Transactions on Information and Communications Technologies vol 16, © 1996 WIT Press, www.witpress.com, ISSN 1743-3517

A Julia Set Methodology for the Detection of Transient Chaotic Oscillations in Continuous Control Systems

David W. Russell & James J. Alpigini [1]
Department of Electrical Engineering
Penn State Great Valley,
Malvern PA 19355, USA
Email: rzn@psugv.psu.edu

[1] *Lockheed Martin Corporation*
Camden NJ, USA

Abstract

The paper defines and demonstrates the occurrence of chaos in the iteration of mathematical formulae and shows that such systems can be exposed by their dependence on the initial values of variables and parametric coefficients. Filled Julia sets, although computationally intensive, give visual explanation of the fate of an iteration and indicate regions of crossover between stable and unstable operation. The paper then demonstrates that even the simplest process controllers may be subject to chaos in their component parts while still maintaining control of the primary outputs of the system. A variation on the Julia set is then described and offered as a possible "chaotic function analyzer" for such systems. The paper closes with some illustrations of phase plots in the stable and chaotic regions for a nonlinear plant that is examined by the proposed method.

1 Introduction

The term "chaos" has taken on new meaning in the last decade. It is now not so much considered a synonym for random, erratic and unrepeatably confused behavior, but rather as a commonly seen and somewhat predictable dynamic phenomenon. Furthermore, in the past few years, it has been reputed that the trajectories of state variables in many dynamic systems may not only demonstrate unstable, but truly, chaotic motion. In modeling a process it often becomes apparent that certain combinations of values can be shown to drive the system into chaotic motion. What is more surprizing is the fact that the system outputs may remain stable, while the feedback signal is exhibiting chaotic behavior. In other words, a control mechanism may be successfully performing its assigned task while generating wildly varying values of control parameters in order to handle the chaos. Chaotic transients have been detected in real-life process control systems (e.g. Swinney et al [1], Corcoran and Sieradzki [2].) It is the purpose of this paper to study the ease with which chaos can be induced into a system and to describe a methodology that may prove useful in the detection of the possibility of occurrence of chaotic motion. The paper includes sections on the occurrence of mathematical chaos, the familiar Julia set, chaos and process controllers, computer simulations describing a Julia-like method for an example continuous system, and finishes with some concluding comments and a statement of future research

2 Mathematical Chaos

Chaotic behavior can occur in very simple mathematical iterations. Consider the rule: $y = y^2 + c$. In order to perform iterations on this, values for c and y_0 (the initial "seed value" of y) must be known. Figures 1 (a)-(c) show the first sixty iterations of $y = y^2 + c$ for various values of c, with seed $y_0 = -1.0$. The data points are connected purely for clarity of observation. What is immediately observable is that the initial value of y and the parameter c, are highly significant in the *fate* of the iteration. Figure 1(a) shows the fate of the iteration approaching a single value for a value of $c = -0.5$. In Figure 1(b) the fate of the iteration for $c = -1.0$ exhibits two values. In this case the orbit or trajectory of the iteration is said to have *bifurcated*. Figure 1(c) shows the fate of the iteration for $c = -1.9999$. In this case the orbit has become chaotic.

The bifurcations which occur across a range of initial values may be plotted as shown in Figure 2, which shows the fate of the iterations of $y^2 + c$ for $y = y_0 = -1.0$ across a range of c from -2.0 to 0. The plot clearly shows the orbits undergoing bifurcation (period doubling) until the onset of chaos.

The importance of initial conditions in transient behavior is one of several indicators that chaotic processes may be present (Moon [3]) Another facet of chaos is the observation of period doubling, which can often be detected in real systems as unstable oscillation and "non-harmonic ringing". The presence of chaotic elements in a control algorithm has obvious detrimental effects on the dynamic performance of a system. Chaotic values in variables occur by successive entry and redefinition of those variables within a module, which is exactly what happens in an in-line process control system.

Transactions on Information and Communications Technologies vol 16, © 1996 WIT Press, www.witpress.com, ISSN 1743-3517

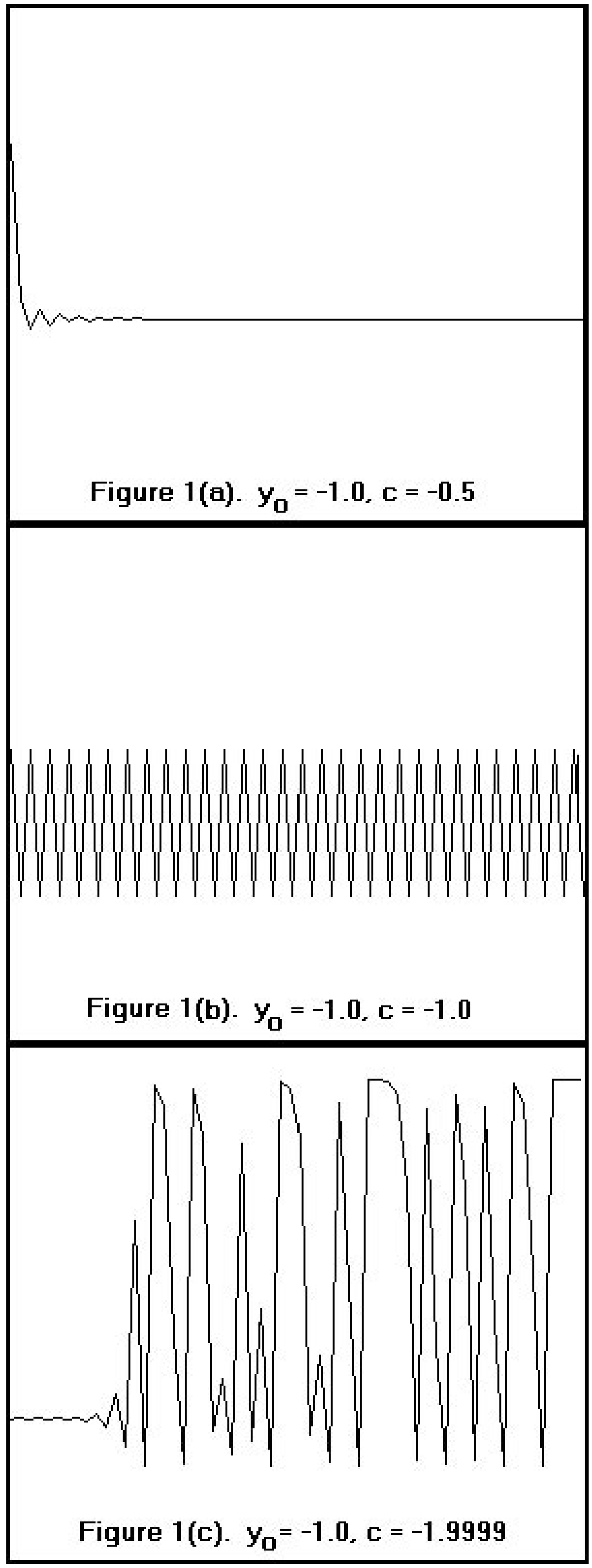

Figure 1 (a)-(c): Iterations of $y = y^2 + c$.

Mareels & Bitmead [4] present some very disturbing and well substantiated findings in this area and conclude:

> "..*under different conditions the adaptive controller gain behaves chaotically but still regulates the plant output*..."

In other words, the plant may be in control even though the control parameters are adapting to inhibit chaotic transients. Consequently, an adaptive control algorithm attempts to mathematically filter out such non-linearities which perhaps explains the irregular forms and trajectories of computed controller outputs generated within the control algorithms themselves, that are required in practice in order to maintain stable operations.

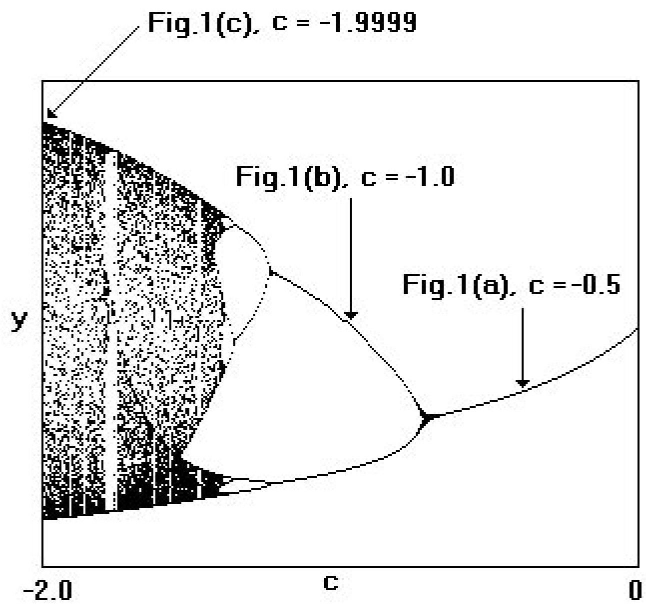

Figure 2: Plot of the orbit fates across a range of c for $y = y^2 + c$, with $y_0 = -1.0$.

3 Julia Sets

The Julia Set provides a graphical representation of the of stable iterative orbits for a given range of some function f. If the variable y in the equation $y = y^2 + c$ is replaced by $\boldsymbol{Z}$, where $\boldsymbol{Z}$ is a complex variable, the "*filled-in Julia set* of f contains the points $\boldsymbol{Z}_0$ for which $\boldsymbol{Z}_n$ stays bounded as [iteration] $n \to \infty$." (Strang [5]) The Julia set gives an immediate, graphical map of the stability of an iterated function across a range of $\Delta\boldsymbol{Z}$. By coloring black the stable orbits of function f, and the unstable or chaotic orbits a different color, it is easy to recognize the stable and chaotic orbits. The process is numerically intensive and requires computational assistance. A typical Julia set may take up to 12 hours on a 486 personal computer and require as many as 790 million calculations.

The Julia set evaluation of the iterative mathematical function $\boldsymbol{z} = \boldsymbol{z}^2 + \boldsymbol{c}$ begins with a the selection of a single $\boldsymbol{c}_{complex} = (\boldsymbol{c}_{\mathrm{real}} + \mathrm{i}\ \boldsymbol{c}_{\mathrm{imaginary}})$ and a range of $\Delta\mathbf{z}_{complex} = (\mathbf{z}_{\mathrm{real}} + \mathrm{i}\ \mathbf{z}_{\mathrm{imaginary}})$ seed values. Each $\mathbf{z}_{complex}$ seed value is mapped to the pixel grid of a computer display so that each pixel in the display grid represents a single value of $\mathbf{z}_{complex}$. The function is then iterated a fixed number of times for each new $\mathbf{z}_{complex}$ seed value (pixel) while holding the value of $\boldsymbol{c}_{complex}$ constant. The trajectory, or orbit of the function, is the value that the iterations approach for each $\mathbf{z}_{complex}$ seed. This trajectory will either remain within some bounds, in which case the associated pixel is colored black, or the orbit will escape, in which case the pixel is set to some other color, typically red. Additional colors may be used to show a range of speeds with which the orbit escapes.

A key component of the Julia set paradigm is the determination of the boundary value of the function that constitutes an escape. For example, the function $\boldsymbol{z}^2 + \boldsymbol{c}$, may be split into its real and imaginary components as follows:

$$z = x + y.\mathrm{i} \tag{1}$$
$$z^2 + c = x^2 + 2.x.y.\mathrm{i} - y^2 + c_{\mathrm{real}} + \mathrm{i}.c_{\mathrm{imaginary}} \tag{2}$$

In order to iterate this function beginning with an initial $z_0 = x_0 + y_0$, the function is expressed as follows:

$$x = x_0^2 - y_0^2 + c_{\mathrm{real}} \tag{3}$$
$$y = 2.x_0.\ y_0 + c_{\mathrm{imaginary}} \tag{4}$$

Where x_0 and y_0 are either the seed values of $\boldsymbol{z}$ or the result of the previous iteration.

Devaney [6], shows that, for this function, the value of $\boldsymbol{z}$ approaches infinity if ever the magnitude of either x or y becomes larger than 2.0. A boundary value of 2 may therefore be set for x and y, with the orbit of the function declared as escaped, or tending toward infinity, if this boundary is exceeded for either value. Figure 3 illustrates the Julia set of $\boldsymbol{z} = \boldsymbol{z}^2 + \boldsymbol{c}$, with a boundary of 2.0 used to determine orbit escape. The areas that are black represent the seed values of $\mathbf{z}_{complex}$ for which the function is bounded and therefore stable. The varying colored shades (gray and white) represent orbits that escaped at different speeds.

4 Chaos and Process Controllers

It is proposed that the Julia set methodology can be adapted for use in the detection of chaos in a process control system. A chaotic control system will, by definition, demonstrate extreme sensitivity to the initial conditions, the values of the plant parameters, or both. It is desirable either to find the plant parameters that provide stability for a given set of initial conditions, or to find the initial conditions that provide stability for a set of plant parameters. Unlike a simple function such as $z^2 + c$, the values in a process control system, such as the output or feedback, may oscillate or have an offset while still remaining stable. For this reason, it was decided that the derivative of a variable was probably a better determinant of orbit escape rather than the variable itself.

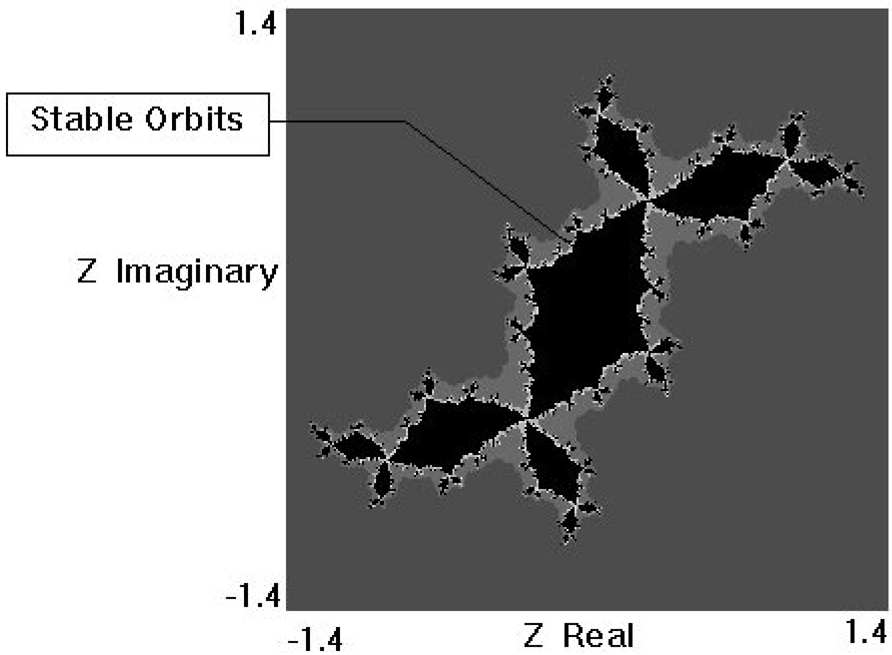

Figure 3: A Julia Set of the complex function $z = z^2 + c$, with $c = 0.1 + 0.8i$, iterated across a Δz range of $-1.4 - 1.4i$ to $1.4 + 1.4i$

Moon [3] identifies *intermittent chaos* as "..periods of regular motion with transient bursts of chaotic motion; where the duration of the regular motion interval is unpredictable." In Russell & Alpigini [7], it was demonstrated that the Julia set methodology could be adapted to give a visual analysis of the motion of the plant values over time for a discreet plant with parameter estimator feedback. In this case, it is demonstrated that the same methodology can be implemented for a continuous system over periods of numeric integration. In the proposed "Julia-like" method, the color map is updated if the any plant variable derivative changes by more than some fixed amount during a time period . In this manner it is possible to record and color-code the transient behavior of the system.

Phase plots of the plant values, derived from control parameters selected from the colored zones in the Julia set, do indeed show chaotic motion in the form of chaotic oscillations, whereas parameters in the black zone give stable performance. The method is demonstrated in the computer simulations that follow.

5 Computer Simulations

Transactions on Information and Communications Technologies vol 16, © 1996 WIT Press, www.witpress.com, ISSN 1743-3517

To demonstrate the use of a Julia set methodology in a continuous control environment, computer simulations were performed for a continuous, nonlinear plant that demonstrates extreme sensitivity to initial conditions. The initial plant, which is illustrated in Figure 4, was proposed by Rubio et al [8], with parametric coefficients fixed at unity and without external perturbations.

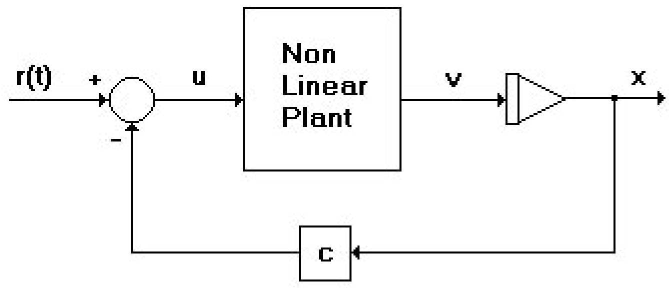

Figure 3: Block diagram of the NonLinear plant used in computer simulation.

In order to demonstrate the Julia set method, the plant is modified to contain an external input of sin(t) and the addition of parametric coefficients a, b and c. The resulting system of equations used in this simulation is:

$$u = a.v^3 - b.v = r(t) - c.x \tag{5}$$
$$v = dx/dt \tag{6}$$

where: r(t) = input, x = output, x, u and v are the state variables, and a, b and c are parametric coefficients.

While holding the plant initial conditions of u_0, v_0, x_0 and c constant, the plant simulation is exercised over a range of Parameter a and Parameter b values. Each a and b value used is mapped to a pixel coordinate on the computer screen. For each set of parameter values, or pixel, the plant algorithm is iterated beginning at the initial (seed) conditions. If the plant orbits remain stable after a fixed number of iterations have occurred, then the pixel is colored black. If the amplitudes of the plant velocity change beyond some fixed amount in a single time unit, a chaotic oscillation or “spike” is declared and the orbit is considered “escaped”. The pixel is then colored according to how many iterations this escape took. The plant values used to determine orbit escape are the rates of change of the state variables u, v and x. A pixel is colored white if the plant is unstable for those parameters. A black pixel indicates that the three state values remain stable, while any other color indicates that at least one plant value evidences chaotic oscillations.

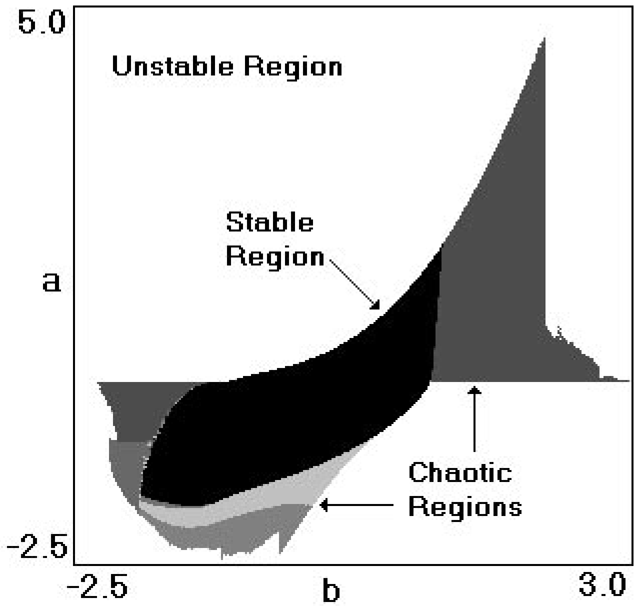

Figure 5: Julia-like Set for the Continuous Plant with initial values of $x_o = v_o = u_o = 0$, c = 1, with Parameter a ranging from -2.5 to 5.0 and Parameter b ranging from -2.5 to 3.0.

Figure 5 is an image of a Julia-like set calculated for this model. The portion which is purely black is a region of stability, where all state variables remain stable. Surrounding the black region are sections which are either white or colored (shaded). The white region is purely unstable, indicating that overflow or other mathematical errors have occurred during iteration. The colored (shaded) regions are chaotic, where the output is controlled, but the feedback elements exhibit chaotic oscillation.

Figure 6 is a normalized phase plot of the plant values of x vs. v, using parameter values selected from a point in the black, stable region. It shows a very well behaved plant. Figure 7 on the other hand, is a normalized phase plot of the plant values of x vs. v, using parameter values selected from a point in a colored (shaded) area of the Julia set. It shows that the the plant values remain bounded, but undergo bifurcations which lead quickly to chaos.

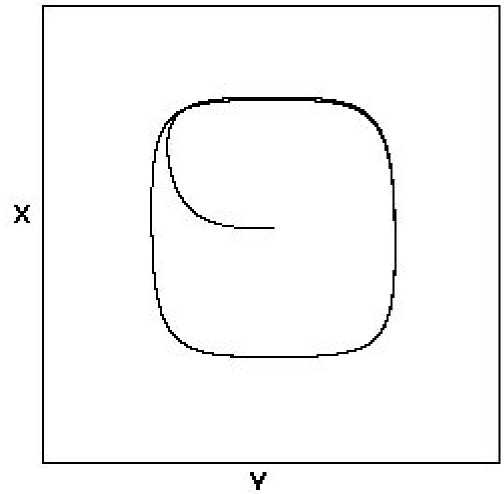

Figure 6: A phase plot of x vs. v selected in the black (stable) region of the Julia like set

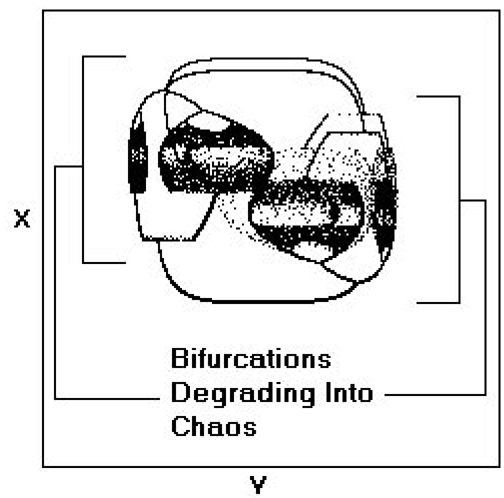

Figure 7: A phase plot of X vs V selected in the colored (chaotic) region of the Julia like set.

6 Conclusion

This paper sounds an alert to process control engineers as to the possible occurrence of chaos in seemingly well-controlled plants. The chaos illustrated in this paper appears to be an propoerty of the iterative nature of the nonlinear plant. However, any methodology that involves models and simulation has latent iterations as part of the numeric processes. With this in mind, it is suggested that chaos may be induced, without detection, in a system and that the net effect is detrimental to the control system's mean-time-to-failure and to its dynamic performance. If the chaotic behavior is intrinsic to the adaptive schema, other non-iterative, possibly AI-based (e.g. Russell [9]), control methodologies may be more desirable. Further research is underway at Penn State Great Valley into the formalization of this "*chaotic function analyzer*" and its import in the control of complex systems, and into how A.I. may provide an inner-control schema that surpresses the development of chaos.

References

1. Swinney H.L., Simoyi R.H. and Roux J.C., Instabilities and Chaos in a Chemical Reaction, *Chaos and Statistical Methods*, ed Y. Kuramoto, pp 244-248, Springer-Verlag. Berlin, Heidelberg, New York, Toronto, 1984.
2. Corcoran, S.G. and Sieradzki., Chaos During the Growth of and Artificial Pit, *Journal of the Electrochem. Soc.*, 139 (6), pp 1568-73, 1992.
3. Moon, F.C., *Chaotic and Fractal Dynamics*, pp 47-53, John Wiley & Sons, 1992.
4. Mareels I.M.Y. & Bitmead R.R., Non-linear Dynamics in Adaptive Control: Chaotic and Periodic Stabilization. *Automatica*, 22 (6), pp 641-655, 1986.
5. Strang, G., *Introduction to Applied Mathematics*, pp 507-510, Wellesley-Cambridge Press, 1986.
6. Devaney, Robert L., *Chaos, Fractals, and Dynamics, Computer Experiments in Mathematics*, pp 86-87, Addison-Wesley, 1990.
7. Russell, David W. & Alpigini, James J., Detection of Chaos in Process Control Systems Using a Julia Set Methodology, *9th International Conf. on Ind & Eng Applications of AI*, Fukoka, Japan, 1996.
8. Rubio, F.R., Aracil, J. & Camacho, E.F., Chaotic Motion in an Adaptive Control System, *International Journal of Control*, V42, No. 2, pp 353-360, 1985.
9. Russell, David W., Further Studies in A.I. Augmented Process Control using the BOXES Methodology, *Artificial Intelligence in Real Time Control*, pp 75-80, Pergamon Press, 1991.