HOW STABLE IS A FUZZY LINEAR SYSTEM?

Hung T. Nguyen[1], Vladik Kreinovich[2]

[1] *Department of Mathematical Sciences, New Mexico State University, Las Cruces, NM 88003*
[2] *Computer Science Department, University of Texas at El Paso, El Paso, TX 79968*

Abstract. One of the main problems to which control is applied is stabilizing a plant. In this case, it is sufficient to consider only small deviations from the ideal state (if the deviations become large, this means that we are not in control anymore). For small deviations, the dynamics of the plant can be described (with great accuracy) by *linear* differential equations. For this same reason, for such problems, an arbitrary control strategy can be approximated by a control that is a *linear* function of the state.

If we know *exactly* the coefficients of the linear equations that describe the plant's dynamics, then we can use the standard methods of control theory to check whether the control makes the plant stable or not. In many real-life cases, however, we do not know the exact values of the coefficients that describe the plant's dynamics. Instead, we have *expert estimates* for these values that can be represented as *fuzzy numbers*. Since we do not know the exact value, we cannot be 100% sure that the plant is stable. *What is the degree of belief that it is stable?*

In the present paper, we:

- describe the relevant mathematical problem;
- (briefly) describe the standard stability techniques of control theory (this part can be skipped by those who know these techniques); and then
- present an algorithm that estimates the desired degree of belief.

In other words, *this algorithm describes how stable is a given fuzzy linear system.*

1. INTRODUCTION TO THE PROBLEM

Why are linear systems important? One of the main problems to which control is applied is stabilizing a plant. In this case, it is sufficient to consider only small deviations from the ideal state (if the deviations become large, this means that we are no longer in control). For the same reason, it is sufficient to consider only small values of control. These restrictions are known to simplify the resulting mathematics.

Indeed, assume that a state $\mathbf{s}$ is described by n numbers: $\mathbf{s} = (s_1, ..., s_n)$. Let's denote the ideal state by $(s_1^{(0)}, ..., s_n^{(0)})$. Instead of s_i, we can use new variables $x_i = s_i - s_i^{(0)}$ to describe a state. The advantage of using new variables is that now, a stable state is described as $(0, 0..., 0)$.

In general, the evolution of a state $\mathbf{x}$ under a control $\mathbf{u}$ is described by a differential equation $\dot{\mathbf{x}} = \mathbf{E}(\mathbf{x}, \mathbf{u})$, where $\mathbf{E}$ is a smooth function (in general, non-linear). Since we are considering states that are close to the equilibrium (0,...,0) (i.e., we are considering small values of x_i), and small values of control, we can expand the function $\mathbf{E}$ into Taylor series, and leave only the main (linear) terms. As a result, we get a linear system of differential equations: $\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}$, where $\mathbf{A}$ and $\mathbf{B}$ are matrices (these matrices are called *constant* because they depend neither on the state $\mathbf{x}$, nor on time t).

Whether a plant is stable or not depends on the control. So, to formulate the question of stability, we must fix some control strategy $\mathbf{u} = \mathbf{U}(\mathbf{x})$ (in other words, we must describe for every possible state $\mathbf{x}$, what control $\mathbf{u}$ to apply). Since we consider only states with small x_i, we can retain only linear terms in the expansion of $\mathbf{U}$, and thus approximate the control strategy by a linear formula $\mathbf{u} = \mathbf{K}\mathbf{x}$ (for some matrix $\mathbf{K}$).

Substituting this expression for control into the equations that describe the plant's dynamics, we get the following system of linear differential equations: $\dot{\mathbf{x}} = \mathbf{C}\mathbf{x}$, where $\mathbf{C} = \mathbf{A} + \mathbf{B}\mathbf{K}$ (this system is called a *closed-loop system.*) Stability means that if we start with a state that is not an equilibrium, then eventually, the plant will return to an equilibrium state $(0, ..., 0)$.

Ideal case: we know the coefficients precisely; how to check whether the plant is stable or not? If we know exactly the coefficients of the system of linear equations that describe the plant's dynamics, then we can use the standard methods of control theory (see, e.g., [1,7]) to check whether the chosen control makes the plant stable or not. These methods are based on the fact that a general solution of a linear system of differential equations with constant coefficients is well known: it is a linear combination of functions of the type $t^k \exp(\lambda t)$, where λ is an eigenvector of the matrix $\mathbf{C}$, and k is a non-negative integer ($k \neq 0$ only if this is a degenerate eigenvalue). For a plant to be stable, we need all these functions to tend to 0 for $t \to +\infty$. From the well-known behavior of an exponential function, we can easily conclude that this occurs if and only if the real parts of all eigenvalues of $\mathbf{C}$ are negative: $Re(\lambda) < 0$.

Eigenvalues are known to be the roots of the polynomial $\det \|\mathbf{C} - s\mathbf{I}\|$ (this polynomial is usually denoted by $p(s) = p_0 + p_1 s + ... + p_n s^n$ [1]). The coefficients p_i of this polynomial can be easily computed if we know $\mathbf{C}$. So, in order to check stability of the plant, it is sufficient to find these roots $s_1, ..., s_n$ and check whether $Re(s_i) < 0$ for all i. Polynomials whose roots satisfy these n inequalities are called *stable* [1].

Interval case: what if we know only the intervals of possible values of coefficients? In real life, we rarely know the exact values of the coefficients that describe the plant's dynamics. In many cases, the plant have been measured, and from these measurements, we can deduce the values of these coefficients. Measurements are never absolutely accurate, there is always some inaccuracy involved. Every measuring device has a guaranteed accuracy (guaranteed by the manufacturer; see, e.g., [3,6]). If we use a device with a guaranteed accuracy Δ to measure a physical quantity x, and the result of this measurement is $\tilde{x}$, this means that the actual value of this quantity x can be any number from the interval $[x^-, x^+]$, where $x^- = \tilde{x} - \Delta$ and $x^+ = \tilde{x} + \Delta$.

So, in these cases, after the measurement, we have intervals of possible values of the components of $\mathbf{C}$. Using these intervals, we can compute the intervals $[p_i^-, p_i^+]$ of possible values of p_i. Based on this information, we would like to know whether we can guarantee the stability of the plant.

One possibility of checking stability is to try all possible values $p_i \in [p^-, p^+]$, and for each of the resulting polynomials, to check whether it is stable. If they are all stable, then the plant is guaranteed to be stable. If one of these polynomials is unstable, then we cannot guarantee stability. This "algorithm" is too time-consuming. A much better method has been proposed in [5] (see also [1]), where it has been proven that it is sufficient to check stability of only 4 polynomials:

$$K_1(s) = p_0^- + p_1^- s + p_2^+ s^2 + p_3^+ s^3 + p_4^- s^4 + p_5^- s^5 + p_6^+ s^6 + ...$$
$$K_2(s) = p_0^+ + p_1^+ s + p_2^- s^2 + p_3^- s^3 + p_4^+ s^4 + p_5^+ s^5 + p_6^- s^6 + ...$$
$$K_3(s) = p_0^+ + p_1^- s + p_2^- s^2 + p_3^+ s^3 + p_4^+ s^4 + p_5^- s^5 + p_6^- s^6 + ...$$
$$K_4(s) = p_0^- + p_1^+ s + p_2^+ s^2 + p_3^- s^3 + p_4^- s^4 + p_5^+ s^5 + p_6^+ s^6 + ...$$

Comment. This method is related to the techniques originally described in [7].

Fuzzy case: what if we only have expert information? In many cases, we do not even have the measurement results. Instead, we have experts' description of the plant. In our terms, we have expert estimates of the coefficients of the matrix $\mathbf{C}$ that describes the plant's dynamics. An "expert estimate" means a statement like "approximately 0.3". Fuzzy methodology provides us with a way to represent such statements in a form understandable by a computer: namely, we represent an expert's statement by a function (called *membership function*) that assigns to every real number x a degree of belief that x satisfies this (informal) statement. For the majority of statements, there is one "most possible" value (e.g., 0.3 for the above statement), and the degree of belief monotonically (and continuously) decreases if we go further away from that value. In other words, we represent each value as a *fuzzy number* (see, e.g., [2,4,8]).

From the expert estimates for the components of $\mathbf{C}$, we can derive the expert estimates for the coefficients p_i of the polynomial. So, we have a polynomial whose coefficients are fuzzy numbers P_i characterized by membership functions μ_i. Is this polynomial stable?

If all the fuzzy numbers are located in some intervals, then we can apply the above-described interval-case algorithm, and check whether the stability of the plant can be guaranteed. If it can be, great. But what if it cannot? This does not necessarily mean that the control strategy that we are using is bad: although it is possible that the plant is unstable, our degree of belief that the plant is unstable can be very small. To make a meaningful decision about the quality of the chosen control, we must be able to estimate this degree of belief. But first, we must define it.

2. MOTIVATIONS OF THE FOLLOWING DEFINITIONS

How to define the degree of belief that the linear system is stable? We need two steps to explain our future definition:

Step 1. A plant is stable if and only if it is not true that it can go unstable. So, following fuzzy interpretation of negation [2,4,8], we define the desired degree of belief S (that a plant is stable) as $1 - C$, where C is a degree of belief that a plant *can* go unstable.

Hence, in order to define S, we must define C.

Step 2. Definiting C. A plant can go unstable if and only if it is unstable for some possible values $p_0, ..., p_n$ of the coefficients. In other words, a plant *can be* unstable if an only for one of the vectors $\mathbf{p} = (p_0, ..., p_n)$, all p_i *are* possible *and* the corresponding polynomial $p(s)$ is unstable. In other words,

$$\text{A plant can go unstable} \leftrightarrow$$
$$\vee(p_0 \text{ is a possible value of } P_0 \;\&\; ... \;\&\; p_n \text{ is a possible value of } P_n \;\&\; p(s) \text{ is unstable}),$$

where $\vee$ means "or" extended to all combinations $(p_0, ..., p_n)$ of real (crisp) numbers.

Let's use this formula to describe the degree of belief C that a plant can go unstable.

- The degree of belief that p_i is a possible value of P_i is described by the membership function $\mu_i(p_i)$.
- For given (crisp) numbers p_i, the stability of a polynomial $p(s)$ is a crisp statement, so its degree of belief is either 1 (=true) or 0 (=false).
- Following standard fuzzy methodology, we interpret & as min, and $\vee$ as sup [2,4,8].

As a result, we arrive at the following formula:

$$C = \sup[\min(\mu_0(p_0), ..., \mu_n(p_n), s(p_0, ..., p_n))],$$

where $s(\mathbf{p})$ is equal to 1 if the polynomial is unstable and to 0 if it is stable, and sup is taken over all possible vectors $\mathbf{p}$.

The fact that s is a crisp function enables us to simplifyy this formula. Indeed:

- When $s = 0$, then $\min = 0$, so we do not need to consider these values when we compute sup.
- When $s = 1$, then (due to the fact that $\min(d, 1) = d$ for every degree of belief $d \in [0, 1]$) we do not need to consider this term in the min.

So, we can simplify the above formula into the following one:

$$C = \sup[\min(\mu_0(p_0), ..., \mu_n(p_n))],$$

where sup is taken over all $\mathbf{p}$ for which the polynomial $p(s)$ is unstable.

3. DEFINITIONS AND THE DESCRIPTION OF AN ALGORITHM

Definitions. By a *fuzzy number*, we mean a continuous function $\mu : R \rightarrow [0, 1]$ that attains its maximum at some value c (called its *center*), and is mononotically decreasing for $x > c$ and monotonically increasing for $x < c$. By a *fuzzy polynomial*, we mean a tuple $(P_0, P_1, ..., P_n)$ of fuzzy numbers. Corresponding centers will be denoted by $c_1, ..., c_n$. A fuzzy polynomial will also be denoted by $P_0 + P_1 s + P_2 s^2 + ... + P_n s^n$. By a *degree of stability* of a fuzzy polynomial we mean a number $S = 1 - C$, where

$$C = \sup[\min(\mu_0(p_0), ..., \mu_n(p_n))],$$

and sup is taken over all vectors $\mathbf{p} = (p_0, ..., p_n)$ for which the polynomial $p(s)$ is unstable.

THEOREM. *Let a non-negative integer k be given. The following algorithm computes S with an accuracy 2^{-k}:*

Algorithm. This algorithm is of binary search type. It consists of k iterations. On $j-$th iteration, we have an interval $[a^-, a^+]$ of length 2^{-j} that contains C. After k iterations, we can take any endpoint of the resulting interval as the desired estimate for C, and compute $S = 1 - C$.

Initially, we set $a^- = 0$, $a^+ = 1$. On each step j, we do the following:

- compute $a = 1/2(a^- + a^+)$;
- for every i, find the values p_i^- and p_i^+ for which $\mu_i(a_i^-) = \mu_i(a_i^+) = a$;
- check whether an "interval" polynomial with intervals $[p_i^-, p_i^+]$ is stable. If it is stable, we take a as a new value of a^+. Else, we take a as a new value of a^-.

Comment. The most time-consuming part of this algorithm is checking whether an interval polynomial is stable. On every iteration, we check stability of one interval polynomial; that means that we check stability of 4 crisp polynomials. Totally, we have $4k$ polynomials to check. To determine the degree of belief S with accuracy 0.1, we need $k = 4$ and 16 polynomials. To determine S with an accuracy 0.01, we need $k = 7$ and 28 polynomials.

Idea of the proof. Because of the definition of C as the supremum, for every real number a, we have:

$$C > a \leftrightarrow \exists p_0, ..., p_n(\min(\mu_0(p_0), ..., \mu_n(p_n)) > a \,\&\, p(s) \text{ is unstable}).$$

Minimum of several numbers is greater than a iff each of them is $> a$. So,

$$C > a \leftrightarrow \exists p_0, ..., p_n(\mu_1(p_0) > a \,\&\, ... \,\&\, \mu_n(p_n) > a \,\&\, p(s) \text{ is unstable}).$$

Since the two statements (one of them $C > a$) are equivalent, their negations are also equivalent, hence:

$$C \le a \leftrightarrow \forall p_0, ..., p_n((\mu_0(p_0) > a \,\&\, ... \,\&\, \mu_n(p_n) > a) \rightarrow p(s) \text{ is stable}).$$

So we reduced checking whether $C \le a$ is true to checking whether it is true that

$$\forall p_0, ..., p_n((\mu_0(p_0) > a \,\&\, ... \,\&\, \mu_n(p_n) > a) \rightarrow p(s) \text{ is stable}).$$

This is (almost) what we check using the interval-case algorithm. The only difference is that the interval-case algorithm actually checks whether the following is true:

$$\forall p_0, ..., p_n((\mu_0(p_0) \ge a \,\&\, ... \,\&\, \mu_n(p_n) \ge a) \rightarrow p(s) \text{ is stable}).$$

The difference ($\ge a$ instead of $> a$) can be taken care of because of the assumptions that the functions μ_i are continuous and monotonic.

As soon as we are able to check, for any given real number a, whether $C \le a$ or not, we can use binary search to find C (actualy, to find C with k binary digits in k steps). And from C, we get S as $1 - C$. Q.E.D.

Acknowledgments. This work was partially supported by NSF grant No. CDA-9015006, and a Grant No. PF90–018 from the General Services Administration (GSA), administered by the Materials Research Institute and the Institute of Manufacturing and Materials Management. The authors are thankful to Reza Langari for valuable discussions.

REFERENCES

1. B. Ross Barmish, *New tools for robustness of linear systems*, McMillan, N.Y., 1994.
2. D. Dubois and H. Prade. *Possibility theory. An approach to computerized processing of uncertainty.* Plenum Press, N.Y. and London, 1988.

3. W. A. Fuller, *Measurement error models*, J. Wiley & Sons, New York, 1987.
4. A. Kauffman, M. M. Gupta. *Introduction to fuzzy arithmetic. Theory and applications.* Van Nostrand, N.Y., 1985.
5. V. L. Kharitonov, *Asymptotic stability of an equilibrium position of a family of systems of linear diferential equation*, Differential Equations, 1978, Vol. 14, pp. 2086-2088.
6. S. Rabinovich, *Measurement errors: theory and practice*, American Institute of Physics, N.Y., 1993.
7. L. A. Zadeh, C. A. Desoer, *Linear systems theory: a state-space approach*, McGraw-Hill, N.Y., 1963.
8. H. J. Zimmerman. *Fuzzy set theory and its applications*, Kluwer–Nijhof Publ., Boston–Dordrecht, 1985.