# Hyperbolic Approach to Fuzzy Control Is Optimal

Hung T. Nguyen
Mathematics, New Mexico State University, Las Cruces, NM 88003, USA, hunguyen@nmsu.edu

Vladik Kreinovich
Computer Science, University of Texas, El Paso, TX 79968, USA, vladik@cs.utep.edu

Michael Margaliot and Gideon Langholtz
Electrical Engineering, Tel Aviv U., Israel 69978, michaelm@eng.tau.ac.il

***Abstract*—In a series of papers and a book, M. Margaliot and G. Langholz proposed a hyperbolic approach to fuzzy control, in which they apply a certain hyperbolic non-linear transformation to the original variables. In this paper, we consider all possible non-linear transformations of this type and show that this hyperbolic transformation is indeed optimal.**

**Many Practical Control Situations Can Be Adequately Described by Linear Systems.** One of the main problems to which control is applied is stabilizing a plant. In this case, it is usually 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: $\mathbf{s} = (s_1, \ldots, s_n)$. Let's denote the ideal state by $(s_1^{(0)}, \ldots, s_n^{(0)})$. 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 $(0, 0, \ldots, 0)$.

In general, the evolution of a state $\mathbf{x}$ under a control $\mathbf{u} = (u_1, \ldots, u_p)$ is described by a system of differential equations

$$\dot{x}_i = E_i(x_1, \ldots, x_n, u_1, \ldots, u_p), \qquad (1)$$

where the functions $E_i$ are, 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 functions $E_i$ into Taylor series, and leave only the main (linear) terms. As a result, we get a linear system of differential equations:

$$\dot{x}_i = \sum_{j=1}^{n} a_{ij} \cdot x_j + \sum_{k=1}^{p} b_{ik} \cdot u_k, \qquad (2)$$

where $\mathbf{A} = \|a_{ij}\|$ and $\mathbf{B} = \|b_{ik}\|$ are matrices.

For such *linear systems*, there exist well-developed optimal control techniques; see, e.g., [2]

**In Many Other Practical Problems, Non-Linear Terms Must Be Taken Into Consideration.** In many practical problems, before we achieve the desired trajectory, we must first reach it from the original state. On this stage, the deviations $x_i$ of the actual state from the desired one are not necessarily small, so we no longer can neglect non-linear terms in the dependence of $E_i$ on $x_1, \ldots, x_n$; we must take non-linear terms into consideration.

If we do not take these terms into consideration, the linear description (2) stops being adequate. From the physical viewpoint, one of the main inadequacies of a linear representation is that it assumes the linear dependence between the deviation $x_i$ and its rate of change: the larger the deviation $x_i$, the more it changes (i.e., the larger its rate $\dot{x}_i$). For example, in a linear system, if we increase $x_i$ twice, we double the rate. Similarly, if we have proportionally larger deviations, all we have to do is proportionally increase the control values. Real-life systems are not that simple. In most practical systems, there is a limitation on the rate of change, due to which, e.g., no matter how large the control values are, we never exceed a certain influence level.

The existence of such bounds is in perfect agreement with common sense. For example, when we try to accelerate a car, at first, the more we press the accelerator, the faster the car moves; however, after a certain level, when we reach this particular car's limit, further pressing does not make the car go faster.

**Nonlinear Control Problems Are, In General, Computationally Difficult.** How can we control nonlinear systems? In contrast to linear systems (for which optimal control can be easily computed by known formulas), for non-linear systems, no such easily computable general formulas exist. Moreover, the problem of the optimal control of a non-linear system is known to be computationally intractable (NP-hard); see, e.g., [1], [5].

Since it is computationally difficult to find the optimal control for an arbitrary nonlinear control system, it is desirable to look for suboptimal ("good enough") methods. A natural approach to finding such suboptimal controls is to find a class of systems for which there are explicit formulas for optimal control, and then approximate the actual nonlinear system by an appropriate system from this class. Then, the optimal control to an approximating system can serve as a suboptimal control to the original hard-to-solve nonlinear control system.

This idea is exactly what we do when we approximate the original nonlinear system by its linear approximation. So, what we really want is to find a class of nonlinear systems, a class which is more general than the class of linear system, and which will still allow us to have explicit expressions for optimal control.

**Saturation Approach to Nonlinear Control: Main Idea.** Since one of the main features of the real-life system which is not reflected in a linear-system description is saturation, a natural idea is to replace the original lin-

ear equations (2) with a new system of equations:

$$\dot{x}_i = \sum_{j=1}^{n} a_{ij} \cdot z_j + \sum_{k=1}^{p} b_{ik} \cdot u_k, \qquad (3)$$

where $z_i = f_i(x_i)$, and the non-linear transformations $f_i(x)$ are selected in such a way that when even the values $x_i$ tend to infinity, the new variables $z_i$ stay within a limited range.

It is reasonable to start with the simplest possible implementation of this saturation idea, when we use the same non-linear function $f(x)$ for all the variables $x_i$. To be more precise, we should take into consideration the fact that different components of the state vector $(x_1, \ldots, x_n)$ represent different physical quantities (e.g., location and speed), so different measuring units can be used for different components. When we change a measuring unit, the numerical value of the measured quantity gets changed by a multiplicative constant $x_i \to k_i \cdot x_i$. Thus, "the use of the same function $f(x)$" can be formally described as using the new variables $z_i = f(k_i \cdot x_i)$ for some constants $k_i$.

For small $x$, the linear system representation works just fine, and no non-linear transformation is needed. Therefore, a natural requirement on the function $f(x)$ is that for small values $x \approx 0$, we should have $f(x) \approx x$, and the closer we are to 0, the better this approximation. In precise terms, we should have $f(x) = x + o(x)$.

Also, since we have no a priori reason to prefer positive or negative values of $x$, it is reasonable to make this non-linear transformation sign-independent, i.e., independent with respect to $x \to -x$. In more precise terms, we require that $f(-x)$ is exactly the same as $-f(x)$.

**Hyperbolic Approach to Nonlinear Control and Its Relation to Fuzzy Control.** In [6], [7], it was proposed to use $f(x) = \tanh(x)$. It turns out that not only this selection leads to a good quality control, but that the resulting optimal control coincides with a specific case of fuzzy control and thus, provides a commonsense interpretation of the corresponding optimal control.

**Which Nonlinear Transformation Is The Best? Open Problem.** The original linear model corresponds to the selection of a function $f(x) = x$. Since switching from $f(x) = x$ to $f(x) = \tanh(x)$ drastically improves the control quality, it is natural to ask: will other choices of a transformation $f(x)$ provide even better control results? And the final question: Which non-linear transformation is the best?

In this paper, we will give an answer to both questions by showing that the hyperbolic transformation is indeed the best (in some reasonable sense).

**Why Is This Problem Difficult?** We want to find a transformation function $f(x)$ for which some characteristics $J$ of the resulting control, such as average relaxation time or the average computation time, is optimal (in these cases minimal). The problem is that even for the hyperbolic function $f(x)$, we do not know how to compute any of these possible characteristics. How can we find $f$ for which $J(f)$ is optimal if we cannot compute $J(f)$ even for a single $f$? At first glance, there does not seem to be a likely answer.

However, we will show that this problem is solvable (and give the solution) using advanced math, namely, group theory.

**First Comment: We Must Choose a Family of Functions, Not a Single Function.** At first glance, our goal is to select the best transformation function $f(x)$. We speak about choosing the transformation function $f(x)$, but the expression for $f(x)$ will change if we change the measuring unit for time.

If we change a unit used to measure time by a one which is $k$ times larger, then the numerical value of time will decrease by $k$, and the numerical values of the rate of increase $\dot{x}_i$ will increase by $k$. Thus, instead of the original term $z_i = f(x_i)$, we now have the term $k \cdot f(x_i)$. So, this change of a time unit is equivalent to using, instead of the original transformation function $f(x)$, a new transformation function $k \cdot f(x)$. Since we did not change anything neither in the actual system nor in the control, all we did was change the measuring unit for time, the quality of a control cannot change. Thus, if the transformation function $f(x)$ provided the optimal control, so would the transformation function $k \cdot f(x)$ which represents exactly the same control.

Hence, we cannot select a unique transformation function $f(x)$ and claim it to be the best, because for every function $f(x)$, the function $k \cdot f(x)$ leads to a control of exactly the same quality.

In other words, different transformation functions can be naturally grouped into *families* of functions so that different functions form the same family represent the same transformation, but in different units. In particular, if a function $f(x)$ belongs to a certain family, then this family must also contain the function $k \cdot f$ for positive real numbers $k$ – this corresponds to changing a time unit. (As we will see later, we can have other re-scalings which lead to the appearance of even more functions in the same family.)

In view of this family structure, instead of formulating a problem of choosing the best transformation *function*, it is more natural to formulate a problem of choosing the best *family* of functions.

What other transformations enable us to stay within the same family? To answer this question, let us recall that one of the most natural examples of a state variable $x_i$ is a spatial coordinate. In this case, the derivative $\dot{x}_i$ is a velocity. The numerical value of the velocity changes not only if we change the time unit (as we did before), but also if we go from the original coordinate system to a new inertial coordinate system which moves with respect to the original one with the constant speed. (The fact that we can express any physical law in terms of the moving system as well as in terms of the original immobile system was known as early as Galileo; it is one of the main principles behind Einstein's Relativity Theory.)

In Newtonian physics, when we move to a moving system, all the velocity values are changed by an additive constant. Thus, in terms of the transformation function,

the possibility to use the moving system is equivalent to replacing the original function $f(x)$ by a new function $f(x) + c$.

Hence, the functions $f(x)$ and $f(x)+c$ express the same transformation and the same control – but expressed in different coordinate systems. Therefore, if a family of physically equivalent transformation functions contains a transformation function $f(x)$, this family must also contain functions $f(x) + c$ for all possible constants $c$.

Since we are talking about non-linear systems, we can also assume that some non-linear "rescaling" transformations $x \to g(x)$ are also applicable, i.e., the family may include the composition $g(f(x))$ for each of functions $f(x)$. For example, according to relativity theory, turning to a new coordinate system which moves with a constant velocity $v_0$ with respect to the old one means that we replace each original velocity $v$ by a new value

$$v' = \frac{v + v_0}{1 + \frac{v \cdot v_0}{c^2}}.$$

This change is non-linear in $v$ and thus, corresponds to a *non-linear* change in the transformation function $f(x)$ (which expresses the velocity $v = \dot{x}$).

It is reasonable to assume that there is some finite-parametric family of re-scalings, and that each family is obtained from a single transformation function $f(x)$ by applying all these rescalings.

If $z \to g_1(z)$ and $z \to g_2(z)$ are possible rescalings, then it is natural to require that this composition $z \to g_1(g_2(z))$ and the inverse $z \to g_1^{-1}(z)$ are also possible rescalings (by $g_1^{-1}$ we denoted an inverse function $g_1^{-1}(z) = w$ if and only if $g_1(w) = z$). In mathematical terms, this requirement can be expressed by saying that the class of possible rescalings is *closed* under composition and inverse. Such a class is called a *group*. Thus, we conclude the following:

> *A family is obtained by applying, to some transformation function $f(x)$, all rescalings from some finite-dimensional group $G$ which includes all linear transformations (and maybe some non-linear ones).*

From the physical viewpoint, if we have two different rescalings, there must be a smooth transition between them. In mathematical terms, the existence of this continuous transition is expressed by saying that the group is *connected*, and the fact that both the transformations and the transitions are smooth is expressed by saying that this is a *Lie group* (for a description in the context of computer science and fuzzy, see, e.g., [8]).

**Which Family Is the Best?** Among all such families, we want to choose the best one. In formalizing what "the best" means we follow the general idea outlined in [8]. The criteria to choose may be computational simplicity, average relaxation time, or something else.

In mathematical optimization problems, numerical criteria are most frequently used, when to every alternative (in our case, to each family) we assign some value expressing its performance, and we choose an alternative (in our case, a family) for which this value is the largest. However, it is not necessary to restrict ourselves to such numerical criteria only.

For example, if we have several different families that have the same average relaxation time $T$, we can choose between them the one that has the minimal computational complexity $C$. In this case, the actual criterion that we use to compare two families is not numerical, but more complicated: *a family $F_1$ is better than the family $F_2$ if and only if either $T(F_1) < T(F_2)$ or $T(F_1) = T(F_2)$ and $C(F_1) < C(F_2)$*. A criterion can be even more complicated. What a criterion *must* do is to allow us for every pair of families to tell whether the first family is better with respect to this criterion (we'll denote it by $F_1 > F_2$), or the second is better ($F_1 < F_2$) or these families have the same quality in the sense of this criterion (we'll denote it by $F_1 \sim F_2$). Of course, it is necessary to demand that these choices be consistent, e.g., if $F_1 > F_2$ and $F_2 > F_3$ then $F_1 > F_3$.

Another natural demand is that this criterion must choose a *unique* optimal family (i.e., a family that is better with respect to this criterion than any other family). The reason for this demand is very simple.

If a criterion does not choose any family at all, then it is of no use.

If several different families are "the best" according to this criterion, then we still have a problem to choose among those "best". Therefore, we need some additional criterion for that choice. For example, if several families turn out to have the same average relaxation time, we can choose among them a family with minimal computational complexity. So what we actually do in this case is abandon that criterion for which there were several "best" families, and consider a new "composite" criterion instead: $F_1$ is better than $F_2$ according to this new criterion if either it was better according to the old criterion or according to the old criterion they had the same quality and $F_1$ is better than $F_2$ according to the additional criterion. In other words, if a criterion does not allow us to choose a unique best family it means that this criterion is not final; we have to modify it until we come to a final criterion that will have that property.

The next natural condition that the criterion must satisfy is connected with the fact that the numerical value of a variable $x_i$ depends on the choice of the starting point. We have already mentioned that in Newtonian physics, when we turn to a moving coordinate system, the numerical value of the velocity changes by an additive constant: $v \to v' = v + c$. Previously, we applied this fact to the situation when one of the state variables was a spatial coordinate. However, when we express Newton's laws in terms of first-order differential equations (1), it is not enough to have spatial coordinates as state variables; we must also have *velocities* as state variable. (In this case, the typical Newton's law $\ddot{x} = F(x)$ can be expressed as $\dot{x} = v$ and $\dot{v} = F(x)$.) For such state variables, a new coordinate system means changing $x$ by $x + c$. Since the acceleration $\ddot{x}$ does not change under such change, this change means, in terms of a transformation function $f(x)$, that we replace $f(x)$ with $f(x + c)$.

Since this change is simply a change in a coordinate

system, it does not change the actual system or its control, it is reasonable to require that going from $f(x)$ from $f(x+c)$ should not change the *relative* quality of the transformation functions, i.e., if a family $\{f(x)\}$ is better that a family of $\{g(x)\}$, then for every $c$, the family $\{f(x+c)\}$ must be still better than the family $\{g(x+c)\}$.

So, we arrive at the following definitions.

**Definition 1.** *By a rescaling we mean a smooth (differentiable) function from real numbers into real numbers.*

**Definition 2.** *By an appropriate rescaling group $G$ we mean a finite-dimensional connected Lie group of transformations.*

**Definition 3.** *By a family of functions we mean the set of functions that is obtained from a smooth (everywhere defined) non-constant function $f(x)$ mapping the real line into a bounded interval by applying all the transformations from some appropriate rescaling group $G$. Let us denote the set of all the families by $\mathcal{F}$.*

**Definition 4.** *A pair of relations $(<, \sim)$ is called consistent* [8] *if it satisfies the following conditions: (1) if $a < b$ and $b < c$ then $a < c$; (2) $a \sim a$; (3) if $a \sim b$ then $b \sim a$; (4) if $a \sim b$ and $b \sim c$ then $a \sim c$; (5) if $a < b$ and $b \sim c$ then $a < c$; (6) if $a \sim b$ and $b < c$ then $a < c$; (7) if $a < b$ then $b < a$ or $a \sim b$ are impossible.*

**Definition 5.** *Assume a set $A$ is given. Its elements will be called alternatives. By an optimality criterion we mean a consistent pair $(<, \sim)$ of relations on the set $A$ of all alternatives. If $a > b$, we say that $a$ is better than $b$; if $a \sim b$, we say that the alternatives $a$ and $b$ are equivalent with respect to this criterion.*

**Definition 6.** *We say that an alternative $a$ is optimal (or best) with respect to a criterion $(<, \sim)$ if for every other alternative $b$ either $a > b$ or $a \sim b$.*

**Definition 7.** *We say that a criterion is final if there exists an optimal alternative, and this optimal alternative is unique.*

**Definition 8.** *Let $c$ be a real number. By the c-shift $T_c(f)$ of a function $f(x)$ we mean a function $(T_c(f))(x) \stackrel{\text{def}}{=} f(x+c)$.*

**Definition 9.** *By the c-shift $T_c(F)$ of a family $F$, we mean the set of the functions that are obtained from $f \in F$ by c-shift.*

In this paper, we consider optimality criteria on the set $\mathcal{F}$ of all families.

**Definition 10.** *We say that an optimality criterion on $F$ is shift-invariant if for every two families $F$ and $G$ and for every number $c$, the following two conditions are true:*

i) *if $F$ is better than $G$ in the sense of this criterion (i.e., $F > G$), then $T_c(F) > T_c(G)$;*

ii) *if $F$ is equivalent to $G$ in the sense of this criterion (i.e., $F \sim G$), then $T_c(F) \sim T_c(G)$.*

As we have already remarked, the demands that the optimality criterion is final and shift-invariant are quite reasonable. The only problem with them is that at first glance they may seem rather weak. However, they are not, as the following Theorem shows:

**Theorem.** *If a family $F$ is optimal in the sense of some optimality criterion that is final and shift-invariant, then every transformation function $f$ from this optimal family $F$ which has the properties $f(x) = x + o(x)$ and $f(-x) = -f(x)$ is equal to $f(x) = a \cdot \tanh(x/a)$ for some real number $a$.*

**Proof.** The proof is similar to the proofs given in [4], [8] that a similar function is the optimal activation function for neural networks. Since $G$ is a a connected finite-dimensional Lie group of rescalings of the set of real numbers $R$ onto itself that contains all linear transformations, $G$ consists only of fractionally linear functions [8]. Similar to [8], we then prove that the optimal family $F_{opt}$ exists and is *shift-invariant* in the sense that $F_{opt} = T_c(F_{opt})$ for all real numbers $c$. Thus, we get a functional equation with known solutions.

**Conclusion.** Since we use the transformation $f(x)$ to generate values $c \cdot f(k \cdot x)$, the use of $\tanh(x/a)$ is equivalent to the use of the transformation function $\tanh(x)$. Thus, our theorem says, in effect, that for every reasonable criterion, among all non-linear transformation functions, the hyperbolic function is indeed optimal.

*Acknowledgments.* This work was supported in part by NASA grants NCC5-209 and NCC 2-1232, NSF grants CDA-9522207 and 9710940 Mexico/Conacyt, Air Force Office of Scientific Research grant F49620-00-1-0365, and by Grant No. W-00016 from the U.S.-Czech Science and Technology Joint Fund.

The authors are thankful to anonymous referees for their valuable comments.

## References

[1] J. Abello, V. Kreinovich, H. T. Nguyen, S. Sudarsky, and J. Yen, "Computing an appropriate control strategy based only on a given plant's rule-based model is NP-hard", *Proc. NAFIPS'94*, IEEE Press, pp. 331–332.

[2] B. D. O. Anderson and J. B. Moore, *Optimal Control: Linear Quadratic Methods*, Prentice Hall, Englewood Cliffs, NJ, 1990.

[3] G. Klir and B. Yuan, *Fuzzy Sets and Fuzzy Logic: Theory and Applications*, Prentice Hall, Upper Saddle River, NJ, 1995.

[4] V. Kreinovich and C. Quintana. "Neural networks: what non- linearity to choose?," *Proc. 4th University of New Brunswick Artificial Intelligence Workshop*, Fredericton, New Brunswick, Canada, 1991, pp. 627–637.

[5] V. Kreinovich, A. Lakeyev, J. Rohn, and P. Kahl, *Computational complexity and feasibility of data processing and interval computations*, Kluwer, Dordrecht, 1997.

[6] M. Margaliot and L. Langholz, "Hyperbolic optimal control and fuzzy control", *IEEE Trans. SMC, Part A*, 1999, Vol. 29, No. 1, pp. 1–10.

[7] M. Margaliot and L. Langholz, *New Approaches to Fuzzy Modeling and Control: Design and Analysis*, World Scientific, Singapore, 2000.

[8] H. T. Nguyen and V. Kreinovich, *Applications of continuous mathematics to computer science*, Kluwer, Dordrecht, 1997.

[9] H. T. Nguyen and E. A. Walker, *First Course in Fuzzy Logic*, CRC Press, Boca Raton, FL, 1999.

[10] L. A. Zadeh, "Fuzzy Sets", *Information and Control*, 1965, Vol. 8, pp. 338–353.