World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:11, No:11, 2017

Implementation of a Paraconsistent-Fuzzy Digital PID Controller in a Level Control Process

H. M. Côrtes, J. I. Da Silva Filho, M. F. Blos, B. S. Zanon

Open Science Index, Computer and Information Engineering Vol:11, No:11, 2017 publications.waset.org/10008182/pdf

Abstract**—**In a modern society the factor corresponding to the increase in the level of quality in industrial production demand new techniques of control and machinery automation. In this context, this work presents the implementation of a Paraconsistent-Fuzzy Digital PID controller. The controller is based on the treatment of inconsistencies both in the Paraconsistent Logic and in the Fuzzy Logic. Paraconsistent analysis is performed on the signals applied to the system inputs using concepts from the Paraconsistent Annotated Logic with annotation of two values (PAL2v). The signals resulting from the paraconsistent analysis are two values defined as Dc - Degree of Certainty and Dct - Degree of Contradiction, which receive a treatment according to the Fuzzy Logic theory, and the resulting output of the logic actions is a single value called the *crisp value*, which is used to control dynamic system. Through an example, it was demonstrated the application of the proposed model. Initially, the Paraconsistent-Fuzzy Digital PID controller was built and tested in an isolated MATLAB environment and then compared to the equivalent Digital PID function of this software for standard step excitation. After this step, a level control plant was modeled to execute the controller function on a physical model, making the tests closer to the actual. For this, the control parameters (proportional, integral and derivative) were determined for the configuration of the conventional Digital PID controller and of the Paraconsistent-Fuzzy Digital PID, and the control meshes in MATLAB were assembled with the respective transfer function of the plant. Finally, the results of the comparison of the level process control between the Paraconsistent-Fuzzy Digital PID controller and the conventional Digital PID controller were presented.

Keywords**—**Fuzzy logic, paraconsistent annotated logic, level control, digital PID.

I. Introduction

MOTIVATED by the need to consider real situations that do not fall into the strict rules of classical logic, parallel studies have been created and have resulted in other logics that are alternatives to the classical one [1]. Non-classical logics investigate, among other things, the situations excluded from the classical logic which are, for example, the existing logical values other than the "True" and the "False", allowing better adaptation of concepts such as; uncertainties, ambiguities, paradoxes and inconsistencies [2], [3].

Classical logic considers only two possible states "true" or "false". However, by considering electronic devices, there are many data noises (inconsistencies and ambiguities) that classical logic using the law of excluded middle is unable to be applied in the face of these situations, at least directly [4].

Paraconsistent logic (PL) [5] was first proposed in the works published by S. Jáskowski (in 1948) and by Da Costa (in 1954), independently, its main property is to accept the contradiction in its foundations and thus to allow a non-trivial treatment of contradictory signals. Paraconsistent logics for accepting contradictions as theses in their structure can be applied in several areas of knowledge, being a good alternative of application in systems of Control, Artificial Intelligence and Automation [6], [7].

The research developed in the area of Automation and Artificial Intelligence (AI), in the sense of implementing circuits and control systems using non-classical logics as alternatives of classical, have shown interesting results creating a more effective way to model complex systems based on a "behavior" more close to the human being [8], [9]. The use of the *Fuzzy* control to model complex processes of indefinite nature has been well accepted in industrial applications. PL and Fuzzy Logic (FL) are among those that allow the implementation of the systems of structured controls in actions that more fully express the real world [9].

PLs and FLs provide an effective means of capturing approximate values of the inaccurate nature of the real world. FL brings new forms of control that allow complex systems to be designed based on the knowledge of a specialist, that is, a human operator with heuristic knowledge and not necessarily have underlying knowledge of dynamic control processes [10]. The theory of classical logic does not allow the treatment of contradictory signals in a non-trivial way. From this perspective, the paper presents the junction of the theories of the two non-classical logics; PL and FL in a control system. This junction enables control systems work with contradictory signals without output trivialization [8], [10].

II. Paraconsistent Annotated Logic with Annotation of Two Values - PAL2v

In the Paraconsistent Annotated Logic (PAL), investigated in [1], annotations can be interpreted as degrees of evidence to the proposition. In [2], inconsistencies appear with information that is represented in the form of propositions accompanied by degrees of evidence. To understand the Para-

H. M. Côrtes is a student from the Master of Mechanical Engineering of University of Santa Cecília, Santos, SP, Brazil (phone: +55-27-999441960; e-mail: hyghorcortes@gmail.com).

J. I. Da Silva Filho, is a professor of Paraconsistent Logic at the University of Santa Cecília, is the Coordinator of the Paraconsistent Annotated Logic Group PALG and member of the Group of the Logical and Science theory - IAS (Institute of Advanced Studies of the University of São Paulo (USP)), Santos, SP, Brazil ((phone: +55-13-32027259, e-mail: inacio@unisanta.br).

M. F. Blos is the Coordinator of the Master's Degree course in Mechanical Engineering of University of Santa Cecília and is researcher of the Paraconsistent Annotated Logic Group PALG, Santos, SP, Brazil (phone: +55-41-996870429; e-mail: mauricioblos@unisanta.br).

B. S. Zanon is a student from the Master of Mechanical Engineering of University of Santa Cecília, Santos, SP, Brazil (phone: +55-22-999645182; e-mail: bzanon05@hotmail.com).

Open Science Index, Computer and Information Engineering Vol:11, No:11, 2017 publications.waset.org/10008182/pdf

Fuzzy control system, is presented a summary of the PAL with the annotation of two values (**PAL2v)** and some associated results and concepts seen in more depth in [2], [5].

In **PAL2v** each proposition P is accompanied by an annotation with two components (μ, λ). The information is of type P (μ, λ) where: μ symbolizes the degree of Favorable evidence and λ symbolizes the degree of Unfavorable evidence, both attributed to proposition P [2].

In PAL2v exists an associated Lattice, as show in Fig. 1:

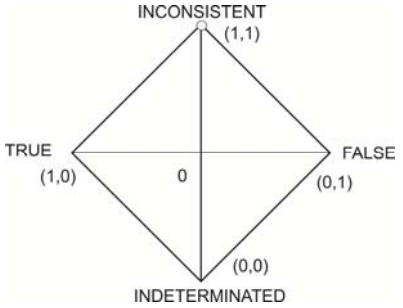


Fig. 1 Associated Lattice of the PAL with annotation of two-values (PAL2v)

The values of the degrees of evidence are within a closed interval between 0 and 1 belonging to the set of real numbers.

The representation of the logical states can be made by a lattice, where four extreme vertices go determine the attributions of logical values given to the proposition considered as: True (t), False (F), Inconsistent ⊤ or Indeterminate ⊥ [2], [5].

We can consider the values of the degree of certainty D_c and the degree of contradiction D_{ct}, by applying the algorithm "**Para-Analyzer**" [2] as explained below:

Algorithm "**Simplified PARA-ANALYZER - Paraconsistent Annotated Logic with annotation of two values-PAL2v**"

** / Input variables * /*

μ */Degree of favorable evidence attributed to Proposition P (value between 0 and 1) * /

λ */Degree of unfavorable evidence attributed to Proposition P (value between 0 and 1) * /

** / Mathematical expressions * /*

Being:

$0 \leq \mu \leq 1$ and $0 \leq \lambda \leq 1$

$\boldsymbol{D_{ct}} = \mu + \lambda - 1$

$\boldsymbol{D_c} = \mu - \lambda$

/Output variables/

$\boldsymbol{D_{ct}}$ */ Degree of contradiction attributed to Proposition *P* (value between -1 and +1) */

$\boldsymbol{D_c}$ */ Degree of certainty assigned to Proposition *P* (value between -1 and +1) */

$\boldsymbol{D_{ct}} = S_{2a}$

$\boldsymbol{D_c} = S_{2b}$

*/ **END** */

The degree of certainty D_c is related to the interval between the extreme assignments of "Truth" and "Falsehood" to the proposition, and the degree of contradiction D_{ct} is related to the extreme assignments of "Inconsistency" and "Indetermination" given to the proposition. The values of D_c (degree of certainty) and of D_{ct} (degree of Contradiction) obtained through the equations presented in the simplified *Para-Analyzer* algorithm are normalized between -1 and +1. The normalization of these values, which are the result of the paraconsistent analysis of two contradictory signals, allows the construction of two classes of *fuzzy* sets where the pertinence functions are elaborated, causing the signals to receive a treatment according to FL considerations [2], [5].

III. Fuzzy Set and FL

In this section, are presents the basic concepts of *Fuzzy* set theory and FL. The considerations presented here are those necessary to understand this text, in [11] and [12] more detailed information is found.

Fuzzy Sets - Terminology

Being **U** a collection of objects denoted generically by $\{u\}$ which may be discrete or continuous, **U** is called the "Universe of discourse" and u generically represents an element of **U**.

Definition 1 – *Fuzzy* Set

A Fuzzy set **F** in the Universe of discourse **U** is characterized by a Membership function u_f that has values in the interval [0,1]. Nominally: u_f: **U** → $[0, 1]$.

A Fuzzy set can be seen as a generalization of the concept of an ordinary set whose *Membership* function has only two values { 0,1 }. Therefore, a *Fuzzy* set **F** in **U** can be represented by a set of ordered pairs of a generic element u and its kind of Membership function by (1):

$$F = \left\{\left(u, u_f(u)\right) \mid u \in \boldsymbol{U}\right\} \quad (1)$$

When **U** is continuous, a Fuzzy set **F** can be written as (2):

$$F = \int U\, u_f(u)/u \quad (2)$$

When **U** is discrete, a Fuzzy set **F** can be written as (3):

$$\sum u_f(u_i)/u_i \quad (3)$$

Definitions

a- The support of a *Fuzzy* set **F** is the unique value set (*crisp*) of all points u in **U** such that $u_f(u) > 0$. In particular, the element u in **U** to which u_f=0.5 is named *single point* and the *Fuzzy* set whose support is the *single point* in **U** with u_f=1,0 is referred to as a *Fuzzy singleton*.

b- Let **A** and **B** two *Fuzzy* sets in **U** with *Membership* functions u_a and u_b, respectively, the set of operations of; Union, Intersection and Complement, for the Fuzzy sets are defined through *Membership* Functions.

Union in (4)

$$\boldsymbol{U}_{a \cup b}(u) = \mathbf{max}\{u_a(u),\, u_b(u)\} \quad (4)$$

Open Science Index, Computer and Information Engineering Vol:11, No:11, 2017 publications.waset.org/10008182/pdf

Intersection in (5)

$$\boldsymbol{U}_{a\cap b}(u) = \mathbf{min}\{u_a(u),\, u_b(u)\} \quad (5)$$

Complement in (6)

$$\boldsymbol{U}_{\bar{a}}(u) = 1 - u_a(u) \quad (6)$$

The procedures according to *Fuzzy* set theory provide a basis for a systematic way of manipulating concepts of vagueness and imprecision. These procedures allow us to use set theory to represent linguistic variables [11], [13]. A linguistic variable can be considered as a variable whose value is a *Fuzzy* number, or as a variable whose value is defined in linguistic terms.

The linguistic variable is characterized by a quintuple $(x, \boldsymbol{T}(x), \boldsymbol{U}, \boldsymbol{G}, \boldsymbol{M})$ where:

a) x is variable name.
b) $\boldsymbol{T}(x)$ is the set of terms of x, that is, the set of names of linguistic values of x with each value being a Fuzzy number defined in U.
c) U is the Universe of discourse.
d) G is the syntactic rule for generating the names of the value of x.
e) M is the semantic rule to associate with each value its meaning.

In the definition of a *fuzzy* implication, *fuzzy* reasoning mechanisms are analyzed and in this process the *Membership* functions generate **n** *Fuzzy* propositions of the type, (7):

$$((x_1, x_2, \ldots, x_n)\ is\ \boldsymbol{R}) \quad (7)$$

where: x_1, x_2, ..., x_n are the names of **m** linguistic variables whose discourse universes are respectively, X_1, X_2, ..., X_n and $\boldsymbol{R}$ is a *Fuzzy* relation defined in X_1x X_2x ...x X_n. Depending on the process combinations of **n** *Fuzzy* statements can be made using *fuzzy* operators presented as control alternatives in various studies [13]. Some can operate by maximizing action by the connective **OR** or by Minimization by the connective **AND** [14].

In a control project these propositions are combined through the FL operators generating new *Fuzzy* propositions that can be described in terms of *Fuzzy* relations. A classic way to find this relationship is with the rules of *Fuzzy* inference. *Fuzzy* inference rules are the core of the *Fuzzy* system where a dynamic behavior is obtained by a set of rules of the form; **If** (a set of conditions are satisfied) **Then** (a set of consequences can be inferred). The **IF** clause is an antecedent and **THEN** is a consequent clause in a control action for a given process under control. With a set of *Fuzzy* rules you can derive a control action for a given set of input values [13], [14].

The approximation used in a *fuzzy* control is based on the approximate reasoning method *(Modus Ponens Generalized* (MPG)). A system with two inputs and one output generates **n**-rules of the fuzzy system, and the states MPG are:

Premise 1 (*fact*): $\boldsymbol{If}\ x\ is\ \boldsymbol{A'}\ and\ y\ is\ \boldsymbol{B'}$
Implication R_1: $\boldsymbol{If}\ x\ is\ \boldsymbol{A_1}\ and\ y\ is\ \boldsymbol{B_1}\ \boldsymbol{then}\ Z\ is\ C_1$
Also R_i: $\boldsymbol{If}\ x\ is\ \boldsymbol{A_i}\ and\ y\ is\ \boldsymbol{B_i}\ \boldsymbol{then}\ Z\ is\ C_i$
Also R_n: $\boldsymbol{If}\ x\ is\ \boldsymbol{A_n}\ and\ y\ is\ \boldsymbol{B_n}\ \boldsymbol{then}\ Z\ is\ C_n$

Conclusion: $Z\ is\ C'$

where x, y and Z are linguistic variables and x, y represent two inputs that may be process states or measurements from sensors.

After the inference of the *fuzzy* control action, an action represented by a single value (*crisp value*) that best represents the *fuzzy* decision is determined.

The process that transforms the *fuzzy* control actions that were inferred into a single value at the output is called defuzzification. There is no systematic procedure to choose the defuzzification method, the most common being:

a) the maximum criterion (Max), which chooses the point where the Membership function has its maximum.
b) the average of the maximum (AOM) that represents the average value of all the maximum points (when there is more than one maximum).
c) the center of area method (COA), which returns the area center of the inferred function.
d) Center of Gravity (COG). This method among those presented is very efficient. Single-value computing (crisp Value) is made by (8):

$$Z_i = \frac{\sum \mu_i(z_k) Z_i}{\sum \mu_i(z_k)} \quad (8)$$

where: $\mu_i(z_k)$ is the area consisting of the output *membership* functions applied by the weights μ_i of all inferential rules (which may have been obtained for example by the minimum operator).

A simplified version of COG defuzzification can be made assuming that all forms of membership functions are symmetric and models the output as a singular function (*Singleton*) in the center Z. With this simplification the result is (9):

$$Z_i = \frac{\sum \mu_i Z_i}{\sum \mu_i} \quad (9)$$

The computational algorithm for all fuzzy rules fired is:

For all fuzzy rules fired:

Step 1. ***Compute:***

a) Sum $\sum \mu_i$ for all weights and;
b) Product $\mu_i Z_i$, for all weights and their corresponding central values of their corresponding output membership functions.

Step 2. Compute the sum of product $\sum \mu_i\, Z_i$ for all $\mu_i Z_i$obtained in *step* 1.

Step 3. Compute the division $\sum \mu_i\, Z_i \,/\, \sum \mu_i$ based on the results obtained in the steps 1 and 2.

IV. Paraconsistent/Fuzzy Analysis

First, the degrees of evidence are analyzed by the simplified "**Para-analyzer**" algorithm, as seen in section II, which transforms them into two values called degree of certainty D_c

Open Science Index, Computer and Information Engineering Vol:11, No:11, 2017 publications.waset.org/10008182/pdf

and degree of contradiction D_{ct}.

The lattice of PAL2v can be represented in degrees of Certainty and Contradiction values according to Fig. 2.

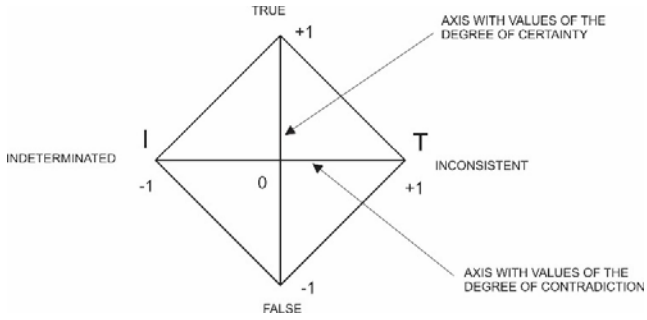


Fig. 2 PAL2v-Lattice with the values of the Degree of Certainty (D_C) and Contradiction (D_{ct})

These two signals, one related to the notion of certainty (Truth and Falsehood) and another related to contradiction (Inconsistency and Indetermination) have their values confined to a normalized discourse universe between -1 and +1.

The normalization of these values is treated in "*Memberships*" of FL, with the fuzzy rules based on paraconsistent logic, and the corresponding logical state can be found.

A. Subdivision of the PAL2v Lattice and Definition of the Inputs for μ and λ in the Lattice for Control

It is then possible to divide the lattice into several parts [2]; thus, determining regions that represent the results of the analysis of the situations in the form of degrees of evidence μ and λ applied at the input. A paraconsistent analysis done in this way allows Control over the situation analyzed.

The control, deducted from the mathematical treatment given to Degree of evidence in the "*Para-analyzer*" algorithm, can be refined by modifying the granulation of the lattice and, as a consequence, an increase in the number of regions. By obtaining a greater number of regions, a fine adjustment in the decisions resulting from analyses is achieved. To better define the regions in the lattice, reflecting on better decision-making after paraconsistent analyzes, the theory of *fuzzy* sets is applied, obtaining a significant improvement in the responses of the control.

According to Fig. 3, the lattice was divided into 260 parts (regions) through the fuzzification. These 260 regions present 260 logical states.

The logical control will be done with the search for stability (balance) system in point equidistant from four vertex of the PAL2v-Lattice. This control is obtained for analysis that considers the degree of favorable Evidence (μ) as the set-point (SP) and the degree of unfavorable Evidence (λ) as a signal from the process variable (PV).

V. Development of the Membership Functions

The FL methodology applied in the lattice of **PAL2v** generates Membership functions for two classes of Fuzzy set; one class provides Membership functions in the axis of the degrees of Contradiction and the other provides the *Membership* functions in the axis of the degrees of Certainty.

In order to construct the Membership functions in the axes of Certainty and Contradiction, we will use the triangular representation type, where the supports of the *Fuzzy* sets and crossover points are identified in Figs. 4 and 5. Fig. 4 shows the lattice of PAL2v and the m*embership* functions generated in the axis of the degrees of Contradiction resulting in the *fuzzy* set related to the uncertain values.

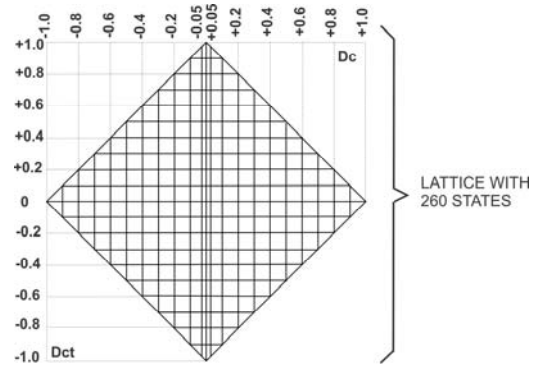


Fig. 3 PAL2v-Lattice divided into 260 regions that produce 260 logical States

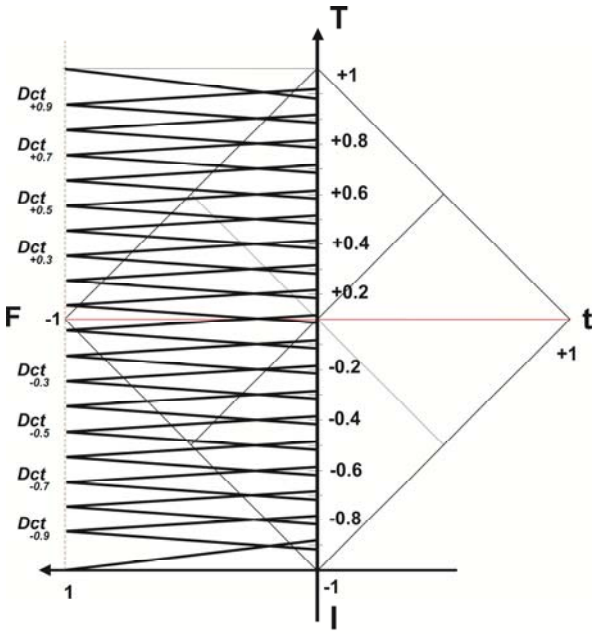


Fig. 4 Membership Functions elaborated on the axis of contradiction

Fig. 5 shows the lattice of **PAL2v** and the Membership Functions of the axis of degrees of certainty resulting in the Fuzzy set related to the certain values.

Open Science Index, Computer and Information Engineering Vol:11, No:11, 2017 publications.waset.org/10008182/pdf

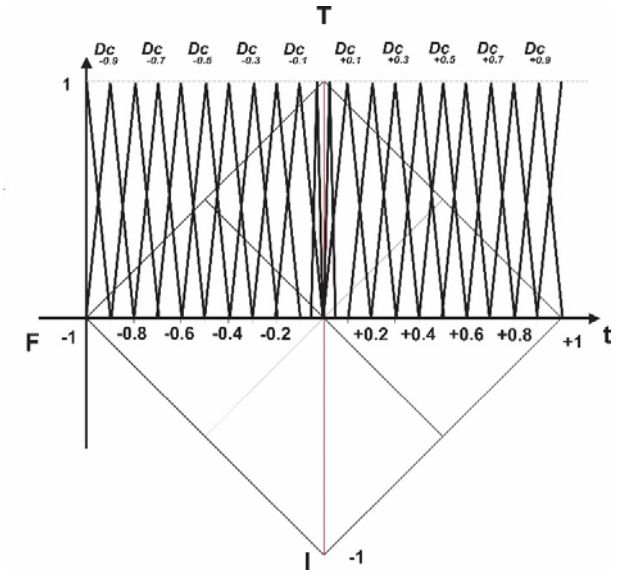


Fig. 5 Membership Functions elaborated on the axis of certainties

VI. Definitions of the Para-Fuzzy Project

With the membership functions obtained, the procedures for the definition of the project of the Para-*Fuzzy* signal analysis system follow the FL methodology.

In this work are included the phases of a *Fuzzy* project that deal with the strategies of fuzzification and defuzzification. For this, it is necessary to apply the techniques to make the base derivation of *fuzzy* control rules. Firstly, the proposed problem and the objective of the project are described (in Fig. 6):

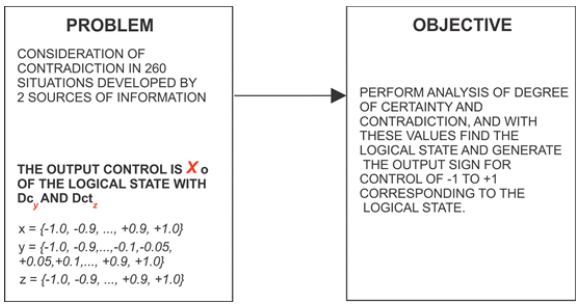


Fig. 6 Problem to analyze and proposed objective

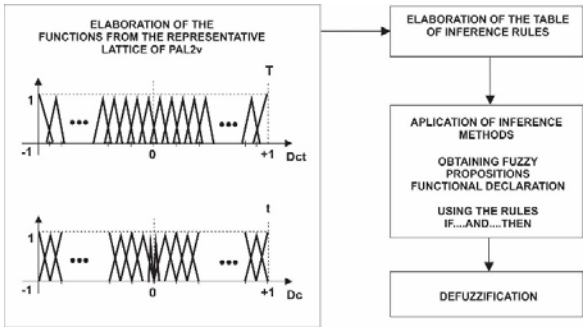


Fig. 7 The phases of the Paraconsistent-*Fuzzy* project

The phases of the Paraconsistent-*Fuzzy* project are summarized as in Fig. 7.

The organization of inference rules obtained from membership functions is shown in Fig. 8.

Dc / Dct	$Dc_{-1.0}$	$Dc_{-0.9}$	...	$Dc_{-0.1}$	$Dc_{-0.05}$	$Dc_{+0.05}$	$Dc_{+0.1}$	...	$Dc_{+0.9}$	$Dc_{+1.0}$
$Dct_{+1.0}$	*	*		-0.1o	-0o	+0o	+0.1o		*	*
$Dct_{+0.9}$	*	*		-0.1o	-0o	+0o	+0.1o		*	*
$Dct_{+0.8}$	*	*		-0.1o	-0o	+0o	+0.1o		*	*
$Dct_{+0.7}$	*	*		-0.1o	-0o	+0o	+0.1o		*	*
$Dct_{+0.6}$	*	*		-0.1o	-0o	+0o	+0.1o		*	*
$Dct_{+0.5}$	*	*		-0.1o	-0o	+0o	+0.1o		*	*
$Dct_{+0.4}$	*	*		-0.1o	-0o	+0o	+0.1o		*	*
$Dct_{+0.3}$	*	*		-0.1o	-0o	+0o	+0.1o		*	*
$Dct_{+0.2}$	*	-0.9o		-0.1o	-0o	+0o	+0.1o		+0.9o	*
$Dct_{+0.1}$	-1.0o	-0.9o		-0.1o	-0o	+0o	+0.1o		+0.9o	+1.0o
$Dct_{-0.1}$	-1.0o	-0.9o		-0.1o	-0o	+0o	+0.1o		+0.9o	+1.0o
$Dct_{-0.2}$	*	-0.9o		-0.1o	-0o	+0o	+0.1o		+0.9o	*

Fig. 8 Representation of the inference rules

With the inference rules presented in Fig. 8, the classical procedures of inferential methods presented in detail in [2] and [7] are made. As well as the choice of the inference methods, the defuzzification in the *membership* functions of the output is made choosing the most appropriate method to the application of the control system. The choice is made in a heuristic way, so it depends on the expert who will apply the Para-*Fuzzy* system. The crisp value of the output will indicate the percentage to be sent to the controller according to the reflex proposition of positioning in one of the 260 regions of the **PAL2v** lattice described as follows:

1) Situation D_{c-1}, $D_{ct+.1}$, then -1.0o
2) Situation D_{c-1}, $D_{ct-.1}$, then -1.0o
3) Situation $D_{c-.9}$, $D_{ct+.2}$, then -0.9o
4) Situation $D_{c-.9}$, $D_{ct+.1}$, then -0.9o
5) Situation $D_{c-.9}$, $D_{ct-.1}$, then -0.9o
6) Situation $D_{c-.9}$, $D_{ct-.2}$, then -0.9o

...

138) Situation $D_{c+.05}$, $D_{ct+.3}$, then+0.0o
139) Situation $D_{c+.05}$, $D_{ct+.2}$, then +0.0o
140) Situation $D_{c+.05}$, $D_{ct+.1}$, then +0.0o

...

260) Situation D_{c+1}, $D_{ct-.1}$, then +1.0o

The *Fuzzy* set of output will therefore have 21 membership functions representing the 260 described situations. Each situation with its Membership functions, are shown in Fig. 9.

It is verified that the single output signal will vary between the maximums and minimums of "Truth", "Falsehood",

Open Science Index, Computer and Information Engineering Vol:11, No:11, 2017 publications.waset.org/10008182/pdf

"Indetermination" and "Inconsistency" by bringing aggregate of action percentage of the controller to the corresponding logical state.

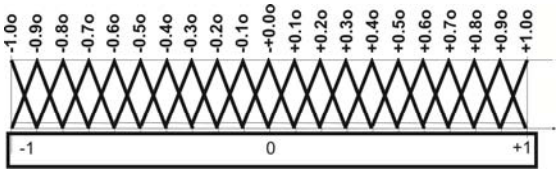


Fig. 9 Single output signal (*crisp value*) after applying of the Paraconsistent and Fuzzy Logics

And for the identification of the logical state is still considered the block "***logical state identifier***" that uses the degree of certainty and degree of contradiction. The system presents a total of two output signals: the *crisp value* and *current logical state*.

VII. Para-*Fuzzy* Digital Controller

A block with "Para-*Fuzzy*" algorithm is used in order to act on the error in the input of the proportional, derivative and integral arguments as shown in Fig. 10.

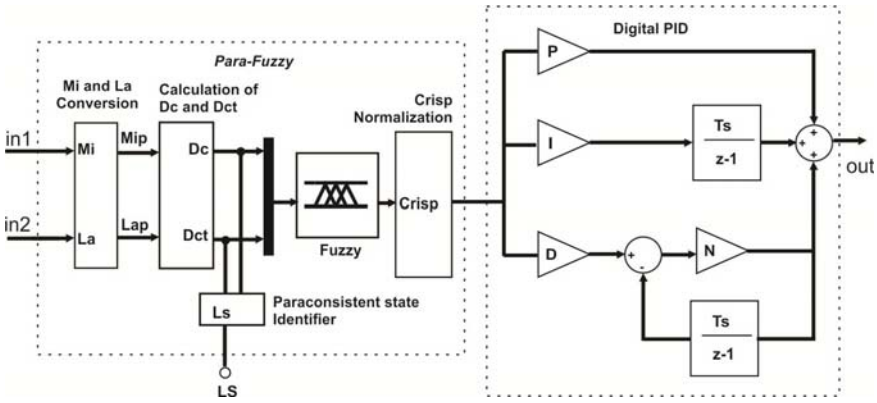


Fig. 10 Para-*Fuzzy* Digital PID block diagram for lattice of 260 states

In the Para-*Fuzzy* subsystem we have the "***μ and λ converter block***" that uses a 1st degree function to transform the set point and process variable values in the range of 0 to 1. The "***D_c and D_{ct} calculation block***" that through μ and λ extracts the degrees of Certainty D_c and Contradiction D_{ct}, the "***state identifier logical block***" that through D_c and D_{ct} computes the paraconsistent logical state in the range 1 to 260. The "***fuzzy block***" that generates the *crisp value* from -1 to + 1 according to the logic state of the lattice and the "***crisp value normalization block***" that uses a 1st degree function to return the value to the real environment.

Fig. 11 shows the comparisons of the conventional Digital PID controller with the Para-*fuzzy* Digital PID controller in response to the unitary step for ***P=12, I=4, D=1.5 e N=100***.

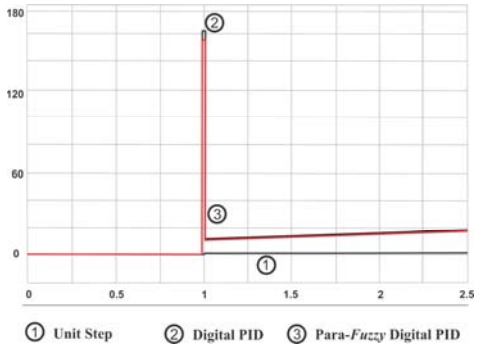


Fig. 11 Comparison of the Conventional Digital PID with Para-*Fuzzy* Digital PID in response to the unitary step. (1) Unitary Step; (2) Response of the Conventional Digital PID; (3) Response of the Para-*Fuzzy* Digital PID

In Fig. 11, a 2.5 seconds simulation was performed in response to the unitary step for the Para-*Fuzzy* Digital PID and Digital PID, both adjusted with proportional gain at 12, integral at 4, derivative at 1.56 and ***derivative filter*** at 100. The Para-*fuzzy* conversion functions were maintained from 1 to 1 (for ***μ and λ conversion, and crisp normalization***). It was observed a difference of 6 units between the pulses resulting from the derivative action, being the controller 1 (Digital PID) at $\boldsymbol{pd_{cc}}$= 162 and the controller 2 (Paraconsistent-*Fuzzy* Digital PID) at $\boldsymbol{pd_{cpf}}$ = 156. A small difference was observed in response to the integral action of 0.05 being the controller 1 at $\boldsymbol{id_{cc}}$= 17.40 and the controller 2 at $\boldsymbol{id_{cpf}} = 17.35$.

Although they have these differences, in a real simulation, a fine adjustment to the conversion function can be made by ***removing*** the gap presented. It was shown the case of maximum conversion, having the consequence of the difference.

A. Example of Application in a Level System

Consider the process composed of two tanks according to Fig. 12. The volumetric flow rate in tank 1 is q_{in}, the volumetric flow rate of tank 1 for tank 2 is q_1, and the volumetric flow rate of tank 2 is q_0. The height of the liquid level is h_1 in tank 1 is h_2 in tank 2. Both tanks have the same cross-sectional area, the area of tank 1 is A_1 and the area of tank 2 is A_2.

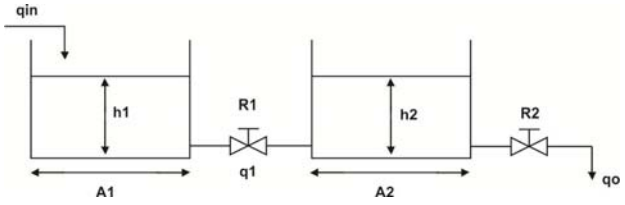


Fig. 12 The block diagram of two coupled liquid tanks

For tank 1:

$$A_1 \frac{dh_1}{dt} = q_{in} - q_1 \quad (10)$$

Assuming a linear resistance to flow, it has:

$$q_1 = \frac{h_1 - h_2}{R_1} \quad (11)$$

and,

$$A_1 \frac{dh_1}{dt} = q_{in} - \frac{h_1 - h_2}{R_1} \quad (12)$$

$$R_1 A_1 \frac{dh_1}{dt} = R_1 q_{in} - h_1 + h_2 \quad (13)$$

Applying Laplace on both sides of (13), it has:

$$R_1 A_1 s h_1(s) + h_1(s) - h_2(s) = R_1 q_{in}(s) \quad (14)$$

$$h_1(s)(R_1 A_1 s + 1) - h_2(s) = R_1 q_{in}(s) \quad (15)$$

For tank 2:

$$A_2 \frac{dh_2}{dt} = q_1 - q_0 \quad (16)$$

Assuming a linear resistance to flow, it has:

$$q_0 = \frac{h_2}{R_2} \quad (17)$$

and,

$$A_2 \frac{dh_2}{dt} = \frac{h_1 - h_2}{R_1} - \frac{h_2}{R_2} \quad (18)$$

$$R_2 A_2 \frac{dh_2}{dt} + h_2 + \frac{R_2}{R_1} h_2 = \frac{R_2}{R_1} h_1 \quad (19)$$

Applying Laplace on both sides of (19), it has:

$$R_2 A_2 s h_2(s) + h_2(s) + \frac{R_2}{R_1} h_2(s) = \frac{R_2}{R_1} h_1(s) \quad (20)$$

$$h_2(s)\left(R_2 A_2 s + \frac{R_2}{R_1} + 1\right) = \frac{R_2}{R_1} h_1(s) \quad (21)$$

To obtain $h_2(s)/q_{in}(s)$, cancels $h_1(s)$ in (14) and (19), therefore:

$$\frac{h_2(s)}{q_{in}(s)} = \frac{R_2}{A_1 R_1 A_2 R_2 s^2 + s(A_1 R_1 + R_2 A_1 + A_2 R_2) + 1} \quad (22)$$

Using the values shown in Table I, the following transfer function is obtained in (23).

$$\frac{h_s(s)}{q_{in}(s)} = \frac{642.86}{199.42 s^2 + 40.04 s + 1} \quad (23)$$

TABLE I
PARAMETERS VALUES FOR TWO TANK [14]

Parameter	*Value*	*Unit*
A_1	0.0145	m^2
A_2	0.0145	m^2
R_1	1478.57	sec/m^2
R_2	642.86	sec/m^2
h_1	0.50	m
h_2	0.50	m

With the transfer function, the "*tune*" feature was used for automatic adjustment in MATLAB of the PID. The following values were found: proportional at 0.007661, integral at 0.000284, derivative at 0.018419 and derivative filter at 1.734662.

Fig. 13 shows the diagram of blocks made in MATLAB for control of the level plant, with the performance of the conventional Digital PID controller and the Para-*Fuzzy* Digital PID for comparison of the answers.

By applying the Para-*Fuzzy* Digital PID controller and comparing with the conventional Digital PID are obtained the answers in Fig. 14.

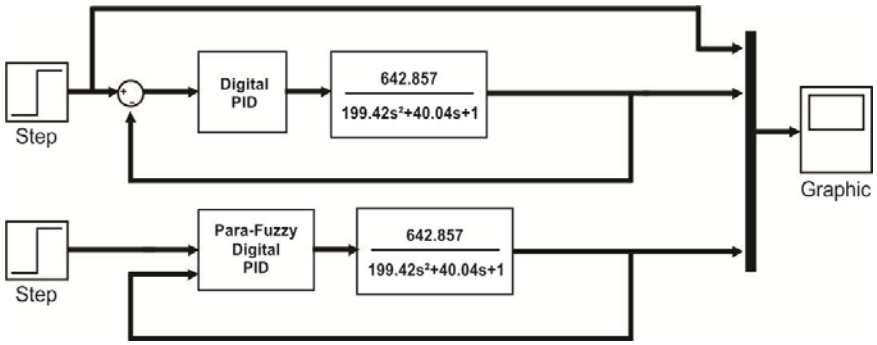


Fig. 13 Block diagrams of conventional Digital PID and Para-*Fuzzy* Digital with proportional value at 0.007661, integral at 0.000284, derivative at 0.018419 and derivative filter at 1.734662

Open Science Index, Computer and Information Engineering Vol:11, No:11, 2017 publications.waset.org/10008182/pdf

Open Science Index, Computer and Information Engineering Vol:11, No:11, 2017 publications.waset.org/10008182/pdf

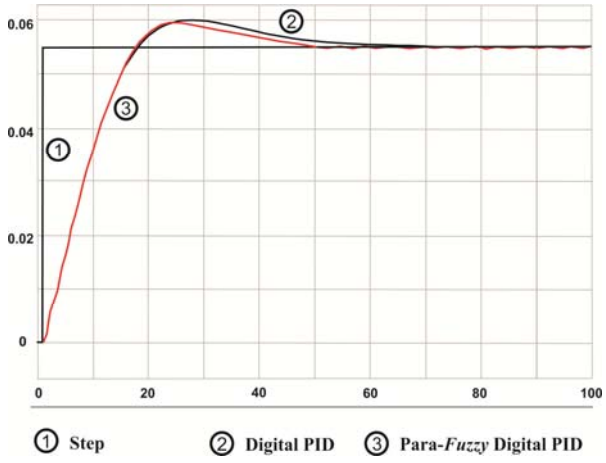


Fig. 14 Comparison of the Para-*Fuzzy* Digital PID with conventional Digital PID in response to 0.055 m step. (1) 0.055 m step; (2) Response of conventional Digital PID; (3) Response of Para-*Fuzzy* Digital PID

In Fig. 14, we have applied the 0.055 step in the controllers for the TF of (23). The duration of the simulation was 100 seconds, the step was applied at time 5 s and the controller's gains were: proportional with 0.007661, integral with 0.000284 and derivative with 0.018419. The 0.055 m step was used in the Para-*Fuzzy* Digital PID with the conversion function from 1 to 10 (***μ and λ conversion, and crisp normalization***); in a real application this conversion function must be well calculated by the designer.

For controller 1 (Digital PID), it was obtained: $\boldsymbol{rt_{cc}}$ (rise time) at 12.73 s, $\boldsymbol{st_{cc}}$ (settling time for +-2%) at 38.05 s, $\boldsymbol{overshoot_{cc}}$ at 8.15%, $\boldsymbol{undershoot_{cc}}$ at 1.99% and $\boldsymbol{of_error_{cc}}$ (off-set error) at 0.00. For controller 2 (Para-*Fuzzy* Digital PID), it was obtained: $\boldsymbol{rt_{cpf}}$ (rise time) at 12.50 s, $\boldsymbol{st_{cpf}}$ (settling time for +-2%) at 34.90 s, $\boldsymbol{overshoot_{cpf}}$ at 8.15%, $\boldsymbol{undershoot_{cpf}}$ at 1.99% and $\boldsymbol{of_error_{cpf}}$ (off-set error) at +-0,002. For close loop without controller it was obtained: $\boldsymbol{rt_{cl}}$ (rise time) at 0.58 s, $\boldsymbol{st_{cl}}$ (settling time for +-2%) at 38.24 s, and $\boldsymbol{overshoot_{cl}}$ at 84.259%.

VIII. Conclusion

In this work, a study was developed to demonstrate the real possibility of applying the two non-classical Logics, the Paraconsistent Logic (PL) and the FL. Furthermore, it was possible to join the two non-classical concepts applied in the control with the realization of a simulation and it was resulted the implementation of a Paraconsistent-*Fuzzy* Digital PID controller. The results of the application of the Para-*Fuzzy* Digital PID controller in a two-tank level plant presented satisfactory results, which allow us to assure new studies in this field of knowledge. The technique shows that it is possible to treat the error within the lattice, generating results for the controller through the corresponding logical state and still brings the possibility of mapping the control region as output to the programmer based on the contradiction and certainty for actions of correction in the mesh, signaling for other meshes or even operating the current protection system of the current mesh, among other applications of the designer. In future works, it is possible to use more *fuzzy* outputs that can respond still better to each logical state of the PAL2v-lattice. Under these conditions, the control still can be improved by archiving the integer value of the paraconsistent logical state and predicting a behavior of the plant taking a predictive action.

References

[1] Abe, J. M. "Fundamentos da Lógica Anotada" Thesis PhD, in portuguese FFLCH/USP - São Paulo, 1992.
[2] Da Silva Fiho, J. I., Lambert-Torres, G. and Abe, J. M. "Uncertainty Treatment Using Paraconsistent Logic - Introducing Paraconsistent Artificial Neural Networks". IOS Press, p.328 pp. Volume 211, Netherlands, 2010.
[3] Alchourrón, C. & D. Makinson, 1981, "Hierarchies of regulations and their logic", In R. Hilpinen, editor, New Studies in Deontic Logic, pp. 123-148, D. Heidel.
[4] Anand, R. & Subrahmanian, V.S. "A Logic Programming System Based on a Six-Valued Logic" AAAI/Xerox Second Intl. Symp. on Knowledge Eng. - Madri-Espanha, 1987.
[5] Da Silva Filho, J.I. "Treatment of Uncertainties with Algorithms of the Paraconsistent Annotated Logic", Journal of Intelligent Learning Systems and Applications, vol. 4, pp. 144-153, May, 2012.
[6] Cruz C. M. et al. "Application of Paraconsistent Artificial Neural network in Statistical Process Control acting on voltage level monitoring in Electrical Power Systems", 18th International Conference on Intelligent System Application to Power Systems (ISAP), pp. 1-6, Porto–PT, Sep, 2015
[7] Da Silva Filho, J.I. "Métodos de Aplicações da Lógica Paraconsistente Anotada de anotação com dois valores LPA2v com construção de Algoritmo e Implementação de Circuitos Eletrônicos". PhD Thesis, in portuguese EPUSP, São Paulo, Brazil.
[8] Subrahmanian, V.S "On the semantics of quantitative Lógic programs" *Proc. 4 th. IEEE Symposiumon Logic Programming, Computer Society press*, Washington D.C, 1987.
[9] Da Costa, N.C.A. & Abe, J.M. &Subrahmanian, V.S. "Remarks on Annotated Logic" Zeitschrift fur Mathematische Logik und Grundlagen der Mathematik, Vol.37, pp.561-570, 1991.
[10] Da Costa, N.C.A., "On the theory of inconsistent formal systems", Notre Dame J. of Formal Logic, 15, 497-510, 1974.
[11] Chen, C. Lee., "Fuzzy Logic in Control Systems: Fuzzy Logic Controller - Part I" – IEEE – Transaction on Systems, Mam and Cybernectics, vol-20, No-2, p.p. 404-435, March/April,1990.
[12] Sakaguchi, S. & Sakai, I. & Haga, T. "Application of Fuzzy Logic Scheduling Method for Automatic Transmission" IEEE Technology Update Series Select Conference Papers-pag-94-100.USA, 1993.
[13] Zadeh, L. "Outline of a New Approach to the Analysis of Complex Systems and Decision Processes" –IEEE- Transaction on Systems, Mam and Cybernectics, vol. SMC-3, No-1, p.p. 28-44, January,1973.
[14] Changela, M., Kumar A. "Design a Controller for Two Tank Interacting System" – IJSR International Journal of Science and Research, vol. 4, Issue 5, May, 2015.



Hyghor Miranda Côrtes is graduated in Automation and Control Engineering from the Faculty UCL, in the city of Serra-ES, Brazil, in 2011. He is currently a master's student in Mechanical Engineering, in the area of Automation and Control Systems, at Santa Cecília University (UNISANTA) in the city of Santos-SP, Brazil.



João Inácio da Silva Filho is coordinator of the PALG – Paraconsistent Annotated Logic Group. In 1999 he obtained a doctorate in Electrical Engineering from POLI / USP in the area of Digital Systems, and a master's degree in Microelectronics from the same Institution. He is currently a Research Professor at Santa Cecília University - UNISANTA in Santos-SP and is researching applications of Paraconsistent Artificial Neural Networks in Expert Systems and Robotic.



Maurício Fontoura Blos obtained his postdoctoral at the Polytechnic School of the University of Sao Paulo (USP), Department of Mechatronics, Laboratory of Automation Systems (Brazil, 2015). D.Eng. in Information Science and Control Engineering from Nagaoka University of Technology (Japan, 2010).



Bruno dos Santos Zanon is graduated in Mechanical Engineering from Redentor College, in the city of Itaperuna-RJ, Brazil, in 2015. He is currently a Master's Degree in Mechanical Engineering, in the area of Automation and Control Systems, at Santa Cecília University (UNISANTA) in the city of Santos-SP, Brazil.

Open Science Index, Computer and Information Engineering Vol:11, No:11, 2017 publications.waset.org/10008182/pdf