COMPUTER EXPERIMENTS ON ANALOG AND DISCRETE SYSTEM STABILITY

Ravi P. Ramachandran and Jason Childs, Department of Electrical Engineering, Rowan University

Abstract-- **An important concept in the study of both continuous (or analog) and discrete (or digital) control systems is that of stability in that engineers must always design a stable system. The criteria for ensuring stability for an analog system is that the poles lie in the left half plane. For a discrete system, the poles must lie within the unit circle. Checking for stability without factorizing the system polynomial has been a topic heavily researched since factorization is computationally expensive. Stability checks are based on recursive arithmetic operations involving the system polynomial coefficients. The Routh-Hurwitz criterion is used for analog systems and the Jury criterion is used for discrete systems. For students to appreciate the significance of these stability checks, computer experiments have been devised. Students continue to refine their programming skills and apply it to learning about stability. Further depth into the study of system polynomials is achieved by randomly generating polynomials and checking for stability.**

Index terms—**stability, Routh-Hurwitz criterion, Jury criterion, random, control systems, transfer function, impulse response.**

I. INTRODUCTION

A linear system is completely described by its impulse response h (.) in that given any input, the output can be calculated using convolution [1][2]. A continuous time or analog system will have impulse response h(t) and a discrete time or digital system will have impulse response h(n). A system is causal if the output only depends on present and previous inputs and outputs. Causality is necessary for realizing a system and will be assumed to hold. For an analog system, the transfer function is the Laplace transform [1][2] of the impulse response denoted as H(s). For a discrete system, the transfer function is the z-transform of the impulse response denoted as H(z) [3][4][5]. In either case, the transfer function is commonly expressed as the ratio of two polynomials. The roots of the numerator polynomial are the zeros of the transfer function and the roots of the denominator polynomial are the poles of the transfer function. The pole positions reveal whether or not the system is bounded-input, bounded-output (BIBO) stable. Stability in the BIBO sense means that any bounded input results in a bounded output. This is crucial to system implementation since any control system must be BIBO stable. For an analog system, the poles must be within the left half plane in that the real part of every pole must be negative [1][2]. For a digital system, the poles must be within the unit circle in that the magnitude of every pole must be less than one [3][4][5].

In practice, the denominator polynomial of the transfer function is known in terms of the polynomial coefficients only. From the coefficients, the issue of whether the system is stable or not must be determined. An obvious method is to use polynomial factorization to get the poles and then, test for stability. Numerical factorization routines like Laguerre's method [6] or those based on finding the eigenvalues of a Hessenberg matrix containing the polynomial coefficients [6] are computationally intensive. The Routh-Hurwitz criterion [1] for analog systems and the Jury criterion [3][4][5] for digital systems detect the stability purely based on arithmetic operations involving the polynomial coefficients and hence, avoid polynomial factorization. Therefore, both criteria are much less computationally demanding.

II. COMPUTER EXPERIMENTS

Students gain an appreciation for the computational efficiency of the Routh-Hurwitz and Jury criteria by doing the following computer experiment.

1. Write a MATLAB [7] code to implement the Routh-Hurwitz and Jury criterion.

2. Compare the floating point operations (flops) required to check stability by doing root finding and implementing the Routh-Hurwitz and Jury stability checks. The number of flops can be easily obtained in MATLAB.
3. Do the comparison for polynomial orders ranging from 1 to 20.

Students continue to refine their programming skills and apply it to learning about stability. It was found that as the polynomial order increased, the Routh-Hurwitz and Jury stability checks became more and more efficient than root finding. Figures 1 and 2 show the results. In Figure 1, the ratio of the number of flops required for polynomial factorization to the number of flops required for the Routh-Hurwitz criterion is shown for varying polynomial orders. In Figure 2, the ratio of the number of flops required for polynomial factorization to the number of flops required for the Jury criterion is shown for varying polynomial orders.

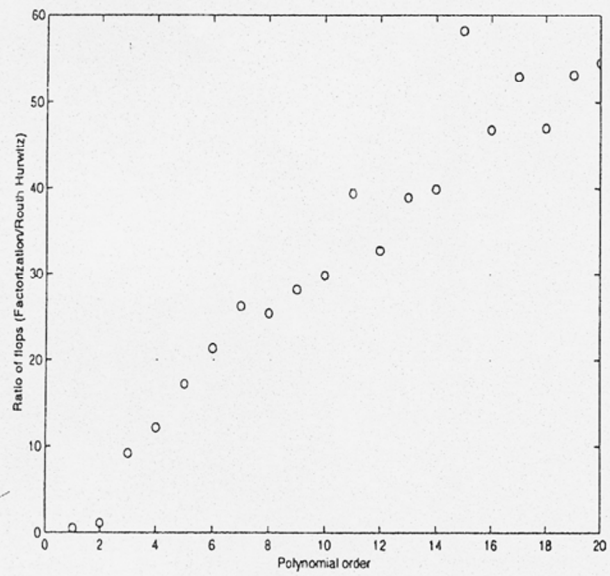

Figure 1 Comparison of polynomial factorization with the Routh-Hurwitz criterion

Further depth into the study of system polynomials is achieved by doing the following for polynomial orders from 1 to 10. The students perform this experiment for an analog system only as the concept easily carries over to digital systems.

1. Randomly generate polynomials and check for stability using the Routh-Hurwitz criterion.
2. From the above, calculate the probability [8] of getting a stable system from a set of random coefficients.
3. Analytically verify the result for orders 1 and 2.

As the polynomial order increases, the probability of getting a stable polynomial decreases. This explains the

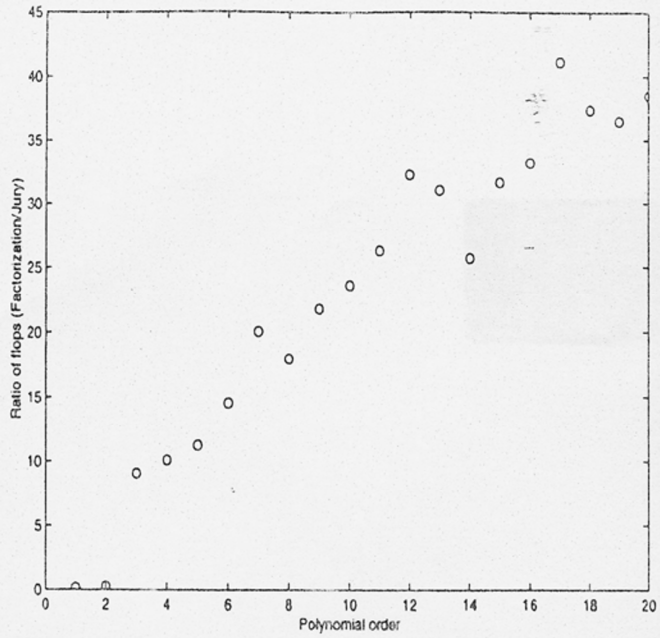

Figure 2 Comparison of polynomial factorization with the Jury criterion

reason why control engineers want to model systems with as low order as possible. A first order system polynomial is given by

$$D(s) = as + b$$

By using the Routh-Hurwitz criterion, the conditions on a and b for left half plane roots which lead to stability are

$$a > 0,\ b > 0 \quad \text{or} \quad a < 0,\ b < 0$$

If the coefficients a and b are uniformly distributed in two dimensional Euclidean space, the probability of randomly generating a stable polynomial is 0.5. A second order system polynomial is given by

$$D(s) = as^2 + b + c$$

By using the Routh-Hurwitz criterion, the conditions on a, b and c for left half plane roots which lead to stability are

$$a > 0,\ b > 0,\ c > 0 \quad \text{or} \quad a < 0,\ b < 0,\ c < 0$$

If the coefficients a, b and c are uniformly distributed in three dimensional Euclidean space, the probability of randomly generating a stable polynomial is 0.25. As the polynomial order increases, there are more nonlinear restrictions on the coefficients thereby diminishing the probability of randomly obtaining a stable polynomial. This probability is computed using computer simulations. One thousand polynomials were randomly generated for each of polynomial orders 1 to 10. For each of the 1000 polynomials, the Routh-Hurwitz criterion was used to test for stability. The number of occurrences of stability divided by 1000 is the numerical calculation of the probability of obtaining a stable polynomial. This probability is computed for polynomial orders ranging from 1 to 10. Figure 3 shows the results.

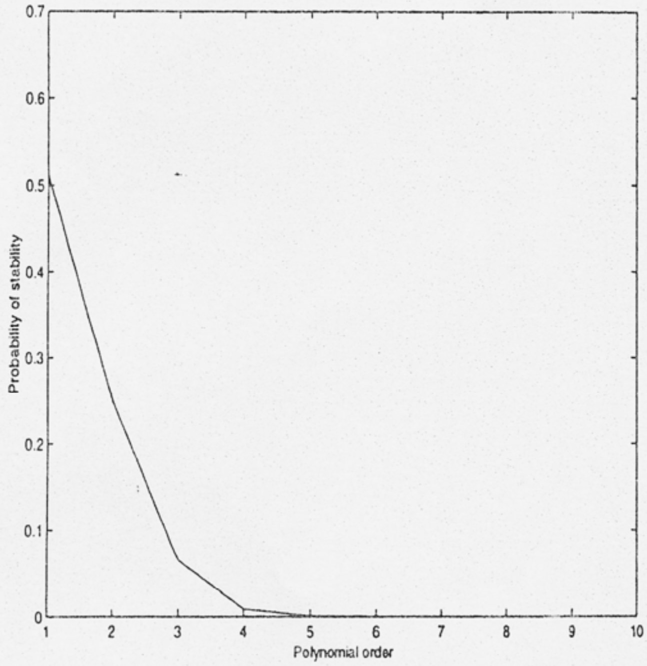

Figure 3 Probability of stability versus polynomial order

It can be seen from Figure 3 that for orders 1 and 2, the probabilities are about 0.5 and 0.25 respectively. This is in accordance with the analytical results. It is also noted that for orders greater than or equal to 5, the chances of obtaining a stable polynomial is practically zero. This experiment was repeated with 6000 randomly generated polynomials. However, the results did not change significantly.

III. SUMMARY

In this set of experiments, students perform and benefit from the following:

1. A study of control system stability in greater depth and an appreciation of stability checks that avoid polynomial factorization.
2. A refinement of programming skills.
3. Application of the knowledge gained in mathematics (particularly probability theory) to observe how stability and polynomial order are related.

IV. REFERENCES

1. R. C. Dorf and R. H. Bishop, *Modern Control Systems*, Addison Wesley Longman Inc., 1998.
2. C. T. Chen, *Introduction to Linear System Theory*, Holt, Rinehart and Winston, 1970.
3. A. Antoniou, *Digital Filters: Analysis, Design and Applications*, McGraw Hill, 1993.
4. E. I. Jury, *Theory and Application of the z-Transform Method,* John Wiley and Sons, 1964.
5. E. I. Jury, *Inners and Stability of Dynamical Systems,* John Wiley and Sons, 1974.
6. W. H. Press, B. P. Flannery, S. A. Teukolsky and W. T. Vetterling, *Numerical Recipes: The Art of Scientific Computing,* Cambridge University Press, 1989.
7. W. J. Palm III, *MATLAB for Engineering Applications*, McGraw-Hill, 1999.
8. S. Ross, *A First Course in Probability*, Macmillan Publishing Company, 1988.