SPIE conference on Mathematics and Control in Smart Structures, March 1998

Pattern control in unstable structures

Oliver Guenther, Tad Hogg and Bernardo A. Huberman

Xerox Palo Alto Research Center

3333 Coyote Hill Road, Palo Alto, CA 94304

ABSTRACT

A major aim of smart structures is to reduce the amount of materials used to build them while achieving the same amount of dynamic control. This reduction can however lead to a loss of stiffness and a consequent instability of the structure in the presence of stresses. We have turned this problem around by examining how the use of dynamical instabilities can lead to improved control of given patterns in smart structures. In particular, we studied the possibility of dynamically switching between two configurations of a mechanical system, one in which the desired patterns are stable and one in which they are unstable. By comparing the performance of a distributed control in terms of time to switch and power consumption we established that unstable patterns often switch faster among themselves than stable ones without consuming more power.

Keywords: distributed control, pattern formation, multiagent systems

1. INTRODUCTION

The control of stable dynamical systems is a well studied field with a variety of applications in different areas. More recently control techniques have been extended to the so called smart matter structures, which consist in physical systems with microscopic sensors, computers and actuators embedded into them. This enables them to actively monitor and respond to their environments in precisely controlled ways [1, 2]. More recently, several methods have been developed so as to control unstable structures [3, 4]. A particularly robust approach to controlling such systems resorts to autonomous agents, each of which can deal with a limited aspect of the overall control problem. Individual agents can be associated with each sensor or actuator in the material and used to control distributed systems [5] as well.

Another interesting aspect of the distributed control of unstable systems is the possibility of switching between different patterns which can correspond to either stable or

O.G.: Email: guenther@parc.xerox.com; WWW: http://www.parc.xerox.com/guenther
T.H.: Email: hogg@parc.xerox.com; WWW: http://www.parc.xerox.com/hogg
B.A.H.: Email: huberman@parc.xerox.com; WWW: http://www.parc.xerox.com/spl/groups/dynamics

unstable fixed points. This desirable and challenging goal has applications that range from switching between on/off state of an unstable mechanical structure and pattern controls to changes among a discrete set of different shapes of smart surfaces or airplane wings.

In this paper we study the advantages and disadvantages of using instabilities for pattern control, and present simulation results for a simple smart structure model system. An obvious advantage of controlling an unstable fixed point is its ability to start a self-enforcing movement with only little effort. This is particularly so when compared to stable states which require a finite and positive amount of energy to leave their energy minimum. in order to move towards other fixed points. Our simulations focus on switching between two patterns that represent the same type of fixed point, i.e. they are either stable or unstable. Furthermore we assume that they are symmetric. Moving from one unstable fixed point to another one requires a control that must be able to stay at the desired unstable pattern after reaching it. This is equivalent to the requirement of staying in the initial unstable pattern if movement is not desired yet.

The paper is organized as follows. We first describe a model system that has the characteristic properties of smart structures, and then design a symmetric potential based on a given pattern that can be either stable or unstable. In the following section we describe two different control algorithms that we use to simulate the switching dynamics. In the results section we compare the performance of both pattern control algorithms for unstable and stable patterns. Finally we summarize and discuss our results and give an outline of the possible extensions that can be made to our work.

2. SYSTEM DESCRIPTION

The model that we use in our simulations on pattern control is an expansion of the unstable chain model that we used in previous studies [4, 6]. It consists of n mass points that are connected to their neighbors by springs. As a boundary condition the first and the last mass point are linked to a wall by a spring. We assume that the mass points can only move in vertical direction and no horizontal movement is possible. The configuration when all mass points are lined up with each other and the connections anchored in the walls represents a fixed point of the system that corresponds to a displacement coordinate of $\mathbf{x}=\{\mathrm{x}_1, \mathrm{x}_2, ..., \mathrm{x}_\mathrm{n}\} = \mathbf{0}$. The system dynamics is described by the displacements x_i and the velocities v_i for $i = 1, ..., n$. In addition, an external potential V_i is used for each mass point to introduce two more fixed points that define the patterns between which the switching takes place.

The parameters that describe the dynamical behavior of that system are given by

1. number of mass points: n
2. spring constants: k

3. damping coefficient: g
4. symmetric pattern position: $\pm\hat{x}_i$ for $i = 1, ..., n$
5. potential value at pattern position: a_i for $i = 1, ..., n$

We assume that all mass points have unit mass. The dynamics of the unstable chain is then given by

$$\begin{aligned} \frac{dx_i}{dt} &= v_i \\ \frac{dv_i}{dt} &= k(x_{i-1} - x_i) + k(x_{i+1} - x_i) - gv_i - \frac{dV_i^*}{dx_i} + H_i \end{aligned} \tag{1}$$

for $i = 1, ..., n$ where we use $x_0 = x_{n+1} = 0$ as the boundary condition. The potential V_i^* is the general potential that also compensates existing spring forces in the given pattern configuration. The negative derivative of V_i^* describes the force applied to mass point i by the potential and H_i is the calculated control force which depends on the used control algorithm.

We introduce two symmetric patterns between which the switching is going to take place by the vectors

$$\mathbf{P}_- = \left(-\hat{\mathrm{x}}_1, -\hat{\mathrm{x}}_2, ..., -\hat{\mathrm{x}}_\mathrm{n}\right), \quad \mathbf{P}_+ = \left(\hat{\mathrm{x}}_1, \hat{\mathrm{x}}_2, ..., \hat{\mathrm{x}}_\mathrm{n}\right) \tag{2}$$

and define a potential V, shown in Fig. 1, that has those patterns as either stable or unstable fixed points. We use

$$V_i^* = q_{1,i}x_i^4 + q_{2,i}x_i^2 \quad \rightarrow \quad -\frac{dV_i^*}{dx_i} = -4q_{1,i}x_i^3 - 2q_{2,i}x_i \tag{3}$$

as the form of the general potential for each mass point that also compensates forces created by the springs. The parameters $q_{1,i}$ and $q_{2,i}$ are calculated by means of the pattern positions $\pm\hat{x}_i$ and an absolut value of a_i for the potential V_i at its maximum position for the unstable patterns and its minimum position for the stable patterns. By using $s_i = k \cdot (2x_i - x_{i-1} - x_{i+1})$ as the resulting force on mass point i caused by the springs to its two neighbors, we derive

$$\begin{aligned} q_{1,i}^{unstable} &= \frac{\frac{s_i}{2}\hat{x}_i - a_i}{\hat{x}_i^4} & q_{2,i}^{unstable} &= \frac{2a_i}{\hat{x}_i^2} - \frac{s_i}{2\hat{x}_i} \\ q_{1,i}^{stable} &= \frac{\frac{s_i}{2}\hat{x}_i + a_i}{\hat{x}_i^4} & q_{2,i}^{stable} &= -\frac{2a_i}{\hat{x}_i^2} - \frac{s_i}{2\hat{x}_i} \end{aligned} \tag{4}$$

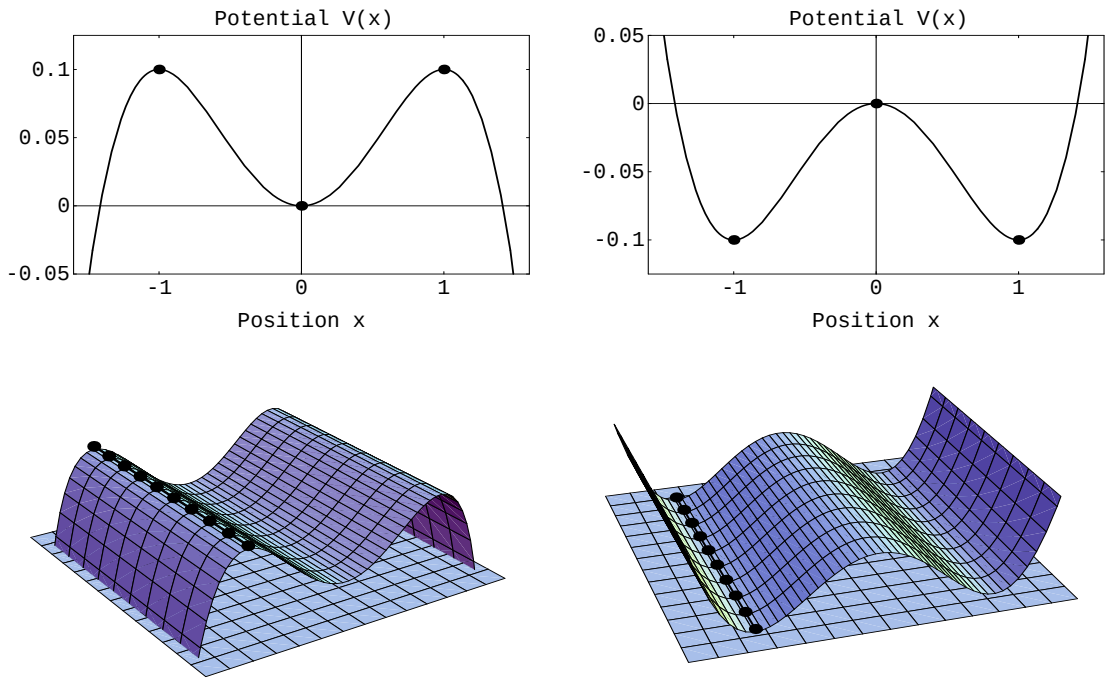

Fig. 1. The top row shows the potential $V(x)$ that we use to define the unstable patterns (left) and the stable patterns (right) for each mass point in our simulations of pattern control. The pictures on the bottom show how the system described by Eq. 1 can be interpreted as a chain of mass points that moves in a potential landscape. The boundary condition of walls at both ends of the chain: $x_0 = x_{n+1} = 0$, that, in respect to the shown initial condition, would modify the potentials $V_1 \to V_1^*$ and $V_n \to V_n^*$ near the walls is not shown here for reasons of a better representation.

We use simple patterns in our simulations where $\hat{x}_i = 1$ and $a_i = 0.1$ for $i = 1, ..., n$. This causes the s_i in Eq. 4 to be zero for $i = 1, ..., n$ except at the positions $i = 1$ and $i = n$, which correspond to the two mass points next to the wall.

For this model system, the pattern control problem consists in determining how hard to push on the various mass points to move them from one pattern to the other one and keeping them there if the corresponding fixed point is unstable. We choose a simple, regular pattern to focus on the effect of the the control algorithm without introducing a complicated dynamics caused by complex spring interactions. We further assume that the system is not disturbed by external forces and that sensors are accurate and supply displacement and velocity information without delay.

3. PATTERN CONTROL

We now introduce and investigate the performance of two different control approaches that are both local and decentralized in nature. They differ in the use of

pre-specified information given by an external model of a desired switching dynamics. The first approach, a commonly used proportional-derivative control (PD-control), makes no use of such a model whereas the second control algorithm uses a simple definition of a desired control dynamics. One can sense intuitively that an instability could be used beneficially for the pattern control example that we introduced with the chain model system. It is in no way clear how a PD-control will make use of the instabilities in the system and how it will compare with other approaches that might be more tailored for the opportunity given by the instability.

We study[1] a chain composed of 10 mass points, all of them having unit mass and connected by springs with a spring constant of value 0.1 and a damping coefficient 0.1. The potential is characterized by $\hat{x_i} = 1$ and $a_i = 0.1$ for $i = 1, ..., n$ and corresponds to the potential shown in Fig. 1. For all the simulations in this paper we start with pattern $\mathbf{P}_-$ as an initial condition and use pattern $\mathbf{P}_+$ as the desired pattern towards which the initial one has to switch as the control goal.

PD-Control

In this well-known control technique [8–10] the control force is calculated by multiplying the distance between the desired and the actual position by a factor b_x and the desired minus the actual velocity by a factor b_v. This leads to a control term that is given by

$$\begin{aligned} H_i &= -b_x \cdot \left(x_i - x_i^{desired} \right) - b_v \cdot \left(v_i - v_i^{desired} \right) \\ &= -b_x \Delta x_i - b_v \Delta v_i \end{aligned} \tag{5}$$

where $b_x \geq 0$, $b_v \geq 0$ and $v_i^{desired} = 0 \ \rightarrow \ \Delta v_i = v_i$. The introduction of a $b_v > 0$ can be interpreted as a virtual viscosity term, added to the system by the control algorithm. Our main reason to study the performance of that algorithm for the given pattern control problem is its known ability to perform well in many other control tasks [8, 10].

In Fig. 2 we compare the switching performance for using stable and unstable patterns. We can see that the power consumption is higher in the unstable pattern case, especially in the initial control phase. The switching time is lower than in the stable pattern case as can be seen in the plot of the average displacement $\overline{x} = \frac{1}{n} \sum_{i=1}^{n} |x_i|$ and no relaxing oscillations into the desired pattern are observed. The observation of a higher maximum power consumption for the unstable pattern control seems to be counterintuitive at first but can be understood by following the first steps of the control dynamics. The

[1] We used a standard ordinary-differential-equation solver for integrating the equations of motion [7].

largest distance Δx_i between actual and desired pattern occurs when the control just starts so the term $b_x \cdot \Delta x_i$ is high and causes a high power request. For switching between unstable patterns, b_x has to be large enough to make sure that the control won't overshoot when approaching the desired pattern. Because of the strong increase in velocity in the beginning based on a high value of $b_x \cdot \Delta x_i$ and on the topology of the downhill potential landscape, this leads to the neccessity of a high virtual viscosity term b_v as well, which in turn requires a high enough b_x near the desired pattern. Those requests constrain the control in a way that can be solved eventually by a set of (b_x, b_v) which then leads to a relatively high energy consumption.

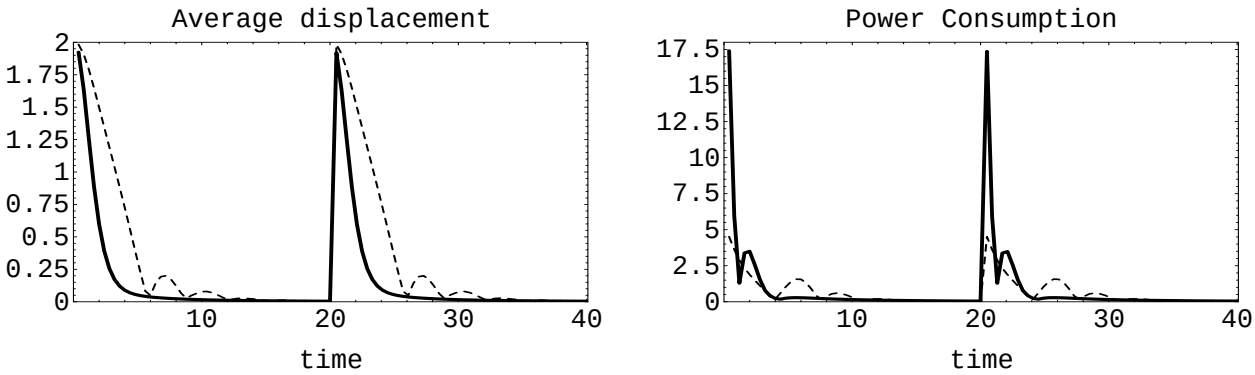

Fig. 2. The plots compare the performance of a PD-control for switching two times between unstable patterns (solid curve) and stable patterns (dashed curve). On the left we see the average displacement and on the right the corresponding power consumption. The parameters used are $b_x = 1.25$ and $b_v = 1.75$ for unstable patterns and $b_x = 0.25$ and $b_v = 0.5$ for stable patterns. The energy consumption is 17.0 energy units for switching between unstable patterns and 13.2 energy units for switching between stable patterns.

This fact shows that a PD-control might not be the appropriate one for taking advantage of the instabilities in the system. To get a better idea of the performance of the PD-control for stable und unstable patterns we investigate how the energy consumption P and the switching time t_s vary for different values of b_x and b_v. This is shown in Fig. 3 where the points (t_s, P) are plotted for the pairs (b_x, b_v) with b_x and b_v varying from 1.5 to 6.5 with step size 0.5. The switching time is defined as the first time during the control when the average displacement $\overline{x} = \frac{1}{n}\sum_{i=1}^{n} |x_i|$ is lower or equal 0.01 and the average velocity $\overline{v} = \frac{1}{n}\sum_{i=1}^{n} |v_i|$ is lower or equal 0.01 for the previous 10 time units.

A closer look at the control performance in terms of switching time and energy consumption shows that the minimum switching time for the unstable patterns is $t_s = 12.4$ $(P = 38.78)$ for $b_x = 6.0$ and $b_v = 4.0$ and the minimum energy consumption is $P = 17.38$ $(t_s = 23.4)$ for $b_x = 1.5$ and $b_v = 2.0$. The equivalent values for the stable patterns are $t_s = 12.4$ $(P = 40.36)$ for $b_x = 6.5$ and $b_v = 4.5$ and a minimum energy consumption of $P = 15.74$ $(t_s = 22.9)$ for $b_x = 1.5$ and $b_v = 4.0$.

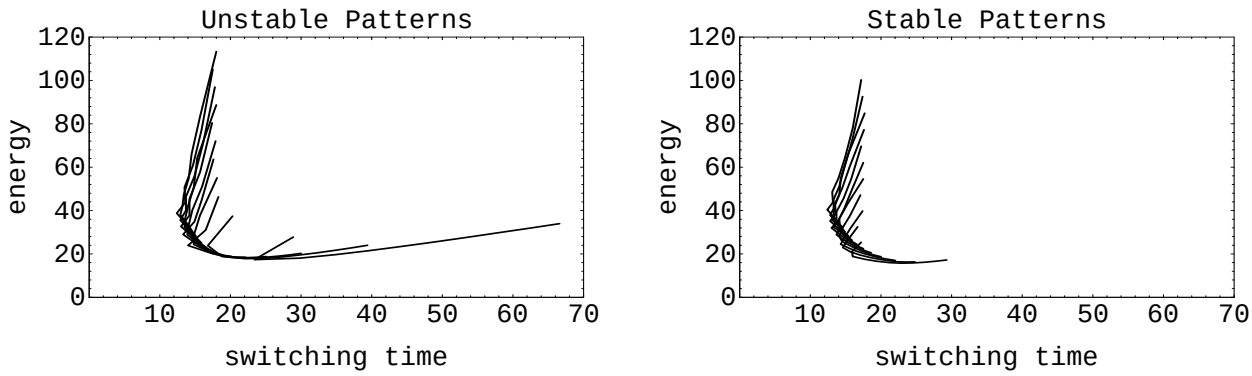

Fig. 3. Comparison of pattern control performance for switching between unstable patterns (left) and stable patterns (right) by using a PD-control. We calculate energy consumption and switching time for different pairs of b_x- and b_v-factors. Connected by lines are those pairs that correspond to a fixed b_x while b_v is varied. Both factors are choosen in the range from 1.5 to 6.5 with a step size 0.5.

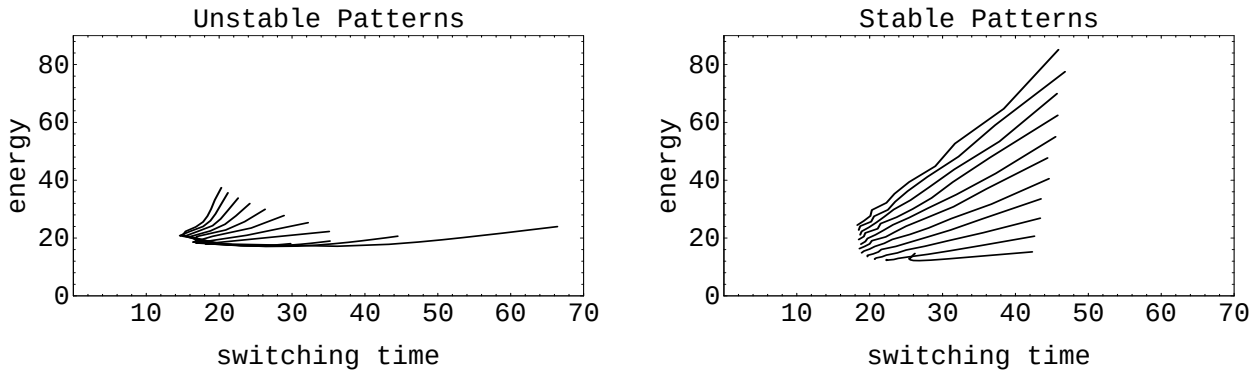

Fig. 4. A similar plot to that of Fig. 3, except for the fact that the factors b_x and b_v are further reduced so as to approach the limits of a successful control. As a result the energy consumption for switching among stable patterns is lower than that required for switching between unstable ones.

Surprisingly we can see that there is not much difference between the minimum values of energy consumption and switching time for switching between unstable patterns and stable patterns. Only if we further decrease the possible values of b_x and b_v, so they will approach the limits of successful pattern control, we observe a slightly changed picture. We vary b_x from 1.0 to 2.0 for the unstable and from 0.2 to 1.2 for the stable case and b_v from 1.5 to 2.5 for the unstable and 0.2 to 1.2 for the stable case with step size 0.1 for all simulation runs. In Fig. 4 we see that the energy consumption can be lower for stable patterns but only for the price of a higher switching time.

A quantitative look shows that the minimum switching time for the unstable patterns is $t_s = 14.6 (P = 20.82)$ for $b_x = 1.9$ and $b_v = 2.1$ and the minimum energy consumption is $P = 17.09$ $(t_s = 26.4)$ for $b_x = 1.3$ and $b_v = 1.7$. The equivalent values for the stable

patterns are $t_s = 18.3$ $(P = 24.48)$ for $b_x = 1.2$ and $b_v = 1.2$and a minimum energy consumption of $P = 12.18$ $(t_s = 26.8)$ for $b_x = 0.2$ and $b_v = 0.6$.

In terms of energy consumption and switching time, we can see in Fig. 3 and Fig. 4 that the PD-control performance is not significantly different for unstable patterns and stable patterns except for a small parameter range. This is a surprising result because the potential and the resulting forces are quite different for stable and unstable patterns. Following our intuition that the instability in a system can be used more effeciently for switching between patterns, we now study the performance of the second pattern control algorithm that uses a pre-defined trajectory in state space for the control.

Trajectory Control

This approach uses known information about the system and puts a strong constraint on the switching dynamics. Ideally the dynamics of each mass point follows a pre-defined trajectory in state space that corresponds to an optimal switching performance. In cases where some knowledge of the underlying potential is available and used in the defintion of the desired trajectory, one would expect a better switching performance for unstable patterns than for stable ones. We illustrate this by means of the potentials in Fig. 1. For unstable patterns and a small damping coefficient one could move from one pattern to the other one by just pushing the mass point slightly towards the right direction. By rolling down the potential hill of the initial pattern at $x = -1$ the mass point stores almost as much kinetic energy as it would need to climb the potential of the desired pattern at $x = +1$. This scenario would consume almost no power and the power needed would only be used to compensate for the friction. Stable patterns on the other hand need to push the mass point over the potential hill at $x = 0$ and either slow it down or let it a perform a damped oscillation in the desired pattern on the other side. The trajectory control approach seems therefore reasonable for cases in which enough information about the system is known and used in the control and the effect of external pertubations and inaccuracies is small.

Another aspect of the trajectory control is its property of using a desired, defined state, characterized by the displacements and velocities of all mass points. This, in a way, makes the system less flexible but offers the advantage of being more predictable, the possibility of calculating a performance measure based on the actual and the desired position in state space and a foundation for building a more complex system on top of highly predictable modules. In an optimal case the trajectory used in this kind of control corresponds to the optimal trajectory concerning a performance measure like power consumption or switching time for example. In all other cases the control has to counteract a non-optimal choice of the pre-specified state space trajectory which will lower the performance.

We define a desired switching dynamics by a simple free fall situation where the control parameter c can be interpreted as a gravitation constant. This choice of a pre-specified trajectory is motivated by a principally similar order of the different dynamical phases that can be grouped into a rolling down a potential hill and rolling up another potential hill. The following equations describe a free fall dynamics with a gravitation constant c

$$\frac{d^2x}{dt^2} = -c \qquad (6)$$
$$v(x) = v_0 - \sqrt{2c \cdot (x - x_0)}$$

Different cases have to be distinguished for the description of a complete switching dynamics to make sure that the argument of the square root has a positive sign. We assume that the free fall starts with an initial velocity $v_0 = 0$ and the corresponding inverse fall that follows the same equation of motion starts with an initial velocity $v_0 > 0$ which causes the resulting trajectory for the complete switching dynamics to be asymmetric. In Fig. 5 we show an example of such a free fall trajectory in state space for a parameter value of c=0.175.

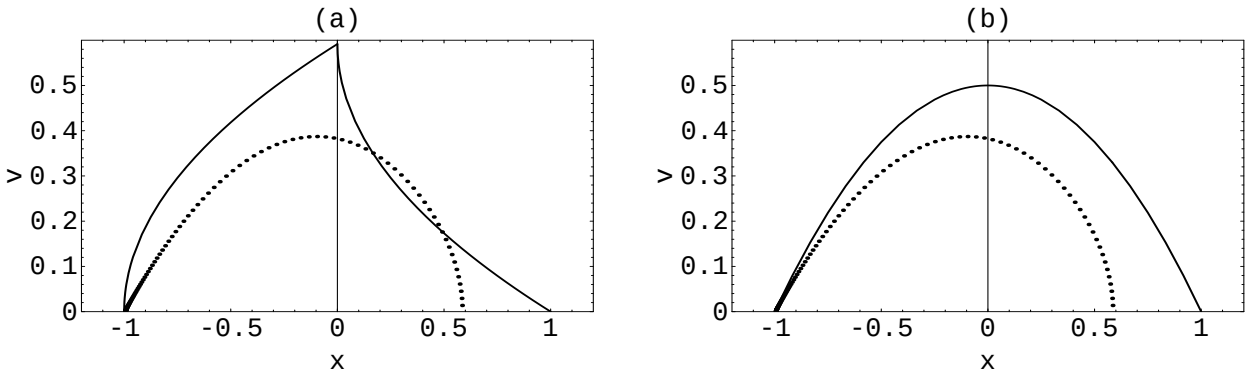

Fig. 5. Comparison between state space trajectories for two different pre-specified trajectories (solid curves) and the uncontrolled dynamics for switching between unstable patterns for one mass point with damping (g=0.1) when no springs are considered (dashed curve). Plot (a) shows an example of a pre-defined trajectory (solid) which is based on free fall dynamics described in Eq. 6. The parameter c for this example is c=0.175. In (b) we see a pre-specified trajectory of quadratic form given by Eq. 7 with a parameter c=0.5 that seems to cover the real dynamics in a better way.

The control acts in such a way that whenever the actual position $(\mathrm{x_i}, \mathrm{v_i})$ of a mass point *i* in state space is different from the desired trajectory $\left(\mathrm{x_i}, \mathrm{v_i^{desired}}(\mathrm{x_i})\right)$, it tries to reduce this differenc by applying a control force $H_i = -b_v \cdot \left(v_i - v_i^{desired}\right)$. To get a starting push when the system is exactly in the initial fixed point pattern and the control

produces no force because of $v_i = 0$ and therefore $H_i = 0$, we apply a small initial force of $H_i^{initial} = b_v \cdot v_i^{desired}\left(x_i - 0.01 \cdot \left(x_i - x_i^{desired}\right)\right)$.

We cannot expect the power consumption to be zero for the unstable pattern control case because the free fall assumption is not true for the potential that we use. It corresponds to a potential landscape with a constant slope. Nevertheless it is a simple assumption with the advantages of predictability and the possibility of building more complex systems with a module architecture and it has approximately the right form. In Fig. 6 we compare the performance of a free fall trajectory control for different values of the parameter c that varies from 0.025 to 1.25 with a step size of 0.025. The b_v factor is kept constant for the unstable and the stable pattern control. We choose a large value of $b_v = 50.0$ to make sure that the control force is large enough so that the switching dynamics follows the pre-specified trajectory within reasonable limits.

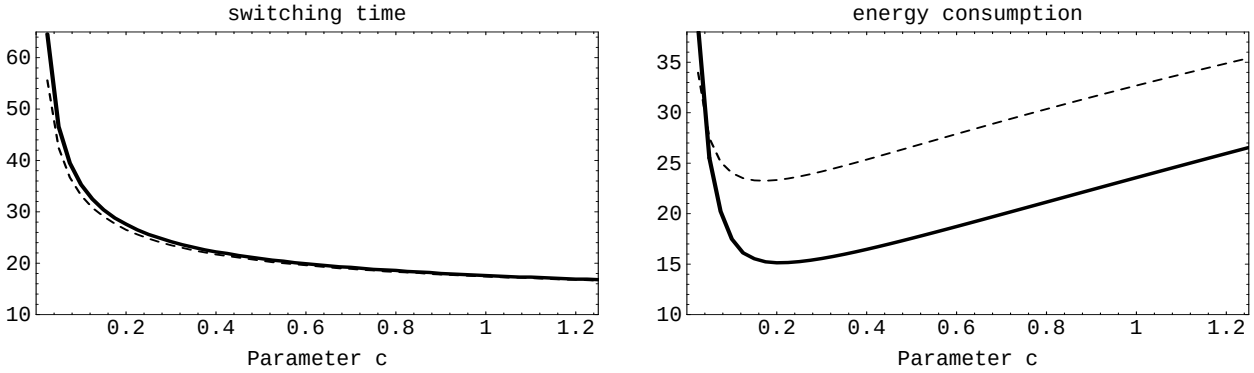

Fig. 6. The performance of a free fall trajectory control for different values of the parameter c in Eq. 6 varying from 0.025 to 1.25 with step size 0.025. On the left we display the switching time and on the right we show the energy consumption. In both plots the solid curve describes a switching between unstable patterns and the dashed curve refers to switching among stable patterns.

Since the control dynamics follows a pre-specified trajectory in state space, we expect the switching time to be the same independent of using unstable or stable patterns. The observation of this expected behavior is a sign that we choose a b_v factor that is high enough to control the desired trajectory. This is also confirmed by the time evolution of the average displacment shown in Fig. 7 for a parameter of c=0.175. This value of c results in the lowest power consumption for both, unstable and stable patterns as we can see in Fig. 6.

To show that one can indeed achieve a fast and successful control with a low energy consumption we use a quadratic function of the form

$$c \cdot \left(1 - x^2\right) \tag{7}$$

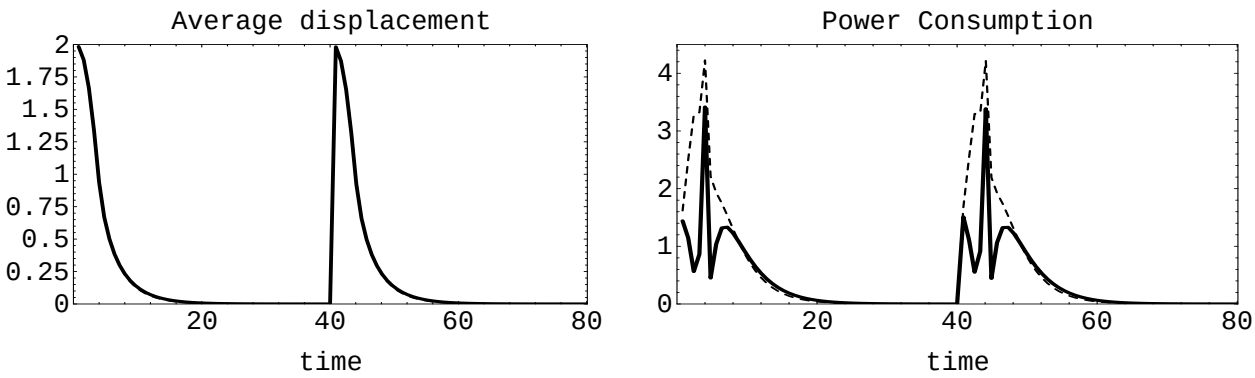

Fig. 7. The control performance of a free fall trajectory control for switching two times between unstable patterns (solid curve) and stable patterns (dashed curve) for a parameter value c=0.175. In the left plot we show that the time evolution of the average displacement is the same but the corresponding power consumption, shown in the right plot, is lower for unstable patterns than for stable ones. The energy consumption is 15.23 energy units for the unstable system and 23.26 energy units for the stable system.

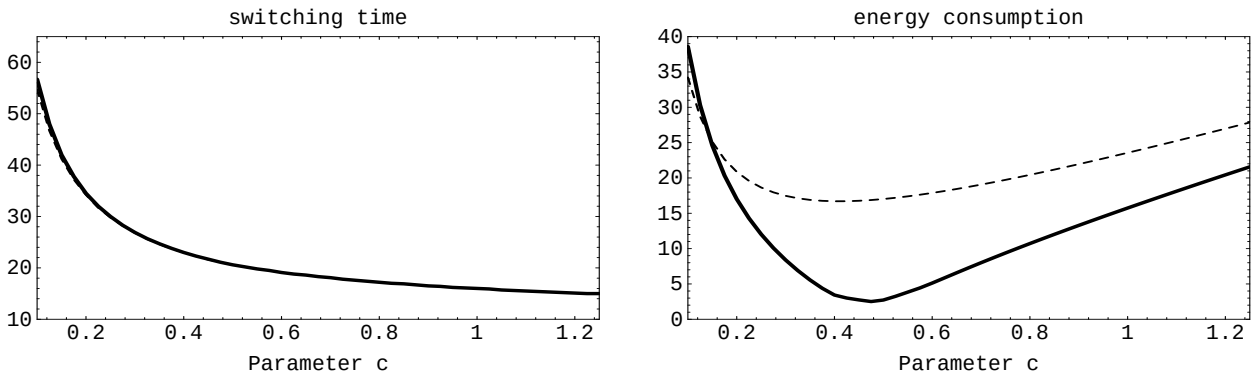

Fig. 8. The performance of a trajectory control for different values of the parameter c in Eq. 7, varying from 0.1 to 1.25 with step size 0.025. On the left is the switching time and on the right the energy consumption. In both plots the solid curve describes switching between unstable patterns and the dashed curve refers to switching between stable patterns.

as the desired trajectory which is shown for c=0.5 in Fig. 5b. To provide an initial push we use a slightly modified force compared to the free fall trajectory control which is based on the term $-0.01 \cdot x_i$ that we add to the desired velocity given by Eq. 7 as long as $\left|x_i - x_i^{desired}\right| > 1$.

The choice of this trajectory is only motivated by the good coverage of the real state space trajectory and assumes that the designer of the pattern control algorithm would have detailed knowledge about the systems dynamics. To get an idea of how the control performance depends on the parameter c, we take a closer look on energy consumption and switching time for different values of c, as it is shown in Fig. 8.

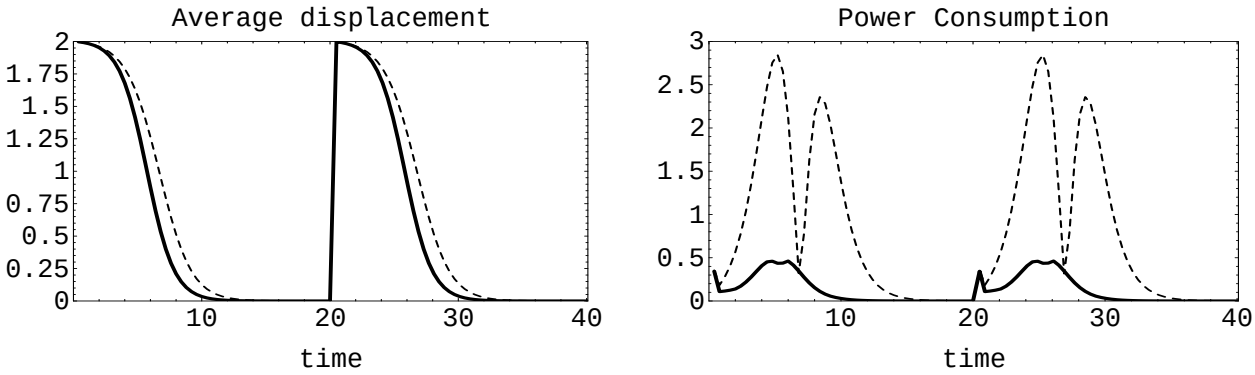

Fig. 9. Comparison of the performance of a trajectory control that uses a function $v_i^{desired} = c \cdot \left(1 - x_i^2\right)$ for switching between unstable patterns (solid curve, c=0.475) and stable patterns (dashed curve, c=0.4). The left plot shows the average displacement. In the right plot we exhibit the corresponding power consumption. We used a b_v factor of 50 for the simulations. The energy consumption is 2.51 energy units for the unstable and 16.70 energy units for the stable system.

Compared to Fig. 6 we see that the minimal energy consumption for the pattern control is in general lower but we also observe a significant decrease for the switching between unstable patters around a parameter value of c=0.5. In Fig. 9 we show the time evolution of switching time and power consumption for the c values that correspond to the lowest energy consumption in Fig. 8. The outstanding performance of this control in connection with the use of unstable patterns is obvious. We observe the lowest energy consumption of $P = 2.51$ compared with the PD-control and the free fall trajectory approach. In case of stable patterns we find a much higher energy consumption of $P = 16.70$.

4. DISCUSSION

In this paper we presented a comparison between the pattern control performance for unstable and stable pattern by applying two different control algorithms. Except for a small parameter range, the PD-control is not significantly different for unstable and stable patterns. One plausible explanation of this result is the existence of the counteracting effects of the displacement- and velocity-proportional control forces. Another reason which also explains the better performance of the trajectory control is the fact that the dynamical landscape of our model system is non-linear, so that different locations in phase space have different optimal parameters b_x and b_v values. Thus, optimality cannot be achieved by fixed parameters in a PD-control and a pre-specified state space trajectory will perform better as long as the pre-specification is reasonable and within valid boundaries. We point out however that the used trajectory might not be the optimal one if one uses a performance measure like power consumption.

In the case of trajectory control for unstable patterns, switching can principally be done with no power consumption if zero friction and perfect information is assumed. For a desired reduction of switching time under a non-zero power limit one has different possibilities and it is not a priori clear which trajectory to choose. A simple one, like the free fall assumption, will in general use more power than in the optimal case but it offers the advantages of being simple in form and implementation. A big advantage of the simple approach is its potential use in module architectures as a fundamental part for controlling more complex systems. By using a pre-specified state space trajectory one automatically chooses the switching time as we saw in Fig. 6 and Fig. 8 which, by turning this statement around, can then be used to choose a desired switching time.

In this paper, we assumed that external forces are not present or that they are so small that they would not have any impact on the switching dynamics. An extension of our research towards more realistic scenarios would consider inaccurate and delayed sensor information as well as the effect of external noise or vibrations. Another interesting question concerns highly irregular patterns in which case the springs have a considerable effect on the control performance. We expect that for more complicated patterns the choice of the control algorithm will have a much more significant effect on the switching performance, dependent on how reasonable it would make use of the power stored and released by the springs. In this respect we believe that an application of a market-based control which we have used previously to control unstable patterns [4, 6] seems to be a promising approach. In the same direction the implementation of learning algorithms might lead to a further performance improvement as well [11].

Acknowledgment

Oliver Guenther acknowledges support by a scholarship from the *Deutscher Akademischer Austauschdienst* (DAAD).

References

[1] Janusz Bryzek, Kurt Petersen, and Wendell McCulley. Micromachines on the march. *IEEE Spectrum*, pages 20–31, May 1994.

[2] K. J. Gabriel. Microelectromechanical systems (MEMS). A World Wide Web Page with URL http://eto.sysplan.com/ETO/MEMS/index.html, 1996.

[3] A. A. Berlin, H. Abelson, N. Cohen, L. Fogel, C. M. Ho, M. Horowitz, J. How, T. F. Knight, R. Newton, and K. Pister. Distributed information systems for MEMS. Technical report, Information Systems and Technology (ISAT) Study, 1995.

[4] Oliver Guenther, Tad Hogg, and Bernardo A. Huberman. Controls for unstable structures. In V. V. Varadan and J. Chandra, editors, *Proc. of SPIE Conf. on Smart Structures and Materials*, volume 3039, pages 754–763. SPIE, 1997.

[5] Tad Hogg and Bernardo A. Huberman. Controlling chaos in distributed systems. *IEEE Trans. on Systems, Man and Cybernetics*, 21(6):1325–1332, November/December 1991.

[6] Oliver Guenther, Tad Hogg, and Bernardo A. Huberman. Market organizations for controlling smart matter. In R. Conte, R. Hegselmann, and P. Terna, editors, *Simulating Social Phenomena*, pages 241–257, Berlin, 1997. Springer.

[7] L. F. Shampine and M. K. Gordon. *Computer Solution of Ordinary Differential Equations: The Initial Value Problem*. W. H. Freeman, San Francisco, 1975.

[8] J. C. Doyle, B. A. Francis, and A. R. Tannenbaum. *Feedback Control Theory*. Macmillan, New York, 1992.

[9] M. Vidyasagar. *Nonlinear Systems Analysis*. Prentice Hall, Englewood Cliffs, NJ, 1993.

[10] J. E. Slotine and W. Li. *Applied Nonlinear Control*. Prentice Hall, Englewood Cliffs, NJ, 1991.

[11] Oliver Guenther, Tad Hogg, and Bernardo A. Huberman. Learning in multiagent control of smart matter. In S. Sen, editor, *Proc. of the AAAI97 Workshop on Multiagent Learning*, pages 36–40, Menlo Park, CA, 1997. AAAI Press.