REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	March 1991	memorandum

4. TITLE AND SUBTITLE

Control Algorithms for Chaotic Systems

5. FUNDING NUMBERS

N00014-85-K-0124
N00014-86-K-0180

6. AUTHOR(S)

Elizabeth Bradley

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Artificial Intelligence Laboratory
545 Technology Square
Cambridge, Massachusetts 02139

8. PERFORMING ORGANIZATION REPORT NUMBER

AIM 1278

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Office of Naval Research
Information Systems
Arlington, Virginia 22217

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

AD-A260067

11. SUPPLEMENTARY NOTES

None

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Distribution of this document is unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

This paper presents techniques that actively exploit chaotic behavior to accomplish otherwise-impossible control tasks. The state space is mapped by numerical integration at different system parameter values and trajectory segments from several of these maps are automatically combined into a path between the desired system states. A fine-grained search and high computational accuracy are required to locate appropriate trajectory segments, piece them together and cause the system to follow this composite path. The sensitivity of a chaotic system's state-space topology to the parameters of its equations and of its trajectories to the initial conditions make this approach rewarding in spite

(continued on back)

14. SUBJECT TERMS (key words)

chaos
nonlinear dynamics
control
scientific computation

15. NUMBER OF PAGES 20

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. Z39-18
298-102

Block 13 continued:

of its computational demands. Boundaries of basins of attraction can be breached, vastly altering both global and local convergence properties. Strange attractor bridges can be found that connect previously unreachable points. Examples of both are shown.

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
ARTIFICIAL INTELLIGENCE LABORATORY

A.I. Memo No. 1278 March 1991

Control Algorithms for Chaotic Systems

Elizabeth Bradley

Abstract

This paper presents techniques that actively exploit chaotic behavior to accomplish otherwise-impossible control tasks. The state space is mapped by numerical integration at different system parameter values and trajectory segments from several of these maps are automatically combined into a path between the desired system states. A fine-grained search and high computational accuracy are required to locate appropriate trajectory segments, piece them together and cause the system to follow this composite path. The sensitivity of a chaotic system's state-space topology to the parameters of its equations and of its trajectories to the initial conditions make this approach rewarding in spite of its computational demands. Boundaries of basins of attraction can be breached, vastly altering both global and local convergence properties. Strange attractor bridges can be found that connect previously unreachable points. Examples of both are shown.

Copyright © Massachusetts Institute of Technology, 1991

This report describes research done at the Artificial Intelligence Laboratory of the Massachusetts Institute of Technology. Support for the laboratory's artificial intelligence research is provided in part by the Advanced Research Projects Agency of the Department of Defense under Office of Naval Research contracts N00014-85-K-0124 and N00014-86-K-0180.

This paper was presented in March 1991 at the First European Conference on Algebraic Computing in Control and will appear in Lecture Notes in Control and Information Sciences.

Control Algorithms for Chaotic Systems

Elizabeth Bradley

Department of Electrical Engineering and Computer Science

Massachusetts Institute of Technology

March 27, 1991

Abstract

This paper presents techniques that actively exploit chaotic behavior to accomplish otherwise-impossible control tasks. The state space is mapped by numerical integration at different system parameter values and trajectory segments from several of these maps are automatically combined into a path between the desired system states. A fine-grained search and high computational accuracy are required to locate appropriate trajectory segments, piece them together and cause the system to follow this composite path. The sensitivity of a chaotic system's state-space topology to the parameters of its equations and of its trajectories to the initial conditions make this approach rewarding in spite of its computational demands. Boundaries of basins of attraction can be breached, vastly altering both global and local convergence properties. Strange attractor bridges can be found that connect previously unreachable points. Examples of both are shown.

1 Introduction

This paper presents control techniques that can be applied to chaotic systems. The unique attributes of chaos are exploited by these algorithms in order to perform control tasks that could not otherwise be accomplished. Programs utilizing extensive simulation and global reasoning about state-space features identify nonlinear trajectory segments that can be pieced together into a path between the specified states. The driving concept behind this approach to control of nonlinear and chaotic systems is to combine fast computers with deep knowledge of nonlinear dynamics to improve performance in systems whose performance is rich but whose analysis is mathematically and computationally demanding.

In a nonlinear system, the distance between neighboring state-space trajectories can grow exponentially with time, so small trajectory perturbations can have serious global repercussions. The Voyager missions used Jupiter as a slingshot for just this reason: near the point of closest approach, a small change in angle, via a short rocket burn, drastically changed the spacecraft's overall path in a fashion simply unobtainable in a linear system. Small *errors* in that course correction can, however, have equally dramatic effects. This leverage is the power of and, paradoxically, the difficulty with nonlinearity.

The system's state space is explored for different parameter values. Since nonlinear systems are exquisitely sensitive to these parameters, a small range of parameter variation can give the control algorithm a large range of behaviors to exploit. Taking advantage of this range and understanding these behaviors, the automatic path-finding algorithm selects a set of trajectory segments from the maps and combines them to form a path through the state space between the desired system states. The controller causes the system to follow this path by monitoring the system state and switching parameter values when the segment junctions are reached. Striking results are achieved with this technique: a very small control action, delivered precisely at the right time and place, can accurately direct the system to a distant point on the state space. In one of the examples in this paper, a small control action briefly pushes a system in a counterintuitive direction in order to reach a path that travels directly to the goal state. In another example, an equally small change is used to move a particular state from the basin of attraction of one fixed point to the basin of another.

These techniques can be applied to any system — linear or nonlinear — but chaotic systems have several properties that make them particularly useful from a control standpoint. Trajectories in such systems cover a subset of the state space densely, visiting arbitrarily small neighborhoods of every point in that subset. This denseness has obvious implications for reachability:

chaotic attractors can be used as bridges between otherwise-unconnected points. Furthermore, such attractors contain an infinite number of unstable periodic orbits that can be located and stabilized.

The rest of this paper is organized as follows. Section 2 briefly reviews pertinent aspects of nonlinear dynamics theory. Section 3 outlines the control algorithms that are used here to find and follow paths through state space. Section 4 illustrates the algorithm with several numerical examples and section 5 summarizes the work, its implications and its connections to previous research.

2 Theory

The equations for an n-dimensional nonlinear system can be written

$$\frac{d\vec{x}}{dt} = F(\vec{x}, k_1, \ldots, k_p, t) \tag{1}$$

where $\vec{x}$ is an n-vector whose elements are the system's state variables and F is a nonlinear function of the state $\vec{x}$ and the parameters k_i. Necessary conditions for chaos are, in addition to the nonlinearity of F, that $n \geq 3$ and $F(\vec{x}, k_1, \ldots, k_p, t)$ be non-integrable[9]. The state-space trajectories of a dissipative chaotic system[1] separate exponentially over time and yet remain on a bounded fractal subset of the state space, called a chaotic or strange attractor[15], within which are embedded an infinite number of unstable periodic orbits. The distance between nearby trajectories grows as $O(e^{\lambda t})$, where λ is the largest positive Lyapunov exponent[6] of the system. On a surface of section through the attractor, the unstable periodic orbits appear as fixed points at the intersections of the stable and unstable manifolds of the system (1) above.

The direction and magnitude of the vector field described by the system (1) depend strongly on the equations' parameters. Locally, this sensitivity can be defined as

$$V_{k_i}(\vec{x}, k_1, \ldots, k_p, t) = \frac{\partial F(\vec{x}, k_1, \ldots, k_p, t)}{\partial k_i} \tag{2}$$

Changes in parameter values (Δk_i) also affect the large-scale features of the state-space plot, causing fixed points to split and give birth to other

[1]Hamiltonian or non-dissipative systems do not have attractors, as their equations preserve state-space volumes.

fixed points, limit cycles or chaotic attractors. These topological changes are known as bifurcations. Between bifurcations, the Δk_i can also cause dramatic changes in the position, shape and size of existing attractors.

Both small- and large-scale changes can be exploited by control algorithms if the responsible parameters are accessible. For example, V_{k_i} can be used to identify regions of state and parameter space where small Δk_i have locally large effects. Changes in attractor size or position can make a target state reachable from different areas of state space. Slow, roundabout paths and fast, direct paths between two points are often separated only by a small parameter difference.

If a stable fixed point can be found near the target state for some parameter value, the control problem is more or less solved, provided that the initial state is in its basin of attraction. However, while the stable fixed points of a chaotic system do move about the state space as the parameters are varied, they might only wander over a small region before bifurcating into more complex attractors. In general, it is unlikely that *any* choice of parameter value would place a fixed point near the target state, unless the path of the fixed point were fortuitous. Moreover, convergence to such points is often slow.

If no suitable fixed points exist, stabilization of the system state at a particular point is, in the classic sense, impossible. An alternative control objective is a steady-state orbit that returns to the target point every m cycles. The unstable periodic orbits embedded within a chaotic attractor can be located using the method of Gunaratne *et al*[8]. Points on a section that return to their own small neighborhoods after m piercings are assumed to be very close to m-cycles; averages of tight bunches of such points are taken to be good approximations to unstable fixed points.

These unstable periodic orbits can be stabilized using the control scheme developed by Ott *et al*[13], wherein the system's dependence upon the parameter is linearized about the fixed point on the $n-1$-dimensional surface of section. This works where the linearization is a good approximation: in the $n-1$-dimensional "control parallelogram" around the point, whose size is determined by the control parameter's range, its effects on the orbit and the orbit's unperturbed stability properties. A small change in k causes a $k = k_0$ periodic orbit to return, after m cycles, not to its original coordinates $\vec{\mathcal{P}}_0$, but to some nearby point $\vec{\mathcal{P}}$. The vector $\vec{g}_k$ measures this effect:

$$\vec{g}_k \equiv \frac{\partial \vec{\mathcal{P}}_0}{\partial k}|_{k_0} \approx \frac{1}{k - k_0}(\vec{\mathcal{P}} - \vec{\mathcal{P}}_0) \tag{3}$$

The stability properties are determined by integrating the variations

$$\frac{d\delta_{ij}}{dt} = \sum_k \frac{\partial f^j}{\partial x_k}\delta_{ik}$$

around the orbit. f^j is the j^{th} component of the system equations (1) and the δ_{ik} are variations around $\vec{\mathcal{P}}_0$. The unstable eigenvector $\hat{e}_u$ and eigenvalue λ_u of this matrix, together with the admissible variation of the parameter k^* around k_0 and the vector $\vec{g}_k$, determine P_{c_k}, the size of the control parallelogram, according to

$$P_{c_k} = k^*|(1 - \lambda_u^{-1})\vec{g}_k \cdot \hat{e}_u| \tag{4}$$

Details about the derivation of these formulae are given in [13].

Since the control parallelogram surrounds a point that is embedded within a chaotic attractor, all trajectories will eventually enter the controller's domain, be driven to the orbit, and, in the absence of noise, remain there indefinitely. The denseness of these orbits makes this technique very practical if a chaotic attractor overlapping the target state exists; nevertheless, target acquisition is a problem. The delay before any particular trajectory wanders into the parallelogram is unpredictable, although it does depend stochastically on the ratio of the areas of the parallelogram and of the entire attractor.

3 The Control Algorithm

The following algorithm can be used to find a path between two state-space points **A** and **B** to within a tolerance $\mathcal{T}$. The system is stabilized either at the target state or upon a nearby periodic orbit. Some restrictions on **B** do apply; these are discussed at the end of the section. This algorithm applies to linear, nonlinear and chaotic systems, but only with the latter can it exploit the dense unstable periodic orbits that are embedded within chaotic attractors. For expositional clarity, this presentation assumes that the system has a single parameter k; more parameters would simply increase the size of the search space and the number of indices needed to keep track of it.

1. Map the state space for different parameter values:
   - Pick an initial range (Δk) and interval $[k_{low}, k_{high}]$ for k.
   - Construct state-space portraits at each $k_{low} + n\Delta k$ for $(k_{low} + n\Delta k) \in [k_{low}, k_{high}]$, using **A** as one of the initial conditions.

- Construct portraits with a smaller Δk in ranges where successive plots exhibit large differences (e.g., large V_{k_i}, stable fixed points near **B**, bifurcations or changes in areas of chaotic zones.)

2. Establish the goal state:

   - Examine the collection of state-space maps and locate a stable fixed point that lies within $\mathcal{T}$ of **B**.
   - Failing that, identify parameter values that create chaotic attractors overlapping **B**. Choose a surface of section $\mathcal{S}$ through **B**. Locate an unstable periodic orbit within $\mathcal{T}$ of **B**. Determine the size P_{c0} of the parallelogram of control around the point where that orbit pierces $\mathcal{S}$.

3. Choose an initial grid size ϵ and find the best (e.g., fastest, shortest euclidean distance) path between the grid squares containing **A** and **B**. This segment can be a portion of any trajectory on any of the state-space maps constructed in step 1. It is designated S^0, starts at S^0_{init} and ends at S^0_{final} with $k = k_0$.

4. Reduce ϵ and find the best path between the grid squares containing **B** and S^0_{final}. Iterate on successively smaller scales until $\epsilon \leq P_{c0}$. Record the k_i value and the starting and ending states S^i_{init} and S^i_{final} of each segment.

5. Similarly, find the best path between S_0 and **A**. The final ϵ depends not on P_{c0} but on the largest positive Lyapunov exponent of the attractor of which S^0 is a segment.

The system state can be caused to evolve along a trajectory consisting of a series of path segments $\{S^0, \ldots, S^l\}$ via the following set of control actions. Because of the recursive, longest-first nature of the path-finding algorithm, the segments are not followed in the order in which they are found, so the list must first be sorted into the proper order $\{S'^0, \ldots, S'^l\}$. Beginning at **A**, the parameter is set to k'_0 to initiate the first segment S'^0 and the state is monitored until $\vec{x} = S'^0_{final}$. The parameter is then changed to k'_1, rerouting the system onto S'^1. Clearly, it is vital that parameter switches take place much faster than the system's time scales. This procedure is repeated through all segments in the path. After the final switch, k is set to the value, determined in step 2, that creates the desired fixed point or unstable periodic orbit; in the case of the latter, the linearized control scheme of [13] is then activated.

This algorithm finds globally good paths that have locally bad segments (e.g., driving east to an airport to catch a westward flight) — the sort of path that purely-local control schemes miss. Even paths between regions of state space that are apparently not connected can be found; the control parameters add dimensions to the space that can open conduits between those regions. The examples presented in the next section illustrate both of these cases.

This particular version of the path-finding algorithm, simplified for presentation, does not apply to nonautonomous systems, problems where the final state is unspecified, or problems that require a specific path to be followed. However, adapting the algorithm to fit these cases requires only simple modifications. Time dependence simply adds a dimension to the problem. Where the state can acceptably settle anywhere in a given range, control problems could be solved with a broadened fixed-point search in step 2. Matching a specified path would require a different criterion for path choice in step 3 — not the fastest or most direct segment, but the best match to a specified path.

Several caveats accompany this method:

- **B** may not fall near an unstable periodic orbit or a fixed point for any value of k, in which case the algorithm will fail. The non-zero fractal dimension of a chaotic attractor and the denseness with which trajectories cover it make the former less likely.
- This method applies to systems of any dimension, but the periodic orbit stabilization method of [13] requires that there be at least as many accessible parameters as there are unstable eigenvalues of the orbit.
- If state variables are not directly accessible, information about the system state must be synthesized from outputs and other accessible signals. Systems in which this is not possible cannot be controlled using this approach.
- Slight timing or parameter value errors (e.g., quantization error) can be magnified exponentially, particularly if they occur at the beginning of a long segment or in an area of large $\frac{\partial F(\vec{x})}{\partial x}$. The property elucidated in the Beta Shadowing Lemma[2] keeps these errors from being truly disastrous, but they still place a fundamental upper bound on realizable path length.

[2]Because "with high probability, the sample paths of the problem with external noise follow *some* orbit of the deterministic system closely"[7] and the deterministic orbit lives on a bounded attractor.

Constructing and examining state-space portraits is time-consuming. This motivates the attempt in step 1 of the algorithm to restrict attention to the useful ones — new maps only being considered useful if their portraits differ from the existing ones, regardless of the k-interval between plots. The region of state-space considered is also restricted: it must be somewhat larger than the bounding box of **A** and **B** in order to allow for locally counterintuitive moves. The size of this region is determined heuristically and can be varied if the algorithm fails to find a path on its first pass. There are many obvious points in this procedure where computation can be traded for accuracy: a smaller Δk and a larger range $[k_{low}, k_{high}]$, location and study of different unstable periodic points to find the one that is closest to the target state or whose control parallelogram is largest, variational analysis around each segment to check whether, for example, the $k = 49.9$ path is better than the $k = 50$ path, etc.

Preliminary versions of these path-finding and control algorithms have been implemented and tested. The programs are written in the Lisp dialect Scheme[14] and run on an HP series-300 workstation. State-space portraits are computed from a given set of system equations using a fourth-order Runge-Kutta adaptive time-step integrator. Trajectories are indexed on a state-space grid and represented as lists of the grid squares that they enter. The grid size ϵ is manipulated as part of the search in step 3 of the path-finding algorithm.

The next section of this paper illustrates these techniques with several numerical examples.

4 An Example

The Lorenz equations[11] are:

$$F(\vec{x}, a, r, b) = \begin{bmatrix} \frac{dx}{dt} \\ \frac{dy}{dt} \\ \frac{dz}{dt} \end{bmatrix} = \begin{bmatrix} a(y - x) \\ rx - y - xz \\ xy - bz \end{bmatrix} \tag{5}$$

These well-known equations approximately describe convection in a sheet of fluid heated from below. The state variable x is proportional to convection intensity; y and z quantify temperature variations. a and r are physical parameters of the fluid — the Prandtl and Rayleigh numbers — and b is an aspect ratio. Numerical integration of these equations from an initial state

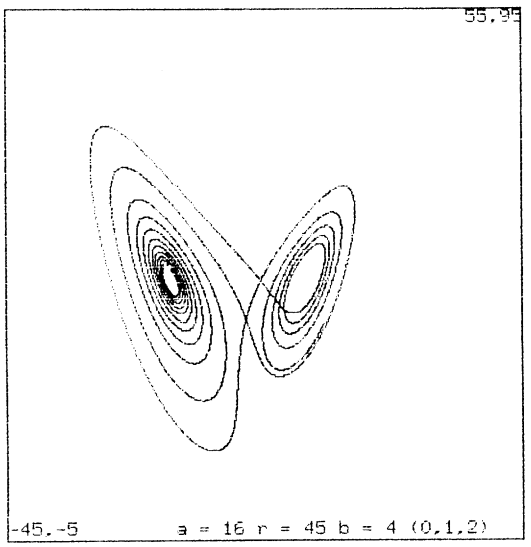


Figure 1: Lorenz Attractor for a = 16, r = 45 and b = 4

$\vec{x}_0 = (x_0, y_0, z_0)$ yields a time-parametrized trajectory $\vec{x}(t)$ in state space. F is dissipative and nonlinear. It is also non-integrable for some parameter values; for these values, its trajectories converge to a chaotic attractor. A particular case of this, shown in figure 1, occurs for $a = 16$, $r = 45$ and $b = 4$. As this figure is an $x - z$ projection of a three-dimensional object, the apparent trajectory crossings do not represent uniqueness violations. The parameter values and the initial state are shown at the bottom of the figure. The values in the upper right and lower left corners are normalized axis coordinates. An $x - z$ section[3] of the same attractor at $y = -15$ is shown in figure 2. More details about the structure and properties of Lorenz attractors may be found in [19].

Consider the task of navigating between the two points marked by crosses in figure 3, starting at the rightmost (**A**) and ending at the leftmost (**B**.) On the axes of the figure, the coordinates (x, y, z) of these points are (8, 29, 64) and (-24.5, -20, 68). r is used as the control parameter and a and b are fixed. We make no assertions about whether changing this parameter is either physical or practical; this is purely a mathematical example[4].

As set out in step 1 of the algorithm in section 3, the state space is mapped with an initial r-step of five and an r-range of [20,60]. Portraits are

[3]Points appear on this section where the trajectory pierces the $y = -15$ plane with a positive velocity.

[4]Lorenz himself explored the parameter space outside the range ($r \approx 1$) within which the equations are considered to be an accurate physical model.

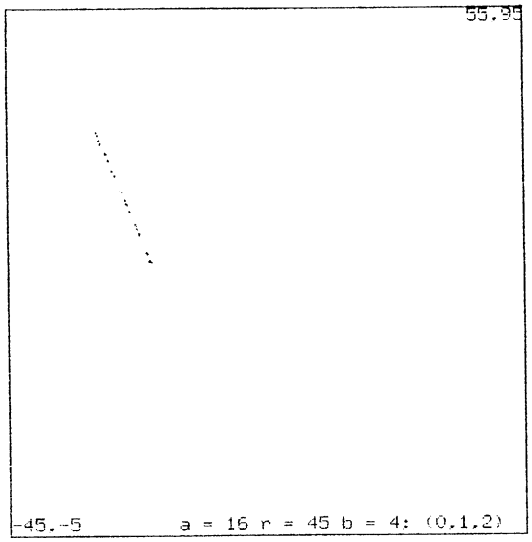


Figure 2: x-z Section of Lorenz Attractor for a = 16, r = 45 and b = 4

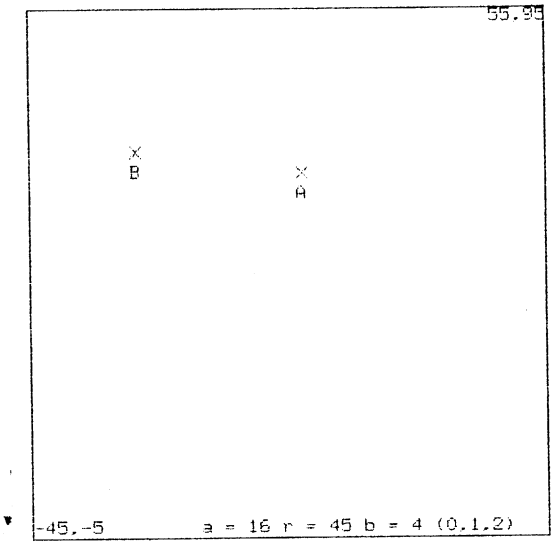


Figure 3: Origin and Destination Points

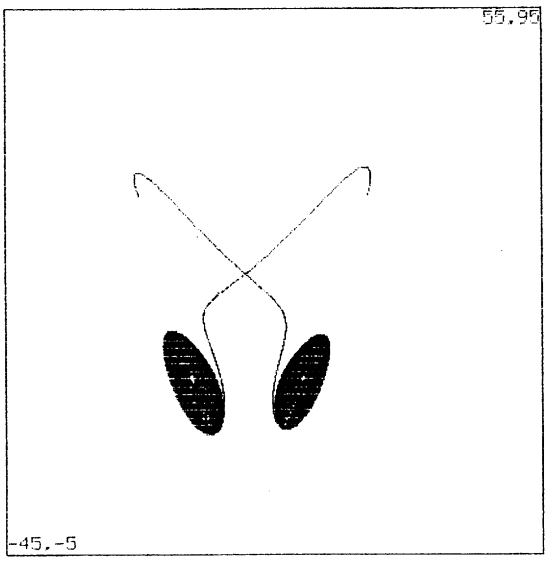


Figure 4: Two Stable Fixed Points

also constructed on a finer grain ($\Delta r = 1$) near, for example, the bifurcation at $r = 24$ that creates a chaotic attractor and in the range where that attractor expands to cover **B**.

Proceeding to the first part of step 2, the maps are examined for stable fixed points of the system (5) at or near **B**. The Lorenz attractor collapses to a pair of stable fixed points at low r, as demonstrated by the $r = 20$ trajectories in figures 4(a) and 4(b). Note that the two starting points are in different basins of attraction. Both of the fixed points — whose coordinates are (8.72, 8.72, 19.00) and (-8.72, -8.72, 19.00)[5] — move as r is varied, but neither approaches **B** for any value of r.

Since no appropriate fixed points exist, the second part of step 2 indicates that the maps should be examined for nearby chaotic attractors and unstable periodic orbits. Figure 5 shows a trajectory for $r = 30$, just above the bifurcation that changes the fixed points of figure 4 into a chaotic attractor. Neither **A** nor **B** happens to lie upon this particular attractor, so it cannot be used as a bridge between them or as a source of nearby unstable periodic orbits. As r is raised further, however, the attractor expands; one of its lobes overlaps **B** when $r = 42$ (figure 6). For $r = 50$, an unstable seven-cycle is found on the $y = -20$ $x - z$ surface of section at $\vec{\mathcal{P}}^7$ = (-24.673, 68.207). This orbit is found by computing 3000 piercings of the $y = -20$ plane, identifying, sorting into bunches and averaging those that return after m cycles, and then choosing the one closest to **B**. We assume, for the purposes

[5]The symmetry is not coincidence; see [19] for details.

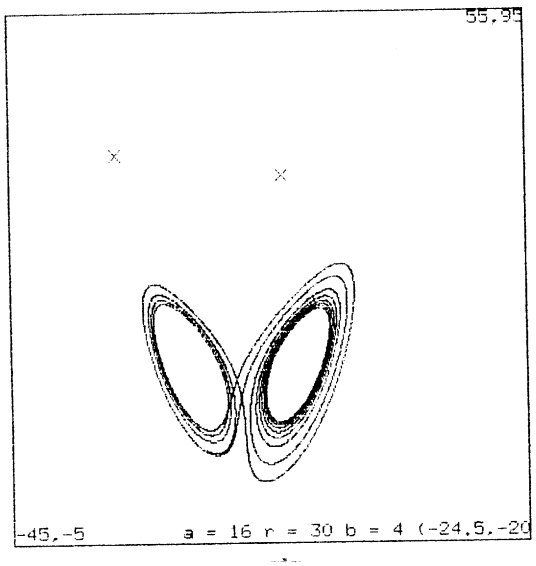


Figure 5: Lorenz Attractor for a = 16, r = 30 and b = 4

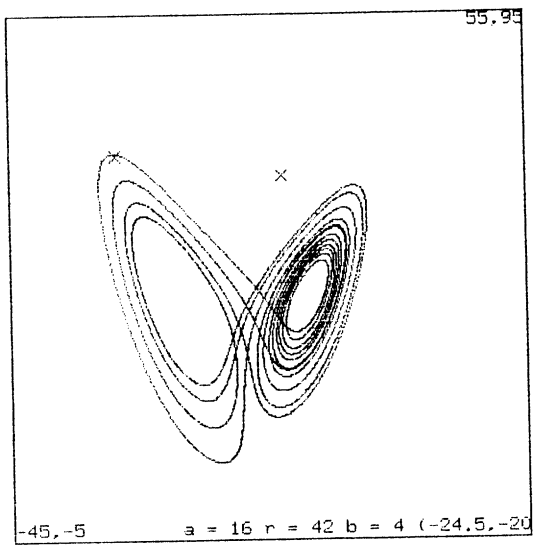


Figure 6: Lorenz Attractor for a = 16, r = 42 and b = 4

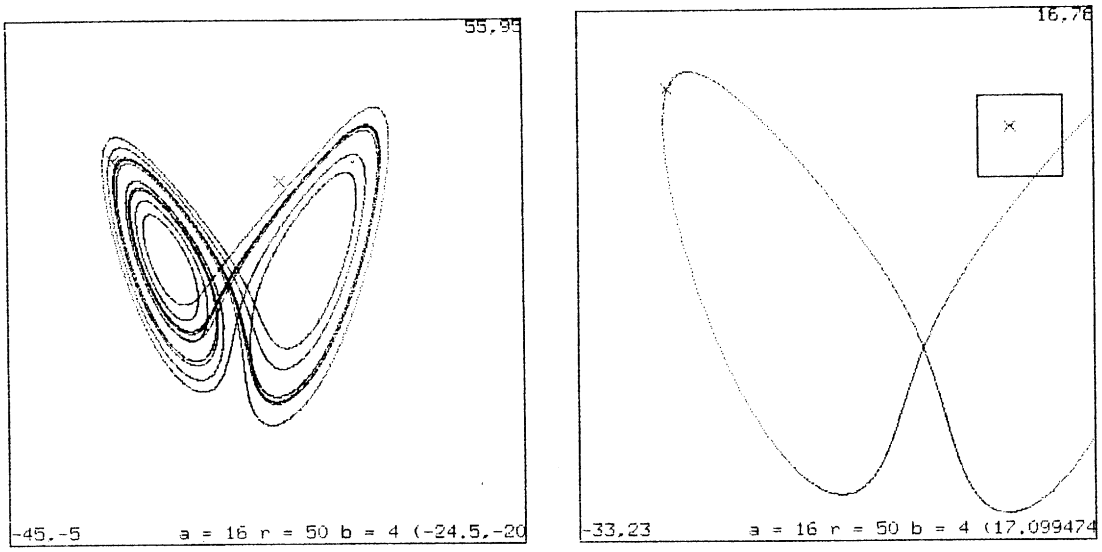


Figure 7: Lorenz Attractor for a = 16, r = 50 and b = 4

of this problem, that errors smaller than 1% of the total signal amplitude $\sqrt{x^2+y^2+z^2}$ are admissible, so $\vec{\mathcal{P}}^7$ meets the specifications. The eigenvalues of the variational system integrated around the seven-cycle are $\lambda_u = 0.734$ and $\lambda_s = -502.077$, with the associated eigenvectors $\hat{e}_u = 0.834\hat{x} + 0.552\hat{z}$ and $\hat{e}_s = 0.685\hat{x} + 0.728\hat{z}$. A 1/2% change in r causes the point $\vec{\mathcal{P}}^7$ to return not to itself (-24.673, 68.207), but to $\vec{\mathcal{P}}^{7'}$ =(-23.929, 66.498), so

$$\vec{g}_r \equiv \frac{\partial \vec{\mathcal{P}}^7}{\partial r}|_{r=50} \approx \frac{1}{\Delta r}(\vec{\mathcal{P}}^{7'} - \vec{\mathcal{P}}^7) = 70.8\hat{x} - 170.9\hat{z}$$

The allowed range of variation of r, for a 1% control tolerance, is ± 0.0075. Using these values in equation (4), we obtain $P_{c_r} = 0.096$ to complete step 2.

Using an initial choice of $\epsilon = 20$, the collection of portraits is examined, as described in step 3, for paths between the grid squares containing **A** and **B**. The $r = 50$ portrait of figure 7(a) contains one such segment, hereafter designated S^0. An edited version of this portrait, plotted on a smaller region around the two points and showing only the useful part of the trajectory, is shown in figure 7(b). S^0 pierces the $y = -20$ plane at (-24.51, -19.99, 68.05), which is actually within the control parallelogram, so no additional path segments need be found to connect S^0 and **B**.

To find trajectory segments that connect point **A** to the other end of S^0, step 4 is iterated in the region outlined by the square in figure 7(b). The nature of the projection makes the distances deceptive; **A** is actually quite far above the nearest threads of the $r = 50$ attractor. The segment S^1, found

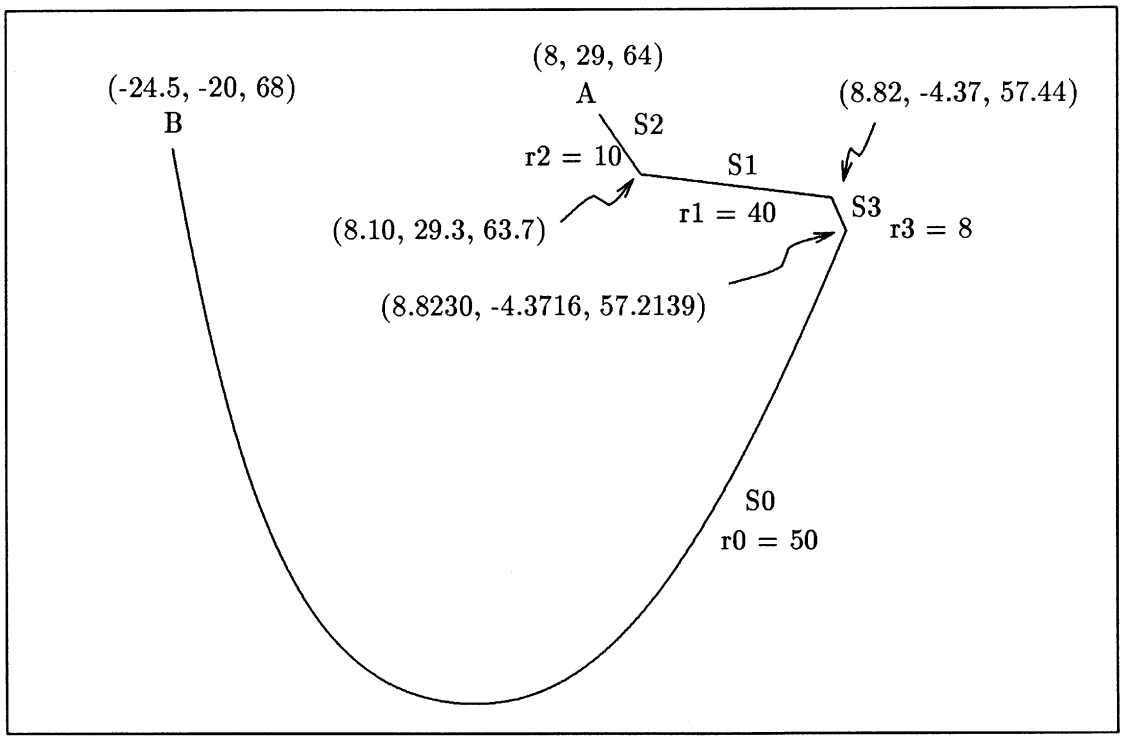


Figure 8: Schematized Segmented Path

on the $r = 40$ map with $\epsilon = 1$, spans most of the distance from **A** to S^0. One more iteration is required on a yet-finer scale ($\epsilon = 0.0001$) to find the two very short segments S^2 and S^3, at $r = 10$ and $r = 8$, needed to connect S^1 to **A** and to S^0. Since S^0 actually enters the control parallelogram around **B**, this process of connecting to it amounts to a solution to the "target acquisition problem" of [13].

A schematized version of the overall path $\{S^2, S^1, S^3, S^0\}$ is shown in figure 8. It is composed of the four segments discussed in the previous two paragraphs. The two longer segments are segments of chaotic attractors; the two shorter ones, enlarged so as to be visible, are sections of transient trajectories ultimately destined for one of the system's fixed points. The former are examples of a "strange attractor bridge", connecting two otherwise-unconnected points. The values for the r_i and the S^i_{final} are shown next to the segments and the transition points where they meet. The actual numerical integration of the path from **A** to **B** along $\{S^3, S^1, S^4, S^0\}$ is shown in figure 9. On this scale, the smaller connecting segments are invisible. Note that S^1 actually moves the system state *directly away from* **B**. This locally

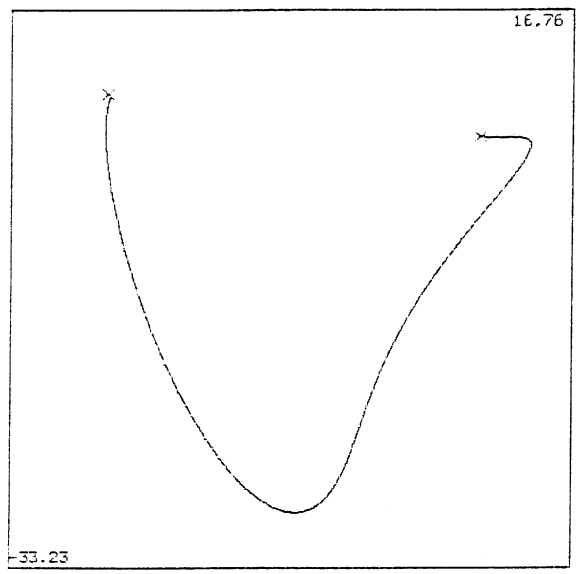


Figure 9: Numerical Integration of Segmented Path

counterintuitive move is made in order to reach a globally good path.

The control program causes the system to follow this segmented path by monitoring the system state and switching the parameter accordingly. The transition that initiates S^0 is the most critical of the three, as nonlinear expansion along its great length can severely magnify any error; this is where the four decimal place accuracy becomes important. The path length between **A** and **B** is 130.1 normalized distance units, requiring 0.3567 normalized time units to traverse. The contrast to the case *without* active target acquisition (see [13]) is striking: if a trajectory is started at **A** and simply allowed to evolve with $r = 50$, it enters the control parallelogram around **B** after traveling 25223 normalized distance units around the attractor in 104 normalized time units. See figure 10.

If **B** were near one of the system's low-r fixed points, the first part of step 2 would succeed. Though a single segment might then converge to the desired state, use of a segmented path can alter macroscopic quantities like convergence speed and reachability. For example, a trajectory starting from the point (22.4, 30.5, 60) at the value $r = 25$ would normally converge to the left hand fixed point (-9.80, -9.80, 24.00) along the tightly-wound spiral at the bottom left of figure 11. The right-hand path in the figure was found by a single pass of the first two steps of the algorithm with $\epsilon = 5$. This path contains two segments: an $r = 60$ trajectory that travels most of the way from (22.4, 30.5, 60) to the *other* fixed point, which lies at (10.25, 10.31, 26.77), and a short section of the $r = 25$ spiral that surrounds the second fixed point. Not only has this manipulation allowed the trajectory to jump the basin boundary and converge to the opposite fixed point, but it has also

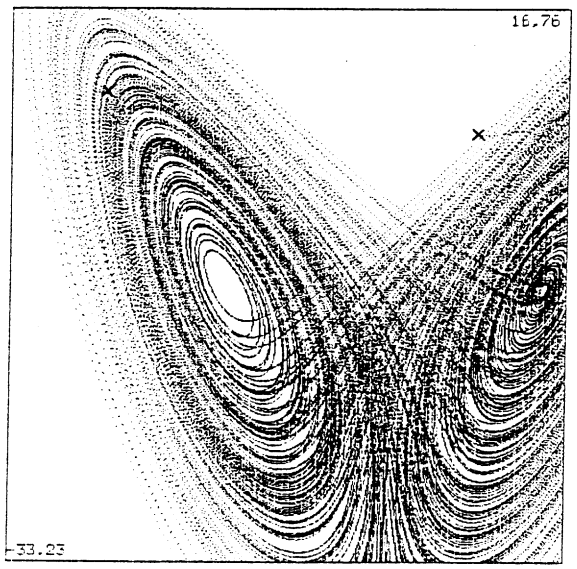


Figure 10: Without Target Acquisition

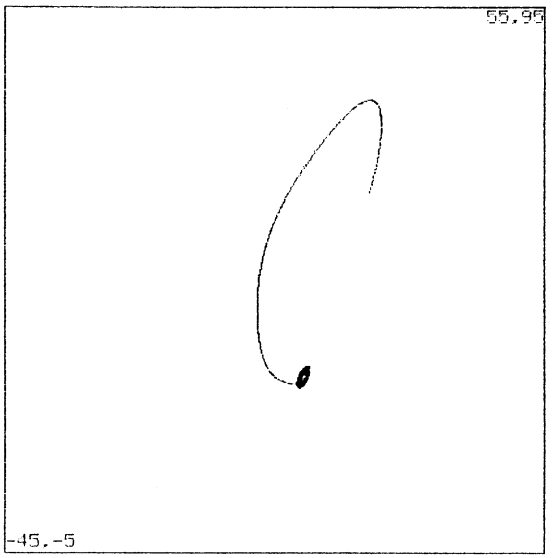


Figure 11: Segmented Path to a Fixed Point

bypassed much of the slowly-converging spiral around that point.

5 Conclusion

The state-space features of chaotic systems are strongly affected by parameter values. Moreover, the trajectories that make up those features separate exponentially over time. Small changes in parameters or state can thus have large and rapid effects; this leverage is a powerful tool for control algorithms. In this paper, we have indicated how fast and accurate computation can be used to synthesize paths through a chaotic system's state space that exploit this leverage to accomplish otherwise-unattainable control tasks. Nonlinear dynamics provides the mathematical tools used to choose values, tolerances, heuristics and limits for the algorithms that select and piece together trajectory segments to create these paths. The Lorenz system is only one of the many systems to which these techniques can be applied. Other examples, with perhaps more practical benefits, are phase-locked loops[4], robot manipulators and spacecraft controllers[18], and single and double pendulums[5]. This brief list is far from complete.

The algorithm is not yet fully automated; for example, the determination of when an attractor "covers" a point was made by eye. Algorithms that produce qualitative descriptions of state space, like the Bifurcation Interpreter[1], KAM[20] or PLR[16] will be involved in the ultimate mechanization of this. Mathematical models are currently used to construct state-space portraits; errors in these models can cause spectacularly bad control decisions. Experimentally exploring and mapping the state space of a physical system [10] would probably be much faster and would also obviate modeling error. Although these techniques have, to date, only been applied to simulated devices, the ultimate aim of this project is to control actual physical systems. Instrumented versions of several chaotic systems — a double pendulum, a driven single pendulum, and several phase-locked loops — have been built. The I/O channel that transmits the state information and control parameter values between controller and system is under construction. These tools will be used to obtain experimental verification of the results presented in this paper.

In the most general terms, the implications of this work are that:

- A broader view and understanding of chaotic state-space features and the effects of parameters upon them is a powerful tool, but its application requires great computational effort.

- Understood and controlled, chaotic behavior can be profitably used to improve a system's design and performance.

This approach can be thought of as a new flavor of adaptive control — one that takes a global viewpoint and eschews almost all linearization[6]. It extends the active use of chaos in control, which presently consists of (1) using knowledge of chaotic zones' boundaries to site an operating point in the middle of the widest part of a system's largest stable zone to maximize noise immunity[17] and (2) the recent work on unstable periodic orbits [13] discussed at various points in this paper. The main difference between this work and [13] lies in the breadth of the aims: we wish, rather than to stabilize a system on a particular type of orbit, to navigate dynamically through *all* of state space with minimal restrictions. This project also extends the ongoing program of research in our group[3] with the overall goal of investigating the use of combined numerical and symbolic methods in scientific and engineering computing.

The complexity of the tasks that are executed by this control program and the accuracy with which it must perform make computation speed a vital issue. The program must compute, store, search through and recognize features in a large number of intricate state-space portraits whose topologies are extremely sensitive to parameter variations. At the same time, many computational approximations are out of bounds because the small errors that they introduce can be amplified exponentially. Physical constraints require that control actions take place about an order of magnitude faster than the actions of the system itself, which further exacerbates the demands on the program's speed. All of this work is worth it: allowing a system to operate in its chaotic regimes opens up new possibilities for better designs. Faster computers[2] or special-purpose hardware might be part of the ultimate solution, together with the understanding and algorithms gained in the course of this research, to attaining novel and effective control of useful systems via knowledge of nonlinear theory and intensive computation.

[6]Some authors in the adaptive control literature have hinted in the direction of controlling chaos, but they fall short of active pursuit of the idea, admitting only that "chaotic parameter estimates are not necessarily a bad thing to have"[12].

Acknowledgements

I would like to thank Hal Abelson for encouragement, proofreading and suggestions that vastly improved the content and presentation of this paper.

This report describes research done at the Artificial Intelligence Laboratory of the Massachusetts Institute of Technology. Support for the laboratory's artificial intelligence research is provided in part by the Advanced Research Projects Agency of the Department of Defense under Office of Naval Research contracts N00014-85-K-0124 and N00014-86-K-0180.

This paper was presented in March 1991 at the First European Conference on Algebraic Computing in Control and will appear in Lecture Notes in Control and Information Sciences.

References

[1] H. Abelson. The Bifurcation Interpreter: A step towards the automatic analysis of dynamical systems. *International Journal of Computers and Mathematics with Applications*, 20:13, 1990.

[2] H. Abelson, A. A. Berlin, J. Katzenelson, W. H. McAllister, G. J. Rozas, and G. J. Sussman. The Supercomputer Toolkit and its applications. Technical Report AI Memo 1249, M.I.T. Artificial Intelligence Lab, July 1990.

[3] H. Abelson, M. Eisenberg, M. Halfant, J. Katzenelon, G. J. Sussman, and K. Yip. Intelligence in scientific computing. *Communications of the ACM*, June 1989.

[4] G. Bernstein, M. A. Lieberman, and A. J. Lichtenberg. Nonlinear dynamics of a digital phase-locked loop. *IEEE Transactions on Communications*, 37:1062, 1989.

[5] E. Bradley. *Taming Chaotic Circuits.* PhD thesis, M.I.T., May 1990. Proposal.

[6] R. L. Devaney. *An Introduction to Chaotic Dynamical Systems.* Benjamin/Cummings, Menlo Park, CA, 1986.

[7] J.-P. Eckmann. Roads to turbulence in dissipative dynamical systems. *Reviews of Modern Physics*, 53:643, 1981.

[8] G. H. Gunaratne, P. S. Linsay, and M. J. Vinson. Chaos beyond onset: A comparison of theory and experiment. *Physics Review Letters A*, 63:1, 1989.

[9] R. H. G. Helleman. Self-generated chaotic behavior in nonlinear mechanics. *Fundamental Problems in Statistical Mechanics*, 5:165, 1980.

[10] M. Lee. Summarizing qualitative behavior from measurements of Nonlinear Circuits. Technical Report AI-TR 1125, M.I.T. Artificial Intelligence Lab, May 1989.

[11] E. N. Lorenz. Deterministic nonperiodic flow. *Journal of the Atmospheric Sciences*, 20:13, 1963.

[12] I. M. Y. Mareels and R. R. Bitmead. Nonlinear dynamics in adaptive control: Chaotic and periodic stabilization. *Automatica*, 22:641, 1986.

[13] E. Ott, C. Grebogi, and J. A. Yorke. Controlling chaos. In *Chaos: Proceedings of a Soviet-American Conference.* American Institute of Physics, 1990.

[14] J. Rees and W. Clinger. The revised3 report on the algorithmic language Scheme. *ACM SIGPLAN Notices*, 21:37, 1986.

[15] D. Ruelle. Strange attractors. *The Mathematical Intelligencer*, 2:126, 1980.

[16] E. P. Sacks. Automatic qualitative analysis of ordinary differential equations using piecewise-linear approximations. Technical Report AI-TR 1031, M.I.T. Artificial Intelligence Lab, March 1988.

[17] F. M. Salam and S. Bai. Complicated dynamics of a prototype continuous-time adaptive control system. *IEEE Transactions on Circuits and Systems*, 35:842, 1988.

[18] J.-J. E. Slotine. Putting physics back in control. In J. Descusse, editor, *New Trends in Nonlinear Control Theory.* Springer-Verlag, 1988.

[19] R. F. Williams. The structure of Lorenz attractors. *Publications Mathematiques de l'IHES*, 50:101, 1979.

[20] K. M.-K. Yip. KAM: Automatic planning and interpretation of numerical experiments using geometrical methods. Technical Report AI-TR 1163, M.I.T. Artificial Intelligence Lab, August 1989.

CS-TR Scanning Project
Document Control Form

Date : 10/06/94

Report # AIM-1278

Each of the following should be identified by a checkmark:
Originating Department:

☒ Artificial Intellegence Laboratory (AI)
☐ Laboratory for Computer Science (LCS)

Document Type:

☐ Technical Report (TR) ☒ Technical Memo (TM)
☐ Other:_____

Document Information

Number of pages: 21
Not to include DOD forms, printer intstructions, etc... original pages only.

Originals are:
☒ Single-sided or
☐ Double-sided

Intended to be printed as :
☐ Single-sided or
☒ Double-sided

Print type:
☐ Typewriter ☐ Offset Press ☒ Laser Print
☐ InkJet Printer ☐ Unknown ☐ Other:_____

Check each if included with document:

☒ DOD Form 2(PGS) ☐ Funding Agent Form ☐ Cover Page
☐ Spine ☐ Printers Notes ☐ Photo negatives
☐ Other: _____

Page Data:

Blank Pages(by page number):_____

Photographs/Tonal Material (by page number):_____

Other (note description/page number):

Description : Page Number:

PAGES 9,10,11,12,13,15,16 (IMAGES 10,11,12,13,14,16,17)
HAVE CUT & PASTE FIGS.

Scanning Agent Signoff:

Date Received: 10/06/94 Date Scanned: 10/11/94 Date Returned: 10/13/94

Scanning Agent Signature: Michael W. Cook

Scanning Agent Identification Target

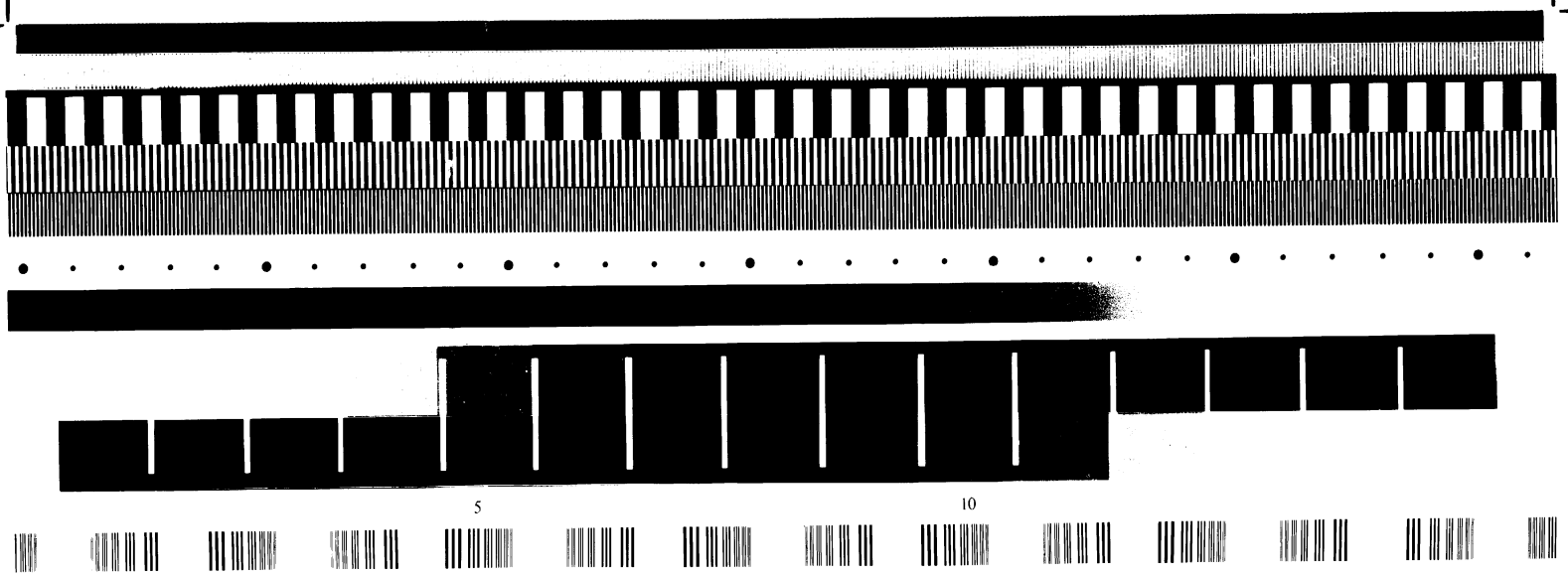
Scanning of this document was supported in part by the **Corporation for National Research Initiatives,** using funds from the **Advanced Research Projects Agency** of the **United states Government** under Grant: **MDA972-92-J1029.**

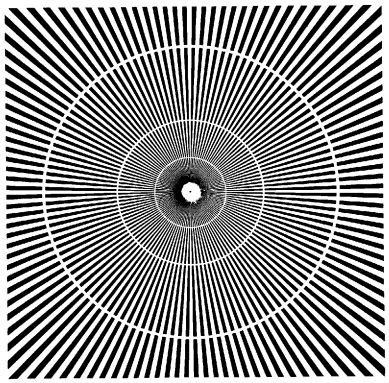
The scanning agent for this project was the **Document Services** department of the **M.I.T Libraries.** Technical support for this project was also provided by the **M.I.T. Laboratory for Computer Sciences.**

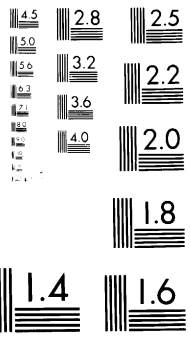
Scanned

Date: 10/11/94

M.I.T. Libraries
Document Services





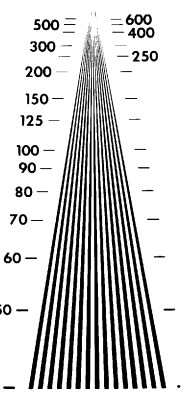


NMA MICROFONT QJKLPYZ
6BS12GH5D4X7U3W8V9E
PQR45DE9UV670FG8STHIJNOWXABYZ
3KLM12C

ABCDEFGHIJKLMNOPQRSTUVW
XYZabcdefghijklmnopqrst
uvwxyz0123456789 OCR-B

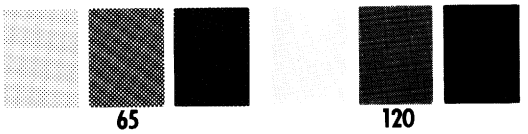
ABCDEFGHIJKLMNOPQRSTUV
WXYZabcdefghijklmnopqr
stuvwxyz1234567890PICA

ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz
1234567890 Elite



ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz
1234567890 Spartan Medium 6 pt

ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz
1234567890 Spartan Medium 4 pt



ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz
1234567890 Spartan Medium 8 pt

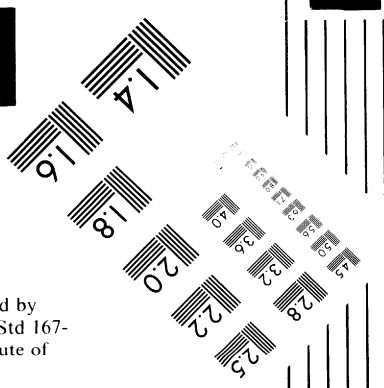
ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz
1234567890 Spartan Medium 10 pt

ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz
1234567890 Spartan Medium 12 pt

IEEE Std 167A-1987

FACSIMILE TEST CHART

Prepared by the IEEE Facsimile Subcommittee and printed by Eastman Kodak Company. Use in accordance with IEEE Std 167-1966, Test Procedure for Facsimile. Copyright 1987, Institute of Electrical and Electronics Engineers.



AIIM SCANNER TEST CHART #2

Spectra

4 PT ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz;:'',../?$0123456789

6 PT ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz;:'',../?$0123456789

8 PT ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz;:'',../?$0123456789

10 PT ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz;:'',../?$0123456789

Times Roman

4 PT ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz;:'',../?$0123456789

6 PT ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz;:'',../?$0123456789

8 PT ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz;:'',../?$0123456789

10 PT ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz;:'',../?$0123456789

Century Schoolbook Bold

4 PT ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz;:'',../?$0123456789

6 PT ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz;:'',../?$0123456789

8 PT ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz;:'',../?$0123456789

10 PT ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz;:'',../?$0123456789

News Gothic Bold Reversed

4 PT ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz;:'',../?$0123456789

6 PT ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz;:'',../?$0123456789

8 PT ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz;:'',../?$0123456789

10 PT ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz;:'',../?$0123456789

Bodoni Italic

4 PT *ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz;:'',../?$0123456789*

6 PT *ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz;:'',../?$0123456789*

8 PT *ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz;:'',../?$0123456789*

10 PT *ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz;:'',../?$0123456789*

Greek and Math Symbols

4 PT ΑΒΓΔΕΞΘΗΙΚΛΜΝΟΠΦΡΣΤΥΩΧΨΖαβγδεξθηικλμνοπφρστυωχψζ≧∓'',../≦±=≠°><∌∢∌≤≡

6 PT ΑΒΓΔΕΞΘΗΙΚΛΜΝΟΠΦΡΣΤΥΩΧΨΖαβγδεξθηικλμνοπφρστυωχψζ≧∓'',../≦±=≠°><∌∢∌≤≡

8 PT ΑΒΓΔΕΞΘΗΙΚΛΜΝΟΠΦΡΣΤΥΩΧΨΖαβγδεξθηικλμνοπφρστυωχψζ≧∓'',../≦±=≠°><∌∢∌≤≡

10 PT ΑΒΓΔΕΞΘΗΙΚΛΜΝΟΠΦΡΣΤΥΩΧΨΖαβγδεξθηικλμνοπφρστυωχψζ≧∓'',../≦±=≠°><∌∢∌≤≡

White

Black

Isolated Characters

e	m	1	2	3	a
4	5	6	7	o	°
8	9	0	h	I	B

A4 Page 6543210

MESH HALFTONE WEDGES

65

85

100

110

133

150

A4 Page 6543210

Scanning Record
Department:ai
Document Number:1278
Type:memo
Pages in Document:\177\17721
Date Scanned:10/11/94
Paper Type:bond
Print Type:laser print
Original input is single-sided:y_ or double-sided:_
Intended output is single-sided:_ or double-sided:y_
Scanner Used:Fujitsu 3096g
Resolution(dpi):400
Greyscale depth(bits):8
Scanner settings:default
Blank pages(page numbers seperated by commas or none):none
Photographs or tonal materials page numbers:none
Page number(s) of color materials:
Cut and Paste material page numbers:9,10,11,12,13,15,16
DoD form present:y
Target IEEE present:y
Target AIIM present:y
Targent Agent ID present:y
Other pages scanned (cover sheet etc...):none