Why Fuzzy Control Is Often More Robust (and Smoother): A Theoretical Explanation

1st Orsolya Csiszár
Research Center for Complex Systems
Aalen University
orsolya.csiszar@hs-aalen.de
and *Institute of Applied Mathematics*
John von Neumann Faculty of Informatics
Óbuda University
csiszar.orsolya@nik.uni-obuda.hu

2nd Gábor Csiszár
Biomatics and Applied AI Institute
John von Neumann Faculty of Informatics
Óbuda University
csiszar.gabor@uni-obuda.hu

3th Olga Kosheleva
Department of Teacher Education
University of Texas at El Paso
El Paso, Texas, USA
olgak@utep.edu

4th Martine Ceberio
Department of Computer Science
University of Texas at El Paso
El Paso, Texas, USA
mceberio@utep.edu

5th Vladik Kreinovich
Department of Computer Science
University of Texas at El Paso
El Paso, Texas, USA
vladik@utep.edu

***Abstract*—In many practical situations, practitioners use easier-to-compute fuzzy control to approximate the more-difficult-to-compute optimal control. As expected, for many characteristics, this approximate control is slightly worse than the optimal control it approximates. However, with respect to robustness or smoothness, the approximating fuzzy control is often better than the original one. In this paper, we provide a theoretical explanation for this somewhat mysterious empirical phenomenon.**

***Index Terms*—fuzzy control, robust control, smooth control**

I. Formulation of the Problem

Fuzzy control: a brief reminder. In the early 1960s, Lotfi Zaheh, one of the world's leading specialists in control, and a co-author of the most popular textbook on optimal control, noticed that sometimes, when experts control a plant, they get better results than supposedly optimal automatic controllers.

To a pure mathematician, this may sound like a paradox, since optimal means the best. However, from the engineering viewpoint, this was not really a paradox: optimal means optimal with respect to a given model of a plant. Models are approximate. Because of the difference between the approximate model and the actual system, optimal control based on the approximate model may not be optimal for the real-life system.

What this advantage of human controllers indicated was that expert controllers have some knowledge that was not implemented in the models. A natural idea is thus to elicit this additional knowledge and to use it when controlling a plant. Here he encountered a problem:

- most expert controllers were willing to share their additional knowledge,
- but they shared it by using words from natural language like "small", and
- it was not clear how to incorporate this imprecise ("fuzzy") knowledge into a mathematical model.

To make such an incorporation possible, Zadeh came up with a methodology that he called *fuzzy logic*; see, e.g., [1]–[5], [7].

One of the main ideas behind this methodology is that:

- in contrast to well-defined mathematical properties which are always either true or false,
- properties like "small" are not that precise.

It is not that control engineers had an exact threshold:

- below which everything is absolutely small, and
- above which everything is absolutely not small.

Yes:

- some values are considered to be absolutely small,
- some as absolutely not small,
- but for many intermediate values, the experts are not sure, their answers – like "somewhat small" – are somewhere in between.

In a computer:

- "true" is usually represented as 1, and
- "false" as 0.

It is therefore reasonable to use numbers between 0 and 1 to represent intermediate opinions. The idea of using such intermediate values, while it was new for engineering applications,

This work was supported in part by the National Science Foundation grants 1623190 (A Model of Change for Preparing a New Generation for Professional Practice in Computer Science), HRD-1834620 and HRD-2034030 (CAHSI Includes), EAR-2225395, and by the AT&T Fellowship in Information Technology. It was also supported by the program of the development of the Scientific-Educational Mathematical Center of Volga Federal District No. 075-02-2020-1478, and by a grant from the Hungarian National Research, Development and Innovation Office (NRDI).

was not fully new: the so-called Likert scale – marking a value between 0 and 5 or 10 (or some other number) – is something we use all the time:

- whether we evaluate the quality of our hotel stay or
- whether students evaluate how well we teach.

Control based on such incorporated knowledge is what became known as *fuzzy control.*

Fuzzy control is often used as an approximation to the actual control. One of the advantages of fuzzy control is that most of its techniques are computationally easy. In contrast, optimal control often requires a lot of computational resources to compute – and in many real-time and/or embedded systems, our computational ability is limited. In such situations, sometimes, fuzzy control is used as an easier-to-compute approximation to the optimal control strategy.

Let us describe this idea in precise terms. In control situations, we want, for each state x of the plant, to provide an appropriate control y. In these terms, a control strategy is an algorithm $f(x)$ that, given the current state x, returns the control value(s) $y = f(x)$.

When the algorithm $f(x)$ is difficult to compute, what we can do is pre-compute the control $y_i = f(x_i)$ corresponding to several typical states x_i. Once we have this information, we can formulate natural common-sense rules of the type

"if the state x is close to x_i, then the applied control should be close to $y_i = f(x_i)$."

The resulting approximate fuzzy control is often smoother and more robust than the original control, but why? In the use of fuzzy control as an approximation to a known difficult-to-implement control, an unexpected phenomenon occurred:

- Since the fuzzy control is an approximation, one would expect that it will be somewhat worse than the original control.
- With respect to some objective functions, this is indeed true: e.g., if we want to reach the destination in the shortest time or with the smallest possible use of fuel, the approximating fuzzy control will be slightly worse than the optimal control it approximates.
- However, surprisingly, it turned out that the approximating control is often more smooth and more robust than the control it approximates.

This was shown in many applications: e.g., elevators or trains controlled by fuzzy control run smoother. Zadeh himself liked to tell what he called an elevator speech about fuzzy control. As you all know, the elevator speech is a short presentation that we need to make when riding an elevator with a colleague (or better a boss or an investor) to whom we need to describe what we are doing. In Zadeh's elevator speech, he recalled how at one of the conferences in Japan, he happened to enter the same elevator as one of the control experts who was skeptical about fuzzy. It so happened that this elevator was using fuzzy control. So, when Zadeh pressed the button and they did not feel the usual jerk, his opponent proudly remarked: I told you so, fuzzy control does not work. Immediately after that, the elevator door opened, and, to this sceptic's surprise, they turned out to be exactly on the top floor where they planned to be.

This smoothness of fuzzy control is well-known; it is used in situations when we want smoothness – e.g., in many car models, fuzzy control is embedded in automatic transmissions to make the ride smoother. The fact that this phenomenon is well known – and actively used – does not make it less mysterious.

Similarly, fuzzy control is known to be more robust than the original control – in the sense that when the situation changes somewhat, the control provided by a fuzzy controller does not change as much as the control provided by the original controller.

What we do in this paper. In this paper, we provide a theoretical explanation for the smoothness and robustness of fuzzy control.

II. Let Us Formulate the Problem in Precise Terms

How do we gauge (and compare) robustness. In order to provide the desired explanation, we need to be able to compare the robustness of different controllers. To be able to make this comparison, we need to come up with a reasonable numerical measure of robustness.

Intuitively, robustness means that if we change the state x a little bit, the control $y = F(x)$ should not change much. Let us consider the case when we change the value of one of the characteristics of a system. In this case, the control strategy $y = F(x)$ is a function of one variable. When we change the input x, from x to $x + \Delta x$, the control value changes from $F(x)$ to $F(x + \Delta x)$, so the change in control is equal to $\Delta y = F(x + \Delta x) - F(x)$.

For small changes Δx, we can expand this expression in Taylor series and take into account the fact that for small Δx, terms which are quadratic (or higher order) in Δx are much smaller than linear terms. For example, if $\Delta x = 1\%$, then $(\Delta x)^2 = 0.01\%$ which is much smaller. Thus, we can safely ignore quadratic and higher-order terms, and keep only linear terms in this expression, i.e., take $\Delta y = F'(x) \cdot \Delta x$.

Robustness means that the difference Δy should be small, i.e., close to 0. Since the value Δx is fixed, this is equivalent to requiring that the derivative $F'(x)$ is small – i.e., close to 0.

We want to have $F'(x) \approx 0$ for all x. From the purely mathematical viewpoint, there are infinitely many possible values x. However, from a practical viewpoint, values which are very close to each other are indistinguishable. Let h be the smallest distinguishable difference. From this viewpoint, there are only finitely many distinguishable states X_1, $X_2 = X_1 + h$, $X_3 = X_2 + h, \ldots$, all the way to some large value x_N. (We use capital letters to distinguish between these very dense values and much more spare values $x_1, x_2, \ldots$ used in the above formulation of fuzzy control rules.)

In these terms, robustness means that the corresponding N derivatives should be all close to 0:

$$F'(X_1) \approx 0, \quad F'(X_2) \approx 0, \quad \ldots, \quad F'(X_N) \approx 0.$$

In other words, the tuple

$$(F'(X_1), F'(X_2), \ldots, F'(X_N))$$

formed by these derivatives should be close to the zero tuple

$$(0, 0, \ldots, 0).$$

Each tuple can be naturally represented by a point in the N-dimensional space. In this space, the natural Euclidean distance d between these two points is equal to

$$d = \sqrt{(F'(X_1))^2 + (F'(X_2))^2 + \ldots + (F'(X_N))^2}.$$

Minimizing this distance is equivalent to minimizing the square of this distance – i.e., the sum

$$d^2 = (F'(X_1))^2 + (F'(X_2))^2 + \ldots + (F'(X_N))^2.$$

This expression can be further simplified if we take into account that if we multiply this sum by the difference $h = X_{i+1} - X_i$, we get an integral sum:

$$d^2 \cdot h = (F'(X_1))^2 \cdot h + (F'(X_2))^2 \cdot h + \ldots + (F'(X_N))^2 \cdot h.$$

For small h, the integral sum is close to the corresponding integral; after all:

- this is how the integral is defined, as the limit of integral sums, and
- this is how integrals are often computed – by computing the appropriate integral sum.

So, we have

$$d^2 \cdot h \approx \int (F'(x))^2 \, dx.$$

Thus, minimizing the desired distance d is equivalent to minimizing the integral

$$\int (F'(x))^2 \, dx. \qquad (1)$$

So, this integral provides a natural measure of non-robustness – whichever control has the smaller value of this integral has the smaller value of the distance d between the actual and ideal difference, which is exactly what robustness is about.

Which fuzzy control methodology we will use. There are several different techniques for transforming natural rules into a precise control strategy. In this paper, we use the most widely used – and the least computationally intensive – Takagi-Sugeno methodology, according to which the control strategy generated by the system takes the form

$$F(x) = \frac{\sum C(x - x_i) \cdot y_i}{\sum C(x - x_i)} = \frac{\sum C(x - x_i) \cdot f(x_i)}{\sum C(x - x_i)}, \qquad (2)$$

where $C(\Delta x)$ describes the degree (from the interval $[0, 1]$) to which x is close to x_i. This degree:

- is equal to 1 ("absolutely close") when $\Delta x = 0$, and
- decreases to 0 when the absolute value of the difference Δx increases.

Resulting question. We need to explain why the degree of non-robustness (1) corresponding to the fuzzy control (2) is smaller than the degree of non-robustness corresponding to the original control strategy $f(x)$.

III. Our Explanation

To formulate our result, let us first simplify the expression (2). To make the desired comparison possible, let us first simplify the expression (2) for fuzzy control.

To perform this simplification, we can use the same idea as in the previous section – namely, we can take into account that if we multiply both the numerator and the denominator of the formula (2) by the difference $H \stackrel{\text{def}}{=} x_{i+1} - x_i$, then both the numerator and the denominator become integral sums:

$$F(x) = \frac{\sum C(x - x_i) \cdot f(x_i) \cdot H}{\sum C(x - x_i) \cdot H}.$$

As we have mentioned, integral sums are a good approximation to the corresponding integrals. So, within this approximation, we can write that

$$F(x) = \frac{\int C(x - z) \cdot f(z) \, dz}{\int C(x - z) \, dz}. \qquad (3)$$

Now, we are ready to formulate our result.

Definition.

- *By a* control strategy*, we mean a differentiable non-constant function $f(x)$ defined on a bounded interval.*
- *By the* degree of non-robustness *of a control strategy $f(x)$, we mean the value* (1). *We will denote this degree by $N(f)$.*
- *By a* fuzzy control strategy corresponding to control strategy *$f(x)$, we mean a function* (3)*, where $C(x)$ is a continuous function whose values are from the interval $[0, 1]$; we will denote this function* (3) *by $S(f)$.*

Proposition. *For each control strategy, the degree of non-robustness $N(S(f))$ of a fuzzy control strategy $S(f)$ corresponding to $f(x)$ is smaller than the degree of non-robustness $N(f)$ of the original control strategy:*

$$N(S(f)) < N(f).$$

Discussion. Thus, indeed, fuzzy control based on some control strategy is always more robust than the original control.

Proof.

1°. Let us first simplify the denominator of the expression (3).

Indeed, by replacing the original variable z with a new variable $t = x - z$, we can check that the integral in the denominator is simply equal to the integral $\int C(t) \, dt$ and thus, does not depend on x at all. Let us denote the value of this integral by $I \stackrel{\text{def}}{=} \int C(t) \, dt$. Thus, the expression (3) takes a simplified form

$$F(x) = \int c(x - z) \cdot f(z) \, dz, \qquad (4)$$

where we denoted $c(x) \stackrel{\text{def}}{=} C(x)/I$. By definition of I, we can conclude that

$$\int c(x) \, dx = 1. \qquad (5)$$

2°. To prove our result, we will use the properties of the Fourier transform; see, e.g., [6]. Fourier transform of a function $a(x)$ is defined as

$$\hat{a}(\omega) = \int a(x) \cdot \exp(\mathrm{i} \cdot \omega \cdot x)\, dx, \tag{6}$$

where, as usual, i means the square root of -1: $\mathrm{i} \stackrel{\text{def}}{=} \sqrt{-1}$. In particular, for $\omega = 0$, we get

$$\hat{a}(0) = \int a(x)\, dx. \tag{7}$$

In our proof, we will use the following three properties of the Fourier transform.

- The first property of the Fourier transform that we will use is the so-called *Parceval theorem*, according to which the integral of the square of a function is proportional to the integral of the square of the absolute value of its Fourier transform:

$$\int (a(x))^2\, dx = \frac{1}{2\pi} \int |\hat{a}(\omega)|^2\, d\omega. \tag{8}$$

- The second property of the Fourier transform that we will use is that once we know the Fourier transform $\hat{a}(\omega)$ of a function $a(x)$, the Fourier transform of its derivative $b(x) = a'(x)$ can be obtained from $\hat{a}(\omega)$ by multiplying by $\mathrm{i} \cdot \omega$:

$$\hat{b}(\omega) = \mathrm{i} \cdot \omega \cdot \hat{a}(\omega). \tag{9}$$

- Finally, the third property of the Fourier transform is that if we know the Fourier transforms $\hat{a}(\omega)$ and $\hat{b}(\omega)$ of two functions $a(x)$ and $b(x)$, then the Fourier transform of their convolution

$$c(x) \stackrel{\text{def}}{=} \int a(x-y) \cdot b(y)\, dy \tag{10}$$

is equal to the product of their Fourier transforms:

$$\hat{c}(\omega) = \hat{a}(\omega) \cdot \hat{b}(\omega). \tag{11}$$

3°. Now, we are ready for the proof.

3.1°. For the original control $f(x)$, its degree of non-robustness is equal to $N(f) = \int (f'(x))^2\, dx$. We want to use Parceval theorem to evaluate this integral. For this, we need to know the Fourier transform of the derivative $f'(x)$ – which, according to (9), is equal to $\mathrm{i} \cdot \omega \cdot \hat{f}(\omega)$. Thus, we conclude that

$$N(f) = \frac{1}{2\pi} \cdot \int \omega^2 \cdot \left|\hat{f}(\omega)\right|^2\, d\omega. \tag{12}$$

3.2°. By the formula (4), the function $F(x)$ describing fuzzy control is a convolution of functions $c(x)$ and $f(x)$. Thus, by the third property of Fourier transforms, we conclude that

$$\hat{F}(\omega) = \hat{c}(\omega) \cdot \hat{f}(\omega). \tag{13}$$

For the non-robustness $N(F)$ of the fuzzy control $F = S(f)$, similarly to the formula (12), we get the formula

$$N(S(f)) = N(F) = \frac{1}{2\pi} \cdot \int \omega^2 \cdot \left|\hat{F}(\omega)\right|^2\, d\omega. \tag{14}$$

Substituting the expression (13) into the formula (14), and taking into account that the absolute value of the product of two complex numbers is equal to the product of their complex values, we conclude that

$$N(S(f)) = \frac{1}{2\pi} \cdot \int \omega^2 \cdot |\hat{c}(\omega)|^2 \cdot \left|\hat{f}(\omega)\right|^2\, d\omega. \tag{15}$$

3.3°. Here, due to (7) and (5), we conclude that

$$\hat{c}(0) = 1. \tag{16}$$

For all other values ω, by definition of the Fourier transform, we get

$$\hat{c}(\omega) = \int c(x) \cdot \exp(\mathrm{i} \cdot \omega \cdot x)\, dx. \tag{17}$$

The absolute value of the sum of complex numbers is smaller than or equal to the sum of absolute values:

$$|a + b| \le |a| + |b|,$$

and equality only happens when these two numbers differ by a positive factor – in all other cases, we have strict inequality. This can be easily understood if we recall that the absolute value of a complex number $x + y \cdot \mathrm{i}$ is equal to the length $\sqrt{x^2 + y^2}$ of the vector correcting the point $(0, 0)$ with the point (x, y), and addition of complex numbers is equivalent to adding the corresponding vectors.

In our case, as one can easily check, the values

$$c(x) \cdot \exp(\mathrm{i} \cdot \omega \cdot x)$$

corresponding to different ω are *not* differing by a positive factor. Thus, we have

$$|\hat{c}(\omega)| < \int |c(x) \cdot \exp(\mathrm{i} \cdot \omega \cdot x)|\, dx =$$

$$\int c(x) \cdot |\exp(\mathrm{i} \cdot \omega \cdot x)|\, dx. \tag{18}$$

We know that for any real number a, we have $\exp(\mathrm{i} \cdot a) = \cos(a) + \mathrm{i} \cdot \sin(a)$ and thus,

$$|\exp(\mathrm{i} \cdot a)| = \sqrt{\cos^2(a) + \sin^2(a)} = 1.$$

Thus, the inequality (18) implies that

$$|\hat{c}(\omega)| < \int c(x)\, dx. \tag{18}$$

We already know that the integral on the right-hand side is equal to 1, so we get

$$|\hat{c}(\omega)| < 1. \tag{19}$$

For non-negative values, the function $z \mapsto z^2$ is an increasing function. So we can square both sides of the inequality (19) and get a valid inequality

$$|\hat{c}(\omega)|^2 < 1 \text{ for all } \omega \neq 0. \tag{19}$$

For each $\omega \neq 0$, the integrated non-negative expression

$$\omega^2 \cdot |\hat{c}(\omega)|^2 \cdot \left|\hat{f}(\omega)\right|^2 \tag{20}$$

in the formula (15) is obtained by the integrated non-negative expression

$$\omega^2 \cdot \left|\hat{f}(\omega)\right|^2 \tag{21}$$

in the formula (12) by multiplying by a factor $|\hat{c}(\omega)|^2$. Since this factor is smaller than 1 for all $\omega \neq 0$, we thus conclude that for all such ω, the integrated non-negative expression (20) in the formula (15) that defines $N(S(f))$ is smaller than the integrated non-negative expression (21) in the formula (12) that defines $N(f)$. Thus, the corresponding integrals are also smaller, i.e., indeed, $N(S(f)) < N(f)$.

The proposition is proven.

What about smoothness. For smoothness, the proof is similar, only instead of dependence on a general value x, we need to consider dependence on time.

REFERENCES

[1] R. Belohlavek, J. W. Dauben, and G. J. Klir, *Fuzzy Logic and Mathematics: A Historical Perspective*, Oxford University Press, New York, 2017.

[2] G. Klir and B. Yuan, *Fuzzy Sets and Fuzzy Logic*, Prentice Hall, Upper Saddle River, New Jersey, 1995.

[3] J. M. Mendel, *Uncertain Rule-Based Fuzzy Systems: Introduction and New Directions*, Springer, Cham, Switzerland, 2017.

[4] H. T. Nguyen, C. L. Walker, and E. A. Walker, *A First Course in Fuzzy Logic*, Chapman and Hall/CRC, Boca Raton, Florida, 2019.

[5] V. Novák, I. Perfilieva, and J. Močkoř, *Mathematical Principles of Fuzzy Logic*, Kluwer, Boston, Dordrecht, 1999.

[6] B. G. Osgood, *Lectures on the Fourier Transform and Its Applications*, American Mathematical Society, Providence, Rhode Island, 2019.

[7] L. A. Zadeh, “Fuzzy sets”, *Information and Control*, 1965, Vol. 8, pp. 338–353.