Control and Chaos

Control and Chaos

Kevin Judd
Alistair Mees
Kok Lay Teo
Thomas L. Vincent

Editors

Birkhäuser
Boston • Basel • Berlin

Kevin Judd
Alistair Mees
Kok Lay Teo
Center for Applied Dynamics
and Optimization
Department of Mathematics
The University of Western Australia
Perth, 6907, Australia

Thomas L. Vincent
Aerospace and Mechanical Engineering
University of Arizona
Tucson, Arizona 85721
U.S.A.

Library of Congress Cataloging-in-Publication Data

Control and chaos / Kevin Judd ... [et al.], editors.
p. cm. -- (Mathematical modeling ; no. 8)
"Proceedings of the US-Australia Workshop on Control and Chaos held in Honolulu, Hawaii from 29 June to 1 July 1995" --Pref.
Includes bibliographical references.
ISBN 0-8176-3867-9 (acid-free paper). -- ISBN 3-7643-3867-9 (Basel : acid-free paper)
1. Control theory--Congresses. 2. Chaotic behavior in systems--Congresses. I. Judd, Kevin, 1962- . II. US-Australia Workshop on Control and Chaos (1995 : Honolulu, Hawaii) III. Series: Mathematical modeling (Boston, Mass.) ; no. 8.
QA402.3.C618 1997 97-7728
003'.75--dc21 CIP

Printed on acid-free paper

Birkhäuser

ISBN 0-8176-3867-9
ISBN 3-7643-3867-9
Camera-ready copy provided by the editors.
Printed and bound by Quinn-Woodbine, Woodbine, NJ.
Printed in the U.S.A.

9 8 7 6 5 4 3 2 1

MATHEMATICAL
MODELING

No. 8

Edited by
William F. Lucas, Claremont Graduate School
Maynard Thompson, Indiana University

Birkhäuser
Boston • Basel • Berlin

PREFACE

This volume contains the proceedings of the US-Australia workshop on Control and Chaos held in Honolulu, Hawaii from 29 June to 1 July, 1995. The workshop was jointly sponsored by the National Science Foundation (USA) and the Department of Industry, Science and Technology (Australia) under the US-Australia agreement.

Control and Chaos—it brings back memories of the endless reruns of "Get Smart" where the good guys worked for Control and the bad guys were associated with Chaos. In keeping with current events, Control and Chaos are no longer adversaries but are now working together. In fact, bringing together workers in the two areas was the focus of the workshop.

The objective of the workshop was to bring together experts in dynamical systems theory and control theory, and applications workers in both fields, to focus on the problem of controlling nonlinear and potentially chaotic systems using limited control effort. This involves finding and using orbits in nonlinear systems which can take a system from one region of state space to other regions where we wish to stabilize the system. Control is used to generate useful chaotic trajectories where they do not exist, and to identify and take advantage of useful ones where they do exist. A controller must be able to nudge a system into a proper chaotic orbit and know when to come off that orbit. Also, it must be able to identify regions of state space where feedback control will be effective. Several new methods were presented, including targeting algorithms and the use of Lyapunov functions for identifying controllable targets. There are many direct applications of the methods discussed in this workshop. These include mechanical and electrical systems which are to be controlled, or ecosystems and other biosystems that are to be managed.

The participants of the workshop were from the USA or Australia except for one from England and one from Japan. The participants were more or less balanced according to their interests in dynamics or control systems. Each participant gave a presentation in his or her field of expertise as related to the overall theme. The formal papers (contained in this volume) were written after the workshop so that the authors could take into account the workshop discussions, and relate their work to the other presentations. Each paper was reviewed by other participants who then wrote comments which are attached to the ends of the papers. We feel that the comments form a valuable part of this volume in that they give the reader a share of the workshop experience.

The papers are grouped, as they were presented, according to three classifications: Understanding Complex Systems, Controlling Complex Systems, and Applications. Part I, contains seven papers dealing with modeling, behavior, reconstruction, prediction, and numerics. Part II, contains nine papers on controlling complex systems by means of embedding unstable periodic orbits, targeting, filtering, optimization, and adaptive methods. Part III contains four applications papers including the control of a bouncing ball, evolutionary stability, chaos in ecosystems, and neural networks.

We would like to acknowledge the financial support of NSF and DIST which made this workshop possible. The cooperation and organizational skills of the East-West center where the workshop was held was also most appreciated. Jenny Harris gave invaluable assistance with the Australian end of the workshop organisation. AIM thanks the Institute of Nonlinear Science, U.C. San Diego, for hospitality.

Kevin Judd
Alistair I. Mees
Kok-Lay Teo
Thomas L. Vincent

Perth, Western Australia
Tucson, Arizona

August 1996.

Standing from left: John Pastor, Petar Kokotovic, John Moore, Katie Glass *(in front)*, Masataka Watanabe *(behind)*, Tim Sauer, Constance Schrober, Alistair Mees, Arthur Mazer, Ed Ott, Phil Diamond, Yosef Cohen, Tom Vincent, Eric Kostelich, Iven Mareels.

Seated from left: Kok–Lay Teo, Walt Grantham, Colin Sparrow, Kevin Judd, John Roberts.

Contents

Understanding Complex Dynamics

Controlling Complex Systems

Applications

Triangulating Noisy Dynamical Systems

Stuart Allie, Alistair Mees, Kevin Judd and Dave Watson
Centre for Applied Dynamics and Optimization
Department of Mathematics
The University of Western Australia
Perth, 6907, Australia

Abstract

Triangulation and tesselation methods have proven successful for reconstructing dynamical systems which have approximately the same dynamics as given trajectories or time series. Such reconstructions can produce other trajectories with similar dynamics, so giving a system on which one can conduct experiments, but can also be used to locate equilibria and determine their types, carry out bifurcation studies, estimate state manifolds and so on. In the past, reconstruction by triangulation and tesselation methods has been restricted to low-noise or no-noise cases. This paper shows how to construct triangulation models for noisy systems; the models can then be used in the same ways as models of noise-free systems.

1 Introduction

An important part of controlling systems is modeling them. The system identification and modeling problem is well-studied for linear systems, but there seem to be few results for strongly nonlinear systems. In this paper we present an approach that generalizes the triangulation method of modeling nonlinear dynamical systems, presented at the previous conference in this series [9].

The earlier work applied only to low-noise systems (roughly, where the noise level was small compared with the typical separation of the data points being used to build the model). This paper deals with the case where the noise is non-negligible. We still simplify matters somewhat, as will be explained below.

1.1 Embedding

It may be helpful to say a few words about embedding and reconstruction. Since this is a well-discussed area (see e.g. [1, 16]) we shall be brief. We start with a discrete-time dynamical system with state $x_t \in M$, where M is a d-dimensional manifold, with the dynamics described by

$$x_{t+1} = f(x_t) + \xi_t,$$

where f is unknown and ξ_t is the dynamical noise of the system. We assume that there is a measurement function $\phi : M \rightarrow \mathrm{R}$ such that

$$y_t = \phi(x_t) + \nu_t,$$

with ν_t being the measurement noise. The set $\{y_t\}$ is the scalar time series which is measured. This set is all that is available for constructing the model. From this, we make vectors in R^n of the form

$$z_t = (y_t, y_{t-\tau}, \ldots, y_{t-(n-1)\tau})$$

for an embedding dimension n and lag τ.

By a theorem due to Takens [17], we know that, in general, if n is sufficiently large, then such a mapping is an embedding $h : M \to \mathrm{R}^n$ with $z_t = h(x_t)$. Since h is an embedding, the following diagram commutes.

$$\begin{array}{ccc} x_t & \xrightarrow{f} & x_{t+1} \\ h \big\downarrow & & h \big\downarrow \\ z_t & \xrightarrow{F} & z_{t+1} \end{array}$$

It is the map F which we wish to model. However, because of the form of the embedded states z_t, it is sufficient to estimate the map $\rho : \mathrm{R}^n \to \mathrm{R}$ such that

$$y_{t+1} = \rho(z_t) + \epsilon_t$$

where ϵ_t represents the total noise present.

2 Triangulation approximations

2.1 Triangulating noise-free systems

Before looking at the proposed solution to the general modeling problem let us review the noise-free case. Notice that when $\epsilon = 0$ the problem is to find a map from R^n to R that fits the given data. One of us (AM) has discussed elsewhere [9, 11] how to optimally triangulate a set of points in R^n and then to produce a map $\hat{\rho}$ that is linear on each triangle (or simplex, if $n > 2$), is continuous across triangle boundaries, and is such that for each vertex v_i,

$$\hat{\rho}(v_i) = h_i$$

where h_i is the actual value (the "height") given by the data at v_i. We will see shortly what is meant by an optimal triangulation; probably the best method for constructing it is Watson's [18], and in this paper we will assume that construction of triangulations is a solved problem.

The definition of $\hat{\rho}$ is simple once the triangulation has been constructed: if $x \in T_j$ where triangle T_j has vertices $\{v_k\}$, where k is a set of $n+1$ labels to the set $\{v_i\}$, then there is a unique non-negative solution λ to the equations

$$\sum_{k \in N(x)} \lambda_k(x) v_k = x, \qquad (1)$$

$$\sum_{k \in N(x)} \lambda_k(x) = 1. \qquad (2)$$

Now define

$$\hat{\rho}(x) = \sum_{k \in N(x)} \lambda_k(x) h_k; \qquad (3)$$

that is, we express x as a convex combination of the vertices of the triangle it is in, then use the same convex combination of the vertex heights to approximate the height at x. It is easy to see that $\hat{\rho}$ is continuous and that if the heights did in fact satisfy

$$h = \rho(x)$$

for some smooth function ρ then

$$|\hat{\rho}(x) - \rho(x)| = O(\epsilon^2)$$

where ϵ is the diameter of triangle T_i. (This, incidentally, explains why we do not want to use just any triangulation: we should pick the one that minimizes diameters on average, which turns out to be the Delaunay triangulation [5, 19, 14, 11].)

We can now define an approximation $\hat{F}$ for the dynamical system from R^n to R^n; for example, in the two dimensional case we use

$$\hat{F}(y_t, y_{t-1}) = (\hat{\rho}(y_t, y_{t-1}), y_t) \tag{4}$$

where we are assuming a simple lag-1 embedding. This map takes triangles into triangles; for example, Fig 1 shows a triangulation of an embedding of the Henon

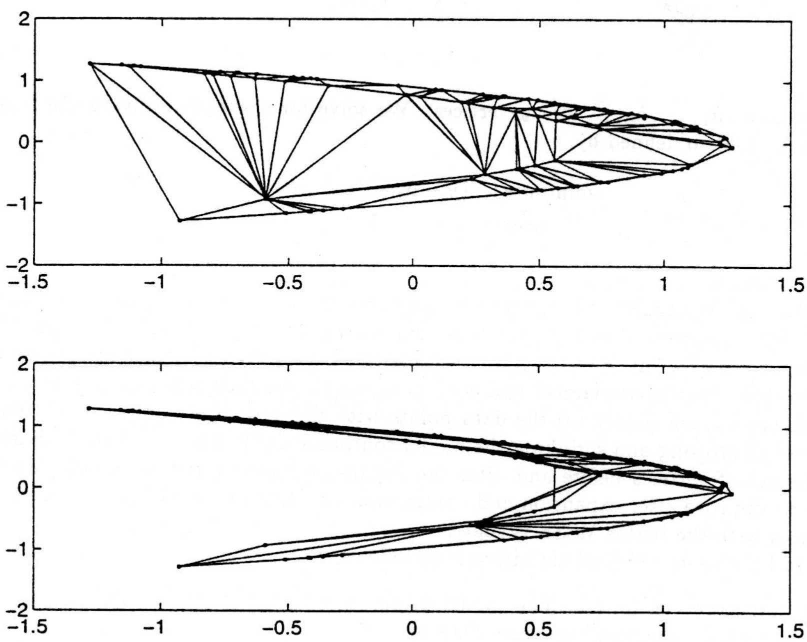

Figure 1: (a) The Delaunay triangulation of some points on an embedding of the x_1 coordinate of a trajectory from the the Henon map. (b) The image of the triangulation under the map $\hat{F}$ defined in (4).

map and the image of the triangulation. This picture alone gives us a good feel for the effect of the map, which is to take a region of the plane and bend and compress it onto a roughly horseshoe-shaped region. For more details of applications, see [9, 10, 11].

2.2 Triangulating noisy systems

The approximation defined by (3) assumes that we know the values of the h_i. In the noisy case, if our v_i are data points then we have noisy data values as an approximation to the h_i. If the v_i are not data points, then we have no a priori information about the $h_i = \rho(v_i)$. We *might* choose the set $\{v_i\}$ to be a subset of our (embedded) data points $\{z_t\}$, but there is no requirement to do so. Our modeling problem may now be expressed as the problem of selecting the v_i and the h_i.

Assuming we have chosen the v_i in some way, we wish to choose the h_i so as to give the best approximation to f given $\{z_t\}$ and $\{y_t\}$. To do this, we note that we can define a matrix Λ with elements $\Lambda_{ij} = \lambda_j(x_i)$ and we define the λ by (1) for $j \in N(x_i)$ and $\lambda_j(x_i) = 0$ for $j \notin N(x_i)$. We can then write the approximation as a matrix equation thus:

$$\hat{\rho}(x_i) = \sum_{j \in N(x_i)} \lambda_j(x_i) h_j \tag{5}$$

$$= \sum_j \lambda_j(x_i) h_j$$

$$= (\Lambda h)_i \tag{6}$$

where $h = (h_1, \ldots, h_{n_v})^T$ for n_v vertices. We solve for the h_i by solving the least squares problem defined by

$$\text{minimise}: \quad (y - \Lambda h)^T (y - \Lambda h) \tag{7}$$

$$\text{over}: \quad h \tag{8}$$

where $y = (y_1, \ldots, y_n)^T$.

All that remains then is to describe a method for choosing the $\{v_i\}$. To do this, we utilize a heuristic based on expansion and contraction of the model. We allow the model to grow by, say, k vertices, by selecting the k data points with the worst fitted data values, that is, the largest values of $|y_t - \hat{f}(x_t)|$. We then remove the k vertices which are (or are closest to) the data points with the *best* fitted data values. This process of growing and shrinking the model continues until either their is no change in the set of vertices before and after the expansion and contraction or the MSSE fails to improve after expansion and contraction (in which case, the previous set of vertices with the better MSSE is kept.)

In brief then, the final algorithm is as follows:

1. Construct an initial affine model using $(n + 1)$ vertices which form a single simplex containing the entire data set.

2. Select the m data points with the worst fitted data values. Bring these points into the triangulation.

3. Select the m data points with the best fitted data values. Remove these points from the triangulation.

4. If the triangulation has changed and the MSSE has improved, go back to 2.
5. Calculate the minimum description length for the current model size.
6. Have we found a local minimum of description length (as a function of number of vertices)? If so, stop now.
7. Increase the model size by one by bringing the worst fitted point into the triangulation and goto 2.

We check for the existence of a (local) minimum of description length as a function of model size (number of vertices) simply by storing the model with the smallest description length to date and declaring it to be a local minimum if we have not found a smaller value after a certain number of increases in model size.

3 Which model is best?

3.1 Model Selection Criteria

Once we are able to construct a model of a given size (that is to say, of a given number of parameters), we must decide how to choose between different sizes. There are various criteria in general use, such as the Akaike Information Criterion (AIC) and the Bayesian Information Criterion (BIC). In this work however, we utilize the criterion of Minimum Description Length (MDL) as described by Rissanen [15]. The description length of a data string is essentially the length of code required, in some optimal coding system, to completely describe the data in terms of a given model. The application of MDL to model selection is discussed in some detail in [7].

3.2 Description Length of Triangulation Models

In order to encode our model, we must transmit sufficient information for the receiver to be able to reconstruct the data completely. One way of doing this is by using what Rissanen calls a "two-part code". Firstly, we encode the information about the residuals of the model as defined by $\epsilon_t = y_t - \hat{\rho}(x_t)$. It is a well-known result of coding theory that one can encode this information in a code of length bounded below by $-\log_2 P(\epsilon)$ bits, where $P(\epsilon)$ is the probability distribution of the residuals, and this length is approachable to better than 1 bit [4]. Secondly, we must encode information about our model. In our case, this means that we must transmit the values of the vertex positions $v_i \in \mathrm{R}^n$ and their heights $h_i \in R$. The values of these parameters are, in principle, known to arbitrary precision and we must truncate them to transmit them in a code of finite length. The important feature of the application of the MDL criterion is that we can optimize over the truncation. That is, we can choose the parameter precisions which give the smallest description length of the data.

In the present case, we *assume* that the residuals have a Gaussian distribution, in order to simplify the calculation of description length. This is a reasonable assumption in light of the fact that we choose the heights h_i by solving a least squares problem as stated above. However, in practice, we find that the residuals are often not sufficiently close to Gaussian and we could improve our models by using a more sophisticated model of the noise.

It is worth noting how the MDL criterion selects the "best" model in some well-defined sense. The term in the description length which describes the residuals will, in general, decrease as the model size increases, due to more accurate fitting of the data. However, as the model size increases, the code needed to describe the model parameters will tend to increase. It is in this way that the MDL criterion selects the model which best achieves a balance between the competing aims of fitting the data as accurately as possible and the desire for parsimony in the model.

4 Results

In this section, we present the results of some preliminary applications of the modeling techniques described here.

4.1 Example of triangulated surface: the Ikeda map

The Ikeda map [3] is a map $\theta : \mathrm{R}^2 \to \mathrm{R}^2$ defined by

$$\begin{aligned} z &= 0.4 - 6/\left(1 + x^2 + y^2\right) \\ \theta(x, y) &= \left(a + b \times (x\cos(z) - y\sin(z))\,, b \times (x\sin(z) + y\cos(z))\right), \end{aligned} \tag{9}$$

where a and b are parameters. For the data used, we have $a = 1$, $b = 0.7$. For this example, we are modeling the x-component of the map. This then gives us a map $\rho : \mathrm{R}^2 \to \mathrm{R}$ which we model by triangulation of the data. The data used to generate this figure was a trajectory of some 500 points with Gaussian noise of standard deviation 0.05 added. Figures 2 and 3 show the surface defined by ρ and an example of the piecewise linear surface which is defined by the approximation $\hat{\rho}$. The model constructed used 25 vertices selected from the data points. Note that although the approximated surface appears to differ considerably from the correct surface, the approximation produces dynamics which are very similar to the original system.

4.2 The Rössler System

The data set chosen was generated by the numerical integration of the equations of motion which produce the Rössler attractor. These equations are

$$\begin{aligned} \dot{x} &= -z - y, \\ \dot{y} &= x + ay, \\ \dot{z} &= b + z(x - c), \end{aligned} \tag{10}$$

where a, b and c are parameters. The original time series consisted of 1000 points of such data. A small amount of normally distributed noise (with a standard deviation of 0.04) was added to this. The data was embedded in 3 dimensions, with a lag of 3, and the prediction "lag" was for one time step. Hence, we are modeling the map f defined by.

$$y_t = f(x_t),$$

where

$$x_t = (y_{t-1}, y_{t-4}, y_{t-7}).$$

The model selected by the MDL principle contained only 7 vertices chosen from the data points, plus 4 vertices placed well outside the convex hull of the data to give

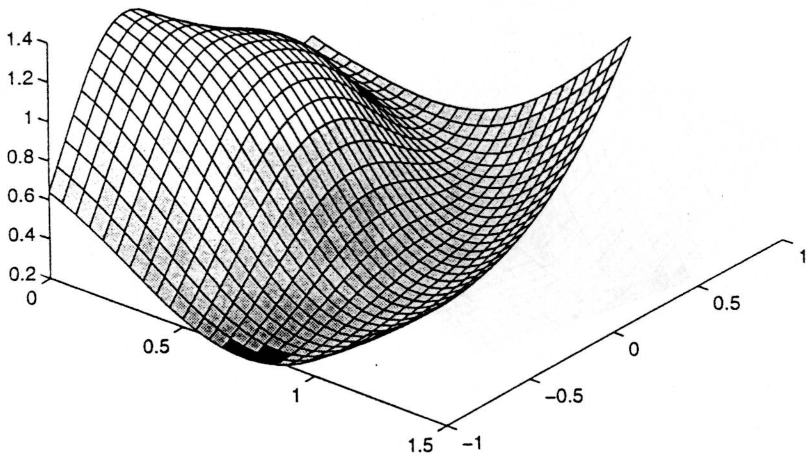

Figure 2: The surface defined by the x-coordinate of the Ikeda map.

an enclosing simplex. Despite this small model (in terms of number of parameters), the model seems to encompass much of the important dynamics of the system. Sample trajectories of the model are shown in figure 4 alongside the trajectory from which the model was constructed. There are two types of trajectories shown, those run without any "dynamical" noise, and those with. The noise added was chosen by random selection from the residuals of the model, hence this type of orbit represents a form of "dynamical bootstrapping." Note that the model reproduces the important folding of the attractor which is the major nonlinearity of this system and the source of the interesting dynamics. In figure 5, we see the accuracy of the model in making one-step predictions.

5 Conclusions

The present paper can best be seen as a research announcement. It is the beginnings of applying to triangulation modeling the methods developed by two of the authors (AM and KJ) for radial basis modeling, as described elsewhere in this volume [6]. A fuller description of this work and, in particular, more detailed applications, will be found in [2].

There are two parts to the modeling process: an easy part, involving linear least-squares fitting (at least when the noise is Gaussian) and a hard part, involving

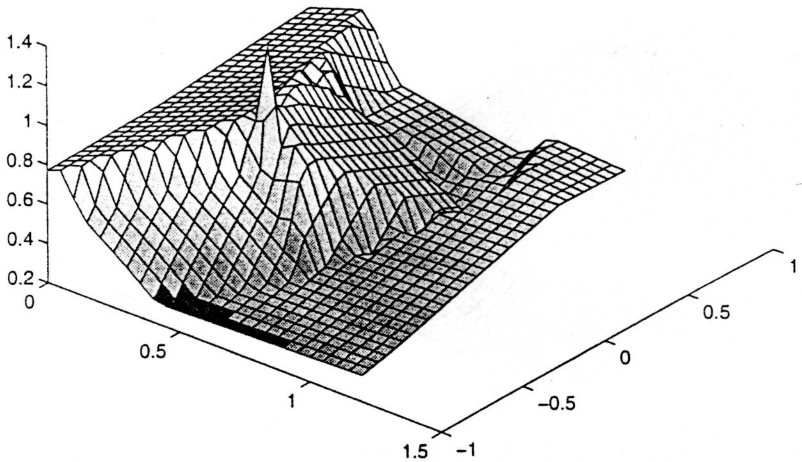

Figure 3: The surface defined by the approximation $\hat{\rho}$ to the x-coordinate of the Ikeda map.

selection of vertex locations. The problem fits into the format we have described elsewhere as *pseudo-linear* modeling [7] and our solution is of the same kind as we used for radial-basis models: we reduce the hard nonlinear part of the problem to a linear (but still NP-hard) subset selection problem which we then solve by a heuristic method which has a history of performing well [13, 12, 7]. In the present case the nonlinear part is the selection of vertices in a triangulation of given size, and we choose some set of candidate vertices such as the data points themselves, then select the subset as described in Section 2.2. The result is a function from R^n to R which models the dynamics of one or more embedded time series.

Selecting the best model from among candidates of different sizes is done using minimum description length, which is described in detail elsewhere. This approach has worked well with other pseudo-linear models of a number of physical systems [8] and we expect it to work well in the present case, though further research is necessary.

The main limitations of the present approach are the inaccuracy of the approximation and the limitations of the noise model used. The former problem is understood, at least in principle, and the main gains are likely to arise from a more sophisticated noise model.