Control of a Noisy Mechanical Rotor

SABINO GADALETA and GERHARD DANGELMAYR
Department of Mathematics
Colorado State University
Engineering E121,
Ft. Collins, CO 80521
USA
{sabino, gerhard}@math.colostate.edu

Abstract: We investigate the control of the dynamics of a periodically kicked mechanical rotor in the presence of noise. It was recently shown that the system dynamics 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 considerable noise level. As control strategy we use a recently developed algorithm for the control of chaotic systems which is based on reinforcement learning. This method finds a global optimal control policy directing the system from any initial state towards the desired state in a minimum number of iterations and stabilizes the system once a neighborhood of the desired state is reached. The algorithm does not use any information about governing equations.

Key-Words: Multistability, mechanical rotor, control, chaos, reinforcement learning

1 Introduction

The long-term behaviour of nonlinear dynamical systems is generally classified as either stationary, periodic, quasi-periodic or chaotic. These types of behaviours and their control are well studied and understood if the available states are well separated and their dimension rather low. In recent years the attention has shifted to systems exhibiting more complex behaviours 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 interrelated parts and possess competing attracting sets. Minor perturbations induced for example, by noise, can cause the system to 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 behaviour is called *multistability* and was first studied experimentally in [3] and since then was observed in a variety of systems from different areas [4, 5, 6]. Adding noise to a multistable system will generate complex behaviour and induce competition between the attractiveness towards regular motion in the neighborhood of an attracting state and the jumping between basins of attraction induced by noise [2]. The dynamics is then characterized by a large number of periodic attractors "embedded" in a sea of transient chaos [1]. This phenonmenon is believed to play a fundamental role in neural information processing [6].

The simplest prototype of a complex multistable system is provided by the model equations of a periodically kicked mechanical rotor which was introduced and studied in this context in [7, 8, 2] and also plays a fundamental role in the study of quantum chaos [9]. Until now control was achieved for low noise levels through a simple feedback mechanism [8] which, however, 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 [8] 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 [10]. In the presence of noise however, this estimation can become very difficult.

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 [11] who applied it to the logisitic map. In [12] we generalized the method and applied it to the control of several discrete and continous low-dimensional chaotic and hyperchaotic systems and recently to coupled logistic map lattices [13].

In this paper we demonstrate for the case of a periodically kicked rotor that the method developed in [12] for chaotic systems is also well suited for the control of complex nonchaotic multistable systems in the presence of significant noise levels.

2 The noisy uncontrolled rotor

The impulsively forced rotor describes a particle rotating on a ring with phase angle θ and angular velocity v subjected to periodic "kicks" or perturbations of size f_0. With damping c and under the influence of noise this *kicked mechanical rotor* can be expressed by the map [2]

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

where δ_θ and δ_v are uniformly and independently distributed random variables bounded by $\sqrt{\delta_\theta^2 + \delta_v^2} \le \delta$. In the following we will use $\mathbf{s}_n = (\theta_n, v_n) \in S = T^1 \times \mathbb{R}$ to denote the state of the rotor at the n-th iteration. The map (1) was extensively studied for $\delta = 0$ in [7] and for $\delta \neq 0$ in [2]. In the Hamiltonian limit, $c = 0, \delta = 0$, it results in the Chirikov standard map and the state space consists of a chaotic sea interspersed with periodic islands and the number of regular periodic orbits is believed to be infinite. For very strong damping ($c \approx 1$) and $\delta = 0$ one obtains the one-dimensional circle map with a zero phase shift, which possesses only one attractor in large regions of the phase space.

By adding a small amount of dissipation to the undamped system, stable periodic orbits turn into sinks and the chaotic motion is replaced by long chaotic transients which occur before the trajectory eventually settles down into one of the sinks [7]. Instead of the very large number of attractors, now, a much smaller number is observed for a given f_0. The number of attractors can be made arbitrarily large by reducing the dissipation. For example for $c = 0.02$, 111 attractors were found numerically [7]. The state in which the system eventually settles down then depends crucially on the initial conditions and the kicked rotor with small dissipation serves therefore as an example of a multistable system.

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

Throughout this work we set $f_0 = 3.5$ and $c = 0.02$. Typical dynamics of this system with these parameters are shown in Figure 1 for a) $\delta = 0.09$ and b) $\delta = 0.3$. The system jumps between different stable states which can be identified as $(\theta, v) = (\pi, \pm 2k\pi)$, $k = 0, 1, 2, \cdots$. 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 then a few iterations.

3 The control algorithm

We control the noisy rotor through small discrete state-dependent perturbations of the applied forcing f_0. The dynamics of the controlled system can

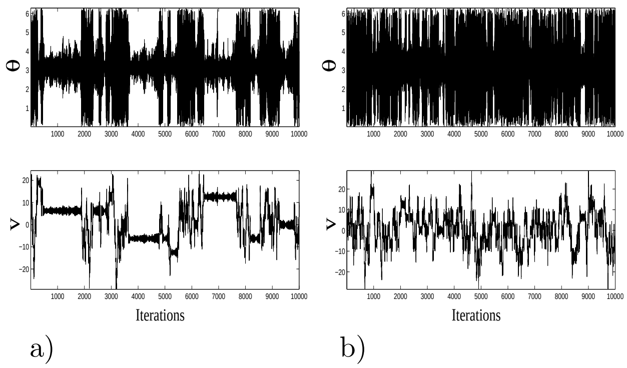

Figure 1: Dynamics of the noisy kicked rotor with $f_0 = 3.5$, $c = 0.02$ for a) $\delta = 0.09$ and b) $\delta = 0.3$.

then be written in the form

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

where $u_n = u(\mathbf{s}_n)$ represents the state dependent control perturbation applied to the external forcing f_0 at the n-th iteration step. To establish control, u_n is chosen from a discrete set U of allowed controls according to a certain control policy $\Pi_\epsilon(\mathbf{s}, u)$ which associates with every state $\mathbf{s}$ a control u. ϵ denotes the probability by which the, according to the policy, best control is chosen. The main task is to find a control policy Π such that a prescribed control goal is achieved in an optimal way. We use reinforcement learning to establish such an optimal control policy as described in [12]. The method requires a discrete representation W of the state space S in terms of a finite set of reference vectors $\mathbf{w} \in W$. To obtain W, in principle any vector quantization technique can be used. We applied the Neural-Gas algorithm [14] with $N = 100$ reference vectors to a set of 50,000 datapoints $\mathbf{s}$ obtained from simulations with $\delta = 0.3$. The outcome of the vector quantization is a set of reference vectors $\mathbf{w} \in W$ which approximates the probability distribution of the presented dataset S in an optimal way [14] and partitions S into so-called Voronoi cells whose centers $\mathbf{w}$ form the necessary discrete state approximation. Each state $\mathbf{s} \in S$ is projected to exactly one $\mathbf{w}(\mathbf{s}) \in W$, where $\mathbf{w}(\mathbf{s})$ is the closest reference vector according to some (usually euclidean) norm:

$$\mathbf{w}(\mathbf{s}) = \arg\min_{\mathbf{w} \in W} ||\mathbf{s} - \mathbf{w}||. \tag{3}$$

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

Given an optimal state-action value function $Q^*(\mathbf{w}, u)$, the optimal control u^* associated to a state $\mathbf{w}$ is chosen according to the rule

$$u^* = \arg\max_{u \in U(\mathbf{w})} Q^*(\mathbf{w}, u). \tag{4}$$

The main task is now reduced to the problem of finding the optimal state-action value function Q^*. If no explicit knowledge about system dynamics is assumed, temporal-difference learning, in particular Watkins' Q-learning [15], which results in the update rule

$$\begin{aligned} \Delta Q_n(\mathbf{w}_n, u_n) = \beta_n \big[& r_{n+1} \\ & + \gamma \max_{u \in U(\mathbf{w}_{n+1})} Q(\mathbf{w}_{n+1}, u) - Q(\mathbf{w}_n, u_n) \big], \end{aligned} \tag{5}$$

where r_{n+1} represents an immediate reward received for performing the control u_n in state $\mathbf{w}_n$, is proven to converge to Q^*. In this work we punish unsuccessful actions by setting $r_{n+1} = -1$ whenever at the $(n+1)$-th iteration the goal was not achieved. Otherwise we set $Q(\mathbf{w}_n, u_n) = 0$, which turned out to improve execution time over setting $r_{n+1} = 0$.

Q-learning can be proven to converge towards Q^*, if the whole state-space is explored and β_n is slowly frozen to zero [16]. In real applications, due to time constraints, it will rarely be possible to satisfy this requirement and one often sets $\beta_n = \beta$. 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 which initially chooses actions stochastically and is slowly frozen into a deterministic policy. This can for example be achieved through ϵ-greedy policies Π_ϵ [16] where an action which is different from the one with maximal estimated action value is chosen with probability ϵ. We call a policy greedy or deterministic if exclusively the best action is chosen ($\epsilon \equiv 0$). An optimal state-action value function Q^* associates with each state $\mathbf{s}$ 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. The goal will be to stabilize a desired state.

4 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 $\mathbf{s}_g = (\pi, g)$, with $g = 0,\ 2\pi$, and 4π (We were also able to stabilize $g = 6\pi$ and $g = 8\pi$). 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 $\delta = 0.09$. As stated in [2], this value of δ marks the transition from a deterministic to a stochastic system. In previous works [1, 8] control could be established only up to very low noise levels ($\delta = 0.01$ in [8]). The parameters of the Q-learning update rule were set to the constant values $\beta = 0.5$ and $\gamma = 1$ but their particular choice does not infuence results much. To measure the quality of an approximated policy defined through the values Q we introduce the quantity λ_Q which measures the average number of iterations per episode, where an episode is an iteration of the system 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. We terminate an episode if the condition $||\mathbf{s}_g - \mathbf{s}_n|| < 1$ is met for 200 consecutive iterations.

4.1 Online Control

Online 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 values Q, are applied to the forcing function and the control algorithm updates the Q-function from immediate rewards r_n, where $r_n = -1$ if $\|\mathbf{s}_g - \mathbf{s}_n\| > 1$ and zero otherwise. Eventually the controller will find a successful control strategy and keep the system stabilized in the desired state. Figure 2 shows online control of the noisy rotor with $\delta = 0.09$. Initially the control goal was to stabilize the system in the state $\mathbf{s}_0$. After 30,000 (60,000) iterations, Q was reset to zero and the control goal changed to the stabilization of $\mathbf{s}_{2\pi}$ ($\mathbf{s}_{4\pi}$). 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.

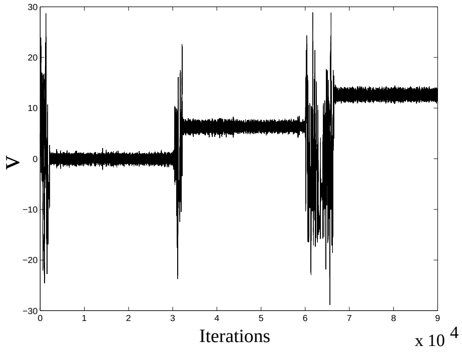

Figure 2: Online control of the rotor dynamics with $\delta = 0.09$ and $U = \{0,\ 0.2,\ -0.2\}$. Initially the control goal is $\mathbf{s}_0$. After 30,000 iterations the control goal is switched to $\mathbf{s}_{2\pi}$ and after 60,000 iterations to $\mathbf{s}_{4\pi}$.

In Table 1 we summarize the performance of the approximated policies for stabilizing the different goals $\mathbf{s}_g$. λ was averaged over 2,000 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 which did not terminate before a total of 2,000 terminating episodes occured. An episode was counted as unterminated if it did not terminate after 10,000 iterations. $P_T(Q)$ is then $100 \cdot 2,000/(2,000 + \lambda_{nt})$. $P_T(u)$ denotes the pobability of satisfying the termination criterion without control. A good policy Q should satisfy $P_T(Q) >> P_T(u)$ and $\lambda_Q << \lambda_u$. These performance measures are shown in Table 1 for online

Goal	λ_u	$P_T(u)$	λ_Q	$P_T(Q)$	λ_{Q^*}	$P_T(Q^*)$
0	524	27%	590	46%	398	98%
2π	582	22 %	557	48%	417	99%
4π	1,700	10%	516	56%	579	98%

Table 1: Comparison of online (Q) and offline (Q^*) controlled systems with the uncontrolled system (u).

(Q) and offline (Q^*) (see next subsection) approximated policies for the three goals. Use of the online approximated policy improves performance considerably over the uncontrolled system, but the policy has low termination probability. To approximate a policy with higher termination probability offline control can be used.

4.2 Offline Control

To better satisfy the requirements of convergence to an optimal policy as stated in Section 3, offline control has to be performed. In offline control, learning is performed ϵ-greedy in many episodes where each episode is started in a new random initial condition. For the learning process, an episode was terminated if the condition $||\mathbf{s}_g - \mathbf{s}_n|| < 1$ was met for 5 consecutive iterations. See Figure 3 and its caption for details and results.

In Figure 4 we show the use of these global policies for a sample control problem. Control is established almost instantaneously and not lost during the interval of control by using the offline approximated policies. In Table 1 the previously mentioned perfomance criteria are summarized and we see considerable improvement of the offline over the online approximated policies. (Note that $\lambda_{Q^*} = 579$ is larger then $\lambda_Q = 516$ since we average λ only over terminating episodes.)

The control for higher noise levels will be discussed elsewhere. Initial results suggest that control up to $\delta = 0.4$ is possible.

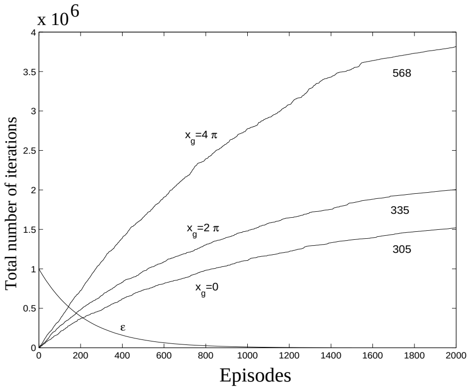

Figure 3: Offline control of the rotor dynamics with $\delta = 0.09$ and $U = \{0,\ 0.2,\ -0.2\}$. The curves labeled $x_g = 0, x_g = 2\pi$ and $x_g = 4\pi$ represent the total number of iterations during the learning process for control of the goal $\mathbf{s}_{x_g}$. ϵ is slowly frozen from initially one to zero during learning as shown in the graph labeled ϵ (rescaled vertical units). 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.

5 Conclusion

In this paper we have demonstrated the control of a "simple" complex system, the kicked mechanical rotor, under 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 considerable noise. A detailed investigation of control under the influence of higher noise levels will be presented elsewhere. Initial results suggest that control can be established even in regions in which the system's dynamics are characterized as stochastic. The presented approach does neither assume knowledge nor require estimation of an analytic description of the system dynamics.

Our results 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

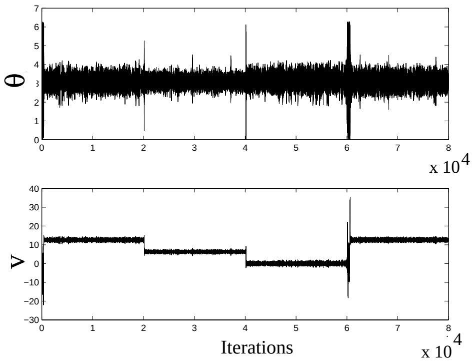

Figure 4: Offline control of the rotor dynamics. The control policy is reset every 20,000 iterations from initially $Q^*_{\mathbf{s}_{4\pi}}$ to $Q^*_{\mathbf{s}_{2\pi}}$, $Q^*_{\mathbf{s}_0}$ and back to the initial policy $Q^*_{\mathbf{s}_{4\pi}}$.

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.

References

[1] L. Poon and C. Grebogi. Controlling complexity. *Physical Review Letters*, **75**:4023–4026, 1995.

[2] S. Kraut, U. Feudel, and C. Grebogi. Preference of attractors in multistable systems. *Physical Review E*, **59**:5253–5260, 1999.

[3] F. T. Arecchi, R. Meucci, G. Puccioni, and J. Tredicce. *Physical Review Letters*, **49**:1217, 1982.

[4] P. Marmillot, M. Kaufmann, and J.-F. Hervagault. Multiple steady states and dissipative structures in a circular and linear array of three cells: Numerical and experimental approaches. *The Journal of Chemical Physics*, **95**:1206–1214, 1991.

[5] F. Prengel, A. Wacker, and E. Schöll. Simple model for multistability and domain formation in semiconductor superlattices. *Physical Review B*, **50**:1705–1712, 1994.

[6] J. Foss, F. Moss, and J. Milton. Noise, multistability, and delayed recurrent loops. *Physical Review E*, **55**:4536–4543, 1997.

[7] U. Feudel, C. Grebogi, B. Hunt, and J. Yorke. Map with more than 100 coexisting low-period periodic attractors. *Physical Review E*, **54**:71–81, 1996.

[8] U. Feudel and C. Grebogi. Multistability and the control of complexity. *Chaos*, **7**:597–604, 1997.

[9] G. Casati. Quantum chaos. *Chaos*, **6**:391–398, 1996.

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

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

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

[13] S. Gadaleta and G. Dangelmayr. Control of 1-D and 2-D coupled map lattices through reinforcement learning. In *Proceedings of Second Int. Conf. "CONTROL OF OSCILLATIONS AND CHAOS" (COC'2000)*, St. Petersburg, Russia, 2000. (To be published).

[14] T. Martinetz and K. Schulten. Topology representing networks. *Neural Networks*, **7**:507–522, 1994.

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

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