Neutrosophic Application for Decision Logic in Robot Intelligent Control Systems

Victor Vlădăreanu
Dept. of Mechatronics and Robotics
Institute of Solid Mechanics, Romanian Academy
Bucharest, Romania

Florentin Smarandache
Dept. of Mathematics and Science
University of New Mexico
Gallup, USA

Luige Vlădăreanu
Dept. of Mechatronics and Robotics
Institute of Solid Mechanics, Romanian Academy
Bucharest, Romania

***Abstract*—The paper presents the research undertaken for extending the functionality of a regular fuzzy multi-input decision support system through the application of basic neutrosophy principles. The application is designed and simulated using tools already available for the fuzzy implementation, adapted to the requirements of the neutrosophic extension. The results show an increase in the decision classification index, while improving the environmental accuracy of the considered model.**

Keywords—robotics, neutrosophic set, intelligent control systems, decision making, computational intelligence

I. INTRODUCTION

The paper presents a proposed solution for the implementation of a generalized neutrosophic inference system (NSIS) by extending the already available tools for the implementation of fuzzy logic, amended to the particularities of neutrosophic logic. The end result is a neutrosophic logic controller (NSLC). As will be discussed in the appropriate section, its implementation can function either on the original neutrosophic principles (T-norms and T-conorms), or on a fuzzy-T2 schema, which can also be seen as a subset of neutrosophic logic. That is to say, the uncertainty dimension can be either used in the original formulae, or directly as a determinant of the fuzzy degree of membership.

The history of fuzzy logic starts with the paper published in 1965 by L.A. Zadeh, entitled ‘Fuzzy sets’, in which the author introduces his new approach to set theory. Essentially, a fuzzy set is an extension of a classical bivalent (crisp) set with ‘a membership function which assigns to each object a grade of membership between 0 and 1 [1]. Subsequent detailed investigations made by Mamdani [2] and Takagi and Sugeno [3] have led to Fuzzy Logic becoming an increasingly appealing alternative to classical control for an array of systems [4].

Fuzzy Logic has long been used in academia and in industry and is one of the more palpable staples of artificial intelligence in use in the world today. Fuzzy logic controllers have been proven to be robust, relatively easy to design [5] and, although a unified algorithm for parameter selection and optimization is still sought after [6, 7], they seem to suffer from no one major flaw while providing a number of important benefits (expert knowledge emulation being perhaps chief among them). There are a number of implementations of various algorithms for the optimization of fuzzy inference systems’ parameters, such as genetic algorithms and neural networks.

Neutrosophy extends fuzzy logic by adding the dimension of uncertainty to the considered model. This is especially useful in information fusion dealing with multiple sources of sensor data. While still in its beginning, neutrosophy enjoys remarkable interest from world-wide research teams due to a proven record of improving inference system models [8, 9]. With applications in artificial intelligence, business, marketing, planning, control theory and image processing, it is one of the fastest developing new fields of study in the world today. Neutrosophic reasoning components work very well with database expert systems and provide a large boost to the current knowledge in decision support systems [10], making them particularly well suited for robotic applications.

Innovative methods were developed, such as the Vlădăreanu - Smarandache method for force-position robot control [11] by applying neutrosophic logic in determining the angular error of the actuator drive control loop in the robot joints.

The generalized area where a robot works can be defined in a constraint space with six degrees of freedom (DOF), with position constrains along the normal force of this area and force constrains along the tangents. On the basis of these two constrains there is described the general scheme of hybrid position and force control in figure 1. Variables XC and FC represent the Cartesian position and the Cartesian force exerted onto the environment. Considering XC and FC expressed in specific frame of coordinates, the selection matrices Sx and Sf can be determined, which are diagonal matrices with 0 and 1 diagonal elements, and which satisfy relation: $Sx + Sf = Id$, where Sx and Sf are methodically deduced from kinematics constrains imposed by the working environment [16-18].

978-1-5090-5795-5/17/$31.00 ©2017 IEEE

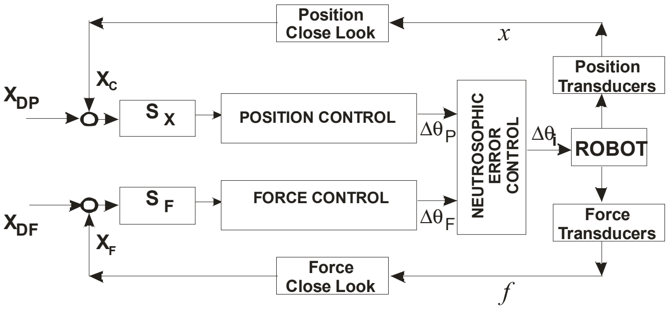

Fig. 1. Architecture of the Robot Neutrosophic Control (RNC)

In order to determine the control relations in this situation, ΔX_P – the measured deviation of Cartesian coordinate command system is split in two sets: ΔX^F corresponds to force controlled component and ΔX^P corresponds to position control with axis actuating in accordance with the selected matrixes S_f and S_x. If there is considered only positional control on the directions established by the selection matrix S_x there can be determined the desired end - effector differential motions that correspond to position control in the relation: $\Delta X_P = K_P\, \Delta X^P$, where K_P is the gain matrix, respectively desired motion joint on position controlled axis: $\Delta\theta_P = J^{-1}(\theta) \cdot \Delta X_P$ [19 - 21].

Now taking into consideration the force control on the other directions left, the relation between the desired joint motion of end-effector and the force error ΔX_F is given by the relation: $\Delta\theta_F = J^{-1}(\theta) \cdot \Delta X_F$, where the position error due to force ΔX_F is the motion difference between ΔX^F– current position deviation measured by the control system that generates position deviation for force controlled axis and ΔX_D – position deviation because of desired residual force. Noting the given desired residual force as F_D and the physical rigidity K_W there is obtained the relation: $\Delta X_D = K_W^{-1} \cdot F_D$.

Thus, ΔX_F can be calculated from the relation: $\Delta X_F = K_F (\Delta X^F - \Delta X_D)$, where K_F is the dimensionless ratio of the stiffness matrix. Finally, the motion variation on the robot axis matched to the motion variation of the end-effectors is obtained through the relation: $\Delta\theta = J^{-1}(\theta)\, \Delta X_F + J^{-1}(\theta)\, \Delta X_P$. Starting from this representation the architecture of the hybrid position – force control system was developed with the corresponding coordinate transformations applicable to systems with open architecture and a distributed and decentralized structure.

This paper presents a robot position control application on a single axis, as hybrid force-position control may be developed in future works by generalizing neutrosophic logic to bi-dimensional space. Respectively for hybrid force-position control on ***n*** degrees of freedom (n DOF) by generalizing NSL to 2***n***-dimensional space.

The designed system is tested on a controller application for the position control of a simple direct current motor, representing the position actuator for one joint of a small robot. The inputs to the motor are the voltage demand used to control the system and the value of the load applied to the motor, which will be a negative value in this layout. The outputs are the speed of the motor, which will later be fed into an Integral block in order to obtain position control, and the value of the armature current. This will be sunk into a scope which allows for monitoring the armature current.

The implementation is done using coded functions and classes, but is also shown in a schematic diagram for purposes of visualization. It should be noted that both coded versions take little time to run, while the models will of course take longer, especially for the double fuzzy version. Both versions of the code draw heavily from the existing fuzzy logic implementations in the Matlab / Octave environment. The opportunity for actual practical implementation of both versions is discussed in the conclusions section.

The rest of the paper is divided as follows. Chapter 2 presents the generalized neutrosophic model with a focus on the elements required for the present application. Chapter 3 outlines the controller logic, implementation of the NSLC (NeutroSophic Logic Controller) and its functionality, while also discussing an experimental application of the controller and details the actual use-case simulation. Chapter 4 discusses the obtained results and attempts to draw conclusion for the present and future research involving neutrosophic models and decision support systems.

II. Neutrosophic Model

The operating algorithm for a fuzzy logic controller (FLC) consists of three stages: Fuzzification, Inference and Defuzzification. All FLCs used for the simulations have a set input and output range of ±1 which is then adjusted to fit the particulars of the system being controlled. Therefore, the FLC is preceded and succeeded by input and output scaling, respectively. Fuzzification is the process of mapping the inputs to ‘linguistic variables’ (fuzzy sets) and determine the degree of membership for each respective set [12]. This membership value is of course entirely dependent upon the shape and layout of the MFs.

The Fuzzy Inference System (FIS) is the key point of fuzzy logic, which is designed to mimic human reasoning [7]. It connects the fuzzified inputs with the fuzzified outputs through a set of “IF – THEN” rules using the previously defined linguistic variables. A ‘rule base’ is merely a table representation of the rules used within a FIS. A number of rule bases are employed depending on the number of inputs, desired controller performance, controller and process particulars and are the defining characteristic of the respective FLCs they are used in.

The FIS rules use *and* and *or* operators to connect the various inputs (linguistic variables) to a prescribed output fuzzy set (also a linguistic variable).The degree of support for each rule and therefore for its respective output fuzzy set is to be found among the degrees of support for the input linguistic variables that are part of that rule. When the *and* operator is used, the minimum of all degrees of support from the input will be used as the degree of support for the output, while the *or* operator will determine the maximum of those values to be used.

The overlapping of MFs means that more than one rule will be used (‘fired’). This leads to a number of output fuzzy sets, which must then be aggregated into a final output fuzzy set. While there are other methods for aggregation, the ‘max’ method is the general standard and as such it was used in all simulations for this paper. It implies mapping the resulted output sets onto the output range and taking the maximum value in areas where they overlap.

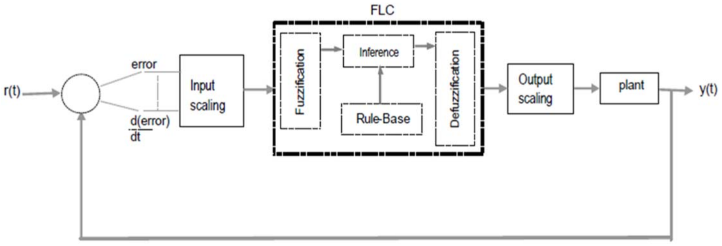

Fig. 2. FLC implementation diagram

Fuzzy set theory expands conventional ('crisp') sets with a 'membership value' (between 0 and 1) which expresses the degree to which a certain element belongs to a set. The relation between a value that is part of a set and its membership value is called a Membership Function (MF). The MFs for a given input must cover the entire universe of discourse [12] and they can and do overlap. Fuzzy Logic Controllers (FLCs) use fuzzy logic to mimic the way an experienced operator would go about controlling the process. A diagram of the implementation of a FLC is shown in Figure 2 [12].

By contrast, neutrosophy allows for an increase in the dimensionality of each input parameter, providing more information that can be coherently modelled and input to the inference system [10].

Neutrosophy is meant to be a unifying theory for the design and implementation of decision support systems. As such, it is a generalization of fuzzy logic. A neutrosophic logic statement includes values for truth, falsehood and indeterminacy, where the appropriate memberships are real values. Similarly to fuzzy logic and in contrast to probability theory, there are no restrictions placed on the sum of the resulting components – in probability theory the sum of all possible outcomes must be 1. The truth and falsehood parameters tie neutrosophy to the existing treatment of decision support problems, to which is added support for modelling the indeterminacy, which expresses the percentage of unknown parameters or states [13].

Inside a universe of discourse *U* and with respect to a set *M* included in *U*, and element x is noted as *x(T,I,F)*, with the following properties [14,15]:

- *x* has a value of truth *t* of belonging to the set *M*
- *x* has a value of indeterminacy *i* of belonging to the set *M*
- *x* has a value of falsehood *f* of belonging to the set *M*

The transformation from a standard operating model using fuzzy logic to a neutrosophic model is done using the formulae set forth in the Dezert-Smarandache Theory (DSmT) [14]. Let $\theta = \{\theta_1, \theta_2, \dots, \theta_n\}$ be a finite set made up of *n* exhaustive elements (this can be assumed without loss of generality, see [14]). The DSmT hyper-power set D^{Θ} is then defined as the set of all composite propositions built from those elements using the *reunion* and *intersection* operators. This means that

$$\emptyset, \theta_1, \dots, \theta_n \in D^{\theta}$$

$$\forall A, B \in D^{\theta} \rightarrow A \cap B \in D^{\theta}, A \cup B \in D^{\theta}$$

By defining an application $m(.) \colon D^{\theta} \rightarrow [0,1]$ within this set, there results $m(\emptyset) = 0$ and $\sum_{A \in D^{\theta}} m(A) = 1$, where *m(A)* is the generalized basic belief assignment of *A* [14,15].

In working with multiple sources of information, the DSm rule of combination states:

$$\forall C \in D^{\theta}, m_{M^{T}(\theta)}(C) \equiv m(C) = \sum_{\substack{A,B \subset D^{\theta} \\ A \cap B = C}} m_1(A) \cdot m_2(B)$$

With D^{θ} closed under the set operators of reunion and intersection, the DSm rule of combination results in *m(.)* being a proper belief mass, with $m(.) \colon D^{\theta} \rightarrow [0,1]$, being commutative, associative and extendable to an unlimited number of sources [14,15].

Alternatively, the transformation can be done using fuzzy-T2 style operators, whereby the third dimension is actually the fuzzification of the fuzzy degree of membership (i.e. the second dimension). This relates well to applications where the uncertainty specifically models the reliability of a sensor network or input device, which can be statically estimated and traced to each individual output [13].

This can be exemplified in the test application model, where feedback inputs are sampled from a Gaussian distribution with an appropriate variance for each for the inputs. This allows the model to incorporate the uncertainty normally found in such practical applications. It is also completely feasible that such values should be known and traced back individually to their respective field equipment. The exact expression of the third dimension of each tuple need not be specifically the variance, as long as it correctly models the uncertainty in the system. However, this is a possible topic for future research and is outside the scope of the current application.

III. Simulation and Testing / Implementation

The implementation of the proposed generalized model is achieved by using the packages already available for fuzzy implementations in a Matlab / Octave environment, both for visualization and coding.

Each input value maps to an object of a class defined to handle the multiple dimensions as properties, thereby forming and coherently expressing the tuple needed. This representation was also chosen for its ease of expansion, as may be the case with future research – neutrosophic logic systems can be extended to four- or five- dimensional logic sets [13]. The coded functions which create the neutrosophic logic objects are implemented to be a parallel representation of those already existing as part of the fuzzy logic toolbox or package, where possible. The second implementation version, using the double-fuzzy style operators, particularly makes full use of the existing *evalfis, readfis,* etc. functions. As such, the *fuzzy-logic-toolkit* package and *fuzzy logic* toolbox can be considered dependencies for the Octave and, respectively, Matlab environments.

The controller schematic implementation is shown in Figure 4. The chosen test application is the position control of a simple direct current motor representing a robotic joint actuator. The two input feeds are the position error and its derivative, calculated independently. This is possible because the derivative of the position is actually the motor speed and is in fact a more accurate representation of an actual practical application, with available date from both types of transducers. The third dimension of indeterminacy takes into account the standard reliability of position and speed sensors along the feedback path.

With a view to making the simulations as realistic as possible, the effects of disturbance and noise along the circuit, integrator windup and maximum armature current demand are included. The system is subjected to a step input of value RV (Reference Value). Also included in the simulation are the Disturbance (*Ds*), Noise (*N*) along the feedback path, Load (*TL*) and the time at which the load occurs (*TLs*) which can be found in the Load Block, and the sample time (*Ts*) in the case of digital systems. These are all declared as global variables and only need to be updated once at the beginning of each session.

The overall transfer function of a DC Motor is:

$$P'(s) = \frac{K}{LJs^2+(RJ+BL)s+RB+K^2}$$

for speed control, which is then divided by s for position control. In the end, the transfer function of the system will be

$$P(s) = \frac{K}{LJs^3+(RJ+BL)s^2+(RB+K^2)s}.$$

All of the parameters are declared within the environment as global variables and are initiated once at the start of the session.

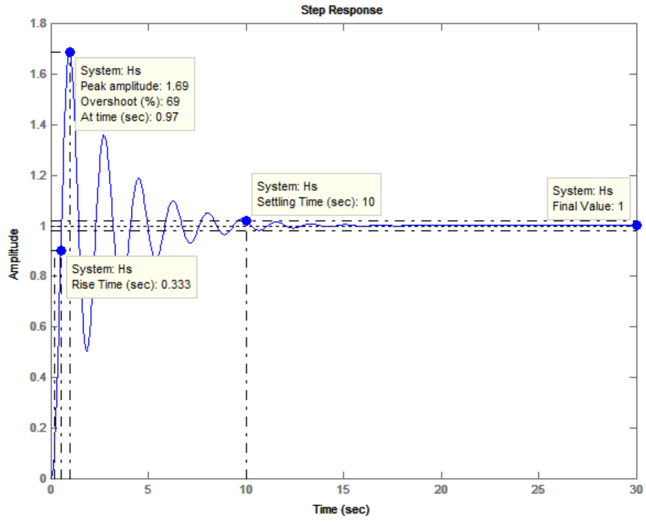

Fig. 3. General stable system response

After a simulation is run, the transient system response is investigated based on the following performance metrics: overshoot, rise time, settling time and steady state error. Figure 3 shows a random stable system response to a step input for 30 seconds, indicating those characteristics.

The actual implementation of the controller schematics is hard coded using the described functions and classes for the neutrosophic logic model. For visualization purposes, Figures 4 and 5 show the controller schematics in Matlab/Simulink for the native norm version and the fuzzy-T2 style version, respectively.

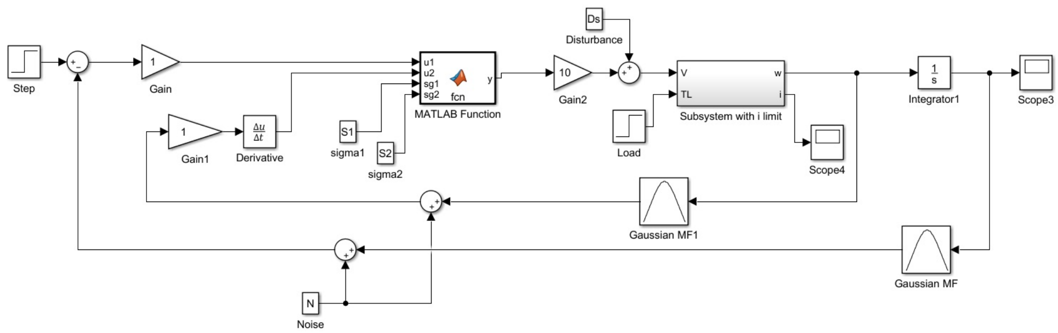

Fig. 4. Controller schematics / native norm

It is worth noting the similarities and discrepancies between the two implementation versions. For both, the position and speed feedbacks are resampled using a Gaussian function form with the expected variance of an actual field transducer. In a practical implementation, the actual values for this are parameterized and can be adjusted according to the known specifications provided by the manufacturer. This allows us to introduce uncertainty into the simulation in a way that corresponds to actual practical scenarios.

The difference between the two versions mainly consists in the treatment of the third dimension, uncertainty, as discussed in the theoretical approach. The known variance of sensor inputs is inserted directly into the coded function in the first version (notice the m-function interpreter block that acts as the central controller of the design). In the second, they are a direct expression of the degree of membership of the second dimension values (which models fuzziness). Each input is treated separately together with its own reliability rating and fuzzified, the output of which is then re-fuzzified into the final FLC controller.

The use-case diagram shown in Figure 6 describes the generalized algorithm for the decision making process.

As much as possible, the implementation is parameterized to allow for a wide array of re-application and future development.

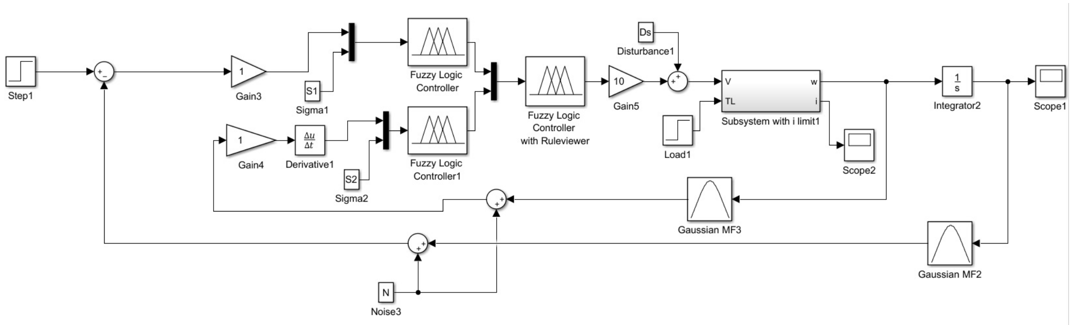

Fig. 5. Controller schematics / fuzzy style norm

It is clear from the onset of neutrosophic theory that, as with fuzzy, various norms and applications can be defined over the considered set and universe of discourse.

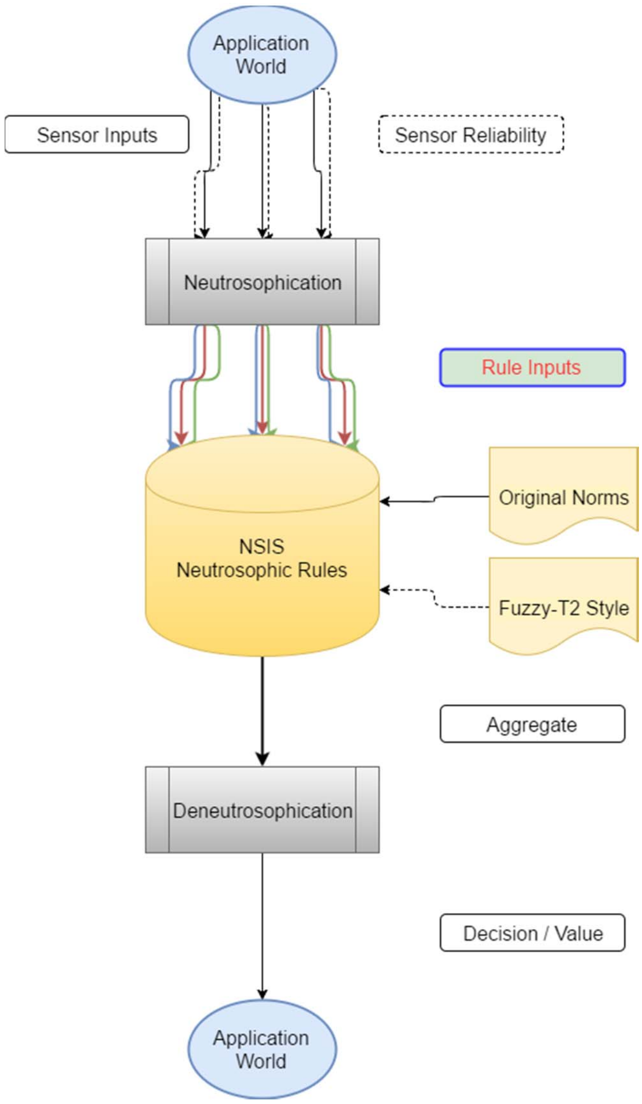

Fig. 6. Use-case diagram

These can be passed as user-defined functions to the object creator of the neutrosophic class, if needed, and could be the object of future research scenarios for different decision support systems and controllers.

IV. Conclusions

The paper shows an inference system application using neutrosophic logic for the control of a small direct current motor used as a robotic actuator. The theoretical aspects of neutrosophy and the practical considerations of decision support systems and controllers are explored, with the application providing a practical demonstration of the proof of concept. Standard controller concerns and design parameters are explained and discussed briefly, so as to give context to the implementation.

Also of note is the programming framework for neutrosophic applications which is implemented, with a view toward generalization, parameterization and reusability. The code builds upon the existing libraries and toolboxes available for fuzzy logic, of which neutrosophy can be seen as a generalized, unifying theory.

As with the more traditional fuzzy controllers, the neutrosophic logic controller needs to be tuned for the particular application and context it is working in [22, 23]. There is currently no single algorithm guaranteed to find the best configuration among the many different options that would map to search space dimensions in an optimization problem.

A possible solution is to use an evolutionary algorithm to tune the scaling gains of the input membership functions, the fuzzy rule base, or both. While there are a number of authors, such as Byrne who look at GA – tuned fuzzy structures, papers using a PSO algorithm are few and far between. Both these approaches provide intriguing options for the future of both fuzzy and neutrosophic logic control, as well as further theoretical and applicative research.

The uniqueness of neutrosophic theory allows possibilities for further increasing the dimensionality of the neutrosophic object. It could be used, for example, for modelling the general noise and disturbance within the system. This would be particularly useful for industrial applications, in which such factors rise in importance considerably.

Acknowledgment

This work was developed with the support of MEN-UEFISCDI, PN-II-PT-PCCA-2013-4, VIPRO project No.009/2014, Romanian Academy and "Joint Laboratory of Intelligent Rehabilitation Robot" collaborative research agreement between Romanian Academy by IMSAR and Yanshan University by China Science and Technical Assistance Project for Developing Countries (KY201501009), China.

REFERENCES

[1] Zadeh, L.A., ‘Fuzzy Sets’, Information and Control, no. 8, 1965

[2] Procyk, T.J., Mamdani, E.H.,‘A linguistic self-organizing process controller’, Automatica, vol. 15, no. 1, 1979

[3] Takagi, T., Sugeno, M., ‘Fuzzy identification of systems and its application to modelling and control’ IEEE Trans. Syst., Man and Cybern., vol. SMC-15, 1985

[4] Thomas, D.E., Armstrong-Helouvry, B., ‘Fuzzy Logic Control – A Taxonomy of Demonstrated Benefits’, Proceedings of the IEEE, vol. 83, no. 3, Mar. 1995

[5] Jantzen, J., ‘Foundations of Fuzzy Control’, John Wiley & Sons, Ltd, Chichester, UK, 2007

[6] Ko, C., Wu, C., “A PSO-tuning Method for Design of Fuzzy PID Controllers”,Journal of Vibration and Control OnlineFirst, published on December 20, 2007

[7] Byrne, J.P., ‘GA-optimization of a fuzzy logic controller’, Master’s Thesis, School of Electronic Engineering, Dublin City University, 2003

[8] Smarandache, F.,, A Unifying Field in Logics: Neutrosophic Logic. Neutrosophy, Neutrosophic Set, Neutrosophic Probability: Neutrsophic Logic. Neutrosophy, Neutrosophic Set, Neutrosophic Probability, Infinite Study, 2005

[9] The optimization of intelligent control interfaces using Versatile Intelligent Portable Robot Platform, Vladareanu, V; Munteanu, RI; Mumtaz, A; Smarandache, F; Vladareanu, L, Procedia Computer Science Volume: 65 Pages: 225-232 DOI: 10.1016/j.procs.2015.09.115, 2015 Accession Number: WOS:000373831000026, ISSN: 1877-0509

[10] Smarandache, F. Neutrosophic Theory and Its Applications, Vol. I: Collected Papers. Infinite Study, 2014.

[11] Smarandache, F, Vlădăreanu, L, "Applications of neutrosophic logic to robotics: An introduction." Granular Computing (GrC), 2011 IEEE International Conference on. IEEE, 2011.

[12] Giaouris, D., ‘SDL: Introduction to Fuzzy Logic’, School of Electrical, Electronic and Computer Engineering, Newcastle University, 2011

[13] Smarandache, F, “Neutrosophy a new branch of Philosophy”, Multi. Val. Logic – Special Issue: Neutrosophy and Neutrosophic Logic, 2002, Vol. 8(3), pp.297-384, ISSN:1023-6627

[14] Florentin Smrandache, Jean Dezert, “Advances and Applications of Information Fusion”, American Research Press, Rehoboth, 2004

[15] Gal, A., Vladareanu, L., Smarandache, F., Yu, H., & Deng, M. (2012). Neutrosophic Logic Approaches Applied to” RABOT” Real Time Control. Neutrosophic Theory and Its Applications. Collected Papers, 1, 55-60

[16] L.D.Joly, C.Andriot, V.Hayward, Mechanical Analogic in Hybrid Position/Force Control, IEEE Albuquerque, New Mexico, pg. 835-840, April 1997

[17] Vladareanu L, The robots' real time control through open architecture systems, cap.11, Topics in Applied Mechanics, vol.3, Ed.Academiei 2006, pp.460-497, ISBN 973-27-1004-7

[18] G. Eason, B. Noble, and I.N. Sneddon, “On certain integrals of Lipschitz-Hankel type involving products of Bessel functions,” Phil. Trans. Roy. Soc. London, vol. A247, pp. 529-551, April 1955. (*references*)

[19] M. Young, The Technical Writer’s Handbook. Mill Valley, CA: University Science, 1989.

[20] Vladareanu L, Sandru OI, Velea LM, YU Hongnian, The Actuators Control in Continuous Flux using the Winer Filters, Proceedings of Romanian Academy, Series A, Volume: 10 Issue: 1 Pg.: 81-90, 2009, ISSN 1454-9069

[21] Yoshikawa T., Zheng X.Z. - Coordinated Dynamic Hybrid Position/Force Control for Multiple Robot Manipulators Handling One Constrained Object, The International Journal of Robotics Research, Vol. 12, No. 3, June 1993, pp. 219-230

[22] N. D. Thanh, M. Ali, L. H. Son, A Novel Clustering Algorithm on Neutrosophic Recommender System for Medical Diagnosis, Cognitive Computation. (2017), pp 1-19, DOI: 10.1007/s12559-017-9462-8

[23] M. Ali, and F. Smarandache, Complex Neutrosophic Set, Neural Computing and Applications, Vol. 25, (2016),1-18. DOI: 10.1007/s00521-015-2154-y.