Model-independent nonlinear control algorithm with application to a liquid bridge experiment

Valery Petrov,* Anders Haaning, Kurt A. Muehlner, Stephen J. Van Hook, and Harry L. Swinney†
Center for Nonlinear Dynamics and Department of Physics, The University of Texas at Austin, Austin, Texas 78712
(Received 11 March 1998)

We present a control method for high-dimensional nonlinear dynamical systems that can target remote unstable states without *a priori* knowledge of the underlying dynamical equations. The algorithm constructs a high-dimensional look-up table based on the system's responses to a sequence of random perturbations. The method is demonstrated by stabilizing unstable flow of a liquid bridge surface-tension-driven convection experiment that models the float zone refining process. Control of the dynamics is achieved by heating or cooling two thermoelectric Peltier devices placed in the vicinity of the liquid bridge surface. The algorithm routines along with several example programs written in the MATLAB language can be found at ftp://ftp.mathworks.com/pub/contrib/v5/control/nlcontrol. [S1063-651X(98)12807-6]

PACS number(s): 05.45.+b, 43.25.Ts, 47.20.Dr, 47.20.Ky

I. INTRODUCTION

Feedback control provides a unique ability to alter the behavior of dynamical systems by selectively stabilizing different unstable states that may coexist in such systems. Stabilization requires an application of very small perturbations, with little modification of the original system. For example, unstable periodic orbits that densely fill a chaotic attractor provide an infinite reservoir of different periodic behaviors and the chaotic system can be forced to follow any of the desired periodic motions using simple linear control methods, known as Ott-Grebogi-Yorke (OGY) control [1]. This approach has been exploited to control chaotic behavior in a variety of physical, chemical, and biological systems [2–6]. OGY control theory and other similar methods assume the availability of linear equations of motion obtained either by linearization of the underlying nonlinear equations of motion or by the method of system identification [7] from laboratory time series. In all the cases the system is described in the small neighborhood of the steady state or fixed point where linear approximation works well. Once the linear dynamics is known the goal of control is to change the eigenvalues of the controlled system, which is usually accomplished using a pole-placement algorithm [8].

Often, however, unstable states are distant in phase space from the system attractor. For example, when a steady state loses stability through a Hopf bifurcation it becomes separated from the stable limit cycle by the nonlinear vector flow. Large perturbations have to be applied in order to move the system from the limit cycle to the unstable steady state through the nonlinear regions of the vector field. Nonlinear targeting can be accomplished in some cases by linear methods [9], when linearized equations provide a fair approximation of the dynamics. In regimes where nonlinearity of the vector flow is strong, however, nonlinear control methods must be used.

A recently introduced model-independent algorithm [10] can control nonlinear systems even when the underlying equations of motion are not available. The algorithm constructs its control law as a look-up table obtained directly from a sequence of system responses to applied perturbations. Because no mathematical transformations are used to define the control law, the algorithm is simple and robust towards identification errors. In contrast, a feedback linearization technique [11], widely adopted for nonlinear control, suffers from structural uncertainties and identification errors that are inevitably present when dynamical equations are reconstructed from time series. Model-predictive control [12] is another nonlinear control approach, especially popular among chemical engineers. Unfortunately, because it requires that the optimization routine be carried out every sampling iteration, it is impossible to use in applications requiring fast response from the controller.

Our model-independent algorithm has been successfully applied in the earlier experiments to stabilize periodic orbits in a liquid bridge [13]. However, complete suppression of time-dependent behavior, i.e., stabilization of the steady state, was not possible with the scalar form of the control algorithm. We present an extended vector form of the nonlinear control algorithm in Sec. II and then demonstrate its use in Sec. III by stabilizing an unstable steady state in liquid bridge convection.

II. DESCRIPTION OF THE CONTROL ALGORITHM

A. Necessary conditions for controllability in linear and nonlinear systems

Controlling a dynamical system entails finding a perturbation sequence that moves the system from its present state $\boldsymbol{\xi}_p$ to some target state $\boldsymbol{\xi}_t$, as schematically shown in Fig. 1. The state vector $\boldsymbol{\xi}$ is the vector in phase space that moves along the system trajectory. We use a discrete description of the system by sampling the dynamics at equal time intervals. This discretization facilitates computerized data processing and simplifies application of the control perturbations. Control perturbations are piecewise, i.e., they are kept constant between sampling points. Formally, an evolution of $\boldsymbol{\xi}$ is governed by some nonlinear function **F**:

$$\boldsymbol{\xi}(i+1)=\mathbf{F}(\boldsymbol{\xi}(i),\mathbf{u}(i+1)), \tag{1}$$

*Electronic address: Val.Petrov@chaos.ph.utexas.edu
†Electronic address: swinney@chaos.ph.utexas.edu

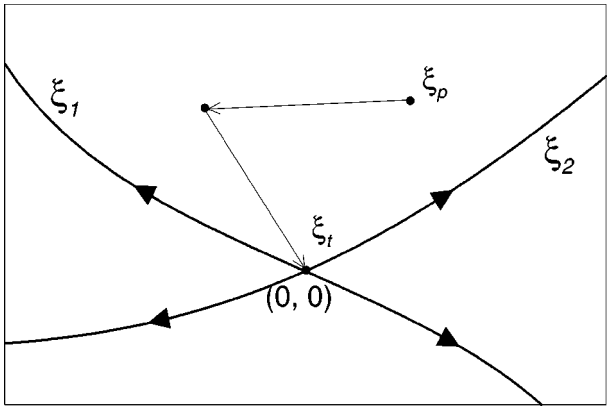

FIG. 1. Phase space representation of the control algorithm. The system trajectory is moved from the present state $\boldsymbol{\xi}_p$ to the target state $\boldsymbol{\xi}_t$, which is often an unstable fixed point as depicted here. Targeting requires two iterations to achieve the goal state in this schematic drawing for a two-dimensional system.

where $\mathbf{u}(i+1)$ is a vector of perturbations that are applied to the inputs of the system between iterations i and $i+1$. If Eq. (1) can be inverted to find $\mathbf{u}$ for an arbitrary present state $\boldsymbol{\xi}_p$ and given target state $\boldsymbol{\xi}_t$, then the control problem is solved. In the linear case when we can approximate Eq. (1) as

$$\boldsymbol{\xi}(i+1)=\frac{\partial \mathbf{F}}{\partial \boldsymbol{\xi}}\cdot\boldsymbol{\xi}(i)+\frac{\partial \mathbf{F}}{\partial \mathbf{u}}\cdot\mathbf{u}(i+1),$$

the solution for $\mathbf{u}$ can always be found if the rank of the system response matrix $\partial\mathbf{F}/\partial\mathbf{u}$ equals the dimensionality of the system $m=\dim(\boldsymbol{\xi})$. This condition is fulfilled when the control parameters are not degenerate and $\dim(\mathbf{u})\geqslant m$. When $\mathbf{F}$ is nonlinear, one cannot guarantee the availability of the solution of Eq. (1) even when $\dim(\mathbf{u})>m$.

When a system does not have sufficient independent control inputs to reach the target state in one step, control is still possible with n controlling iterations applied in succession. This can be seen from the equation for an nth iterate:

$$\begin{aligned}\boldsymbol{\xi}(i+n)&=\mathbf{F}(\ldots\mathbf{F}(\boldsymbol{\xi}(i),\mathbf{u}(i+1)),\ldots\mathbf{u}(i+n))\\&\equiv\mathbf{F}^n(\boldsymbol{\xi}(i),\mathbf{U}(i+1)),\end{aligned}\tag{2}$$

where

$$\mathbf{U}(i)=[\mathbf{u}(i),\ldots,\mathbf{u}(i+n-1)]$$

is a perturbation vector that combines perturbations applied to all the system's controlling inputs at every iteration of the control sequence.

Since the system of equations defined by Eq. (2) has n $\dim(\mathbf{u})$ unknowns, it can be solved for $\mathbf{U}(i+1)$, at least in the linear sense, when $n\geqslant m/\dim(\mathbf{u})$. The exact controllability conditions for nonlinear systems cannot be easily derived, but the general strategy is to lengthen the control sequence until the system becomes controllable. The control law for this case can be found by inversion of Eq. (2) to express $\mathbf{U}(i+1)$ as a function of $\boldsymbol{\xi}(i)\equiv\boldsymbol{\xi}_p$ and $\boldsymbol{\xi}(i+n)\equiv\boldsymbol{\xi}_t$.

In practice, the whole controlling sequence $\mathbf{U}(i+1)$ does not have to be calculated at the ith iteration. Instead, only the first perturbation $\mathbf{u}(i+1)$ is calculated based on $\boldsymbol{\xi}_p=\boldsymbol{\xi}(i)$ and $\boldsymbol{\xi}_t$ as

$$\mathbf{u}(i+1)=\mathbf{C}(\boldsymbol{\xi}_p,\boldsymbol{\xi}_t).\tag{3}$$

On the next controlling iteration $\mathbf{u}(i+2)$ is calculated using $\boldsymbol{\xi}_p=\boldsymbol{\xi}(i+1)$ and the same $\boldsymbol{\xi}_t$. In the absence of noise we expect that these two approaches should give the same result for $\mathbf{u}(i+2)$, but the latter should be more robust in a noisy environment since more recent information is used for calculating the control perturbations.

B. Observability of system dynamics

The control law defined in Eq. (3) requires the knowledge of the system state vector $\boldsymbol{\xi}$, which may not be observable directly in laboratory conditions. Instead, the system is observed by l sensors that form an observation vector $\mathbf{y}$. In general, $\mathbf{y}$ is some nonlinear projection function $\mathbf{P}$ of the system coordinates:

$$\mathbf{y}(i)=\mathbf{P}(\boldsymbol{\xi}(i)).\tag{4}$$

If $l>m$ and the sensors provide independent information, then a one-to-one reconstruction of the system state is possible from instantaneous readings of observables. Otherwise, time-delayed readings of $\mathbf{y}$ must be used to increase the dimensionality of the observation vector.

Combining Eqs. (2) and (4) for k successive iterations $i,\ldots,i+k-1$, the following system of equations is obtained:

$$\begin{aligned}\mathbf{y}(i)&=\mathbf{P}(\boldsymbol{\xi}(i)),\\\mathbf{y}(i+1)&=\mathbf{P}(\boldsymbol{\xi}(i+1))=\mathbf{P}(\mathbf{F}(\boldsymbol{\xi}(i),\mathbf{u}(i+1))),\\\mathbf{y}(i+k-1)&=\mathbf{P}(\boldsymbol{\xi}(i+k-1))\\&=\mathbf{P}(\mathbf{F}^{k-1}(\boldsymbol{\xi}(i),\mathbf{u}(i+1),\ldots,\mathbf{u}(i+k-1))).\end{aligned}\tag{5}$$

If Eq. (5) can be solved for $\boldsymbol{\xi}(i)$, then the system is observable, i.e., one can reconstruct the state of the system from time-delayed observations. The linearized version of Eq. (5) can be written as

$$\begin{pmatrix}\mathbf{y}(i)\\\cdots\\\mathbf{y}(i+k)\end{pmatrix}=\mathbf{O}\cdot\boldsymbol{\xi}(i)+\mathbf{B}\cdot\begin{pmatrix}\mathbf{u}(i+1)\\\cdots\\\mathbf{u}(i+k)\end{pmatrix},\tag{6}$$

where

$$\mathbf{B}=\frac{\partial\begin{pmatrix}\mathbf{P}(\boldsymbol{\xi})\\\mathbf{P}(\mathbf{F}(\boldsymbol{\xi}))\\\cdots\\\mathbf{P}(\mathbf{F}^{k-1}(\boldsymbol{\xi}))\end{pmatrix}}{\partial[\mathbf{u}(i+1)\cdots\mathbf{u}(i+k)]}$$

describes the sensitivity of the observations to applied perturbations and the observability matrix

$$\mathbf{O}=\frac{\partial\begin{pmatrix}\mathbf{P}(\boldsymbol{\xi})\\ \mathbf{P}(\mathbf{F}(\boldsymbol{\xi}))\\ \cdots\\ \mathbf{P}(\mathbf{F}^{k-1}(\boldsymbol{\xi}))\end{pmatrix}}{\partial\boldsymbol{\xi}}$$

defines the relation between time-delayed observations and the system state. Solvability is satisfied when $\mathrm{rank}(\mathbf{O})\geqslant m$. This means that if the number of independent observation channels is less than the dimensionality of the system, then a time-delayed vector with $k=m/l$ delayed coordinates has to be built.

For a nonlinear system, observability cannot be guaranteed based on simple considerations of the linear independence of time-delayed observations. The problem is reminiscent of prediction of a time series generated by a nonlinear system. The longer the length of the observation sequence, the higher the probability that the solution can be found; therefore, in practice k can be increased until the system becomes observable.

From these considerations Eq. (3) can be modified to express $\boldsymbol{\xi}_p$ and $\boldsymbol{\xi}_t$ as a function of $\mathbf{y}$ and $\mathbf{u}$. Equation (5) explicitly relates $\boldsymbol{\xi}(i)$ and the set of time forwarded $\mathbf{y}$ and $\mathbf{u}$. Time-delayed readings can be used as well because $\boldsymbol{\xi}(i+k-1)$ can be calculated from $\boldsymbol{\xi}(i)$ and $\mathbf{U}(i)$ using Eq. (2). Therefore, $\boldsymbol{\xi}_p$ in Eq. (3) can be replaced by the time-delayed vector

$$\mathbf{Y}^d(i)=[\mathbf{y}(i-k+1),\ldots,\mathbf{y}(i),\mathbf{u}(i-k+2),\mathbf{u}(i-k)]$$

and $\boldsymbol{\xi}_t$ can be replaced by the time-forwarded vector

$$\mathbf{Y}^f(i)=[\mathbf{y}(i),\ldots,\mathbf{y}(i+k-1),\mathbf{u}(i+1),\mathbf{u}(i+k-1)].$$

With these new arguments the control function C becomes dependent only on quantities that can be measured directly in experiment.

C. Construction of the control law

The choice of time direction in forming the sequences $\mathbf{Y}^d$ and $\mathbf{Y}^f$ can be understood from Fig. 2. There are two different stages of the algorithm: identification and control. During the identification stage, uniformly distributed, random perturbations are applied to the system and corresponding responses are measured to build a control function. During the control stage, the $\mathbf{u}$ and $\mathbf{y}$ that define the present state are recorded while $\mathbf{u}$ and $\mathbf{y}$ for the target state are preset to values determined by the control objective.

Since all data recorded during the identification stage can be analyzed off line, past and future information relative to any point in the middle of the identification sequence is available. One can use either a time-forwarded or a time-delayed sequence for defining the current state. During the on-line control stage, however, only past information relative to the current iteration is available. This imposes the restriction that only time-delayed observations can be used to define the current state.

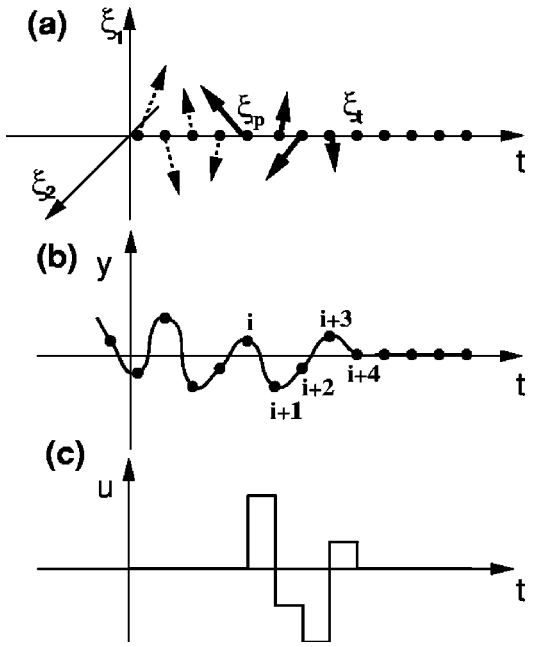

FIG. 2. Schematic controlling sequence showing (a) relations between time series of state vectors, (b) observables, and (c) controlling perturbations. The dashed vectors in (a) correspond to unperturbed dynamics, while the solid vectors represent the system trajectory during the controlling sequence.

The use of time-delayed perturbations in the target state definition requires knowledge of perturbations applied during the control sequence, as can be seen from Fig. 2. The stabilized steady state at $\boldsymbol{\xi}_t=0$ in Fig. 2 corresponds to $\mathbf{y}=\mathbf{0}$. The observer looking at $\mathbf{y}$ will conclude that the target state is reached only after several $\mathbf{y}=\mathbf{0}$ and $\mathbf{u}=\mathbf{0}$ are recorded, i.e., several iterations past the moment of reaching $\boldsymbol{\xi}_t=0$. The use of a time-forwarded sequence in this case allows the convenient definition of the target state in terms of the desired values of the observation variables. For example, to define the steady state $\boldsymbol{\xi}=0$ as the target state, values $\mathbf{y}_{target}=\mathbf{0}$ and $\mathbf{u}_{target}=\mathbf{0}$ are substituted in $\mathbf{Y}^f$.

If the targeted steady state has never been visited before by the system trajectory, then the exact value of the observable $\mathbf{y}_{target}$ may not be known to us. In this case one can use a time derivative and set $\dot{\mathbf{y}}=\mathbf{0}$ as a target. Since in experiment $\dot{\mathbf{y}}$ usually has a low signal-to-noise ratio, the finite differences formulation should be used instead:

$$\mathbf{DY}^f(i)=(\mathbf{y}(i+1)-\mathbf{y}(i),\ldots,\mathbf{y}(i+k)-\mathbf{y}(i+k-1),\mathbf{u}(i+1),\ldots,\mathbf{u}(i+k)).$$

The additional reading $\mathbf{y}(i+k)$ in $\mathbf{DY}^f(i)$ necessitates the inclusion of the additional perturbation term $\mathbf{u}(i+k)$ since $\mathbf{y}(i+k)$ is affected by the perturbation applied during iteration $i+k$. In our experiment the finite-difference form of the state vector was used both for $\mathbf{Y}^f$ and $\mathbf{Y}^d$ in order to cancel low-frequency drift in the temperature sensors.

Note that special care must be exercised during the identification procedure if the controlled dynamical system has multiple steady states. Since $\dot{\mathbf{y}}=\mathbf{0}$ cannot differentiate between coexisting states, the function C will be multivalued

and additional constraints are necessary to specify the target state.

Real-time application of the algorithm requires some time be spent on calculation of the controlling perturbation. This will produce a delay d before the current measurement is taken and the corresponding perturbation is applied. Therefore, the target state will be reached in $d+n$ iterations instead of n iterations. Also, since applied perturbations disturb the corresponding observations d iterations later, the indices of $\mathbf{u}$ in $\mathbf{Y}$ have to be shifted by d iterations. As a result of incorporating delays and finite differences in the state vectors in Eq. (3), the control surface has the form

$$\mathbf{u}(i+1)=\mathbf{C}(\mathbf{DY}^d(i),\mathbf{DY}^f(i+n+d)),$$

$$\mathbf{DY}^d(i)=\begin{pmatrix} \mathbf{y}(i-k+1)-\mathbf{y}(i-k) \\ \cdots \\ \mathbf{y}(i)-\mathbf{y}(i-1) \\ \mathbf{u}(i-k-d+1) \\ \cdots \\ \mathbf{u}(i) \end{pmatrix},$$

$$\mathbf{DY}^f(i)=\begin{pmatrix} \mathbf{y}(i+1)-\mathbf{y}(i) \\ \cdots \\ \mathbf{y}(i+k)-\mathbf{y}(i+k-1) \\ \mathbf{u}(i-d+1) \\ \cdots \\ \mathbf{u}(i-d+k) \end{pmatrix}. \tag{7}$$

The choice of n, k, and d depends on the controlled system and may be not known *a priori*. In linear or weakly nonlinear systems the length of the observation sequence (k) multiplied by the number of observation channels and the length of the control sequences (n) multiplied by the number of feedback elements are usually the same and equal to the dimensionality of the system (m). If m is not known but the system has low noise and the dynamics has a well defined number of degrees of freedom, then the set of data recorded during the identification stage can be used to check the fitting error that the reconstructed surface C produces against recorded perturbations. The optimal choice of n, k, and d will minimize the fitting error. However, if a large amount of noise is present or the controlled system has many degrees of freedom, then the fitting error criteria become less reliable. The best solution is to apply the control algorithm, trying different combinations of n, k, and d, and select the combination that provides the optimal convergence toward the target state.

Once all terms in the control surface C are established, the exact shape can be found using different approximating techniques. The specific choice depends on the quality and size of the data sets, nature of the controlled system, and required speed of the control calculation. Artificial neural networks, for example, can be used to approximate C when fast calculation of the controlling perturbation is required [14]. We found, however, that neural networks have poor approximating properties when used with small data sets and high-dimensional systems.

In the experiment described in this paper a look-up table procedure was used with the reference data set recorded during the identification sequence serving as a basis for approximation of C. At every iteration of the control stage the vector $[\mathbf{DY}^d_{current},\mathbf{DY}^f_{target}]$ is matched with the vectors from the identification data set using minimum distance criteria. Several neighbors are used for approximating the surface at the point of interest. In the absence of noise, $R=\dim(\mathbf{DY}^f)+\dim(\mathbf{DY}^d)+1$ points are sufficient to draw an approximating tangential hyperplane in R-dimensional space. If noise is present, it is desirable to have more data points and use a least-squares fitting procedure to determine the parameters of this plane. We used the technique of singular-value decomposition [15] to find the approximating plane in a manner suggested for prediction of time series generated by nonlinear systems [16], except that here we are trying to ''predict'' controlling perturbations that will bring our system to the desired state.

The optimal number of neighbors used for interpolation depends on the nonlinearity of the surface and density of points in the region of phase space where the approximation is made. Although a larger neighborhood allows more points to be used in calculating the approximation, it may be best to use a smaller neighborhood in regions with high local curvature since the remote points may introduce a systematic error due to the deviation of the surface from the plane. The optimal approximation should therefore take into account different curvatures of the surfaces in the different regions of the phase space. We found that using a constant number of approximating neighbors gives sufficiently uniform error distribution for our weakly nonlinear system. The number of neighbors that minimizes the average error of the approximation in our experiment was found to be approximately $3R$.

D. Using the algorithm

Despite its internal complexity, the final formulation of the algorithm is compact and can easily be used with any multiple-input–multiple-output system. In addition to stabilizing unstable states, the method can be used for targeting objective dynamics or tracking a preset trajectory. The only modification to the equations for the targeting is a different form of $\mathbf{DY}^f(i)$ that uses differences in perturbations instead of differences in observations [10]. The complete implementation of the algorithm incorporating all the different forms of goal dynamics can be found at ftp://ftp.mathworks.com/pub/contrib/v5/control/nlcontrol. The routines are written in MATLAB language and also include sample programs for controlling the logistic map, the Lorenz equations, a four-dimensional model of the liquid bridge, and a physical pendulum.

III. APPLICATION OF THE CONTROL ALGORITHM TO A LIQUID BRIDGE EXPERIMENT

A. Experimental setup

A liquid bridge is a convective system where a drop of fluid is trapped between two coaxial cylindrical boundaries [Fig. 3(a)]. A temperature difference ΔT is imposed verti-

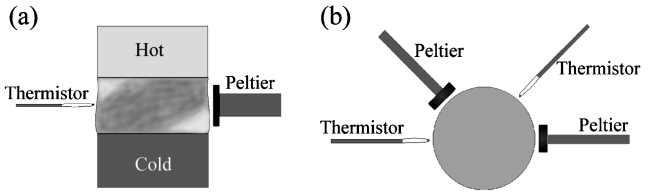

FIG. 3. Sketch of our liquid bridge convection experiment. (a) Side view showing hot and cold boundaries made of coaxial stainless steel cylinders with radii of 0.3 cm. The distance l between cylinders is 0.3 cm. One of the two pairs of sensors and feedback elements is shown. (b) Top view indicating angular location of the sensors and feedback elements. The sensor is a 0.03-cm-diam thermistor that is placed 0.03 cm from the surface of the liquid. The feedback elements are 0.1×0.3 cm^2 thermoelectric devices that are placed at the same height as the temperature sensors.

cally across the drop, with the upper surface heated and bottom surface cooled. Surface-tension gradients due to the imposed temperature gradient drive a steady-state toroidal flow that is downward along the liquid-gas interface and upward in the center of the drop. For large enough ΔT, the axisymmetric toroidal flow becomes unstable to an oscillatory state [17]; additional oscillatory frequencies appear for even larger ΔT [18]. Liquid bridge convection models hydrodynamic effects in the float-zone refinement of crystalline materials, where appearance of the time-dependent convective flow induces undesired variation in the chemical composition of the crystals that are formed [19].

Our working fluid is a purified Dow Corning 200 silicone oil [20] with a Prandtl number of approximately 40 and a volume of 0.065 cm^3. We impose $\Delta T\sim12.5\,^\circ$C with the upper boundary warmer than the lower; the mean temperature of the bottom boundary is 15.0 °C and ΔT is computer controlled to a precision of $\pm0.05\,^\circ$C. Buoyancy effects, which would stabilize convection due to the direction of heating, play little role since the height of the drop is small.

The dimensionless number that characterizes the surface tension driving is the Marangoni number $M\equiv\sigma_T\Delta Tl/\rho\nu\kappa$, with liquid density $\rho=0.89$ g/cm^3, kinematic viscosity $\nu=0.026$ cm^2/s, distance $l=0.3$ cm between the cylinders, thermal diffusivity $\kappa=7.4\times10^{-4}$ cm^2/s, surface tension σ, and $\sigma_T\equiv|d\sigma/dT|=0.068$ dyn/cm K. For small M, the convective flow is time independent. For $M\gtrsim14\,000$, the flow becomes oscillatory and infrared imaging reveals that the flow has the structure of a helical traveling wave [18]. In our experiment, $M\sim15\,000$.

In our previous experiments a single temperature sensor and a single feedback element were sufficient for stabilization of unstable periodic orbits. However, attempts to suppress the helical traveling wave via a simple extension of this scheme were not effective. Instead of stabilizing the steady state in the liquid bridge, the control algorithm was found to change the flow into a standing wave with a node at the sensor location. This failure can be explained by the degeneracy of the stationary state with respect to the direction of wave rotation, counterclockwise and clockwise, resulting in the appearance of rotational degrees of freedom with exactly the same eigenvalues. It is thus impossible to realize and control the direction of wave rotation just by looking at the temperature variation of a single probe since both the counterclockwise and clockwise rotating waves will produce identical temperature variation at any particular location around the liquid bridge surface. In order to remove this degeneracy we use measurements of the surface temperature at two locations around the liquid bridge and apply perturbations by lowering or raising the temperature of two feedback elements positioned as shown in Fig. 3(b).

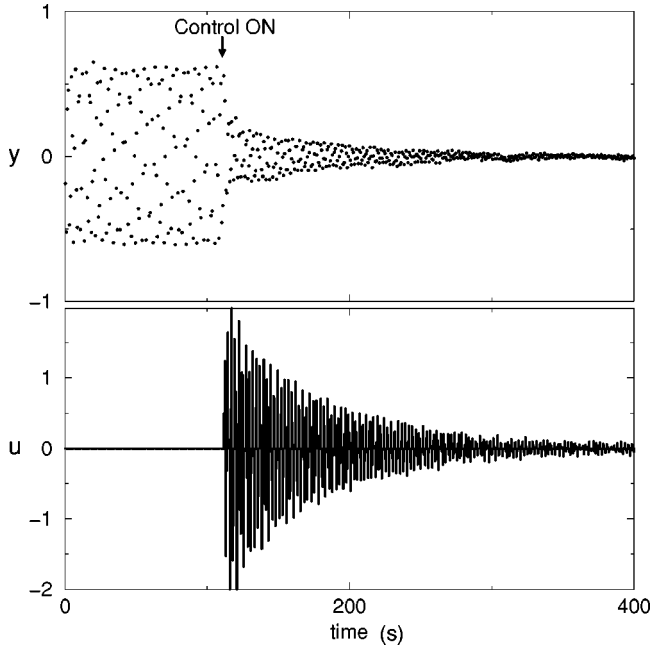

FIG. 4. Time-series recordings of the control experiment. The upper plot shows readings of the first channel sensor. The lower plot shows perturbations applied to the first channel feedback element. Second channel readings look identical.

B. Experimental results

Temperature measurements converted into voltage and digitized every 0.7 s are used to define our observation vector $\mathbf{y}(i)=[y_1(i),y_2(i)]$. To synthesize the state vectors $\mathbf{Y}^d$ and $\mathbf{Y}^f$, the numbers n, k, and d are found using a trial-and-error method. The dimensionality of the liquid bridge is expected to be at least 4 since each traveling wave can be described by two variables with two possible directions of rotation. Fastest convergence of the algorithm is achieved with $k=n=4$, suggesting that the overall dimensionality m of the bridge is $k\times2=n\times2=8$. The four additional degrees of freedom probably correspond to the pair of counterrotating waves that are least stable for the conditions of our experiment. These waves are excited by the applied perturbations and contribute to the observed dynamics.

The optimal delay d is found to be two sampling periods (1.4 s). One delay interval is needed for the sorting procedure on our Pentium 120-MHz computer, which forces the program to skip at least one sampling period before the calculated controlling perturbations are applied to the feedback elements. The second sampling interval delay improves convergence since it takes approximately 0.3 s for changes in the temperatures of the Peltier devices to affect the fluid flow. Selecting $d=2$ makes the overall dimensionality of the control surface C equal to 36 ($m\times4+l\times d=8\times4+2\times2$). Such high dimensionality is a drawback of the reconstruction

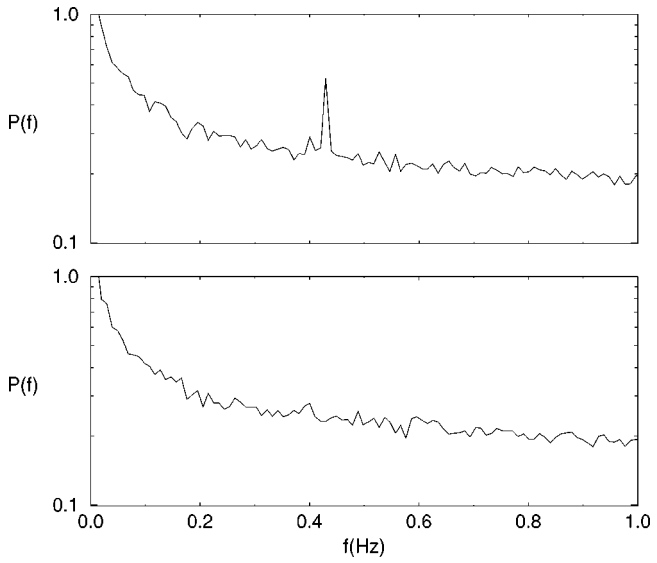

FIG. 5. Power spectra of the temperature before control is applied (top) and after the convergence to the steady state is complete (bottom). The dynamics of 50×100 individual pixels in the infrared image of the liquid bridge is converted to the corresponding power spectra using 512 frames taken with 0.2-s intervals. The power spectra of all pixels are then averaged to improve the signal-to-noise ratio.

since the length of the reference set grows as a power of the embedding dimension. However, since the method does not rely on any model, uncertainties in the controlling surface reconstructions are not amplified by additional transformations and as the experiment shows, the method is robust even with a moderate number of reference data points.

During the first part of the identification stage, 3000 random perturbations and corresponding temperature responses are used to form the reference data set. After the control algorithm is activated, new control perturbations and system responses are added to the reference set. This procedure adaptively decreases the approximating error; as control takes the system closer to the target state, new observations provide a finer interpolation of the control surface around this state. In our experiment an additional 3000 iterations are added to the reference set as the system converges toward the goal.

Figure 4 demonstrates the application of the control algorithm to the liquid bridge after the reference set is built. Control is activated at $t=110$ and slowly suppresses the oscillatory amplitude. Convergence is significantly slower than the expected four iterations since 6000 data points barely provide a satisfactory approximation of a 36-dimensional nonlinear surface. In addition, the maximum range of the controlling perturbations is limited by the power handling capacities of the Peltier devices.

An infrared camera is used to monitor temperature variations on the liquid bridge surface and to confirm that the stabilization is successful at every point of the observed surface of the liquid bridge. Unfortunately, the low signal-to-noise ratio of infrared camera measurements requires filtering of the obtained video sequence and only time-averaged dynamics can be reliably monitored.

A moving window of 512 consecutive recordings is used to calculate power spectra of the bridge dynamics as a function of the frequency, time, and location on the bridge surface. Averaging over the bridge surface produces a power spectrum (Fig. 5) that can be used as an indicator of the overall dynamics of the bridge; i.e., any periodic movement will appear as a peak at the corresponding frequency. The reduction of the fundamental frequency to the noise floor after control is activated confirms the stabilization of the steady state. No additional waves are excited by the control and as a result the controlling perturbations become quite small, as shown in Fig. 4(b).

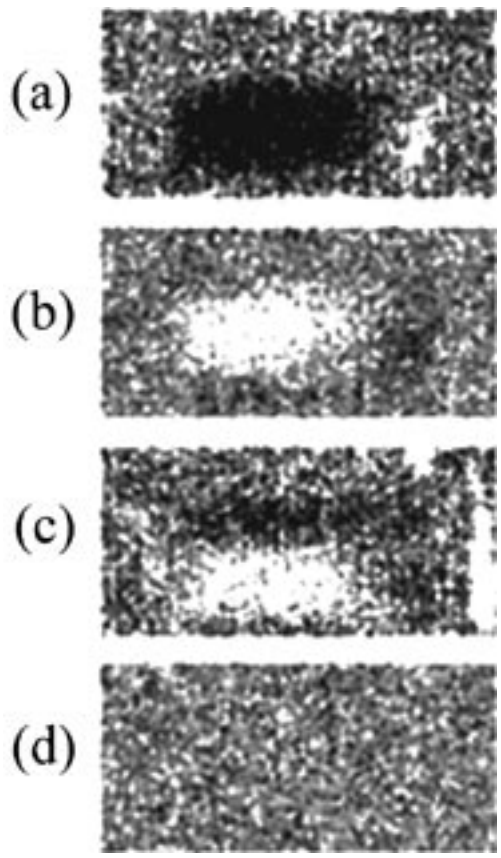

FIG. 6. Infrared images (0.6 cm$\times$0.3 cm) showing temperature distribution in the liquid bridge [18] (a) and (b) before, (c) during, and (d) after the controlling sequence. Darker regions have lower temperatures. The waves in (a) and (b) appear to deviate from the helical form because Peltier devices disrupt the symmetry of the boundary conditions even when the control is not active.

The spatial distribution of the temperature field filtered at the characteristic frequency of 0.42 Hz is shown in Fig. 6 for different stages of the experiment. The first two images show the temperature distribution of the rotating wave before the application of the algorithm. The third image shows a moment during the control sequence, where uneven temperature distribution is produced by the control. The fourth image confirms the complete suppression of the temperature variations in the liquid bridge.

Large perturbations are required to push the system from the stable limit cycle to the unstable steady state at the beginning of the control sequence. Once the steady state is reached, only small perturbations are necessary to maintain the stationary convection and the average power applied to the controllers drops to about 100 μW, less than 1% of the heat flow through the bridge.

Restrictions on the maximum size of the applied perturbations may forbid the targeting of very remote points in phase space. We are not able to suppress time-dependent convection when the temperature difference is more than 1.0 °C above the critical value ($\Delta T = 11.7°$) that corresponds to the Hopf bifurcation in the system. This limitation is mostly due to the weak response of the fluid flow to changes in the temperatures of the Peltier devices, which cannot be cooled more than a few degrees during the application of the pulse.

Our control algorithm works well for Marangoni numbers near the onset of the oscillatory flow; the ability to control a spatially extended system with four unstable degrees of freedom in laboratory conditions proves the robustness of the algorithm. Using high-power feedback elements, one could apply the method to improve quality of the zone refining processes in industry.

ACKNOWLEDGMENTS

The authors thank Michael F. Schatz, Stanislav Shvartsman, William D. McCormick, and Jack B. Swift for useful discussions. This research was supported by the NASA Microgravity Science and Applications Division (Grant No. NAG3-1839), the Office of Naval Research (Grant No. N00014-89-J-1495), a NASA ESS Cooperative Agreement (Contract No. NCCS5-159), and the U.S. Department of Energy (Grant No. DE-FG03-93ER14312).

[1] E. Ott, C. Grebogi, and J. A. Yorke, Phys. Rev. Lett. **64**, 1196 (1990).
[2] W. L. Ditto, S. N. Rauseo, and M. L. Spano, Phys. Rev. Lett. **65**, 3211 (1990).
[3] R. Roy, T. W. Murphy, T. D. Maier, Z. Gills, and E. R. Hunt, Phys. Rev. Lett. **68**, 1259 (1992).
[4] V. Petrov, V. Gáspár, J. Masere, and K. Showalter, Nature (London) **361**, 240 (1993).
[5] A. Garfinkel, M. L. Spano, W. L. Ditto, and J. N. Weiss, Science **257**, 1230 (1992).
[6] S. J. Schiff, K. Jerger, D. H. Duong, T. Chang, M. L. Spano, and W. L. Ditto, Nature (London) **370**, 615 (1994).
[7] L. Ljung, *System Identification: Theory for the User* (Prentice-Hall, Englewood Cliffs, NJ, 1987).
[8] K. Ogata, *Discrete-Time Control Systems* (Prentice-Hall, Englewood Cliffs, NJ, 1987).
[9] D. J. Christini and J. J. Collins, Phys. Rev. E **53**, R49 (1996).
[10] V. Petrov and K. Showalter, Phys. Rev. Lett. **76**, 3312 (1996).
[11] A. Isidori, *Nonlinear Control Systems* (Springer, New York, 1989).
[12] M. A. Henson and D. E. Seborg, *Nonlinear Process Control* (Prentice-Hall, Upper Saddle River, NJ, 1996).
[13] V. Petrov, M. F. Schatz, K. A. Muehlner, S. J. VanHook, W. D. McCormick, J. B. Swift, and H. L. Swinney, Phys. Rev. Lett. **77**, 3779 (1996).
[14] E. R. Weeks and J. M. Burgess, Phys. Rev. E **56**, 1531 (1997).
[15] W. H. Press, W. T. Vetterling, S. A. Teukolsky, and B. P. Flannery, *Numerical Recipes in C* (Cambridge University Press, New York, 1994).
[16] H. D. I. Abarbanel, R. Brown, J. J. Sidorowich, and L. S. Tsimring, Rev. Mod. Phys. **65**, 1331 (1993).
[17] R. Velten, D. Schwabe, and A. Scharmann, Phys. Fluids A **3**, 267 (1991).
[18] K. A. Muehlner, M. F. Schatz, V. Petrov, W. D. McCormick, J. B. Swift, and H. L. Swinney, Phys. Fluids **9**, 1850 (1997).
[19] A. Croll, W. Muller, and R. Nitsche, Microgravity Sci. Technol. **3**, 204 (1991).
[20] M. F. Schatz and K. Howden, Exp. Fluids **19**, 359 (1995).