



Um Sistema de Controle Utilizando Algoritmo NAP Aplicado a uma Unidade Experimental no Controle de Vazão

Abstract—A Lógica Paraconsistente (LP) vem sendo pesquisada para que seus algoritmos possam ser aplicados com êxito em diversos campos da área industrial. Entre estes campos, destacam-se: a robótica, a eletrônica, e controle de processos (que pode associar lógica com técnicas convencionais de controle). Esta pesquisa teve como objetivo principal a implementação de um algoritmo denominado de Nó de Análise Paraconsistente (NAP), em um Controlador Programável (CP), algoritmo este baseado nos princípios e fundamentos da Lógica Paraconsistente Anotada com anotação de dois valores (LPA2v), que é um ramo da LP. Nesta implementação do algoritmo NAP, houve a aplicação direta deste em uma malha de controle real, cuja variável controlada foi a vazão. Detalhando, o NAP foi implantado em uma unidade experimental, composta por: dois reservatórios, transmissor de vazão, bomba, inversor de frequência e *software* supervisão, onde foi possível realizar o controle da vazão em tubulação entre os reservatórios. Para tornar possível o uso do NAP, foi criado um bloco funcional na programação do CP, denominado NAP_PARACON, o qual recebeu o sinal de *setpoint* (SP) – correspondente ao valor de vazão desejado pelo operador na tubulação –, e o sinal da variável de processo (VP), que representou a vazão medida na tubulação pelo transmissor de vazão. Assim, o bloco funcional NAP_PARACON funcionou como um sistema de controle paraconsistente, que recebe os dois sinais (SP e VP) nas entradas, e gera um sinal de saída que é aplicada ao inversor de frequência, tendo este a função de controlar a velocidade da bomba e, conseqüentemente, a vazão pretendida. Os resultados desta pesquisa apresentaram o comportamento do algoritmo NAP confiável e equivalente ao de algoritmo PID, demonstrando uma nova alternativa para o controle de processos em malha fechada, através da ação de um controle paraconsistente. Esta nova alternativa de controle de processos utilizando LPA2v, na forma implementada, expande iniciativa da norma IEC 61131, em seu item 7, de aplicação de lógicas não-clássicas em CPs.

Index Terms—Automação industrial, IEC 61131, Lógicas não-clássicas, Lógica Paraconsistente, Controle de vazão.

I. INTRODUÇÃO

Os diversos processos da indústria têm tornado seu gerenciamento cada vez mais complexo, que aumenta proporci-

onalmente ao crescimento da necessidade de qualidade nos produtos e serviços, assim como a velocidade na logística de produção e entrega ao cliente final. Esses processos geram uma grande quantidade de dados que devem ser geridos e interpretados na etapa de controle, através dos controladores, dispositivos de campo, *software* de monitoramento e sistemas dedicados de IA [1].

No processo de automação industrial, os Controladores Lógicos Programáveis (CLP) são aplicados, na maioria dos casos, à manufatura, e os Sistemas Digitais de Controle Distribuído (SDCD) são utilizados em processos contínuos. Com a evolução acelerada da tecnologia houve a conversão de ambos, para um único computador com as funcionalidades agregadas: foi, então, feito o Controlador Programável (CP) [2].

Segundo [3] as lógicas denominadas de “não-clássicas”, são as que divergem em seus fundamentos de alguma das leis binárias que sustentam a lógica clássica. Atualmente, existem muitas lógicas que são classificadas como não-clássicas e que são utilizadas com sucesso em projetos de engenharia, tais como: Lógica Multivalorada e Lógica Fuzzy (LF), ou nebulosa. Para que os CPs pudessem se adequar a estas novas lógicas, principalmente a LF, houve definição de requisitos na norma IEC 61131, onde, no seu item 7, são estabelecidas as condições, nas quais os algoritmos de uma lógica não-clássica podem ser configurados adequadamente [4].

A LP tem abordagem não-clássica, e tem apresentado resultados satisfatórios em trabalhos utilizando o reticulado dividido em 12 regiões [2] e em controle híbrido [5], [6], [7].

Nesse viés de aplicação de lógica não-clássica em CPs, neste trabalho foi aplicada a Lógica Paraconsistente Anotada com anotação de dois valores (LPA2v), no programa de um CP, para controle de vazão em uma unidade experimental. Para tal, foi desenvolvido um sistema de controle, através de bloco funcional feito em CP, chamado de NAP_PARACON. Neste

sistema de controle, no bloco NAP_PARACON, foi utilizado o algoritmo Nó de Análise Paraconsistente (NAP) em sua *core*, baseado em LPA2v. Todo o desenvolvimento feito em CP utilizou da norma IEC 61131. Desta forma, foi apresentada uma alternativa para controle de processos, que utilizou LPA2v (não-clássica) com norma IEC 61131 em CP.

Em sequência, na seção II deste trabalho é apresentada a revisão teórica; na seção III, a unidade experimental utilizada e algumas informações em relação ao controle de vazão proposto. Na seção IV, é apresentada, em detalhes, a construção do bloco funcional do sistema de controle proposto (NAP_PARACON); na seção V, são apresentados os resultados; e na seção VI, os trabalhos relacionados. Por fim, na seção VII são apresentadas as conclusões e possibilidades de trabalhos futuros.

II. REVISÃO TEÓRICA

Nesta seção, serão apresentadas as teorias da LP, de princípios de controle de processo, e da norma IEC 61131, como embasamento teórico deste trabalho.

A. Lógica Paraconsistente

Atualmente, os grandes centros de pesquisa investem em projetos de sistemas digitais capazes de funcionar com base em novos tipos de lógica, cujos conceitos teóricos básicos são mais flexíveis e adaptáveis. São conceitos que não têm, necessariamente, seus fundamentos presos nas rígidas leis binárias inerentes à Lógica Clássica, mas sim em sistemas não clássicos, impulsionados principalmente pela área de Inteligência Artificial, que resultam em novas formas de obter resultados em projetos de sistemas de tomadas de decisão.

Para [8], os sistemas não clássicos se mostraram de profundo significado não somente do ponto de vista prático, como também teórico, quebrando um paradigma do pensamento humano que vem imperando há mais de dois mil anos. Segundo [9], as Lógicas Não-clássicas estão classificadas em dois grandes grupos: as que complementam a lógica clássica, ampliando o poder de expressão dela, complementando aspectos que ela não é capaz de expressar; e as que rivalizam com a lógica clássica, também conhecidas como lógicas heterodoxas, que restringem ou modificam certos princípios fundamentais da lógica tradicional.

Entre as várias ideias no âmbito das lógicas não-clássicas, criou-se uma família de lógicas denominada Lógica Paraconsistente (LP), que teve como principal fundamento a revogação do princípio do terceiro excluído, que determina que uma coisa deve ser, ou então não ser; não havendo uma terceira possibilidade (o terceiro é excluído). A LP também revoga o princípio da não contradição, em que uma coisa não pode ser e ser ao mesmo tempo, segundo uma mesma perspectiva, e admite o tratamento de sinais contraditórios à sua estrutura teórica.

Segundo [3], a LP pode ser entendida conforme a seguir: - É dito que uma teoria é consistente quando entre seus teoremas não houver contradição, caso contrário, ela é denominada inconsistente. Uma teoria é denominada como trivial se todas as sentenças (ou fórmulas) de sua linguagem forem teoremas;

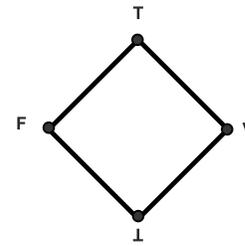


Figura 1. Reticulado representativo de Hasse.

se o contrário ocorrer, ela é classificada como sendo não-trivial. Uma lógica é denominada paraconsistente se puder tratar de inconsistências e de forma não-trivial.

A Lógica Paraconsistente Anotada (LPA) mostra-se mais eficiente para o tratamento de incertezas porque, em sua estrutura, são considerados outros resultados lógicos não permitidos pela Lógica Clássica. É possível a obtenção de estados lógicos que abrangem situações reais como a contradição e a indefinição. A estrutura não clássica da LPA é flexível para tratar informações representativas de situações incertas [8].

A LPA possui um reticulado associado que desempenha um papel importante em sua representação. Segundo [10], na LPA as fórmulas proposicionais vêm acompanhadas de suas respectivas anotações, cada qual é pertencente a um reticulado finito τ , e atribuem-se valores à sua correspondente fórmula proposicional. Uma LPA pode ser representada por um reticulado finito, de “quatro estados”, são eles: Verdade (V), Falsidade (F), Inconsistência (T) e Paracompleteza (\perp) [7], conforme Figura 1.

B. Lógica Paraconsistente Anotada com anotação de dois valores (LPA2v)

LPA2v é um método de análise que possibilita a interpretação e a valoração de estados lógicos representados no reticulado da LPA, por meio da anotação de dois valores. Desse modo, podem-se associar regiões do reticulado com estados lógicos, e extrair equações que fornecem valores para a tomada de decisão. Esses procedimentos permitiram, por exemplo, a construção de um algoritmo baseado nos conceitos da LPA2v, denominado Para-Analisador [8], [11].

Na LPA2v, [10] apresenta o primeiro valor da anotação como a evidência a favor (grau de evidência favorável) à proposição P, e o segundo valor como a evidência contrária (grau de evidência desfavorável) à proposição P. O símbolo do grau de evidência favorável é a letra grega “ μ ” e o símbolo do grau de evidência desfavorável é a letra grega “ λ ”. Cada constante anotacional de um reticulado τ pode ser representada pelo par (μ, λ) conforme Equação 1.

$$\tau = (\mu, \lambda) | \mu, \lambda \in [0, 1] \subset \mathfrak{R} \quad (1)$$

Assim: Se P é uma fórmula básica, o operador $\sim: |\tau| \rightarrow |\tau|$ é definido como: $\sim [(\mu, \lambda)] = (\mu, \lambda)$, onde $(\mu, \lambda) \in \{ x \in \mathfrak{R} | 0 \leq x \leq 1 \}$, e considera-se (μ, λ) como uma anotação de P.

A atribuição dos graus de evidência objetiva solucionar justamente o problema de sinais contraditórios, ou seja,

coletando-se evidências, analisando-as, para assim modificar o comportamento do sistema de forma que as contradições diminuam.

Com base nisso, intuitivamente, conforme ilustrado na Figura 2, para o reticulado de Hasse tem-se:

- $P_t = P(\mu, \lambda) = P(1, 0)$: indicando uma evidência favorável total e evidência desfavorável nula para P, atribuindo uma conotação lógica da *Verdade*.
- $P_F = P(\mu, \lambda) = P(0, 1)$: indicando uma evidência favorável nula e uma evidência desfavorável total de P, atribuindo uma conotação lógica de *Falsidade*.
- $P_{\top} = P(\mu, \lambda) = P(1, 1)$: indicando uma evidência favorável total e uma evidência desfavorável total de P, atribuindo assim uma conotação de *Inconsistência*.
- $P_{\perp} = P(\mu, \lambda) = P(0, 0)$: indicando uma evidência nula favorável e uma evidência nula desfavorável para P, atribuindo uma conotação lógica de *Paracompleteza*.

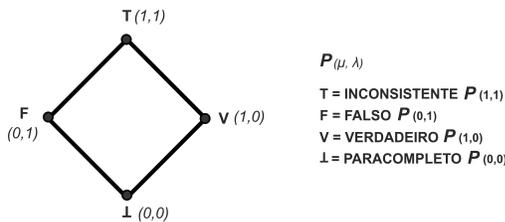


Figura 2. Reticulado de Hasse com "quatro estados".

C. Algoritmo do Nó de Análise Paraconsistente (NAP)

Dos fundamentos da LPA2v, são extraídos algoritmos para: (a) coleta de sinais (na forma de graus de evidência) de informação com contradição; (b) tratamento desses sinais com incertezas pelas equações da lógica; (c) e geração de resultados com base em inferências satisfatórias, dada uma proposição em questão.

Neste trabalho, todos os procedimentos de tratamento lógico paraconsistente são relacionados à análise efetuada por um algoritmo denominado NAP (Nó de Análise Paraconsistente) [8].

Várias versões do algoritmo NAP foram criadas. No entanto, a utilizada neste trabalho foi a versão com as saídas: (i) Grau de evidência resultante real (μ_{ER}); e (ii) Intervalo de Certeza (ϕ). A ilustração do algoritmo encontra-se na Figura 3.

A seguir, é apresentado o pseudocódigo do algoritmo do NAP (Algoritmo 1).

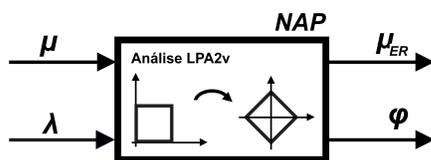


Figura 3. Representação do NAP com saídas: Grau de evidência resultante real e intervalo de certeza.

Algoritmo 1 NAP com saídas μ_{ER} e ϕ - Pseudocódigo

```

1: procedure NAP( $mi, la$ ) ▷ Leitura de  $\mu$  e  $\lambda$ 
2:
3:    $gc \leftarrow mi - la$ ;
4:    $gct \leftarrow (mi + la) - 1$ ;
5:
6:    $phi\_p \leftarrow 1 - |gct|$ ;
7:
8:    $D \leftarrow \sqrt{(1 - |gct|)^2 + (gct)^2}$ ;
9:
10:  if ( $gc \geq 0$ ) then
11:     $gcr = (1 - D)$ ;
12:  end if
13:
14:  if ( $gc < 0$ ) then
15:     $gcr = (D - 1)$ ;
16:  end if
17:
18:  if ( $phi\_p \leq 0.25$ ) || ( $D > 1$ ) then ▷ Cálculo de  $\mu_{ER}$  e  $\phi$ 
19:     $mi\_er = 0, 5$ ;
20:     $phi = phi\_p$ ;
21:  else
22:    if ( $gct < 0$ ) then
23:       $phi = -phi\_p$ ;
24:    else
25:      if ( $gct > 0$ ) then
26:         $phi = phi\_p$ ;
27:      else
28:         $phi = 0$ ;
29:      end if
30:    end if
31:     $mi\_er = (gcr + 1)/2$ ;
32:  end if
33:
34:  retorno( $mi\_er, phi$ ); ▷ Retorna os valores de  $\mu_{ER}$  e  $\phi$ 
35: end procedure

```

D. Princípios de Controle de Processos

Em um sistema de controle típico, a variável do processo (VP) é o parâmetro do sistema que precisa ser controlado, como temperatura (° C), pressão (psi) ou fluido (litros/minuto). Um sensor é usado para medir a VP e fornecer *feedback* para o sistema de controle. O *setpoint* (SP) é o valor desejado ou comando para a VP, tal como 100°C, no caso de um sistema de controle de temperatura. A qualquer momento, a diferença entre a VP e o SP é usada pelo algoritmo do sistema de controle (geralmente denominado “compensador”), para determinar a saída desejada para o atuador (através de uma variável manipulada (VM)) que, por sua vez, irá intervir na planta ou estrutura. Por exemplo, se a VP é a “temperatura” medida, seu valor é 100° C, e o SP de temperatura desejada é 120° C, então a saída do atuador especificada pelo algoritmo do compensador pode mudar a condição de um aquecedor. Controlar um atuador para ligar um aquecedor faz com que a temperatura se eleve, e resulta em um aumento na VP. Isto é chamado de “sistema de controle em malha fechada”, porque o processo de leitura do sensor fornece *feedback* constante, e o cálculo para definir a saída desejada para o atuador se repete continuamente a uma taxa fixa. A ilustração de um sistema de controle típico é apresentada na Figura 4 [12], [13].

Proporcional-Integral-Derivativo (PID) é um dos algoritmos de compensação mais usados na indústria, e tem sido utilizado em todo o mundo para sistemas de controle industriais. O algoritmo PID é composto por três componentes: proporcional, integral e derivativo, que são variados para obter a resposta ideal [12].

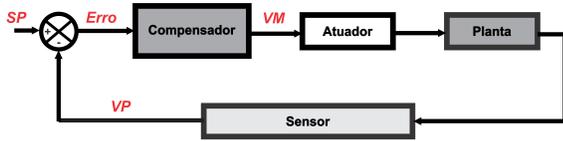


Figura 4. Sistema de controle típico em malha fechada.

A ideia básica por trás do algoritmo PID (que na malha de controle faz a função de compensador) é ler um sensor e o valor que se deseja, e calcular uma resposta, proveniente das somas das influências das componentes proporcional, integral e derivativo.

A componente proporcional (correspondente ao ganho K_p) depende apenas da diferença entre o SP e a VP. Esta diferença é referida como o termo “erro”. O ganho K_p determina a taxa de resposta de saída para o sinal de erro. Por exemplo, se o termo de erro tem uma magnitude de 20, um ganho proporcional de 5 produziria uma resposta proporcional de 100. Em geral, aumentando o ganho proporcional, irá aumentar a velocidade da resposta do sistema de controle. No entanto, se o ganho proporcional é muito grande, a variável de processo começará a oscilar [13], [14].

A componente integral (correspondente ao ganho K_i) soma o termo de erro, ao longo do tempo. Mesmo um erro pequeno fará com que a componente integral aumente lentamente. A resposta integral irá aumentando ao longo do tempo a menos que o erro seja zero. A componente integral é ajustada pelo tempo integral (τ_i), que é o inverso de K_i [13], [14].

A componente derivada (correspondente ao ganho K_d) faz com que a saída diminua, se a variável de processo está aumentando rapidamente. A derivada de resposta é proporcional à taxa de variação de VP. Aumentar o parâmetro do tempo derivativo (τ_d), em que $\tau_d = K_p$, fará com que o sistema de controle reaja mais fortemente a mudanças no parâmetro de erro aumentando a velocidade da resposta global de controle do sistema. Na prática, a maioria dos sistemas de controle utilizam o tempo derivativo (τ_d) muito pequeno, pois a derivada de resposta é muito sensível a ruídos no sinal da VP. Na Figura 5, é ilustrado o diagrama do PID com τ_i , K_p e τ_d [13], [14].

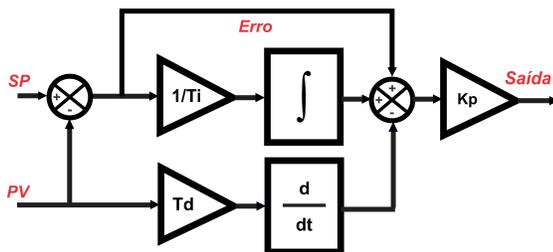


Figura 5. Diagrama do PID com τ_i , K_p e τ_d .

Na atuação de um compensador, a resposta da VP nos dá algumas informações a respeito do desempenho do efeito da compensação. Algumas dessas informações são: (i) tempo de subida (t_s); (ii) tempo de assentamento (t_a); e (iii) sobressinal (*overshooting*).

O tempo de subida (t_s) é definido como o tempo para a resposta da VP alcançar uma faixa de valores de 2% em torno de seu valor final, e ali permanecer. O tempo de assentamento (t_a) é definido como o tempo necessário para que a resposta da VP, partindo de 10% de seu valor final, atinja 90% desse valor. O sobressinal (*overshooting*) é o percentual do valor de pico da resposta da VP em relação ao seu valor final (estacionário ao longo do tempo), que geralmente acontece mediante um estímulo por degrau no SP.

E. Norma IEC 61131

International Electrotechnical Commission (IEC) é uma comissão internacional, não governamental, sem fins lucrativos, formada por representantes de fabricantes, fornecedores, distribuidores e usuários de CPs pertencentes aos Comitês Nacionais dos países membros de todo o mundo. Com sua sede, desde 1948, em Genebra, a IEC possui diversos centros regionais, dentre eles, um na América Latina, no Brasil, na cidade de São Paulo [2].

A norma IEC 61131 estabelece, em seu item 3, a padronização de cinco tipos de linguagens de programação, que são: Sequenciamento Gráfico de Funções (SFC), Lista de Instruções (IL), Diagrama Ladder (LD), Diagrama de Blocos (FBD) e Texto Estruturado (ST) [2], [15].

A norma IEC 61131, no item 3, foi elaborada visando o atendimento às demandas da comunidade industrial e buscando uma padronização, tendo como principais aspectos a adoção de linguagens de programação, recursos multitarefa e reutilização de *software* [2].

Já no item 7 desta mesma norma, foram abordadas funcionalidades de lógica nebulosa dentro de CPs, com blocos funcionais prontos, e com padrões para serem usados [16], [17].

Na próxima seção, serão apresentadas os materiais e métodos.

III. MATERIAIS E MÉTODOS

Nesta seção, são apresentados os materiais e métodos. Na seção A, a unidade experimental é apresentada com seus detalhes. Na seção B, mostra como o desenvolvimento do sistema de controle foi realizado, e como foi testado e validado.

A. Unidade experimental e controle de vazão

A unidade experimental utilizada neste projeto de pesquisa do SENAI-SP, representada na Figura 6, foi adequada às características de um sistema de controle em malha fechada. A grandeza escolhida para ser controlada foi a vazão de uma tubulação entre os tanques (representados por T100 e T200).

No controle da vazão entre os tanques (T100 e T200), a variável manipulada (VM) foi a velocidade do inversor (representado por FZ-200) que, por consequência, atua na rotação da bomba. A leitura da vazão foi feita por transmissor de vazão (representado por FT-200), correspondente à variável do processo (VP). Já a referência de vazão (SP), foi inserida pelo usuário em um *software* supervisor.

Neste contexto, desatacam-se os seguintes componentes da unidade experimental utilizados: um CP, um inversor de

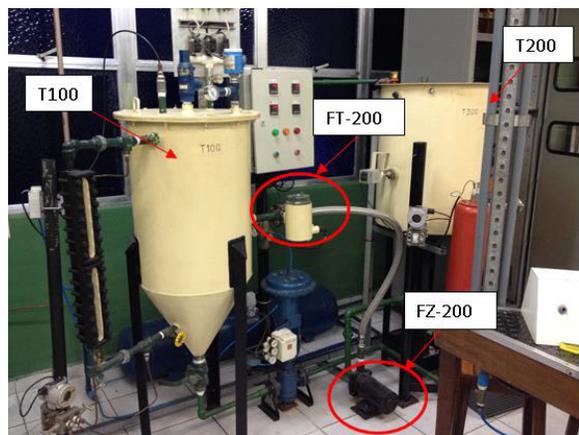


Figura 6. Unidade experimental.

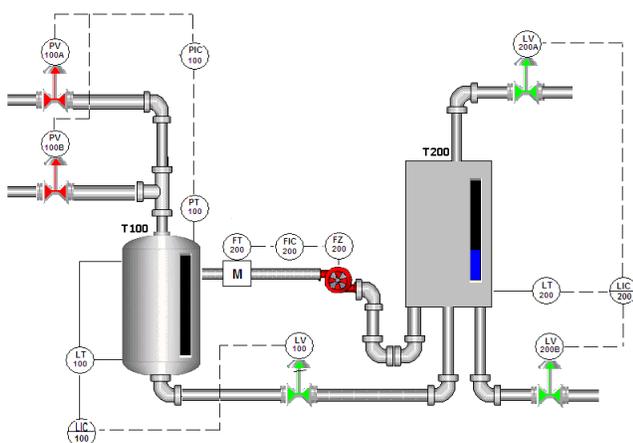


Figura 7. Representação da unidade experimental no InduSoft.

frequência (FZ-200), uma bomba, um transmissor de vazão (FT-200), um *software* supervisor, e dois tanques (T100 e T200). O CP utilizado foi o da marca Altus, modelo DU351. A linguagem de programação do CP foi feita em Texto Estruturado (TE). O *software* supervisor usado foi o InduSoft Web Studio.

Na Figura 7, encontra-se o esquemático da unidade experimental desenvolvido no *software* supervisor InduSoft Web Studio.

Com isso, um sistema de controle em malha fechada foi feito possibilitando o ajuste da vazão entre os tanques, utilizando-se da unidade experimental, com o monitoramento de todas as informações.

B. Desenvolvimentos e testes

Para construção do sistema de controle, foram utilizados conceitos da LPA2v, o que já foi realizado em trabalhos anteriores como em: [7] e [6]. Em detalhes, na idealização do sistema de controle, considerou-se como premissa: (a) receber dados de *setpoints* e de VPs como evidências para a LPA2v; e (b) se usar uma única saída de 0 a 1, proveniente de tratamento da lógica paraconsistente, utilizando algoritmo já existente e difundido da lógica, para atuar em elemento final de controle. Desta forma, optou-se por utilizar o algoritmo NAP.

A malha de controle utilizando um NAP foi elaborada com base na estrutura e disponibilidade de sensores e atuadores já existentes em planta experimental de laboratório do SENAI-SP, o que determinou a aplicação da mesma no controle da vazão entre tanques desta planta.

Para a construção do NAP em microprocessador, foi utilizado o CP - Altus DU351, por: (i) estar disponível para uso no SENAI-SP; (ii) ser equipamento utilizado em plantas industriais; (iii) permitir a utilização da norma IEC 61131 em sua programação; e (iv) ser programável facilmente e possuir bloco PID para possíveis comparações de dados.

Para testes e validação do sistema de controle criado, foram realizadas simulações com variações do tipo degrau no *setpoint* desse sistema de controle proposto e verificadas as respostas. Posteriormente, tais respostas foram comparadas às respostas também ao degrau em *setpoint* utilizando-se de PID convencional. Foram realizados vários ensaios nesse contexto, selecionados três ensaios para o sistema de controle criado, e dois ensaios para o PID convencional para serem apresentados no artigo. Na comparação das respostas, os parâmetros analisados foram: degrau (*step*), sobressinal (*overshooting*), tempo de subida (t_s) e tempo de assentamento (t_a). Ao final, verificou-se a coerência dos parâmetros mencionados nas respostas dos dois agentes de controle, considerando diferentes degraus de *setpoint*.

Na próxima seção, será apresentada, detalhadamente, a construção do bloco funcional do sistema de controle proposto (NAP_PARACON).

IV. CONSTRUÇÃO DO BLOCO FUNCIONAL NAP_PARACON

O bloco funcional NAP_PARACON foi construído utilizando a linguagem de programação em TE, onde houve a inserção do algoritmo do NAP da LPA2v, ação esta realizada utilizando técnicas de programação estruturadas na norma IEC 61131, item 3. O NAP_PARACON, representado na Figura 8 abarcou funções de lógica não-clássica. O bloco constituiu-se de duas entradas – VP e SP –, representando, respectivamente, o grau de evidência favorável (μ) e grau de evidência desfavorável (λ). As saídas do bloco foram compostas pelo grau de evidência resultante real (S1) e Intervalo de Certeza (S2).

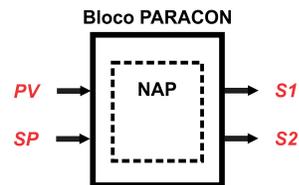


Figura 8. Ilustração do bloco funcional NAP_PARACON.

Para adequação dos valores de engenharia lidos nas entradas deste bloco, advindos do processo e do supervisor (SP e VP), para o formato de graus de evidência favorável e desfavorável (μ e λ) necessários à LPA2v, foram feitas modelagens. Para se obter μ e λ , foram utilizadas duas conversões para cada variável. Nas Tabelas I e II, é possível identificar essas

Tabela I
 MODELAGEM PARA OBTENÇÃO DE μ A PARTIR DE L/H DO TRANSMISSOR DE VAZÃO (FT-200).

Leitura (l/h/mA)	1º Conversão (Inteiro)	2º Conversão (μ)
15/5,3	10	0,1
30/6,9	20	0,2
45/8,5	30	0,3
60/10,2	40	0,4
75/11,7	50	0,5
90/13,2	60	0,6
105/14,8	70	0,7
120/16,5	80	0,8
135/18,1	90	0,9
150/19,6	100	1,0

Tabela II
 MODELAGEM PARA OBTENÇÃO DE λ A PARTIR DO SUPERVISÓRIO INDUSOFT WEB STUDIO.

Leitura (l/h)	1º Conversão (Inteiro)	2º Conversão (λ)
15	10	0,1
30	20	0,2
45	30	0,3
60	40	0,4
75	50	0,5
90	60	0,6
105	70	0,7
120	80	0,8
135	90	0,9
150	100	1,0

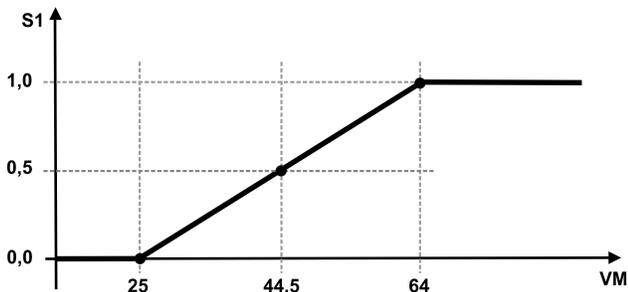


Figura 9. Modelagem de S1 para atuar na VM.

conversões, respectivamente para " μ " e " λ ". As modelagens de μ e λ foram feitas no próprio NAP_PARACON.

O grau de evidência resultante real (μ_{ER}), correspondente à saída S1 do NAP_PARACON, foi utilizado para atuar na VM do processo, enviando ordens para o inversor de frequência (FZ-200) que conseqüentemente mudava a velocidade da bomba. A Figura 9 demonstra, em formato gráfico, a modelagem de S1.

Observa-se, claramente, que o μ_{ER} (correspondente a S1) varia de 0 e 1, enquanto a VM varia de 25 a 64. A Equação 2 representa a modelagem da Figura 9.

$$VM = 25 + (\mu_{ER} * (64 - 25)) \quad (2)$$

Essa modelagem, de S1 para VM, foi feita em sub-bloco funcional auxiliar, denominado Modelagem do Valor Lógico

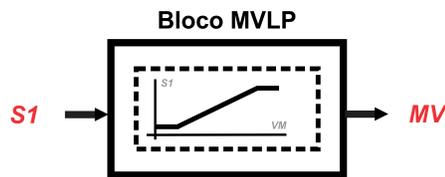


Figura 10. Ilustração do sub-bloco MVLP.

Paraconsistente (MVLP). Na Figura 10 tem-se a ilustração do MVLP.

Na programação do MVLP, utilizou-se também, como no NAP_PARACON, de técnicas de programação estruturadas na norma IEC 61131, item 3.

É importante ressaltar que a modularidade do bloco funcional NAP_PARACON, no formato da Figura 8, possibilitou a sua interação com equipamentos ou sistemas baseados em lógicas clássicas, e sistemas baseados em outras linguagens pertencentes norma IEC 61131, item 3. Ainda, é importante frisar que ambos os blocos necessitaram de pouco recurso computacional, e a administração do algoritmo não clássico (NAP) ficou simplificada.

Na próxima seção, serão apresentados os resultados.

V. RESULTADOS

Apresenta-se na Figura 11 o comportamento do NAP_PARACON para um SP de 80%. Nesta situação, a variável do processo estava em 35% representando uma vazão de 52,5 l/h, e o valor desejado de SP igual a 80% representando uma vazão de 120 l/h. Essa diferença entre os valores foi tratada pelo bloco, resultando na correção do valor da VP.

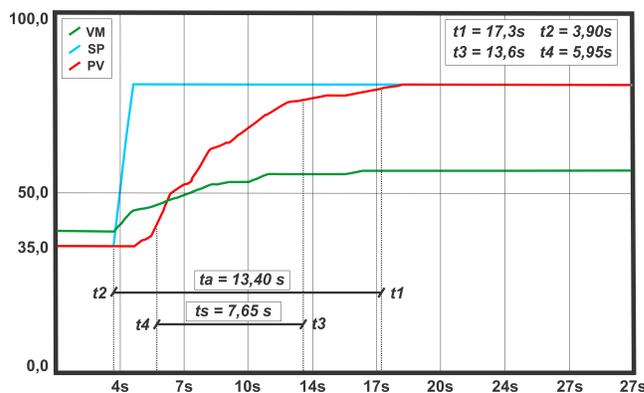


Figura 11. Resposta do bloco NAP_PARACON para um SP = 80%.

Apresenta-se na Figura 12, o teste realizado inicialmente com a VP em 45% (67,5 l/h) e o SP para 35% (52,5 l/h). Na Figura 13, é mostrado o teste com a VP em 80% (120 l/h) e o SP para 45% (67,5 l/h). Em ambos os testes, a VP buscou o SP, após um período de tempo, demonstrando a atuação do bloco NAP_PARACON.

No teste da Figura 11, relacionado ao bloco NAP_PARACON, a resposta da VP teve tempo de subida (t_s) de 7,65 segundos, um tempo de assentamento (t_a) de 13,40 segundos (considerando 2% de erro), e não apresentou

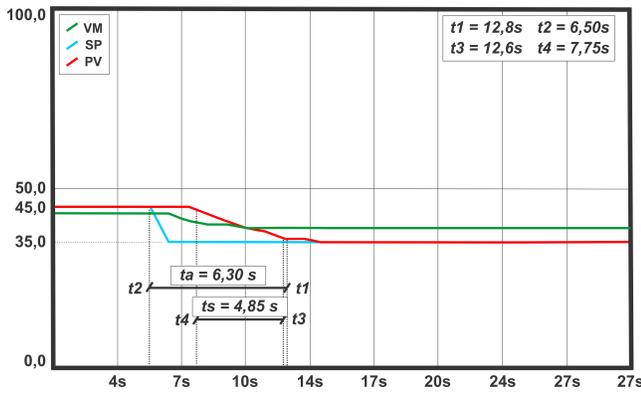


Figura 12. Resposta do bloco NAP_PARACON para um SP = 35%.

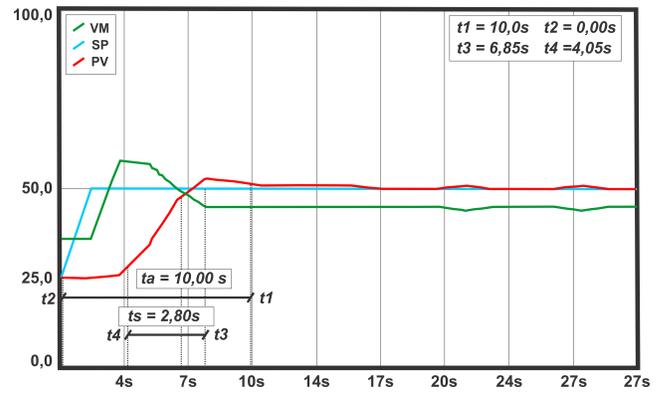


Figura 14. Resposta do bloco PID na malha de vazão para um SP = 50%.

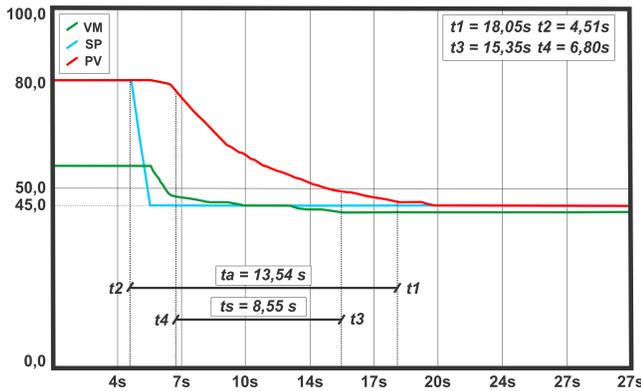


Figura 13. Resposta do bloco NAP_PARACON para um SP = 45%.

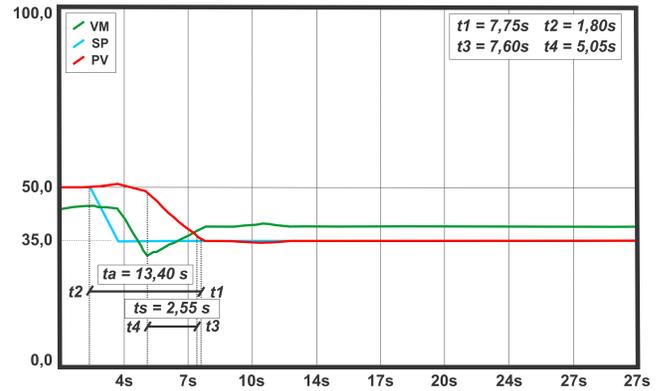


Figura 15. Resposta do bloco PID na malha de vazão para um SP = 35%.

sobressinal (*overshooting*). Já nos testes das Figuras 12 e 13, as respostas de VP, foram, respectivamente para t_s , t_a e *overshooting*: 4,85 segundos, 6,30 segundos, sem sobressinal; 8,55 segundos, 13,54 segundos, sem sobressinal.

Tais testes comprovam que foi possível controlar a vazão na unidade experimental, utilizando o algoritmo do NAP pertencente à LPA2v.

Para comprovar a eficiência NAP_PARACON, foi aplicado na mesma unidade experimental de estudo (na malha vazão), um bloco PID (presente na biblioteca de programação do CP). O bloco PID foi utilizado com os parâmetros $K_p=0.75$, $K_i=3.0$ e $K_d=0.0$.

Com o bloco PID, na Figura 14, é apresentado teste realizado inicialmente com a VP em 25% (37,5 l/h) e SP para 50% (75,0 l/h). Na Figura 15 é mostrado, com o mesmo bloco, teste com a VP em 50% (75 l/h) e SP para 35% (52,5 l/h). Em ambos os testes, a VP buscou o SP após um período de tempo, demonstrando a atuação do bloco PID.

No teste da Figura 14, relacionada ao bloco PID, a resposta da VP teve t_s de 2,80 segundos, um t_a de 10,00 segundos (considerando 5% de erro), e apresentou pequeno sobressinal (*overshooting*). Já no teste da Figura 15, a resposta de VP teve t_s de 2,55 segundos, um t_a de 13,40 segundos (considerando 5% de erro), e também apresentou pequeno sobressinal (*overshooting*).

Observa-se, pela Tabela III, que os resultados do bloco NAP_PARACON foram correlatos aos resultados do PID,

em dinâmica e característica. Nos testes destacados em azul na Tabela III, com degraus (*steps*) de 10% e 15%, foram obtidos os valores de t_s e t_a , respectivamente, de: 4,85 segundos, 6,30 segundos, 2,55 segundos, 5,95 segundos, ou seja, correspondentes entre si, para os diferentes valores de degraus impostos. O mesmo ocorreu, para os testes destacados em vermelho na mesma tabela, com degraus de 45% e 35%, no qual foram obtidos os valores de t_s e t_a , respectivamente, de: 7,65 segundos, 13,40 segundos, 8,55 segundos e 13,54 segundos. Tais dados, convalidam a comprovação dos bons resultados do algoritmo NAP (da LPA2v) no controle da vazão, aplicado à unidade experimental. Os resultados do PID foram bem aproximados aos resultados do NAP_PARACON, o que convalida a comprovação dos bons resultados do algoritmo NAP (da LPA2v) no controle da vazão, na unidade experimental.

Tabela III
COMPARAÇÃO RESPOSTAS DOS BLOCOS NAP_PARACON E PID.

Tipo de bloco	Teste	VP inicial /SP	step	tr	ta	over-shooting
NAP _CONTROL	Fig. 11	35%/80%	45%	7,65s	13,40s	Não
NAP _CONTROL	Fig. 12	45%/35%	10%	4,85s	6,30s	Não
NAP _CONTROL	Fig. 13	80%/45%	35%	8,55s	13,54s	Não
PID	Fig. 14	25%/50%	25%	2,80s	10,00s	Sim
PID	Fig. 15	50%/35%	15%	2,55s	5,95s	Sim

Na próxima seção, serão apresentadas os trabalhos relacionados.

VI. TRABALHOS RELACIONADOS

Em [2] foi apresentado um controlador chamado de BF_Paracon, que utilizou o algoritmo Para-Analisador fundamentado em LPA2v, aplicado a uma planta de controle de temperatura. O BF_Paracon, basicamente: (i) lia como evidências (favoráveis e desfavoráveis) o *setpoint* e a VP; (ii) com o *setpoint* e VP ajustados no *range* de 0 a 1, aplicava o algoritmo Para-Analisador; (iii) e como saída, gerava doze possíveis ações no elemento final de controle (que era uma resistência), com base nos 12 estados lógicos do reticulado da LPA2v. Em [2], destaca-se que o BF_Paracon não utilizou nenhuma associação com as ações PID, convencionalmente usadas para controle de processos, e foi aplicado a malha de temperatura que exigia ações de controle relativamente lentas. Em [2], foi utilizada a norma IEC 61131 na programação do controlador.

Em [7], foi apresentado um controlador que utilizava a lógica Para-Fuzzy com as ações PID, aplicado a um controle de nível. Neste controlador, a lógica Para-Fuzzy utilizou algoritmo específico que: (i) recebia como evidências (favoráveis e desfavoráveis) os valores de *setpoint* e VP; (ii) aplicava o tratamento Para-Fuzzy, que gerava um valor entre 0 e 1 (denominado *crisp*¹); (iii) o valor do *crisp* atuava no diagrama de blocos convencional do PID, que possuía seus valores integral, derivativo e proporcional, pré-configurados; (iv) e, por fim, o valor final proveniente do PID era enviado ao elemento final de controle. Neste formato de controlador, diferente do realizado por [2], há uma conjugação entre LPA2v, LF e as ações PID padrão, e ambas as lógicas no formato Para-Fuzzy otimizaram as ações do controlador. Em [2], as ações foram limitadas em doze possíveis saídas, o que limita também a atuação desse controlador a processos mais lentos e que não necessitam de otimização e refinamento nas ações de controle sobre o elemento final. Todavia, o controlador apresentado por [7] necessita de um considerável poder computacional, comparado ao de [2] e a um controlador PID convencional. Esse poder computacional considerável, principalmente pela forma de uso da lógica Para-Fuzzy no trabalho de [7], que utilizou da subdivisão do reticulado em mais de 250 partes (estados lógicos) para cálculo do *crisp* de 0 a 1 para atuar otimizando as ações do PID. Os resultados em [7] foram satisfatórios, e o controle de nível foi possível.

Em [6], foi apresentado um controlador que utilizava Nós de Análise Paraconsistentes (NAPs) com as ações PI aplicado a uma malha de controle de nível de tanque. Um NAP, de forma geral, utilizando-se de algoritmo específico: (i) lê as evidências (favoráveis e desfavoráveis); (ii) calcula um parâmetro chamado Grau de Certeza Real (G_{cr}), que é um valor de -1 a 1, baseado nas evidências; e (iii) normaliza o valor de G_{cr} , gerando um valor de saída (denominada μ_{ER}) de 0 a 1, que representa uma situação no reticulado. No trabalho apresentado, dois NAPs foram ligados em cascata, o que é possível quando se usa NAPs. Nesta ligação, o primeiro NAP

(NAP1) recebia como evidências (favoráveis e desfavoráveis) leituras de sensores de nível, e sua saída alimentava a evidência favorável do segundo NAP (NAP2), que se caracterizava como o valor da VP da malha de controle de nível. O NAP 2 recebia como evidência desfavorável o *setpoint*. Por sua vez, resultados do NAP2 atuavam em diagrama de blocos de PI convencional, que possuía seus valores integral e proporcional já pré-configurados. Todavia, as saídas usadas do NAP2 foram μ_{ctr} e G_{ct} , que também são saídas possíveis para um NAP além do μ_{ER} , e ambas representavam o nível de contradição (G_{ct}) entre as evidências lidas (a primeira normalizada, e segunda não), ou seja, contradição entre VP e *setpoint*. Em detalhes, o μ_{ctr} atuava na ação proporcional e G_{ct} , na ação integral do diagrama de blocos do PI. Os resultados foram satisfatórios, e foi possível realizar o controle do nível pretendido.

Em [6], há uso de poder computacional bem menor do que o utilizado em [7]. Para além disso, utiliza-se de NAP, diferentemente do Para-Fuzzy, que é uma forma mais simples de aplicação da lógica para inferir resultados do que a forma Para-Fuzzy, o que o torna muito mais fácil de se implementar e administrar. Ou seja, um especialista que esteja implantando a solução não necessitará de um conhecimento mais apurado.

Como em [6], neste trabalho utiliza-se de algoritmo NAP para controle de processos. Entretanto, neste trabalho, a saída do NAP atua diretamente no elemento final de controle, diferentemente de [6], em que a saída do NAP atua otimizando ações PI (proporcional e integral), aprimorando-as. Como em [2], neste trabalho não se conjuga LPA2v com ações PID convencionais. Ainda, neste trabalho, toda a programação do sistema de controle foi feita com base na norma IEC 61131, como em [2], o que nos demais trabalhos não foi feito.

Na próxima seção serão apresentadas as conclusões deste trabalho, e possibilidades de trabalhos futuros.

VII. CONCLUSÕES E TRABALHOS FUTUROS

Nesta pesquisa, verificou-se o desenvolvimento de um método paraconsistente de controle com a implementação de um bloco funcional baseado na LPA2V, utilizando um algoritmo do NAP em um CP fundamentando a programação na norma IEC 61131, em seu item 3. Essa implementação ainda apresenta iniciativa que expande a norma IEC 61131 em seu item 7, de aplicação de lógicas não-clássicas em CPs. Os resultados obtidos ampliam ainda mais o campo de utilização de lógicas não-clássicas aplicadas a CPs.

Os resultados obtidos dos estudos no controle da vazão, na unidade experimental, comprovaram que o uso do NAP é um método alternativo em controle de processos.

Para trabalhos futuros, propõe-se: (i) a comparação do desempenho do NAP_CONTROL deste trabalho com controladores PID que utilizam LF; (ii) a aplicação do sistema de controle deste trabalho em outras malhas de controle e plantas; (iii) a comparação de resultados do NAP_CONTROL com resultados de outros controladores que usam LPA2v; e (iv) aplicação de outros controladores que usam LPA2v em CPs utilizando a norma IEC 61131.

¹trata-se do *crisp* normalizado.

REFERÊNCIAS

- [1] M. T. COELHO, E. M. C. JUNIOR, L. F. QUINTINO, C. A. PIAZZA, and A. A. DE ANDRADE, "A evolução das tecnologias opc como subsídio para as fábricas inteligentes," *XIV CEEL*, 2016.
- [2] C. Fernandes, "Lógica paraconsistente aplicada em sistemas de automação e controle," Ph.D. dissertation, Dissertação de Mestrado em Engenharia Mecânica–Universidade Santa Cecília ..., 2012.
- [3] J. Abe, "Foundations of annotated logics," *Doctor thesis, University of São Paulo, Brazil (in Portuguese)*, 1992.
- [4] A. Kaufman and M. M. Gupta, *Introduction to fuzzy arithmetic*. Van Nostrand Reinhold Company New York, 1991.
- [5] M. Coelho, "Controle pi híbrido com lógica paraconsistente aplicado em malha de controle de nível em vaso pressurizado," Ph.D. dissertation, Dissertação de Mestrado em Engenharia Mecânica–Universidade Santa Cecília ..., 2015.
- [6] M. S. Coelho, J. I. da Silva Filho, H. M. Côrtes, A. de Carvalho Jr, M. F. Blos, M. C. Mario, and A. Rocco, "Hybrid pi controller constructed with paraconsistent annotated logic," *Control Engineering Practice*, vol. 84, pp. 112–124, 2019.
- [7] H. Côrtes, J. Da Silva Filho, M. Blos, and B. Zanon, "Implementation of a paraconsistent-fuzzy digital pid controller in a level control process," in *Paper accepted on the 19th International Conference on Fuzzy Computation and Application (ICFCA)*, 2017.
- [8] J. I. Da Silva Filho, J. M. Abe, and G. L. Torres, "Inteligência artificial com as redes de análises paraconsistentes," *LTC-Livros Técnicos e Científicos Editora SA, São Paulo*, vol. 313, 2008.
- [9] F. R. d. CARVALHO and J. M. Abe, "Tomadas de decisão com ferramentas da lógica paraconsistente anotada," *São Paulo: Editora Edgard Blucher Ltda*, 2011.
- [10] J. I. Da Silva Filho, "Métodos de aplicações da lógica paraconsistente anotada de anotação com dois valores-lpa2v com construção de algoritmo e," *Universidade de São Paulo*, 1999.
- [11] J. I. da Silva Filho, "Métodos de aplicações da lógica paraconsistente anotada de anotação com dois valores-lpa2v," *Revista Seleção documental*, vol. 1, no. 1, pp. 18–25, 2006.
- [12] G. C. Goodwin, S. F. Graebe, M. E. Salgado *et al.*, *Control system design*. Prentice hall New Jersey, 2001, vol. 240.
- [13] K. OGATA, "Engenharia de controle moderno. 4ª edição, 2003."
- [14] K. J. Åström, "Control system design lecture notes for me 155a," *Department of Mechanical and Environmental Engineering University of California Santa Barbara*, vol. 333, 2002.
- [15] M. Tiegelkamp and K.-H. John, *IEC 61131-3: Programming industrial automation systems*. Springer, 2010.
- [16] E. P. Legaspe, "Controlador fuzzy de código aberto para uso em controladores programáveis." Ph.D. dissertation, Universidade de São Paulo.
- [17] P. Cingolani and J. Alcalá-Fdez, "jfuzzylogic: a robust and flexible fuzzy-logic inference system language implementation," in *2012 IEEE International Conference on Fuzzy Systems*. IEEE, 2012, pp. 1–8.

Received: 07 July 2021;
Accepted: 19 August 2022;
Published: 20 August 2022;



© 2022 by the authors. Submitted for possible open access publication under the terms and conditions of the Creative Commons Attribution (CC-BY) license (<http://creativecommons.org/licenses/by/4.0/>).

Control System Using A PAN Algorithm Applied To An Experimental Flow Rate Controller

Abstract—Paraconsistent logic (PL) algorithm applications have been researched for a successful implementation in various industries, including robotics, electronics, and process control (which might associate logic with conventional control techniques). The purpose of this study was to implement a paraconsistent analyzer node (PAN) in a programmable controller (PC), based on annotated paraconsistent logic principles and fundamentals with annotation of two values (PAL2v) — a branch of PL. The PAN algorithm was directly implemented in a real control loop on the flow rate variable. It was implemented in an experimental unit with two reservoirs, a flowmeter, a pump, a frequency inverter, and supervisor software to control the flow rate through a pipeline connecting both reservoirs. The PAN was made possible through a NAP_PARACON functioning block programmed into the PC, which received a setpoint signal (equaling the desired flow rate through the pipeline) and the process variable (PV) signal representing the flow rate from the flowmeter. Thus, the NAP_PARACON functioning block acted as a paraconsistent control system, receiving both input signals (SP and PV) and emitting an output signal to the frequency inverter, which controlled the pump speed, and consequently, the intended flow rate. The results obtained demonstrated reliable PAN algorithm behavior, similar to that of the proportional–integral–derivative algorithm, providing a new alternative for closed-loop processes, based on the paraconsistent control action. This technique, using PAL2v, expands the initiative concurrently with item 3 of the IEC 61131 standard, which specifies the application of nonclassical logics in PCs.

Index Terms—Industrial automation, IEC 61131, Nonclassical logics, Paraconsistent logic, Flow rate control.