor that the method developed in [17] for chaotic systems is also well suited for the control of complex multistable systems in the presence of significant noise levels. Instead of perturbing the system state directly, we apply parametric control.

II. BASIC EQUATIONS

The differential equation describing the temporal evolution of the phase angle θ of a forced damped pendulum with forcing $f(t)$ and damping d is given by

$$\theta''+d\theta'=f(t)\sin\theta. \tag{2.1}$$

If the external force acts periodically and impulsively on the rotor,

$$f(t)=f_1\sum_n \delta(t-n), \tag{2.2}$$

the dynamics is most conveniently described by its return map. Let $\theta(t)$ be the solution at time t and let $\theta_n=\theta(n)$ be its value at the nth kick. Due to the δ-forcing the velocity $\theta'(t)$ shows a discontinuity at $t=n$,

$$\theta'(n+0)-\theta'(n-0)=f_1\sin\theta_n, \tag{2.3}$$

whereas $\theta(t)$ is continuous. The solution between two successive kicks is then given by

$$\begin{aligned}\theta(t)&=\theta_n-\frac{l_n}{d}(e^{-d(t-n)}-1), \quad n\leqslant t\leqslant n+1,\\ l_n&:=\theta'(n-0)+f_1\sin\theta_n.\end{aligned} \tag{2.4}$$

It follows

$$\theta_{n+1}=\theta_n-\frac{l_n}{d}(e^{-d}-1) \tag{2.5}$$

and

$$l_n=l_{n-1}e^{-d}+f_1\sin\theta_n. \tag{2.6}$$

For simplicity we set $c=1-e^{-d}$ $(0<c\leqslant1)$. Equation (2.5) with n replaced by $n-1$ yields $u_{n-1}=(d/c)(\theta_n-\theta_{n-1})$ and from Eq. (2.6) we obtain

$$l_n=\frac{d}{c}(1-c)(\theta_n-\theta_{n-1})+f_1\sin\theta_n, \tag{2.7}$$

which, when substituted back into Eq. (2.5), leads to a ''finite difference form'' for the phase angle θ_n,

$$\theta_{n+1}-2\theta_n+\theta_{n-1}+c(\theta_n-\theta_{n-1})=f_0\sin\theta_n, \tag{2.8}$$

where $f_0=cf_1/d$. By introducing the new variable $y_n=\theta_n-\theta_{n-1}$, we obtain from Eq. (2.8) the *dissipative standard map*

$$\begin{aligned}y_{n+1}&=(1-c)y_n+f_0\sin\theta_n,\\ \theta_{n+1}&=\theta_n+y_{n+1} \pmod{2\pi},\end{aligned} \tag{2.9}$$

which is related to the system just before successive kicks. Introducing the variable $v_n=y_{n+1}$ we can rewrite this map in the form

$$\begin{aligned}\theta_{n+1}&=\theta_n+v_n \pmod{2\pi},\\ v_{n+1}&=(1-c)v_n+f_0\sin(\theta_{n+1}),\end{aligned} \tag{2.10}$$

which describes the state of the system just after two successive kicks. The map (2.10) was extensively studied in [10]. For $c=0$ it results in the *Chirikov standard map* [20]. In this undamped, Hamiltonian limit the state space consists of a chaotic sea interspersed with regular motion represented by stability islands of stable periodic orbits. The largest regions of regular motion are made up by the islands of the primary periodic orbits. These are the fixed points (period 1, $\theta=\pi, v=2m\pi$, $m=0,\pm1,\ldots$) and higher period periodic orbits that are present for $f_0=0$. Further, secondary stable periodic orbits occur for $f_0\neq0$. Their islands are grouped around the primary islands and are in general much smaller, inducing a hierarchical organization of the stable periodic orbits [10,21]. We note that in the undamped case the range of the velocity v can also be identified with a circle ($v \mod 2\pi$), i.e. the infinitely extended cylindrical phase space is compactified to a torus. On the torus the infinite family of primary fixed points is represented by a single point, but the number of all periodic orbits is still assumed to be infinite [21].

On the other hand, for very strong damping ($c\approx1$) one obtains the one-dimensional *circle map* with a zero phase shift,

$$v_{n+1}=v_n+f_0\sin v_n, \tag{2.11}$$

which exhibits the usual Feigenbaum scenario for the transition to chaos. In particular, this map possesses only one attractor in large regions of the phase space [21].

When a small amount of dissipation ($0<c\ll1$) is added to the undamped system, stable periodic orbits turn into sinks or disappear. The parameter dependence of the sinks and their basins of attraction have been studied numerically by Feudel *et al.* [10]. We shortly summarize some of the main results of this study. For $c\neq0$ the range of v can no longer be identified with a circle. The phase space is now an infinitely extended cylinder on which we find an infinite number of stable periodic orbits in the limit $c\to0$, in particular the primary family of fixed points, denoted by P_1. For $c>0$ all trajectories are eventually trapped in the finite cylinder $|v|\leqslant f_0/c$ that contains all attractors. The number of sinks is now finite, but can be made arbitrarily large by choosing c

sufficiently small. Specifically, only a finite number of the formerly infinite P_1 family can be present. These fixed points still have $v=2\pi m$ but the phases are now different and the size of the attraction basins quickly decreases with increasing $|m|$. The main fixed point, P_1^0 $(m=0)$, has the largest basin. In addition, when f_0 is varied one finds births of sinks in saddle node bifurcations and their disappearance in period-doubling sequences, but the resulting chaotic parameter regimes are extremely small.

Concerning basin boundaries, these appear all to be fractalized giving rise to chaotic transients along chaotic saddles and hence to uncertainty in initial conditions. The question as to which extent the basins are riddled has only partly been addressed in [10]. The basins of the P_1 family contain full (though small in size for larger m) neighborhoods of the fixed points and, therefore, cannot be fully riddled, but partial riddling as defined in [22] is not excluded. The basins of other periodic orbits are smaller and full riddling might occur in some cases, but this requires further investigation. In summary, the kicked rotor with small dissipation serves as an example of a multistable system characterized by a complicated coexistence of many periodic sinks with sensitive dependence on initial conditions.

The complexity of multistable systems can further be enhanced through the introduction of noise, which leads to unpredictable transitions between different attracting states revealed by almost periodic motion interspersed by random bursts [2]. In addition Kraut *et al.* [2] observed a decrease in the number of accessible states. Their results indicate that the noise induces a preference of certain attractors.

In the following sections we show that the multistable rotor can be stabilized at a desired attracting state through a control method based on reinforcement learning. We show that control can be achieved up to noise levels as high as $\delta=0.4$ [see Eq. (3.1) below]. As control parameter we choose the forcing f_0. The allowed control actions consist of small discrete perturbations of f_0.

III. THE NOISY UNCONTROLLED ROTOR

The system investigated by Kraut *et al.* [2] has the form

$$\theta_{n+1}=\theta_n+v_n+\delta_\theta \pmod{2\pi},$$
$$v_{n+1}=(1-c)v_n+f_0\sin(\theta_{n+1})+\delta_v, \qquad (3.1)$$

where δ_θ and δ_v are the components of the uniformly and independently distributed noise vector with bounded norm: $\sqrt{\delta_\theta^2+\delta_v^2}\leqslant\delta$. Throughout this section we set the unperturbed forcing to $f_0=3.5$ and the damping to $c=0.02$. For these parameter values Kraut *et al.* [2] found numerically 111 stable periodic orbits in the noiseless limit. Most of these orbits belong to the P_1 family ($\theta=\theta_m, v=2m\pi$) and some of them have period 3. Only 0.01% of all found orbits have periods other than 1 and 3, so these orbits do not play an important role. With noise added, Kraut *et al.* [2] observed three different types of behavior. For small noise level ($\delta\lessapprox 0.05$) the trajectory may be trapped in the open neighborhood of an attractor forever. For intermediate noise level ($0.05\lessapprox\delta\lessapprox 0.1$) attracting periodic orbits still could be identified. However, when an attracted region around such an orbit is reached, the noise will eventually drive the system to the basin boundary where it then follows for a while a chaotic saddle until it reaches a neighborhood of another attractor. The resulting dynamics is characterized by almost periodic motion interrupted by bursts, and the smaller the noise is the larger are the regular phases. This behavior is referred to as attractor hopping and may also be considered as noise-induced chaos. In a sense, sinks of the noiseless system turn into saddles of a ‘‘noise-induced chaotic attractor.’’ Controlling a periodic orbit in this regime resembles the control of an unstable periodic orbit embedded in a chaotic attractor of a deterministic system. However, while in the deterministic case unstable directions have to be avoided by forcing the system to remain on the stable manifold, in the noisy case random pushes towards the basin boundary have to be suppressed. It is therefore questionable whether Ott-Grebogi-Yorke–type methods [13] work in the noisy case because stable and unstable manifolds can hardly be identified.

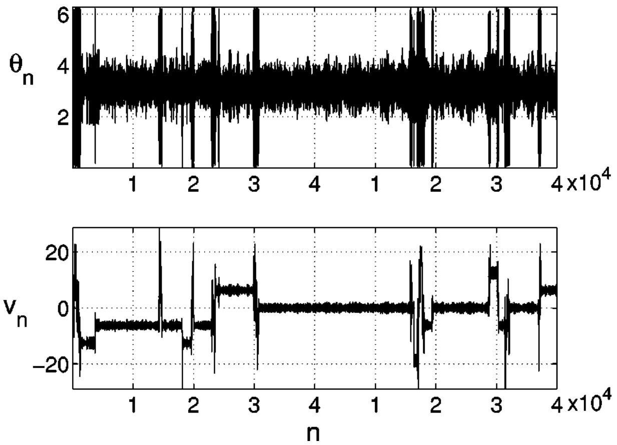


FIG. 1. Dynamics of the noisy kicked rotor for $f_0=3.5$, $c=0.02$, and $\delta=0.09$.

Which of the periodic orbits of the noiseless system are observed when noise is present depends on the sizes of their basins and the noise level. For example, for $\delta=0.09$, we observe hopping between fixed points of the primary family P_1, but no period 3 orbits could be identified, see Fig. 1. When the noise level is reduced, period 3 orbits occur in addition to the primary fixed points. In Fig. 2 we show the probability density $p(v,\theta)$ in the rectangle $[-7\pi,7\pi]\times[0,2\pi]$ for (a) $\delta=0.02$ and (b) $\delta=0.09$. Covering the rectangle with an 80×80 grid the density was numerically generated by iterating 1000 initial conditions for 1000 iterations and counting the visits to each grid cell. In Fig. 2(a) we clearly see small peaks corresponding to period 3 orbits that surround the large peaks corresponding to the primary fixed points. In Fig. 2(b) the small peaks have disappeared and the large peaks are broadened, suggesting that the attraction basins of the period 3 orbits are now absorbed in the basin boundaries of the primary fixed points.

For increasing noise level δ the jumps become more and more frequent up to a point where the system does not remain in a stable state for more than a few iterations. Kraut

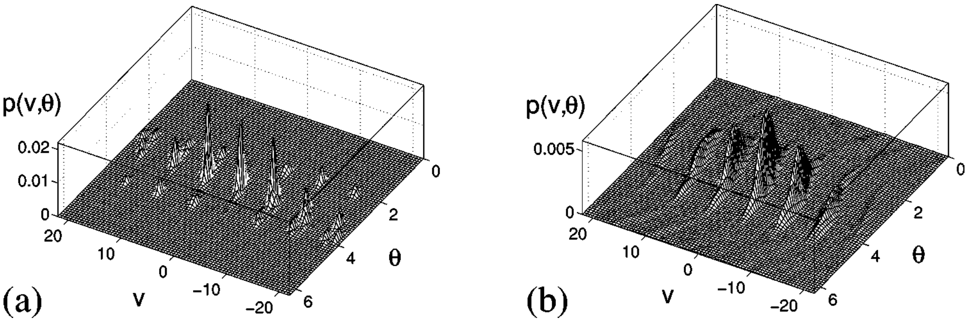


FIG. 2. Probability density $p(v,\theta)$ in the rectangle $[-7\pi,7\pi]\times[0,2\pi]$ for (a) $\delta=0.02$ and (b) $\delta=0.09$.

et al. [2] refer to this type of behavior as predominance of diffusion due to noise, the system behaves now truly stochastically. The transition from hopping to diffusion is marked by a sharp increase of the exponent of the autocorrelation function of a noisy trajectory that occurs in a range about $\delta=0.1$ [2]. In the diffusion-dominated regime the fine structure of the basins is blurred by noise, but some of this structure is still present preventing the motion from being a pure (δ-correlated) random process. A typical time series in the diffusive regime is shown in Fig. 3(a) for $\delta=0.3$. Figure 3(b) shows the corresponding phase plot.

In the following, we will use the notation $\mathbf{x}_n=(\theta_n,v_n)\in X=T^1\times\mathbb{R}$ to denote the state of the rotor at the nth iteration. We will attempt to control the system (3.1) through small discrete perturbations u_n of f_0,

$$f_n=f_0+u_n. \tag{3.2}$$

IV. THE CONTROL ALGORITHM

In this section, we describe the control algorithm used to stabilize the noisy rotor at a prescribed state. As mentioned in the last section, we control the rotor through small discrete state-dependent perturbations $u(\mathbf{x})$ of the applied forcing f_0. The dynamics of the controlled system can then be written in the form

$$\begin{aligned}\theta_{n+1}&=\theta_n+v_n+\delta_\theta \pmod{2\pi},\\ v_{n+1}&=(1-c)v_n+(f_0+u_n)\sin(\theta_n+v_n)+\delta_v,\end{aligned} \tag{4.1}$$

where $u_n=u(\mathbf{x}_n)$ represents the state-dependent control perturbation applied to the external forcing f_0 at the nth iteration step.

In terms of optimal control theory [23], the task is to find an optimal control policy $\pi^*(\mathbf{x},u)$ associating to every state $\mathbf{x}$ a control u such that the control goal is achieved in an optimal way. Concerning the control of the rotor, an optimal policy should allow the stabilization of a prescribed state and in addition allow to reach its neighborhood in a minimum number of iterations from any initial condition.

If analytical knowledge on system dynamics is available, methods from optimal control theory such as dynamic programming [24] or Lagrangian relaxation [25] can be used to establish an optimal control policy. Here we do not assume that such analytical knowledge is available. Then reinforcement learning [26], also known as neurodynamic programming [27], which has been shown to find good solutions to optimal control problems [27], can be applied. Even if analytical knowledge on system dynamics is available, reinforcement learning is often applied instead of analytical techniques since it is easier to implement and computationally often more efficient [27].

A. Reinforcement learning

Reinforcement learning methods offer a solution to the problem of learning from interaction of a decision maker, called agent, with a controlled environment to achieve a goal. At each discrete time step $n=1,2,\ldots$ the agent receives a representation $\mathbf{w}_n\in W$ of the environment's state $\mathbf{x}_n\in X$, where W is the (finite) set of all possible state representations. On the basis of $\mathbf{w}_n$, the agent selects a control (or action) $u_n\in U$, the set of all available actions. The control is selected according to a policy π, which is described by a probability distribution $\pi_n(\mathbf{w},u)$ of choosing $u_n=u$ if $\mathbf{w}_n=\mathbf{w}$. One time step later, as a consequence of the control u_n, the agent receives a numerical reward r_{n+1} and a new state $\mathbf{w}_{n+1}$. The goal is to maximize the accumulated rewards.

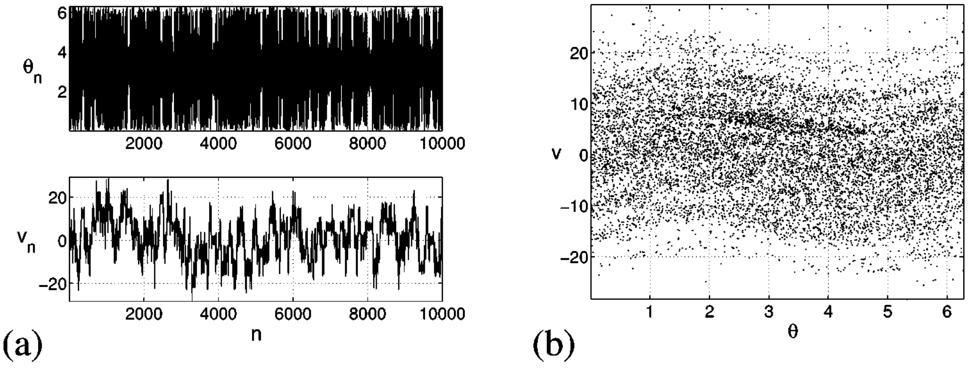


FIG. 3. (a) Dynamics of the noisy kicked rotor for $\delta=0.3$. (b) Corresponding phase plane representation of the dynamics.

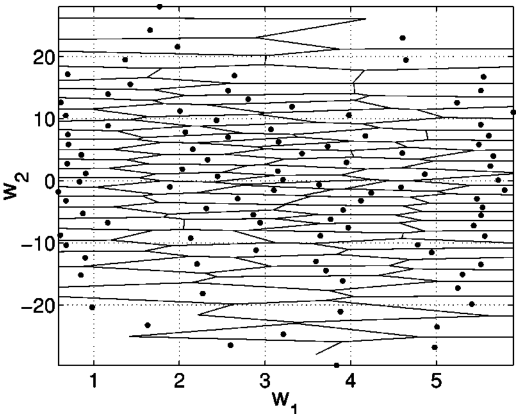
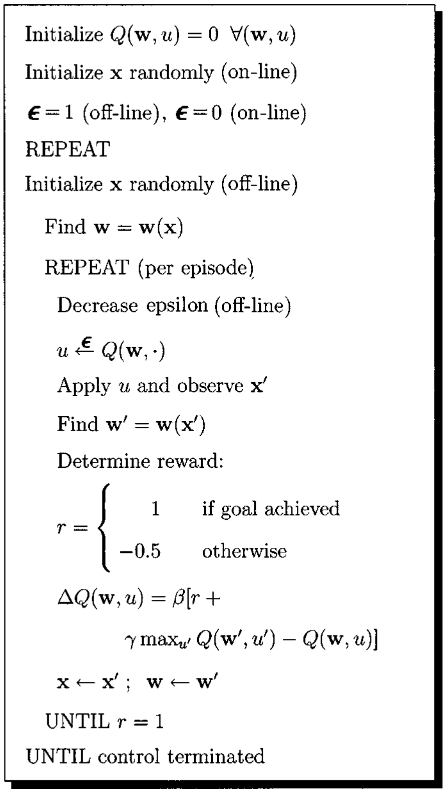


FIG. 4. The set W of 100 reference vectors $\mathbf{w}$ and their corresponding Voronoi cells obtained through a neural-gas vector quantization of 10 000 data points of the uncontrolled dynamics with $\delta = 0.3$.

Reinforcement learning methods offer ways of improving the policy through observations of delayed rewards to find an optimal policy that associates with each state an optimal control such that rewards are maximized over time.

Problems with delayed reinforcements are well modeled as finite Markov decision processes (MDPs). A finite MDP consists of a finite set of states $\mathbf{w} \in W$ and controls $u \in U$, a reward function $R(\mathbf{w},u)$ specifying the expected instantaneous reward of the current state and action, and a state transition function $P^u(\mathbf{w}',\mathbf{w})$ denoting the probability of a transition from state $\mathbf{w}$ to $\mathbf{w}'$ using control u. The model is Markovian if the state transitions depend only on the current state and control.



```
Initialize $Q(\mathbf{w},u) = 0 \ \forall(\mathbf{w},u)$
Initialize $\mathbf{x}$ randomly (on-line)
$\epsilon = 1$ (off-line), $\epsilon = 0$ (on-line)
REPEAT
Initialize $\mathbf{x}$ randomly (off-line)
  Find $\mathbf{w} = \mathbf{w}(\mathbf{x})$
  REPEAT (per episode)
    Decrease epsilon (off-line)
    $u \overset{\epsilon}{\leftarrow} Q(\mathbf{w},\cdot)$
    Apply $u$ and observe $\mathbf{x}'$
    Find $\mathbf{w}' = \mathbf{w}(\mathbf{x}')$
    Determine reward:
    $r = \begin{cases} 1 & \text{if goal achieved} \\ -0.5 & \text{otherwise} \end{cases}$
    $\Delta Q(\mathbf{w},u) = \beta[r +$
          $\gamma \max_{u'} Q(\mathbf{w}',u') - Q(\mathbf{w},u)]$
    $\mathbf{x} \leftarrow \mathbf{x}'$ ; $\mathbf{w} \leftarrow \mathbf{w}'$
  UNTIL $r = 1$
UNTIL control terminated
```

FIG. 5. Summary of the proposed rotor control algorithm based on Q learning. Both the on-line and off-line versions are shown.

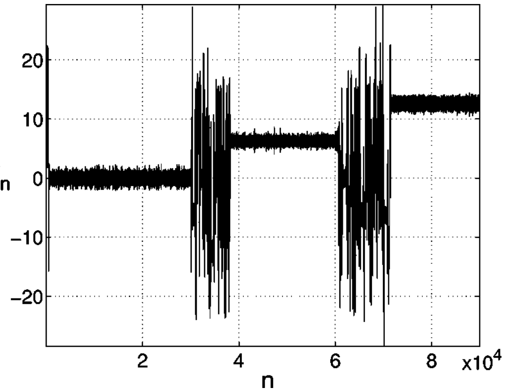


FIG. 6. On-line control of the rotor dynamics with $\delta = 0.09$ and $U = \{0,0.2,-0.2\}$. Initially the control goal is $\mathbf{x}_0$. After 30 000 iterations the control goal is switched to $\mathbf{x}_{2\pi}$ and after 60 000 iterations to $\mathbf{x}_{4\pi}$.

In general one seeks to maximize the expectation value of the discounted return

$$\hat{r}_n = \sum_{k=0}^{\infty} \gamma^k r_{n+k+1}, \quad 0 \leq \gamma < 1. \tag{4.2}$$

Most reinforcement learning algorithms are based on estimating the state-action value function

$$Q^{\pi}(\mathbf{w},u) = E_{\pi}\{\hat{r}_n | \mathbf{w}_n = \mathbf{w}, u_n = u\}, \tag{4.3}$$

which gives the value of a state-action pair under a certain policy. An optimal state-action value function $Q^*(\mathbf{w},u)$ is defined for given $(\mathbf{w},u)$ as the maximum over the set of all policies π,

$$Q^*(\mathbf{w},u) = \max_{\pi} Q^{\pi}(\mathbf{w},u). \tag{4.4}$$

Given an optimal state-action value function $Q^*(\mathbf{w},u)$, choosing in any state $\mathbf{w}$ the action u^* with associated maximal value Q^*,

$$u^*(\mathbf{w}) = \arg\max_{u \in U} Q^*(\mathbf{w},u), \tag{4.5}$$

leads to an optimal control strategy. Here $\arg\max_u$ denotes the value of u at which the expression that follows is maximized.

One can show [28] that Q^* satisfies the Bellman fixed point equation

$$Q^*(\mathbf{w},u) = R(\mathbf{w},u) + \gamma \sum_{\mathbf{w}' \in W} P^u(\mathbf{w}',\mathbf{w}) \max_{u'} Q^*(\mathbf{w}',u'). \tag{4.6}$$

With available analytical knowledge on system dynamics, i.e., R and P are known, the solution Q^* can be found through dynamic programming [23,26]. For unknown R,P

TABLE I. Comparison of on-line (Q) and off-line (Q^*) controlled systems with the uncontrolled system. See text for details.

Goal	λ_u	$P_T(u)$ (%)	λ_u^l	λ_Q	$P_T(Q)$ (%)	λ_Q^l	λ_{Q*}	$P_T(Q^*)$ (%)	λ_{Q*}^l
0	524	27	7441	590	46	5671	398	98	590
2π	582	22	7928	557	48	5467	417	99	512
4π	1700	10	9170	516	56	4688	579	98	767

temporal-difference learning [29] offers a way to estimate Q^*. In temporal-difference learning, Q^* is found through a stochastic Robbins Monro approximation [30] according to an update rule of the form [17]

$$Q_{n+1}=Q_n+\Delta Q_n, \tag{4.7}$$

where the update ΔQ_n is based on observations of the present state $(\mathbf{w}_n,u_n)$ and the next state $(\mathbf{w}_{n+1},u_{n+1})$. A particular choice of ΔQ_n is given by Watkin's Q learning [31],

$$\Delta Q_n(\mathbf{w}_n,u_n)=\beta_n[r_{n+1}+\gamma \max_{u\in U} Q(\mathbf{w}_{n+1},u)-Q(\mathbf{w}_n,u_n)], \tag{4.8}$$

where r_{n+1} represents an immediate reward received for performing the control u_n in state $\mathbf{w}_n$. In this work we punish unsuccessful actions by setting $r_{n+1}=-0.5$ whenever at the $(n+1)$th iteration the goal was not achieved and otherwise we set $r_{n+1}=1$.

Q learning can be proven to converge towards Q^*, if the whole state space is explored and β_n is slowly frozen to zero [32]. In real applications, due to time constraints, it will rarely be possible to satisfy this requirement and one often uses a fixed β, $\beta_n=\beta$ [26]. Then the estimated policy will not be the globally optimal one but an approximation to it. To ensure exploration of the whole state space, control actions are chosen from the corresponding Q values according to a specified policy that initially chooses actions stochastically and is slowly frozen into a deterministic policy. This can, for example, be achieved through ϵ-greedy policies π_ϵ [26]. Here, an action that is different from the greedy action (the one with maximal estimated action value) is chosen with probability ϵ, where ϵ is initially one and then slowly frozen to zero, i.e.,

$$\pi_\epsilon(\mathbf{w},u)=\begin{cases} 1-\epsilon+\epsilon/|U|, & u=u^* \\ \epsilon/|U|, & u\neq u^*. \end{cases} \tag{4.9}$$

In Eq. (4.9), $|U|$ denotes the cardinality of U and $\pi_\epsilon(\mathbf{w},u)$ is the probability of choosing action u in state $\mathbf{w}$. We call a policy greedy or deterministic if exclusively the best action u^* is chosen ($\epsilon\equiv 0$). An optimal state-action value function Q^* associates with each state $\mathbf{w}$ a control u such that when control actions are performed according to a greedy policy from Q^* the goal is achieved in an optimal way. In this work we will measure the optimality of a policy in terms of the average number of iterations λ it takes to achieve the goal when starting from a random initial condition (iterations per episode). The goal will be to stabilize a prescribed state.

B. Representation of state space

The reinforcement learning algorithm presented in the last subsection requires a finite state representation W and a finite action space U. The complexity of the reinforcement learning problem increases exponentially with the size of W and U, and it is desirable to keep the sizes $|W|,|U|$ of the sets W and U small.

Concerning the action space, we choose a minimal possible set of actions $U=\{0,u_{max},-u_{max}\}$ acting on the external forcing: $f_0\rightarrow f_0+u_n, u_n\in U$. u_{max} will be small compared to f_0. In other words, the actions are restricted to either force the system slightly stronger, slightly less, or with an unchanged forcing.

The set W represents a finite approximation to the true state space X and we will construct W through a vector quantization technique. The outcome of the vector quantization is a set W of reference vectors $\mathbf{w}\in W$, which partitions X into so-called Voronoi cells whose centers $\mathbf{w}$ form the necessary discrete-state approximation. Each state $\mathbf{x}\in X$ is projected to exactly one $\mathbf{w}(\mathbf{x})\in W$, where $\mathbf{w}(\mathbf{x})$ is the closest reference vector according to some (usually Euclidean) norm

$$\mathbf{w}(\mathbf{x})=\arg\min_{\mathbf{w}\in W}||\mathbf{x}-\mathbf{w}||. \tag{4.10}$$

To every reduced state $\mathbf{w}$, we associate an allowed set of controls $U(\mathbf{w})$. In this work the set U is fixed for all $\mathbf{w}$. To each possible pair of reduced state $\mathbf{w}$ and allowed control signal $u\in U$, we associate a state-action value $Q(\mathbf{w},u)$ representing the value of performing control u when the system is in state $\mathbf{x}$, such that $\mathbf{w}=\mathbf{w}(\mathbf{x})$. Whenever the system is in a state $\mathbf{x}$, its corresponding reference vector $\mathbf{w}(\mathbf{x})$ is identified and a control $u(\mathbf{x})$ is chosen from the set U according to the policy defined through the values $Q(\mathbf{w}(\mathbf{x}),u)$.

Vector quantization methods have been used in applications such as time series prediction [33] or chaos control [16,17] to construct a finite state-space approximation. Specifically the neural-gas algorithm [33] has been shown to be well suited for the approximation of chaotic attractors since it is topology preserving [34].

To approximate the state space of the noisy rotor we applied the neural-gas algorithm [33] with $N=100$ reference vectors to the set of data points $\mathbf{x}$ obtained from simulations with $\delta=0.3$ shown in Fig. 3(b). The centers resulting from the neural-gas quantization are shown in Fig. 4 together with their Voronoi cells.

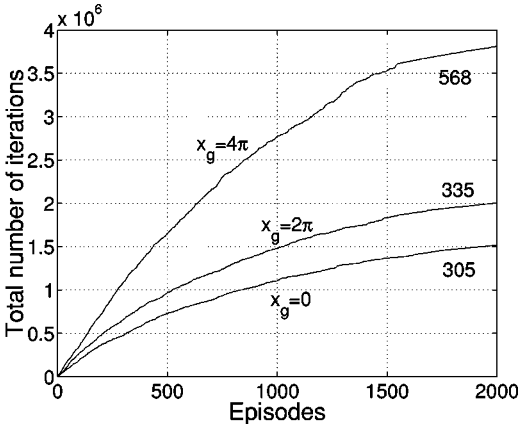


FIG. 7. Off-line control of the rotor dynamics with $\delta = 0.09$ and $U = \{0,0.2,-0.2\}$. The curves labeled $x_g = 0$, $x_g = 2\pi$, and $x_g = 4\pi$ represent the total number of iterations (the sum over the iterations of the performed episodes) during the learning process for control of the goal $\mathbf{x}_{x_g}$. ϵ is slowly frozen from initially one to zero. With increasing number of episodes and decreasing ϵ, a decrease in the slope of the curve shows convergence of the control policy. During the last 500 episodes ϵ was set to zero. The limiting slope (number below the curves) is an estimate of λ for the particular policy.

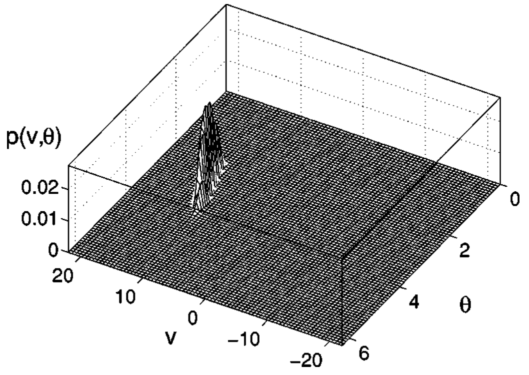


FIG. 9. Probability density $p(v,\theta)$ in the rectangle $[-7\pi,7\pi]\times[0,2\pi]$ when using policy $Q^*4\pi$ for δ=0.09.

We are not certain whether the proposed algorithm can be successfully applied to the control of attractors with fully riddled basins of attraction. The periodic orbits whose control will be demonstrated in the next section possess regions in phase space (in particular regions close to the attractors) that are dominated by points leading to the same attracting state. In these regions the reduced representation can represent the true phase space X as long as the representation is fine enough. We will see in the next section that the codebook of size $|W| = 100$ as shown in Fig. 4 leads to successful control of the fixed points of the P_1 family. For stabilization of period 3 orbits, however, a finer set of size $|W| = 200$ was necessary. We emphasize here that the discrete phase-space representation concerns only the control strategy. The dynamics itself evolves in the full continuous phase space. This is certainly consistent with the requirements of real world applications where a finite state representation is inevitable.

In this paper we treat the construction of W and the approximation of the control policy as separate steps in the algorithm and we choose the size of W in advance. This is not necessary since both steps can be combined. Specifically the combination of growing vector quantization methods (e.g., [35,36]) with the approximation of the control policy can lead to minimal codebooks well suited for specific control problems as initial results concerning chaos control suggest [37].

The complete algorithm for the approximation of a control policy, i.e., a set Q of state-action values, is presented in Fig. 5 for on-line and off-line control as described in the next section. Once a control policy is established, the system is controlled by choosing the perturbations $u_n(\mathbf{x}) \in U$ greedy with respect to the associated state-action values $Q(\mathbf{w}(\mathbf{x}),\cdot)$.

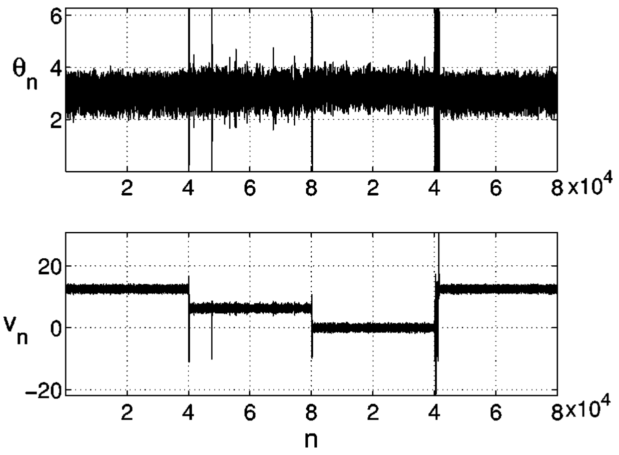


FIG. 8. Off-line control of the rotor dynamics with $\delta = 0.09$ over 80 000 iterations. The control policy is reset every 20 000 iterations from initially $Q^*_{\mathbf{x}_{4\pi}}$ to $Q^*_{\mathbf{x}_{2\pi}}$, $Q^*_{\mathbf{x}_0}$ and back to the initial policy $Q^*_{\mathbf{x}_{4\pi}}$.

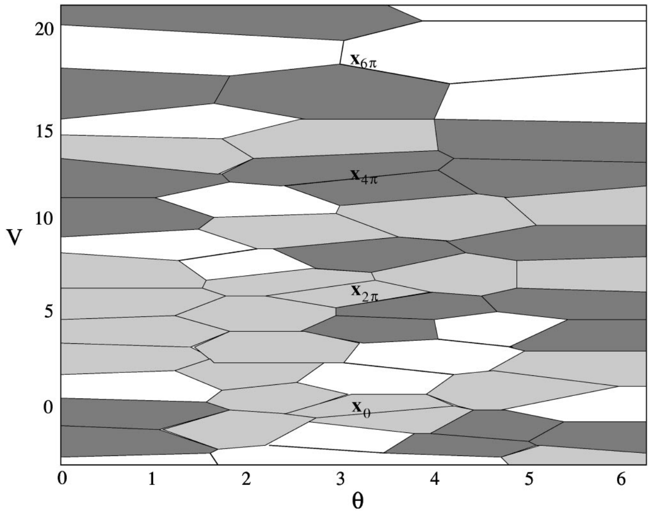


FIG. 10. Visualization of the policy $Q^*_{\mathbf{x}_{4\pi}}$. The dark (light) gray areas correspond to regions where the control $u_n = -0.2$ ($u_n = 0.2$) is associated with a maximal state-action value. In the white areas no control will be applied.

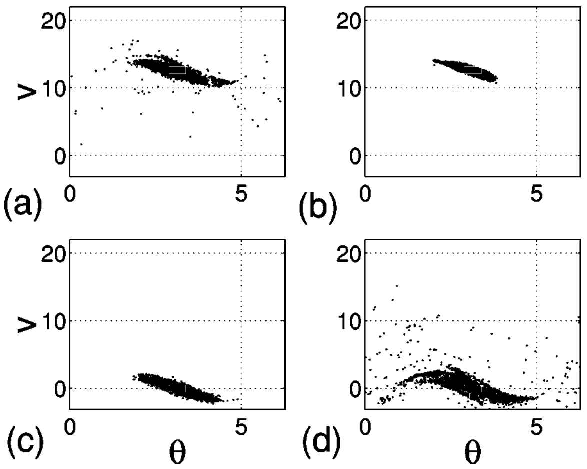


FIG. 11. Local performance of the policy $Q^*_{\mathbf{x}_{4\pi}}$. Shown are the points obtained after 10 iterations ($\delta = 0.09$) with 1000 initial conditions (θ, v) randomly chosen from a small rectangle (as shown) around (a),(b) $\mathbf{x}_{4\pi}$ and (c),(d) $\mathbf{x}_0$. (a),(c) shows uncontrolled dynamics and (b),(d) shows dynamics controlled according to the policy $Q^*_{\mathbf{x}_{4\pi}}$.

V. RESULTS

In this section we present results obtained by applying the algorithm discussed in the previous section to stabilize the rotor at the fixed points $\mathbf{x}_g = (\theta_g, g)$ for $g = 0$, 2π, and 4π and with $\theta_g \approx \pi$. Unless otherwise stated, the control set U was restricted to $U = \{0, u_{max}, -u_{max}\}$ with $u_{max} = 0.2$ and the noise level was set to $\delta = 0.09$. In previous works [1,11] control could be established only up to very low noise levels ($\delta = 0.01$ in [11]). The parameters of the Q-learning update rule were set to the constant values $\beta = 0.5$ and $\gamma = 0.9$ but their particular choice does not influence results much. To measure the quality of an approximated policy defined through the Q values, we introduce the quantity λ_Q that measures the average number of iterations per episode, where an episode is an iteration of the system, controlled on the basis of the Q values, starting at a random initial condition until the criterion for termination of an episode is met. λ_u denotes the same quantity for the uncontrolled system. For computation of λ we terminate an episode if the condition $||\mathbf{x}_g - \mathbf{x}_n|| < 1$ is met for 200 consecutive iterations.

A. On-line control

On-line control refers to learning in one episode. During system dynamics, starting from a random initial condition with all Q values set to zero, control perturbations, chosen greedy from the current set of Q values, are applied to the forcing function and the control algorithm updates the Q function from immediate rewards r_{n+1}, where $r_{n+1} = -0.5$ if $\|\mathbf{x}_g - \mathbf{x}_n\| > 1$ and $r_{n+1} = 1$ otherwise. (Note that we can easily generalize the algorithm to situations where the exact location of the fixed point, i.e., $\mathbf{x}_g$, is unknown. See [17] for details.) Eventually the controller will find a successful control strategy and keep the system stabilized in the desired state. Figure 6 shows on-line control of the noisy rotor with $\delta = 0.09$. Initially the control goal was to stabilize the system in the state $\mathbf{x}_0$. After 30 000 (60 000) iterations, Q was reset to zero and the control goal changed to the stabilization of $\mathbf{x}_{2\pi}$ ($\mathbf{x}_{4\pi}$). We see that the controller is able to stabilize the rotor at the desired location after only a small number of iterations in all three cases. In a similar fashion, we were able to stabilize all fixed points (θ_g, g) with $g = \pm 2m\pi$ for m up to 4.

In Table I we summarize the performance of the approximated policies for stabilizing the different goals $\mathbf{x}_g$. λ was averaged over 2000 terminating episodes. As additional performance criterion, we introduce the probability $P_T(Q)$ that a policy Q will succeed. To determine this probability, we count the number λ_{nt} of episodes that did not terminate before a total of 2000 terminating episodes occurred. An episode was counted as unterminated if it did not terminate after 10 000 iterations. $P_T(Q)$ is then $100 \times 2000/(2000 + \lambda_{nt})$. $P_T(u)$ denotes the probability of satisfying the termination criterion without control. A good policy Q should satisfy $P_T(Q) \gg P_T(u)$ and $\lambda_Q \ll \lambda_u$. A lower limit λ^l for the average number of iterations is given by $\lambda^l = P_T \lambda + 10\,000(1 - P_T)$. These performance measures are shown in Table I for on-line (Q) and off-line (Q^*) (see next subsection) approximated policies for the three goals. Use of the on-line approximated policy improves performance considerably over the uncontrolled system, but the policy has low termination probability. To approximate a policy with higher termination probability, off-line control can be used.

B. Off-line control

To better satisfy the requirements of convergence to an optimal policy off-line control can be performed. In off-line control, learning is performed ϵ-greedy in many episodes where each episode is started from a new randomly chosen initial condition. For the learning process, an episode was

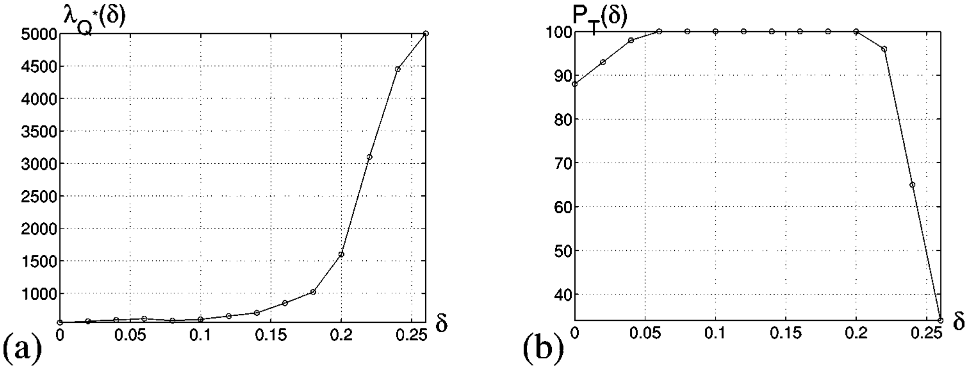


FIG. 12. Performance measures of the controlled rotor as functions of δ when using $u_{max} = 0.2$ and the policy $Q^*4\pi$ learned for δ=0.09. (a) $\lambda_{Q^*}(\delta)$, (b) $P_T(\delta)$. At $\delta \approx 0.2$ a sharp drop in P_T indicates that control with the policy approximated for low noise levels is unable to stabilize $\mathbf{x}_{4\pi}$.

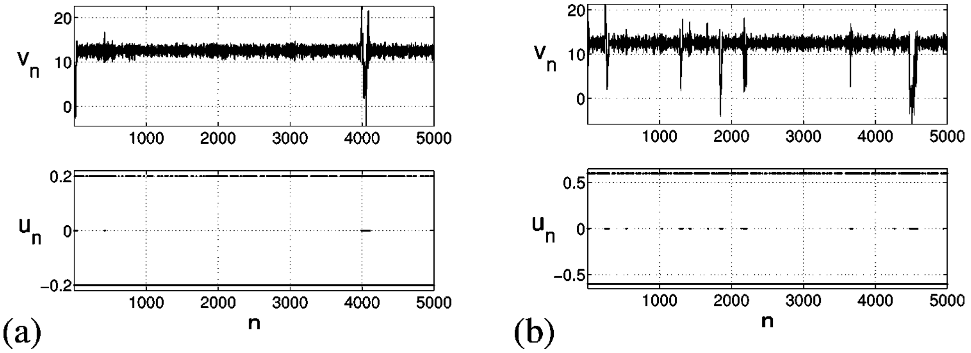


FIG. 13. Controlled dynamics when using the policy $Q^*_{4\pi}$ learned for δ=0.09 for higher noise levels. (a) $\delta = 0.2, u_{max} = 0.2$, and (b) $\delta = 0.3, u_{max} = 0.6$.

terminated if the condition $||\mathbf{x}_g - \mathbf{x}_n|| < 1$ was met for five consecutive iterations. Learning was performed over 2000 episodes and ϵ was decreased from initially one, over 1500 episodes, to zero. During the last 500 episodes ϵ was set to zero. Figure 7 shows the total number of iterations as a function of the episodes for establishing optimal control policies $Q^*_{\mathbf{x}_0}$, $Q^*_{\mathbf{x}_{2\pi}}$, and $Q^*_{\mathbf{x}_{4\pi}}$ for stabilizing the goals $\mathbf{x}_0$, $\mathbf{x}_{2\pi}$, and $\mathbf{x}_{4\pi}$, respectively. A decrease in slope points to convergence of the corresponding policy and we observe convergence in all three cases. The slope of the curves during the last episodes is an estimate for the quality λ^* of the corresponding policy. From the slopes during the last 200 episodes we get $\lambda_0^* \approx 305$, $\lambda_{2\pi}^* \approx 335$, and $\lambda_{4\pi}^* \approx 568$. $\lambda_{2\pi}^* \approx 335$, for example, means that on average, by using the off-line approximated policy $Q^*_{\mathbf{x}_{2\pi}}$ the dynamics of the controlled rotor will be stabilized at the goal $\mathbf{x}_{2\pi}$ after 335 iterations.

In Fig. 8 we show the use of these global policies for a sample control problem. Over 80 000 iterations, the used control policy was switched every 20 000 iterations as described in the caption of the figure. We see that in all cases, control is established almost instantaneously and not lost during the interval of control.

In Table I the previously mentioned performance criteria are summarized. We can see a considerable improvement in the performance of the off-line approximated policies over the on-line approximated policies. Note that for 4π as goal, $\lambda_{Q*} = 579$ is larger than $\lambda_Q = 516$ since we average λ only over terminating episodes. Comparing λ_Q^l and λ_{Q*}^l we clearly see the improvement offered by the off-line approximated policy.

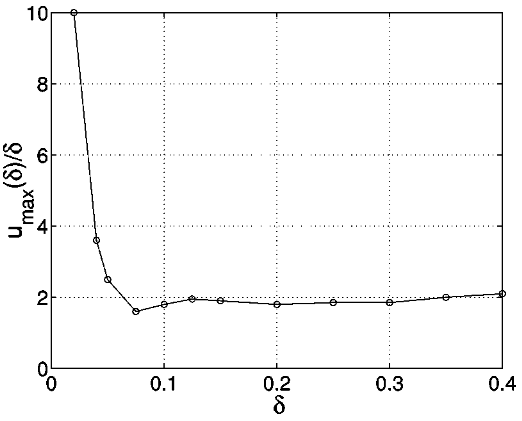


FIG. 14. Learning control in the presence of large noise. The graph shows $u_{max}(\delta)/\delta$, where $u_{max}(\delta)$ denotes the minimum perturbation that achieves on-line control for a given δ.

The global character of the approximated policies becomes clearly apparent by looking at the probability density of the dynamics controlled with the policies Q^*. Figure 9 shows the probability density, approximated as discussed in Sec. III, of the dynamics controlled with the policy $Q^*_{\mathbf{x}_{4\pi}}$. We see that the probability of states not in the neighborhood of $\mathbf{x}_{4\pi}$ is negligible.

The approximated global policy $Q^*_{\mathbf{x}_{4\pi}}$ can be visualized as in Fig. 10. The dark (light) areas correspond to regions where the control $u_n = -0.2$ $(u_n = 0.2)$ is associated with a maximal state-action value. The white areas correspond to regions where no control, $u_n = 0$, is applied. The effect of this policy is that points in the neighborhood of the state $\mathbf{x}_{4\pi}$ are trapped in this region while points in the neighborhood of other attracting states are more probable to leave their neighborhood. This can be seen from Fig. 11, where 1000 random initial conditions from a small rectangle around $\mathbf{x}_{4\pi}$ [Figs. 11(a,b)] and $\mathbf{x}_0$ [Figs. 11(c,d)] were iterated. In Figs. 11(a,c) the states for uncontrolled dynamics are shown after 10 iterations, while Figs. 11(b,d) show the states after 10 iterations for the controlled dynamics. The parameters are as in the last subsection. Comparing Fig. 11(a) with Fig. 11(b), we see that application of the control signals prevents dynamics from leaving the neighborhood of $\mathbf{x}_{4\pi}$. As seen from Fig. 10, control in this region is equivalent to a slight decrease of the forcing. In the neighborhood of other attracting states the controller learned to slightly increase the forcing, which in turn increases the probability of leaving undesired neighborhoods.

C. Control for higher noise levels

1. Using a policy learned for low noise

We tested the control performance of the policy $Q^*_{\mathbf{x}_{4\pi}}$ approximated for $\delta = 0.09$ as described in Sec. V B for larger values of δ. As in the preceding subsection, we measure λ by averaging the number of iterations per episode over 2000 terminating episodes, but here we terminate an episode if $\|\mathbf{x}_g - \mathbf{x}_n\| < 2$ for 200 consecutive episodes.

Figure 12(a) shows $\lambda_{Q^*_{\mathbf{x}_{4\pi}}}(\delta)$ and Fig. 12(b) $P_T(\delta)$ for a range of noise levels, with λ and P_T defined as before and maximal perturbation $u_{max} = 0.2$. The policy $Q^*_{\mathbf{x}_{4\pi}}$ approxi-

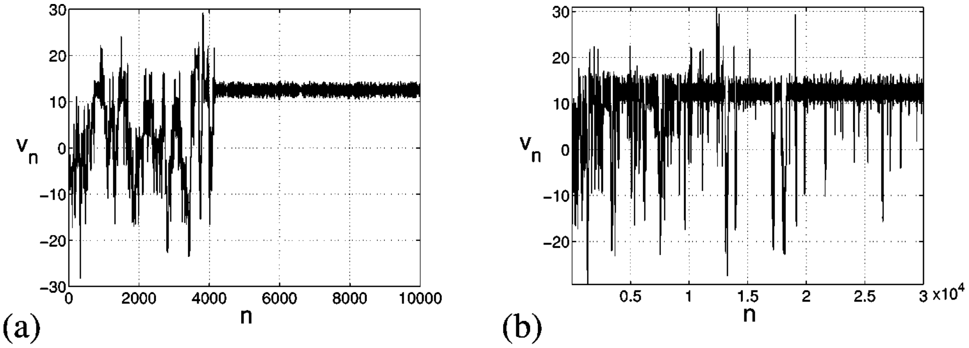


FIG. 15. On-line control of rotor for (a) $\delta=0.2$, $u_{max}=0.4$, and (b) $\delta=0.4$, $u_{max}=0.8$.

mated at $\delta=0.09$ could successfully stabilize $\mathbf{x}_{4\pi}$ over a wide range of noise levels up to $\delta\approx0.2$, at which point P_T begins to drop sharply below 100%. Figure 13(a) shows v_n for the controlled system at $\delta=0.2$ and the applied control signal u_n. For larger noise, the noise often kicks the system out of the desired region and the maximal perturbation $u_{max}=0.2$ is not sufficient. However, allowing a larger maximal perturbation u_{max} and for the same policy, control can be established for $\delta=0.3$ as can be seen from Fig. 13(b), where $u_{max}=0.6$.

2. Learning control in the presence of large noise

Next we investigated if a control policy stabilizing $\mathbf{x}_{4\pi}$ can be approximated in the presence of large noise. To this end we determined the smallest value $u_{max}(\delta)$ of u_{max} for which, for a given δ, $\mathbf{x}_{4\pi}$ can be stabilized through on-line control. Figure 14 shows the ratio $u_{max}(\delta)/\delta$ as function of δ. This ratio approaches a limiting value of approximately 2 and can be used to choose the control parameter for different noise levels. For small δ the ratio is getting larger, since comparably larger perturbations are needed to enter the desired state-space region due to longer transients. In Fig. 15 we show the on-line control of the rotor for (a) $\delta=0.2$, $u_{max}=0.4$, and (b) $\delta=0.4$, $u_{max}=0.8$. It is possible to stabilize the system under the influence of large noise although much larger system perturbations are needed in this case.

D. Destabilization of attracting states

The algorithm, as presented, can be generalized to achieve more general control goals. An example would be the stabilization of fixed points of period 3 or a destabilization of attracting states. To describe a suitable measure for detection if the goal was achieved, we introduce the quantity

$$d_k(\mathbf{x}_n)=\frac{\|\mathbf{x}_n-\mathbf{x}_{n-k}\|}{\exp\left(\sum_{i=1}^{p-1}\ln\left\|\mathbf{x}_n-\mathbf{x}_{n-1}\right\|\right)},\quad k>1,$$

which is minimal only for fixed points of period k. To stabilize a fixed point of period 3 we applied the reward $r_{n+1}=1$ if $d_k(\mathbf{x}_n)<0.02$ and $r_{n+1}=-0.5$ otherwise. This choice of reward rewards a stabilization of any period 3 fixed point. Since the rotor has several of these, if a particular period 3 orbit is desired, we must give positive rewards only if the desired orbit was stabilized. We successfully used this approach to stabilize the period 3 fixed point in the neighborhood of $\mathbf{x}_{2\pi}$ but needed a set of reference vectors of size $|W|=200$.

The other control goal we investigated is related to the problem of maintenance of chaos [38]. The goal here is to prevent the system from a transition of the chaotic state into a stable attracting state. For the rotor a ''maintenance'' of the chaotic state means that the system should be kept from entering the neighborhoods of the stable period 1 or period 3 fixed points. This could certainly be done by adding large noise to the system. Instead, here we will demonstrate that the proposed algorithm is able to ''maintain'' chaos in the rotor through small perturbations to the external forcing by a suitable formulation of the reward function. We chose the reward function $r_{n+1}=1$ if $d_1>2.5$ and $d_3>5$ and $r_{n+1}=-0.5$ otherwise. This punishes all actions leading to states close to period 1 or 3 fixed points. We approximated a maintenance control policy Q^*_{mtn} off-line. The noise was set to $\delta=0.02$, the control set was $U=\{0,0.2,-0.2\}$, and we used the set W of size $|W|=100$ mentioned above. Figure 16(a) shows the negative probability density $p(v,\theta)$ in the rectangle $[-5\pi,5\pi]\times[0,2\pi]$. We see that the regions in the neighborhoods of the attracting states have negligible probability of being visited.

VI. CONCLUSIONS

In this paper we have demonstrated the control of a ''simple'' complex system, the kicked mechanical rotor, un-

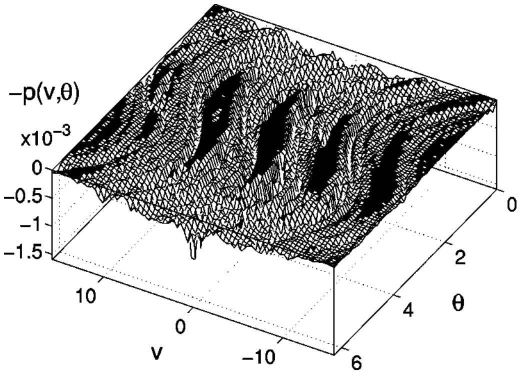


FIG. 16. Negative probability density $-p(v,\theta)$ in the rectangle $[-5\pi,5\pi]\times[0,2\pi]$. It is $\delta=0.02$ and $U=\{0,0.2,-0.2\}$ with perturbations to f_0 chosen from U according to the policy Q^*_{mtn}. The black areas belong to densities with $p<2\times10^{-5}$.

der the influence of noise. Our control method is based on reinforcement learning and establishes an optimal control policy in terms of an optimal state-action value function depending on discretized states and control actions. The approximated control policy acts globally and establishes stabilization of a desired state quickly from any initial condition even under the influence of large noise. It is interesting to note that control could be established even in regions in which the system's dynamics are characterized as stochastic. The presented approach neither assumes knowledge nor requires estimation of an analytical description of the system dynamics. All information required to find an optimal policy is obtained through interaction with the environment. This does, however, not mean that we suggest not to use information about the system when this is available. Reinforcement learning is flexible and allows to include additional constraints in the algorithm, as we did when selecting specific fixed points or period 3 orbits.

In certain applications, such as medical applications, it might be impossible to interact with the system to be controlled. If a good model is available then the policy can be established through an interaction with the model dynamics. If no model is available then it is necessary to combine model approximation techniques with reinforcement learning, i.e. a model of the system is approximated together with the control policy [26]. Of particular interest for future research is the combination of vector quantization and model approximation techniques as suggested in [33,36] with the approximation of the control policy.

The method requires a discrete state representation, which was computed in advance from sample data with the neural-gas algorithm. Many different vector quantization techniques exist in the literature and can be used instead. The question of how to obtain an optimal discrete representation must still be investigated. Future research will attempt to combine the approximation of the finite representation with the policy approximation that can lead to representations suited for the particular control problem. Furthermore, the application to systems with a reconstructed phase space, such as a space of embedding vectors [39], must be investigated to make the approach more suitable for realistic applications.

The presented approach has not yet been applied to systems with riddled basins or more complexly structured phase spaces, which often occur in applications, and it has to be investigated how to deal with systems evolving in an infinite-dimensional phase space. A combination with a dimension reduction technique such as a Karhunen-Loeve decomposition might allow us to reduce the system to a finite-dimensional system in certain situations. Furthermore, the control of more complicated dynamics such as quasiperiodic orbits (invariant circles) or chaotic saddles still has to be investigated. Our results on ''maintenance of chaos'' provide a first step in this direction.

Current research in the reinforcement learning literature is focusing on the development of algorithms suitable in continuous phase and action spaces and on the application of reinforcement learning to high-dimensional systems. Successes in the field of reinforcement learning could be applied to the control of complex systems. On the other hand, the control of complex systems provides an interesting set of problems for the testing of new reinforcement learning control algorithms.

Our results in this paper suggest that the proposed method might lead to a variety of interesting applications in the field of complex dynamical systems. In particular, the combination with other existing methods could lead to more flexible and versatile control techniques. Possible implications for neural information processing have to be investigated and will be a topic of future research. One of the goals here will be the development of an information processing or pattern retrieval device, which is based on the principle of our control strategy.

ACKNOWLEDGMENT

We wish to thank U. Feudel for bringing this problem to our attention and pointing out to us a very interesting area of research.

[1] L. Poon and C. Grebogi, Phys. Rev. Lett. **75**, 4023 (1995).

[2] S. Kraut, U. Feudel, and C. Grebogi, Phys. Rev. E **59**, 5253 (1999).

[3] F. T. Arecchi, R. Meucci, G. Puccioni, and J. Tredicce, Phys. Rev. Lett. **49**, 1217 (1982).

[4] F. Prengel, A. Wacker, and E. Schöll, Phys. Rev. B **50**, 1705 (1994).

[5] M. Brambilla *et al.*, Phys. Rev. A **43**, 5090 (1991).

[6] M. Brambilla *et al.*, Phys. Rev. A **43**, 5114 (1991).

[7] P. Marmillot, M. Kaufmann, and J.-F. Hervagault, J. Chem. Phys. **95**, 1206 (1991).

[8] J. Foss, F. Moss, and J. Milton, Phys. Rev. E **55**, 4536 (1997).

[9] J. Foss, A. Longtin, B. Mensour, and J. Milton, Phys. Rev. Lett. **76**, 708 (1996).

[10] U. Feudel, C. Grebogi, B. Hunt, and J. Yorke, Phys. Rev. E **54**, 71 (1996).

[11] U. Feudel and C. Grebogi, Chaos **7**, 597 (1997).

[12] G. Casati, Chaos **6**, 391 (1996).

[13] E. Ott, C. Grebogi, and J. Yorke, Phys. Rev. Lett. **64**, 1196 (1990).

[14] P. Montague, P. Dayan, C. Person, and T. Sejnowski, Nature (London) **377**, 725 (1995).

[15] P. Montague, P. Dayan, C. Person, and T. Sejnowski, J. Neurosci. **16**, 1936 (1996).

[16] R. Der and M. Herrmann, *Proceedings of the 1994 IEEE International Conference on Neural Networks* (IEEE, New York, 1994), Vol. 4, p. 2472.

[17] S. Gadaleta and G. Dangelmayr, Chaos **9**, 775 (1999).

[18] S. Gadaleta and G. Dangelmayr, in *Proceedings of Second International Conference on Control of Oscillations and Chaos*, edited by F. L. Chernousko and A. L. Fradkov (IEEE, St. Petersburg, Russia, 2000), Vol. 1, pp. 109–112.

[19] C. Lin and C. Jou, IEEE Trans. Neural Netw. **10**, 846 (1999).

[20] B. V. Chirikov, Phys. Rep. **52**, 263 (1979).

[21] A. Lichtenberg and M. Lieberman, *Regular and Chaotic Dynamics* (Springer-Verlag, New York, 1992).

[22] P. Ashwin and J. Terry, Physica D **142**, 87 (2000).

[23] D. Bertsekas, *Dynamic Programming and Optimal Control* (Athena Scientific, Belmont, MA, 1995), Vols. 1 and 2.

[24] R. Bellman, *Dynamic Programming* (Princeton University Press, Princeton, NJ, 1957).

[25] D. Bertsekas, *Nonlinear Programming* (Athena Scientific, Belmont, MA, 1999).

[26] R. Sutton and A. Barto, *Reinforcement Learning: An Introduction* (MIT Press, Cambridge, MA, 1998).

[27] D. Bertsekas and J. Tsitsiklis, *Neuro-Dynamic Programming* (Athena Scientific, Belmont, MA, 1996).

[28] S. Singh, Ph. D. thesis, Department of Computer Science, University of Massachusetts, 1994 (unpublished).

[29] R. Sutton, Mach. Learn. **3**, 9 (1988).

[30] H. Robbins and S. Monro, Ann. Math. Stat. **22**, 400 (1951).

[31] C. Watkins, Ph. D. thesis, University of Cambridge, England, 1989 (unpublished).

[32] L. Kaelbling, M. Littman, and A. Moore, J. Artif. Intell. Res. **4**, 237 (1996).

[33] T. Martinetz, S. Berkovich, and K. Schulten, IEEE Trans. Neural Netw. **4**, 558 (1993).

[34] T. Martinetz and K. Schulten, Neural Networks **7**, 507 (1994).

[35] B. Fritzke, Neural Networks **7**, 1441 (1994).

[36] J. Bruske and G. Sommer, Neural Comput. **7**, 845 (1995).

[37] S. Gadaleta, Ph. D. dissertation, Department of Mathematics, Colorado State University, 2000 (unpublished).

[38] V. In, S. Mahan, W. Ditto, and M. Spano, Phys. Rev. Lett. **74**, 4420 (1995).

[39] H. Abarbanel, R. Brown, J. Sidorowich, and L. Tsimring, Rev. Mod. Phys. **65**, 1331 (1993).