

**UNIVERSIDADE SANTA CECÍLIA  
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA MECÂNICA  
MESTRADO EM ENGENHARIA MECÂNICA**

**RODRIGO SILVÉRIO DA SILVEIRA**

**DESENVOLVIMENTO DE CÉLULA NEURAL ARTIFICIAL PARA CONSISTENTE  
EM CONTROLADOR PROGRAMÁVEL PARA APLICAÇÃO EM TÉCNICAS DE  
APRENDIZAGEM POR DEMONSTRAÇÃO APLICADAS À AUTOMAÇÃO**

**SANTOS/SP**

**2018**

**RODRIGO SILVÉRIO DA SILVEIRA**

**DESENVOLVIMENTO DE CÉLULA NEURAL ARTIFICIAL PARA CONSISTENTE  
EM CONTROLADOR PROGRAMÁVEL PARA APLICAÇÃO EM TÉCNICAS DE  
APRENDIZAGEM POR DEMONSTRAÇÃO APLICADAS À AUTOMAÇÃO**

Dissertação apresentada a Universidade Santa Cecília como parte dos requisitos para obtenção de título de mestre no Programa de Pós-Graduação em Engenharia Mecânica, sob a orientação do Prof. Dr. João Inácio da Silva Filho e coorientação do Prof. Me. Cláudio Luís Magalhães Fernandes.

**SANTOS/SP**

**2018**

Autorizo a reprodução parcial ou total deste trabalho, por qualquer que seja o processo, exclusivamente para fins acadêmicos e científicos.

511.191 SILVEIRA, Rodrigo Silvério.  
S591d Desenvolvimento de célula neural artificial paraconsistente em controlador programável para aplicação em técnicas de aprendizagem por demonstração aplicadas à automação/Rodrigo Silvério da Silveira - 2018.  
72f.: il.

Orientador: Prof. Dr. João Inácio da Silva Filho.  
Coorientador: Prof. Me. Cláudio Luís Magalhães Fernandes.

Dissertação (Mestrado) - Universidade Santa Cecília, Programa de Pós-Graduação em Engenharia Mecânica, Santos, 2018.

1. Lógica Paraconsistente Anotada. 2. Célula Neural Artificial Paraconsistente. 3. Controlador Programável. 4. Aprendizagem por demonstração. 5. Automação  
I. Da Silva Filho, João Inácio II. Desenvolvimento de célula neural artificial paraconsistente em controlador lógico programável para aplicação em técnicas de aprendizagem por demonstração aplicadas à automação.

## DEDICATÓRIA

*Dedico este trabalho a minha família, em especial a minha esposa e as minhas filhas que me apoiaram de diversas maneiras durante esta importante etapa da minha vida.*

## **AGRADECIMENTOS**

Este trabalho só foi possível de ser concluído graças ao sólido alicerce dado pelo apoio técnico e emocional das diversas pessoas citadas a seguir.

Ao meu orientador Prof. Dr. João Inácio da Silva Filho, que através da sua grande inteligência e simplicidade me conduziu a encontrar soluções para os problemas que surgiram ao longo deste trabalho.

Ao meu amigo e coorientador Cláudio Luís Magalhães Fernandes, pelas grandes oportunidades oferecidas e pelas valiosas informações transmitidas.

A todo excelente corpo docente do programa de Pós-Graduação em Engenharia Mecânica da UNISANTA, pelos ensinamentos e dedicação com todos os alunos.

Ao meu amigo Leonardo do Espírito Santo, pela sua fundamental importância no apoio técnico e principalmente emocional durante os estudos.

A escola SENAI -SP, pela estrutura disponibilizada para os ensaios deste trabalho.

A toda minha família, em especial a minha esposa Lislene Silveira e as minhas filhas Yasmim e Isabela que foram fundamentais nesta trajetória, pois tiveram paciência com as minhas ausências e me incentivaram muito, fazendo com que eu conseguisse ter energia para superar todos os desafios.

E claro, agradeço muito a Deus pois é graças a ele que tudo se concretizou.

## RESUMO

Em Inteligência Artificial a “Aprendizagem Por Demonstração-*ApD* (*learning from demonstration - LfD*) é o processo no qual se baseia em ensinar novas tarefas às máquinas por meio de manipulação repetitiva, sem a necessidade de programação. Nesta pesquisa utiliza-se a Lógica Paraconsistente Anotada – LPA para criar um sistema de Aprendizagem capaz de ensinar tarefas a uma máquina através de demonstrações repetitivas. Para isto uma Célula Neural Artificial Paraconsistente de aprendizagem CNAPap foi implementada em um Controlador Programável (CP) constituindo-se em um Sistema Paraconsistente de Aprendizagem Por Demonstração - SPApD-LPA2v construído em um Bloco Funcional denominado FB\_CNAP, que possui características de modularidade com base nas Normas Técnicas IEC 61131-3. O Bloco Funcional FB\_CNAP teve seu algoritmo escrito no software TIA Portal, que pertence ao Controlador Programável (CP) Siemens. Para testes de validação o SPApD-LPA2v foi aplicado em uma máquina com processo de movimentação de peças eletropneumático, composto por atuadores lineares, ventosa e sensores analógicos de posicionamento linear. Os ensaios com o SPApD-LPA2v, foram realizados de maneira que a máquina aprendesse padrões de movimentos executados em modo manual e após o treinamento concluído, os movimentos foram repetidos sem intervenção de um operador. Os testes foram desenvolvidos com movimentos definidos manualmente, com diferentes distâncias e metas de aprendizagem e a introdução de ruídos no processo. As respostas foram bastante eficientes, o que indica que, apesar de a implementação ter sido com uma única CNAPap os resultados obtidos neste trabalho demonstram que o SPApD-LPA2v se apresenta fundamentalmente como uma potencial alternativa para novas implementações de controles em processos industriais e ao campo da robótica, se adequando perfeitamente a estes novos conceitos na automação. A pesquisa traz uma importante contribuição a área da automação industrial, pois devido as suas características de modularidade e seus algoritmos fundamentados em Lógicas não clássicas, o SPApD-LPA2v mostra-se como uma ferramenta inovadora onde se pode obter maior facilidade e flexibilidade na execução de modificações de tarefas sem necessidade de reprogramação.

**Palavras Chave:** Lógica Paraconsistente Anotada. Célula Neural Artificial Paraconsistente. Controlador Programável. Aprendizagem por demonstração. Automação.

## ABSTRACT

In Artificial Intelligence Learning from Demonstration - LfD is the process in which it relies on teaching new tasks to the machines through repetitive manipulation, without the need for programming. In this research we use the Paraconsistent Annotated Logic (PAL) to create a Learning system capable of teaching tasks to a machine through repetitive demonstrations. For this, a Paraconsistent Artificial Neural Cell for learning PANCell<sub>e</sub> was implemented in a Programmable Controller (PC) constituting a System Learning Paraconsistent by Demonstration - SLeP-PAL2v built in a Functional Block denominated FB\_CNAP, that has characteristics of modularity based on the IEC 61131-3 Technical standards. The FB\_CNAP Function Block has its algorithm written in the TIA Portal software, which belongs to the Siemens Programmable Controller (PC). For validation tests the SLeP-PAL2v was applied in a machine with an electro-pneumatic parts movement process, consisting of linear actuators, suction cups and linear positioning analog sensors. The SLeP-PAL2v tests were performed in such a way that the machine learned patterns of movements performed in manual mode and after the training was completed, the movements were repeated automatically without the intervention of an operator. The tests were developed with manually defined movements, with different distances and learning goals and the introduction of noise in the process. The results were quite efficient, which indicates that although the implementation was with a single PANCell<sub>e</sub> the results obtained in this work demonstrate that the SLeP-PAL2v presents fundamentally as a potential alternative for new implementations of controls in industrial processes and in the field of robotics, fitting perfectly to these new concepts in automation. The research brings an important contribution to the area of industrial automation, because due to its modularity characteristics and algorithms based on non-classical Logic, SLeP-PAL2v is an innovative tool where greater ease and flexibility can be achieved in the execution of modifications without the need for reprogramming.

**Keywords:** Paraconsistent Annotated Logic. Paraconsistent Artificial Neural Cell. Programmable Controller. Demonstration learning. Automation.

## LISTA DE ILUSTRAÇÕES

Figura 1 - Reticulado da LPA2v e regiões delimitadas. ....	6
Figura 2 - Símbolo de uma CNAPap. ....	8
Figura 3 - Valores obtidos no processo de aprendizagem e desaprendizagem. ....	11
Figura 4- Controlador Programável S7-1500. ....	15
Figura 5 - <i>Rack</i> da configuração de hardware. ....	18
Figura 6 - Configuração de <i>Hardware</i> . ....	19
Figura 7 – Estrutura da programação no <i>Software</i> TIA Portal. ....	20
Figura 8 – Criação do FB_CNAP. ....	20
Figura 9 - Declaração das variáveis de entrada. ....	21
Figura 10 - Declaração das variáveis de saída. ....	21
Figura 11 - Declaração das variáveis estáticas. ....	21
Figura 12 – Bloco de Conversão SCALE. ....	22
Figura 13 - Normalização dos valores do algoritmo original para 0 a 100. ....	22
Figura 14 – Representação do algoritmo do bloco FB_CNAP. ....	23
Figura 15 – Bloco Funcional FB_CNAP. ....	24
Figura 16 – Manipulador utilizado para os testes. ....	25
Figura 17 – Detalhes do Manipulador eletropneumático de três eixos. ....	26
Figura 18– Alimentação do transdutor Linear. ....	27
Figura 19 - Correspondência deslocamento do cilindro e a tensão do transdutor. ....	27
Figura 20 - Sistema de movimentação de peças. ....	28
Figura 21 – Sistema de movimentação de peças. ....	29
Figura 22 – Tela do Sistema de <i>interface</i> Supervisório. ....	30
Figura 23 – Endereçamentos do bloco FB_CNAP. ....	31
Figura 24 - Valores de aprendizagem do FB_CNAP com $u_1 = 100$ e $F_A = 100$ . ....	34
Figura 25 - Valores de aprendizagem do FB_CNAP com $\mu_1 = 0,0055$ e $F_{DA} = 100$ . ....	35
Figura 26 - Valores de aprendizagem do FB_CNAP com $\mu_1 = 25,275$ e $F_A = 100$ . ....	36
Figura 27 - Resultados de diferentes Fatores de Aprendizagem $F_A$ . ....	37
Figura 28 - Aprendizado do FB_CNAP com desvio de 90,455. ....	38
Figura 29 - Gráfico de aprendizado do FB_CNAP com desvio de 70,1297. ....	39
Figura 30 - Deslocamento do cilindro para a posição 75,15 (526,05 mm). ....	40
Figura 31 - Deslocamento do cilindro para a posição 50,13 (352,1mm). ....	41
Figura 32 - Deslocamento do cilindro para a posição 25,24 (176,68 mm). ....	42
Figura 33 - Deslocamento do cilindro para a posição 100 (700 mm). ....	43

## LISTA DE TABELAS

Tabela 1-Valores do processo de aprendizagem e desaprendizagem de uma CNAPap .....	11
Tabela 2 – Aplicação do FB_CNAP com $\mu_1 = 100$ e $F_A = 100$ .....	34
Tabela 3 – Aplicação do FB_CNAP com $\mu_1 = 0,0055$ e $F_A = 100$ .....	35
Tabela 4 – Aplicação do FB_CNAP com $\mu_1 = 25,275$ e $F_A = 100$ .....	36
Tabela 5 - Comportamento do FB_CNAP com diferentes Fatores de aprendizagem $F_A$ . .....	37
Tabela 6 – Aprendizado do FB_CNAP com desvio de 90,455 .....	38
Tabela 7 – Aprendizado do FB_CNAP com desvio de 70,1297 .....	39
Tabela 8 – Teste de deslocamento do sistema na posição 75,15 (526,05 mm).....	40
Tabela 9 – Testes de deslocamento do sistema na posição 50,13 (352,1 mm).....	41
Tabela 10 – Teste de deslocamento do sistema na posição 25,24 (176,68 mm).....	42
Tabela 11 – Testes de deslocamento do sistema na posição 100 (700 mm).....	43

## LISTA DE QUADROS

Quadro 1 - Dados do Controlador Programável Siemens utilizado .....	15
Quadro 2 - Tabela de endereçamento do CP com as suas aplicações.....	17

## LISTA DE ABREVIATURAS E SIGLAS

ApD	Aprendizagem por Demonstração
<i>LfD</i>	<i>Learning from Demonstration</i>
CNAPap	Célula Neural Artificial Paraconsistente de aprendizagem
CAPb	Célula Artificial Paraconsistente básica
LPA	Lógica Paraconsistente Anotada
LPA2v	Lógica Paraconsistente Anotada com anotação de dois valores
SPApD	Sistema Paraconsistente de Aprendizagem Por Demonstração
CP	Controlador Programável
CLP	Controlador Lógico Programável
IEC	<i>International Electrotechnical Commission</i>

## LISTA DE SÍMBOLOS

$\mu_1$	Grau de crença ou grau de evidência favorável 1.
$\mu_E$	Grau de evidência resultante
$\mu_{ER}$	Grau de evidência resultante real
$\mu_{ctr}$	Grau de contradição normalizado
$G_{ct}$	Grau de contradição
$G_c$	Grau de certeza
$G_{CR}$	Grau de certeza calculado real
$\lambda$	Grau de evidência desfavorável
$\varphi_E$	Intervalo de evidência real
$F_A$	Fator de Aprendizagem
$F_{DA}$	Fator de Aprendizagem
$T$	Inconsistente
$\perp$	Paracompleto ou indeterminado
$\perp \rightarrow f$	Indeterminado, tendendo ao Falso;
$\perp \rightarrow v$	Indeterminado, tendendo ao Verdadeiro;
$T \rightarrow f$	Inconsistente, tendendo ao Falso;
$Qv \rightarrow T$	Quase Verdadeiro, tendendo ao Inconsistente;
$Qf \rightarrow T$	Quase Falso, tendendo ao Inconsistente;
$Qf \rightarrow \perp$	Quase Falso, tendendo ao Indeterminado;
$Qv \rightarrow \perp$	Quase Verdadeiro, tendendo ao Indeterminado.
$V_{scc}$	Variável superior de controle de certeza ( $C_1$ );
$V_{icc}$	Variável inferior de controle de certeza ( $C_2$ );
$V_{scct}$	Variável superior de controle de contradição ( $C_3$ );
$V_{icct}$	Variável inferior de controle de contradição ( $C_4$ ).

## SUMÁRIO

1	INTRODUÇÃO.....	1
1.1	A Aprendizagem por Demonstração - ApD.....	3
1.2	Lógica Paraconsistente Anotada .....	5
1.3	Célula Neural Artificial Paraconsistente com Aprendizagem e Desaprendizagem – CNAPap.....	7
1.4	Norma IEC 61131-3 .....	12
1.5	Objetivos.....	13
2.	MATERIAIS E MÉTODOS .....	14
2.1	Características do Controlador Programável (CP).....	14
2.2	Características do Software utilizado.....	18
2.3	Configuração Inicial do Software TIA PORTAL.....	18
2.4	Elaboração do bloco FB_CNAP .....	19
2.4.1	Programação do bloco FB_CNAP .....	22
2.5	Implementação do Bloco Funcional FB_CNAP para a “Aprendizagem Por Demonstração- APD .....	25
2.5.1.	Funcionamento do Bloco FB_CNAP no controle do sistema .....	30
3.	RESULTADOS E DISCUSSÃO .....	33
4.	CONCLUSÕES.....	45
5.	VERSÃO COMPACTA.....	48
	REFERÊNCIAS.....	53
	APÊNDICE A - Programação do bloco FB_CNAP .....	57
	ANEXO A - Fluxograma do processo de Aprendizagem e Desaprendizagem da CNAPap.....	59

## 1 INTRODUÇÃO

A competitividade industrial cresce a cada dia e com isso as exigências relacionadas a produção se elevam na mesma proporção. Dessa forma a indústria a fim de garantir maior eficiência, maior qualidade com custos reduzidos, busca tornar os seus processos cada vez mais automatizados (FERNANDES et al., 2012).

Atualmente um produto tem como característica possuir um ciclo de vida reduzido, no qual a sua renovação ou até mesmo inovação devem ser constantes, isto faz com que grandes empresas invistam em tecnologias, tornando os seus processos mais flexíveis a modificações. Dessa forma, a automação busca otimizar o processo produtivo, fornecendo um produto com menor custo, em tempos curtos e com a maior quantidade e qualidade possível (ROSÁRIO, 2009). Dentro deste contexto está a robótica e o controle de máquinas, no qual estão cada vez mais sofisticadas proporcionando assim crescentes ganhos produtivos as empresas. No entanto, o alto custo de programação dos equipamentos e o tempo gasto com novas implementações, consomem recursos importantes das empresas e de certa forma causam restrições quando se pensa em modificações no processo fabril (KARLSSON et al.,2017). Desta forma pesquisas avançam a fim de proporcionar que a programação dos robôs possa ser feita de maneira mais rápida, flexível e intuitiva, utilizando para isso conceitos como o de aprendizagem por demonstração. A técnica conhecida como “Aprendizagem por Demonstração- ApD (*Learning from Demonstration - LfD*) é uma das mais difundidas atualmente e propõe que um robô aprenda com as demonstrações feitas por um “professor” (BRENNAN et al., 2008). Com base nesta técnica, novos estudos surgem a fim de contribuir para a melhora dos resultados, como no caso deste trabalho, onde se utiliza a Lógica Paraconsistente Anotada como base para criar um processo de aprendizagem por demonstrações repetitivas. Para isto utiliza-se a Célula Neural Artificial Paraconsistente de aprendizagem CNAPap como elemento base para a construção deste sistema denominado Sistema Paraconsistente de Aprendizagem Por Demonstração - SPAPD-LPA2v. O sistema SPAPD-LPA2v pode ser implementado em diversas plataformas, porém neste trabalho, optou-se pelo uso do Controlador Programável (CP), que é um dispositivo amplamente utilizado em aplicações de controle industrial.

Os Controladores Programáveis (CPs), são dispositivos de fundamental

importância para automação, sendo considerado o “cérebro” do controle de uma planta industrial. O CP evolui constantemente desde a sua criação em 1968, e de acordo com esta evolução, a lógica não clássica e a Inteligência Artificial vêm ganhando espaço em diversos modelos de controladores, com o objetivo de atender de forma mais eficiente a automação de processos industriais (FERNANDES et al., 2012). As lógicas não clássicas desafiam os princípios binários da clássica. Um tipo de lógica não clássica é a Paraconsistente que, por sua vez, trata as regiões não atingidas pela lógica clássica, como os valores intermediários entre verdadeiro e falso. Essa condição possibilita um melhor tratamento de indefinições, ambiguidades e inconsistências (DA SILVA FILHO, 2007).

Com base na Lógica Paraconsistente Anotada (LPA), surgem as Redes Neurais Artificiais Paraconsistentes (RNAPs), formadas basicamente por Células Neurais Artificiais Paraconsistente (CNAP) e que pertencem à área da Inteligência Artificial (IA), cuja característica é ter o comportamento de um neurônio biológico. Apesar das aplicações concretas com Inteligência Artificial e especificamente com Células Neurais Artificiais Paraconsistentes, aumentarem a cada dia, ainda são poucas, principalmente em CPs, ficando muito conhecimento restrito a área acadêmica (DA SILVA FILHO, 2007).

Diante do contexto apresentado, será demonstrado neste trabalho a construção de um Bloco Funcional fundamentado no Sistema Paraconsistente de Aprendizagem Por Demonstração - SPAPD-LPA2v. O seu Bloco Funcional é denominado FB\_CNAP e tem a sua construção baseada na norma IEC 61131-3 e implementação em um Controlador Programável (CP).

Um bloco funcional pode ser construído com características de lógicas não clássicas, de modo a permitir uma maior modularidade em aplicações que envolvam automação industrial baseada em lógicas como a Paraconsistente (FERNANDES, 2012). A criação do bloco funcional conforme a IEC 61131-3 tem como objetivo agregar maior eficiência e flexibilidade aos processos industriais, permitindo que usuários de sistemas de automação industrial utilizem as Células Neurais Artificiais Paraconsistentes de forma modular, o que facilitará implementações diversas, como as que envolvem as Redes Neurais Artificiais Paraconsistentes aplicadas a Aprendizagem por Demonstração. Neste trabalho, para demonstrar a sua aplicação, será utilizado um sistema de movimentação de peças, no qual o bloco FB\_CNAP irá sofrer um processo de treinamento em modo manual através do conceito de

Teleoperação e depois de treinado deverá controlar o sistema sem intervenção de operador.

### **1.1 A Aprendizagem por Demonstração - ApD**

A robótica tem evoluído de maneira significativa ao longo anos, se tornando um elemento cada vez mais fundamental em vários segmentos, como em aplicações espaciais, industriais e até cotidianas. Em sistemas de automação industrial, sempre se buscou agregar os robôs ao controle de processos, com o objetivo de adquirir maior produtividade e qualidade, como na indústria automotiva, onde os robôs estão começando a atuar de modo colaborativo, ou seja, trabalhando em conjunto com seres humanos em aplicações específicas de montagem.

Para o controle de robôs, algoritmos com grande complexidade são criados e implementados de modo a conseguir realizar movimentos com elevada precisão. Em um sistema tradicional, a cada modificação na trajetória do robô, é necessária a intervenção de especialistas para se determinar um novo movimento, o que não proporciona flexibilidade quanto a novas tarefas. Um método que permite proporcionar maior rapidez quanto a atribuição de movimentos é a “Aprendizagem Por Demonstração- ApD (*Learning from Demonstration - LfD*), que tem como fundamento utilizar os conceitos de aprendizagem dos seres humanos, onde se normalmente adquire habilidades por meio de tutela direta, através de observação, emulação de objetivos, imitação e outras interações sociais (SCASSELLATI, 1999b).

Em um sistema ApD, os robôs ou outras máquinas aprendem tarefas por meio de demonstrações, permitindo que novos movimentos sejam implementados sem intervenção direta na sua programação. Este sistema pode proporcionar inúmeras vantagens, como a facilidade de não especialistas atuarem na dinâmica do robô, ou na redução de tempo em lhe atribuir novas tarefas, proporcionando assim maior produtividade no caso de aplicações industriais.

Neste trabalho são utilizados os conceitos da Aprendizagem por Demonstração (ApD) e para o seu desenvolvimento optou-se por utilizar os princípios das redes neurais, mais especificamente da Células Neurais Artificiais Paraconsistentes-CNAPs. Segundo (HAYKIN, 2008), o aprendizado de máquinas com redes Neurais podem ser categorizados de duas formas: aprendizado com um professor ou sem o professor.

Na aprendizagem sem professor, o processo ocorre sem a supervisão do mesmo e dessa forma não possuem exemplos a serem seguidos pela rede neural. Este tipo de aprendizagem pode ser dividido em aprendizagem por reforço e auto-organizada.

Na aprendizagem com professor, também denominada como aprendizagem supervisionada, conceitua-se que o professor detém o conhecimento do ambiente com base em um conjunto de entrada e saída. Em uma etapa de demonstração, considera-se para a rede neural, que o ambiente é desconhecido, mas se o professor e a rede ficarem expostos em um vetor de treinamento, que é um exemplo extraído do ambiente, o professor consegue fornecer a rede neural uma resposta adequada para o vetor.

Para que um processo de aprendizagem ocorra com sucesso, os estados e as ações das execuções no conjunto de dados de aprendizagem devem ser utilizados pelo aluno, ou seja, os estados e as ações do professor mapeiam diretamente o aluno. No entanto em aplicações reais, este conceito pode não ser possível, visto que o aluno e professor se diferem em questões de percepção ou mecânica (BRENNNA et al., 2008). Em uma etapa de demonstração algumas técnicas podem ser utilizadas para a sua execução e registro, neste caso foi destacada uma de comum utilização, a Teleoperação, pois foi à aplicada na implementação deste trabalho. Esta técnica possui a característica de transferir diretamente as informações, no qual sequências de movimentos são definidas e operadas pelo professor, enquanto o aprendiz grava dados a partir de seus próprios sensores. A Teleoperação não é recomendada a todo tipo de sistema, devido a dificuldade em realizar movimentos teleoperados de maior complexidade, como por exemplo, em robôs humanoides, entretanto em aplicações com controles mais simples, como em máquinas industriais, a teleoperação pode facilmente ser utilizada devido as características de movimentação deste tipo de aplicação. (BRENNNA et al., 2008).

## 1.2 Lógica Paraconsistente Anotada

A base tecnológica da indústria atual continua sendo fundamentada na lógica clássica, que utiliza como raciocínio apenas valores binários, ou seja, falso (0) ou verdadeiro (1), não permitindo o uso de valores intermediários, como quase verdadeiro ou quase falso. Porém com o avanço tecnológico, a indústria está cada vez mais automatizada e de certa forma “inteligente”, o que gera a necessidade em determinadas aplicações do emprego de lógicas que consigam interpretar dados mais fiéis as situações reais, que normalmente são carregados de incertezas e contradições.

Para atender essa demanda da tecnologia atual, algumas lógicas não-clássicas estão disponíveis para serem exploradas e aplicadas. Dentre estas, as que mais se destacam são a *Fuzzy*, as Trivalentes, as Multivaloradas e a Paraconsistente (NETTO et al.,2013).

A Lógica Paraconsistente é uma lógica não clássica, proposicional e evidencial e já foi formalizada e descrita com suas equações e todos os seus predicados em (ABE, 1998) e (DA SILVA FILHO et al.,2008).

A Lógica Paraconsistente Anotada com anotação de dois valores (LPA2v) tem como fundamento interpretar duas evidências denominadas como crença ( $\mu_1$ ) e descrença ( $\mu_2$ ), onde uma é considerada favorável e outra desfavorável (DA SILVA FILHO, 2006) a proposição que está sendo analisada. A representação da LPA2v é feita por meio de um reticulado de 4 vértices que através de anotações de dois valores, é possível expressar estados lógicos extremos em cada um de seus vértices. Estas anotações são geradas a partir de medições efetuadas no meio físico que podem, através da representação no Reticulado da LPA2v, atribuir conotação lógica a proposição P. Estas informações são interpretadas como entradas e são de grande importância na tomada de decisão do algoritmo. A interpretação resulta da intersecção dos valores de Graus de Certeza ( $G_c$ ) e Graus de contradição ( $G_{ct}$ ), cujos cálculos são demonstrados a seguir.

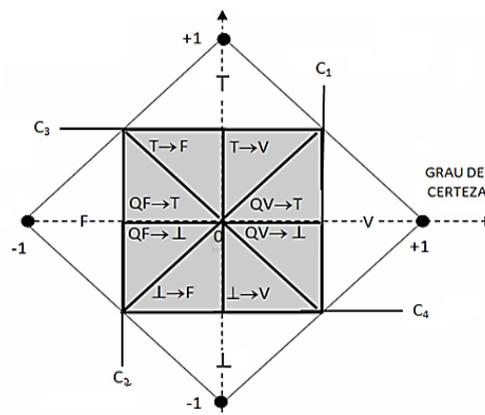
$$G_c = \mu_1 - \lambda \tag{1}$$

$$G_{ct} = \mu_1 + \lambda - 1 \tag{2}$$

Com base nos dois valores gerados de  $G_c$  e  $G_{ct}$ , é formado um par ordenado que fornece um único ponto situado dentro do reticulado definido como estado lógico

paraconsistente.

O reticulado da LPA com anotação de dois valores (LPA2v) pode gerar uma infinidade de pontos e consequentemente infinitos estados lógicos em seu interior. Para possibilitar o controle de algumas aplicações, o reticulado é dividido em apenas 12 regiões, no qual os estados lógicos extremos são delimitados por C1, C2, C3 e C4 definidos como: Verdadeiro, Falso, Indeterminado e Inconsistente (DA SILVA FILHO et al., 2012). Na figura 1 é mostrado o reticulado da LPA2v contendo os 4 estados lógicos paraconsistentes estabelecidos pelos valores extremos e também os 8 valores Não-extremos.



**Figura 1 - Reticulado da LPA2v e regiões delimitadas.**  
Fonte: (DA SILVA FILHO, 2007)

A seguir são apresentados os estados não extremos:

- $\perp \rightarrow \mathbf{F}$  Indeterminado tendendo a Falso
- $\perp \rightarrow \mathbf{V}$  Indeterminado tendendo a Verdadeiro
- $\mathbf{T} \rightarrow \mathbf{F}$  Inconsistente tendendo a Falso
- $\mathbf{T} \rightarrow \mathbf{V}$  Inconsistente tendendo a Verdadeiro
- $\mathbf{QV} \rightarrow \mathbf{T}$  Quase verdadeiro tendendo a inconsistente
- $\mathbf{QF} \rightarrow \mathbf{T}$  Quase – falso tendendo a Inconsistente
- $\mathbf{QV} \rightarrow \perp$  Quase – falso tendendo a Indeterminado
- $\mathbf{QF} \rightarrow \perp$  Quase – verdadeiro tendendo a Indeterminado

A partir dos valores inseridos nas entradas  $\mu$  e  $\lambda$ , o algoritmo da LPA2v os processa fornecendo um dos 12 estados lógicos determinados pelas regiões delimitadas e assim foi possível obter o algoritmo da LPA2v que é denominado Para-Analisador (DA SILVA FILHO et al., 2012).

### 1.3 Célula Neural Artificial Paraconsistente com Aprendizagem e Desaprendizagem – CNAPap

O algoritmo Para-Analisador possui a característica de tratar informações consideradas incompletas e contraditórias e, portanto, demonstra semelhanças com o comportamento humano. Desta forma as informações recebidas pelos sentidos, é um processo mental biológico que pode ser modelado pela Lógica Paraconsistente Anotada de Anotação de dois valores (LPA2v) tornando possível assim, construir algoritmos com características de Células Neurais Artificiais Paraconsistentes (MARIO et al., 2007).

Com base nos fundamentos da LPA2v construiu-se a Célula Artificial Paraconsistente básica (CAPb) que possui um algoritmo simplificado e que a partir dos sinais recebidos efetua a tomada de decisão. Pode-se dizer que uma CAPb possui a capacidade de após receber informações de graus de evidência favorável ( $\mu$ ) e desfavorável ( $\lambda$ ), disponibiliza na sua saída três estados lógicos: Grau de Contradição resultante ( $G_{ct}$ ), Grau de certeza resultante ( $G_c$ ), ou Indefinição (I).

Utilizando como base a Célula Artificial Paraconsistente básica CAPb, foram construídas as Células Neurais Artificiais Paraconsistentes, as quais são de diversos tipos, com finalidades específicas. Com base na Célula Artificial Paraconsistente básica (CAPb), foi formada a Célula Neural Artificial Paraconsistente de Aprendizagem e desaprendizagem (CNAPap). Esta célula após treinada deve ser capaz de aprender padrões entre os valores reais 0 e 1. O padrão em uma Rede Neural Artificial Paraconsistente é definido como um dígito binário, no qual 1 equivale ao estado lógico “verdadeiro” e o valor 0 significa estado lógico “falso”.

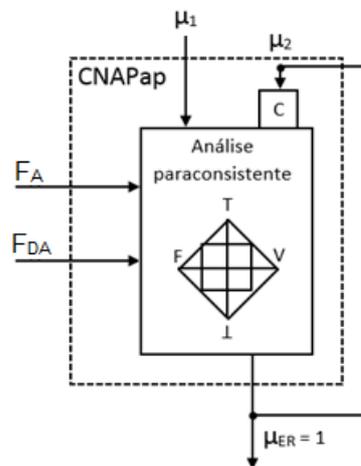
Uma CNAPap pode ser preparada para aprender dois tipos de padrões, um denominado de verdade e o outro falsidade. O aprendizado do padrão verdade ocorre quando a CNAPap recebe em sua entrada o valor 1 repetidas vezes e a cada repetição, o grau de evidência resultante  $\mu_{ER}$  é incrementado de acordo com o algoritmo até atingir o valor 1 (DA SILVA FILHO, 2001).

Assim como no processo de aprendizado do padrão de verdade, o aprendizado do padrão de falsidade acontece quando a CNAPap recebe em sua entrada o valor 0 repetidas vezes até o grau de evidência resultante chegar a 0.

Com o objetivo de proporcionar um controle da rapidez e também da precisão do aprendizado da CNAPap, é inserido nesta, um fator de aprendizado ( $F_A$ )

que pode ser ajustado externamente de acordo com a necessidade da aplicação. O fator de aprendizado possui valores no intervalo entre 0 e 1. Quando o  $F_A$  é ajustado com valor igual a 1, a CNAPap necessitará de mais passos possíveis para concluir um treinamento, neste caso caracteriza-se uma capacidade de aprendizado natural. Ao reduzir o  $F_A$ , o aprendizado natural da célula diminui e quando  $F_A$  é igual 0 a célula perde a capacidade de aprender.

Em um processo de desaprendizagem, ao aplicar um valor repetidas vezes na entrada da CNAPap, o grau de evidência resultante será decrementado chegando a um valor de indefinição. Caso esse valor continue a ser repetido na entrada, o grau de evidência continuará diminuindo até chegar a zero, caracterizando assim o aprendizado de um padrão contrário. A sinalização da CNAPap para um processo de aprendizagem também deve ser 1 e para isso é utilizado um operador NOT (DA SILVA FILHO et al., 2008). Assim como no processo de aprendizagem é utilizado o fator  $F_A$ , no processo de desaprendizagem, utiliza-se o fator  $F_{DA}$ , que é ajustado externamente a célula. O símbolo que representa uma Célula Neural Artificial Paraconsistente de Aprendizagem (CNAPap) é demonstrado na figura 2.



**Figura 2 - Símbolo de uma CNAPap.**  
Fonte: (MARIO et al., 2007)

A capacidade completa de aprendizado e desaprendizado sobre qualquer valor de padrões entre 0 e 1, são características fundamentais da CNAPap, o que proporciona a eficiência necessária para aplicações em redes Neurais Artificiais. Todas essas características foram utilizadas como base para a construção deste trabalho, no qual aplicações reais poderão ser feitas a fim de controlar sistemas de automação.

A construção do bloco FB\_CNAP é fundamentada na Célula Neural Artificial Paraconsistente de aprendizagem (CNAPap), cuja característica é aprender ou desaprender qualquer valor real no intervalo fechado  $[0,1]$  depois de receber um processo de treinamento (MÁRIO et al., 2007). A equação para determinar o grau de evidencia resultante com base em processo de aprendizagem do padrão de verdade, pode ser vista a seguir:

Para:  $0 \leq F_A \leq 1$

$$\mu_{E(K+1)} = \frac{\{ \mu_1 - (\mu_{E(K)c}) F_A \} + 1}{2} \quad (3)$$

Sendo,

$$\mu_{E(K)c} = 1 - \mu_{E(K)}$$

A equação para determinar o padrão de falsidade é demonstrada a seguir:

Para:  $0 \leq F_{DA} \leq 1$

$$\mu_{E(K+1)} = \frac{\{ \mu_{1c} - (\mu_{E(K)c}) F_{DA} \} + 1}{2} \quad (4)$$

Sendo,

$$\mu_{1c} = 1 - \mu_1 \text{ e } \mu_{E(K)c} = 1 - \mu_{E(K)}$$

A Célula Neural Artificial Paraconsistente de Aprendizagem CNAPap deve ter condições ao final dos passos de um treinamento, aprender o padrão apresentado na entrada da célula. Para que o processo de aprendizagem do padrão de verdade aconteça, o padrão aplicado deve ser 1, dessa forma a cada passo realizado é fornecido um valor de grau de evidência resultante  $\mu_{ER}$ , no qual as diferenças geradas entre a entrada e saída são consideradas contradições que se anularão apenas quando  $\mu_{ER}$  chegar a 1. Em um processo de desaprendizagem, ao aplicar um valor 0 repetidas vezes na entrada da CNAPap, o grau de evidência resultante será decrementado chegando a um valor de indefinição (DA SILVA FILHO et al.,2008). Caso esse valor continue a ser repetido na entrada, o grau de evidência continuará diminuindo até chegar a zero, caracterizando assim o aprendizado de um padrão contrário. A sinalização da CNAPap para um processo de aprendizagem também deve ser 1. A seguir é apresentado o algoritmo de aprendizagem do padrão de verdade (DA SILVA FILHO et al.,2008).

- 1) Início  $\mu E = 0,5$  (Célula Virgem)
- 2) Entre com o padrão de entrada inicial:  
 $\mu_i = P_{di} \quad (0 \leq P_{di} \leq 1)$
- 3) Calcule o grau de evidência inicial  

$$\mu E = \frac{\mu_i - 0,5 + 1}{2}$$
- 4) Determine o padrão a ser aprendido pelas condições:  
 Se  $\mu E = 0,5$ , então vá para o item 1 (início)  
 Se  $\mu E > 0,5$ , então vá para o item 6
- 5) Complemente a entrada do Grau de evidência favorável  
 $\mu_i = 1 - P_d$
- 6) Entre com o fator de aprendizagem, FA:  
 $FA = C_4 \quad (0 \leq FA \leq 1)$
- 7) Conecte a saída da célula na entrada do grau de Evidência desfavorável  
 $\mu_2 = \mu E$
- 8) Aplique o Operador Complemento na entrada do grau de Evidência desfavorável  
 $\mu_{2c} = 1 - \mu_2$
- 9) Calcule o grau de evidência  

$$\mu E = \frac{\{\mu_1 - (\mu_{2c})C_4\} + 1}{2}$$
- 10) Determine o próximo passo pelas condições:  
 Se  $\mu E = 1$ , sinalize célula treinada com padrão de Verdade e vá para o item 14.  
 Se  $0 < \mu E < 1$ , então vá para o item 15.  
 Se  $\mu E = 0$ , sinalize célula treinada com padrão de Falsidade e vá para o próximo item.
- 11) Faça a negação lógica na saída:  $\mu E = 1$
- 12) Faça o complemento na entrada do grau de Evidência favorável  
 $\mu_i C = 1 - P_D$
- 13) Troque FA pelo fator de desaprendizagem, FDA :  
 $FDA = C_4$
- 14) Entre com o novo padrão de entrada:  
 $P_n = P_d \quad (0 \leq P_n \leq 1)$
- 15) Retorne ao passo 7

De acordo com os padrões 0 e 1 na entrada da CNAPap e com fator de aprendizagem igual a 1, serão apresentados a seguir na tabela 1 os valores fornecidos pelo algoritmo da CNAPap durante um processo completo de aprendizagem e desaprendizagem.

Tabela 1-Valores do processo completo de aprendizagem e desaprendizagem de uma CNAPap.

Passo	$\mu_1$	$G_c$	$G_{ct}$	$\mu_E$	Observações
0	1	0,0000000000	-1,0000000000	0,5000000000	Início
1	1	0,5000000000	0,5000000000	0,7500000000	Aprendizagem
2	1	0,7500000000	0,2500000000	0,8750000000	
3	1	0,8750000000	0,1250000000	0,9375000000	
4	1	0,9375000000	0,0625000000	0,9687500000	
5	1	0,9687500000	0,0312500000	0,9843750000	
6	1	0,9843750000	0,0156250000	0,9921875000	
7	1	0,9921875000	0,0078125000	0,9960937500	
8	1	0,9960937500	0,0039062500	0,9984687500	
9	1	0,9984687500	0,0019531250	0,9992343750	
10	1	0,9992343750	0,0009765620	0,9996171870	
	<b>1</b>	<b>1,0000000000</b>	<b>0,0000000000</b>	<b>1,0000000000</b>	Término
11	0	-1,0000000000	-1,0000000000	0,5000000000	Início
12	0	-0,5000000000	-0,5000000000	0,2500000000	Desaprendizagem
13	0	-0,7500000000	-0,2500000000	0,1250000000	
14	0	-0,8750000000	-0,1250000000	0,0625000000	
15	0	-0,9375000000	-0,0625000000	0,0312500000	
16	0	-0,9687500000	-0,0312500000	0,0156250000	
17	0	-0,9843750000	-0,0156250000	0,0078125000	
18	0	-0,9960937500	-0,0039062500	0,0039062500	
19	0	-0,9980468750	-0,0019531250	0,0015312500	
20	0	-0,9990234370	-0,0009765620	0,0004882810	
21	0	-1,0000000000	0,0000000000	0,0000000000	Término
<b>22</b>	<b>0</b>	<b>1</b>	<b>0</b>	<b>1</b>	Confirma Falsidade

Fonte: (DA SILVA FILHO et al.,2008)

De acordo com os valores dos graus de evidência obtidos na tabela anterior é apresentado na figura 3, o gráfico correspondente.

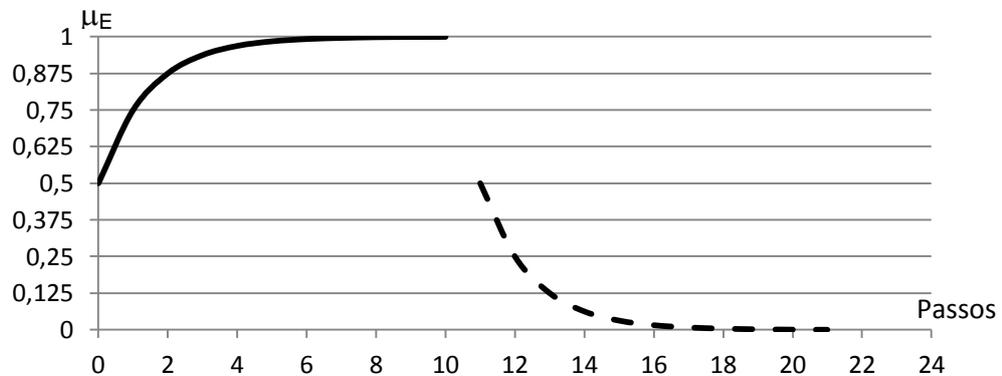


Figura 3 - Valores obtidos no processo de aprendizagem e desaprendizagem.

Fonte: (DA SILVA FILHO et al.,2008)

## 1.4 Norma IEC 61131-3

Neste trabalho foi seguido os conceitos da norma IEC 61131-3, cujo objetivo é estabelecer padrões para Controladores Programáveis (CPs) no que diz respeito a execução do programa, a estrutura de *software* e principalmente para linguagens de programação. Antes de a norma ser criada, cada fabricante de CP adotava a sua, ou seja, não existia um padrão comum definido e isso gerava grandes dificuldades para as empresas. Por exemplo, em um processo industrial totalmente automatizado, significa que há uma grande quantidade de programação de CP envolvida. Se, após alguns anos de operação, esse processo industrial necessitasse de alteração no seu funcionamento, por motivos como aumento de produção e melhoria na segurança, poderia ser necessário modificar a programação dos CPs. E para isso, normalmente, apenas o programador que fez o programa teria condições de alterá-lo, pois naquela época o programa possuía características únicas devido a falta de padronização da linguagem, o que acabava sendo extremamente complicado de se realizar por outras pessoas (SILVEIRA, 2013).

Para resolver esse, dentre outros problemas, em 1992, a IEC publicou a primeira edição da norma IEC 61131-3. A partir daí, foi definido um padrão para linguagens de programação dos Controladores Programáveis e que os fabricantes deveriam seguir. Outro importante benefício da norma e que fora utilizado neste trabalho, está relacionado ao desenvolvimento de programas baseados nos seguintes princípios (JOHN, 2010):

- a) **Modularização:** torna possível decompor um programa simples ou complexo em partes menores, possibilitando maior entendimento e controle sobre ele;
- b) **Estruturação:** possibilita elaborar um programa de forma hierárquica, ou seja, em níveis, o que também proporciona a reutilização de blocos funcionais;
- c) **Tarefas (Tasks):** controla a execução de programas ou de blocos funcionais de forma periódica ou mesmo por eventos. A sua criação é necessária em programas mais complexos, mas principalmente em situações de emergência. Por exemplo, no caso de algum defeito em um equipamento em que deve ocorrer a sua parada imediata, o sistema interrompe o ciclo normal de processamento e atende prioritariamente a linha de programa específico dessa emergência.

## 1.5 Objetivos

Com o objetivo de tornar mais flexíveis novas implementações no que diz respeito à programação de máquinas na área de automação industrial, este trabalho busca novas alternativas no controle da automação, de modo a agregar maior eficiência para a produção industrial. Desta forma este trabalho propõe a inclusão de Células Neurais Artificiais em sistemas de Aprendizagem por Demonstração (ApD) de maneira modular em Controladores Programáveis (CP). Este dispositivo é considerado um elemento essencial para o controle da indústria atual, o que justifica a sua utilização na implementação deste trabalho. Pesquisas que envolvem Inteligência Artificial e sistemas de Aprendizagem por Demonstração com Células Neurais Artificiais Paraconsistentes, são raras ficando muito estudo limitado a área acadêmica (DA SILVA FILHO, 2007).

O bloco FB\_CNAP construído com base na CNAPap possibilitará que usuários da área de automação tenham um acesso prático a uma CNAPap, tornando mais simples as novas implementações industriais, visto que o bloco traz a principal vantagem de possuir modularidade, pois foi construído seguindo os padrões da norma IEC 61131-3.

Contudo a pesquisa se mostra também com grande relevância para pesquisadores, vistos as poucas publicações neste segmento, ou seja, o trabalho realizado cria a perspectiva de que mais estudos e implementações sejam feitas na área de Inteligência Artificial aplicada a automação industrial.

Como objetivo secundário, esta pesquisa pretende contribuir para que novos estudos sejam realizados na área de “Aprendizagem Por Demonstração- ApD (*Learning from Demonstration – LfD*) utilizando como base a Lógica Paraconsistente Anotada.

Outro Objetivo secundário está na possibilidade da construção de Redes Neurais Artificiais Paraconsistentes em Controladores Programáveis.

Outra abordagem está nas aplicações de modularidade da norma IEC 61131-3 na área de Inteligência Artificial.

## 2. MATERIAIS E MÉTODOS

De acordo com os objetivos propostos deste trabalho, será apresentado neste capítulo a construção do Bloco Funcional (FB\_CNAP) fundamentado no algoritmo completo da Célula Neural Artificial Paraconsistente (CNAPap), bem como nas regras determinadas pela norma IEC 61131-3.

O bloco FB\_CNAP possui como característica principal a sua modularidade, no qual pode ser facilmente aplicado de forma isolada ou até mesmo na possibilidade de união a fim de formar lógicas diversas ou até mesmos Redes Neurais Artificiais Paraconsistentes.

Para a realização da implementação, fora utilizado um único fabricante de CP, porém todo o conceito abordado poderá ser facilmente aplicado em qualquer outro, desde que seja feita as adequações peculiares a cada fabricante.

Esta facilidade de adaptação se deve a padronização realizada pela norma IEC 61131-3 e que foi adotada amplamente neste trabalho, possibilitando, por exemplo, que o algoritmo escrito em linguagem *Ladder* possa ser reutilizado de forma simples sem muitas adaptações.

Será apresentado a seguir todo o procedimento para a construção do bloco FB\_CNAP bem como a sua aplicação em um sistema de aprendizado Por Demonstração- ApD para a uma ação de movimentação de peças, a fim de verificar o seu comportamento.

### 2.1 Características do Controlador Programável (CP)

Neste trabalho foi utilizado o Controlador Programável (CP) do fabricante SIEMENS modelo S7-1500 (figura 4), composto por uma Fonte de Alimentação (1) Unidade Central de Processamento (2), um Módulo de Entrada Digital (3), um Módulo de Saída Digital (4), além dos Módulos de Entrada (5) e Saída Analógica(6).



Figura 4– Controlador Programável S7-1500.

O quadro 1 mostra as características técnicas dos módulos que compõem controlador programável utilizado neste trabalho:

Quadro 1 - Dados do Controlador Programável Siemens utilizado

	Módulo	Modelo	Características Gerais
1	Fonte	6EP1333-4BA00	Potência 190 W, 120/230 VAC; Tensão de 24 VDC com fornecimento na parte frontal.
2	CPU	6ES7 516-3AN01-0AB0	CPU com display; memória de trabalho de MB e de dados de 5MB; Tempo de instrução de 10 ns bit; Conceito de proteção em 4 estágios, funções de tecnologia: controle de movimento, controle em circuito fechado, contagem e medição; rastreamento; 1ª interface: controlador PROFINET IO, suporta RT / IRT, atualização de desempenho PROFINET V2.3, 2 portas, I-device, MRP, MRPD, protocolo de transporte TCP / IP, comunicação de usuário aberta e segura, comunicação S7, servidor web, cliente DNS, Acesso de dados do servidor OPC UA, tempo constante do ciclo de ônibus, roteamento; 2ª interface: controlador PROFINET IO, suporta RT, I-dispositivo, protocolo de transporte TCP / IP, comunicação de usuário aberta e segura, comunicação S7, servidor Web, cliente DNS, acesso a dados do servidor OPC UA, roteamento; 3ª interface: mestre PROFIBUS DP, comunicação S7, tempo constante do ciclo de barramento, roteamento; Opções de tempo de execução, firmware V2.0.

**Quadro 1- Dados do Controlador Programável Siemens utilizado (continuação)**

3	Entrada Digital (DI)	6ES7 521-1BL00-0AB0	Módulo de entrada digital DI32 x DC24V; atraso de entrada 0,05 a 20ms; diagnósticos configuráveis; interrupções de hardware; modo isócrono.
4	Saída Digital (DQ)	6ES7 522-1BL01-0AB0	Módulo de saída digital DQ32 x DC24V / 0,5A; diagnósticos configuráveis; valor de substituição configurável para saída; modo isócrono.
5	Entrada Analógica (AI)	6ES7 531-7KF00-0AB0	Módulo de entrada analógica AI8 x U / I / RTD / TC 16 bits; agrupando 8; 4 canais com medição de RTD; tensão em modo comum 10 V; diagnósticos configuráveis; interrupções de hardware.
6	Saída Analógica (AQ)	6ES7 532-5HD00-0AB0	Módulo de saída analógica AQ4 x U / I 16 bits; agrupando 4; diagnósticos configuráveis; valor de substituição configurável para saída.

Para a implementação do bloco FB\_CNAP foram necessárias 9 entradas digitais, 6 saídas digitais e 1 entrada analógica. As entradas digitais foram utilizadas para o recebimento dos sinais provenientes dos botões e sensores montados externamente ao CP. As saídas digitais foram utilizadas para acionar as válvulas solenoides e os sinalizadores. Já a entrada analógica realiza a leitura do transdutor de posição e envia para entrada ( $\mu_1$ ) do bloco FB\_CNAP. Mais detalhes quanto aos endereçamentos e as aplicações das entradas e saídas do CP podem ser observados no quadro 2.

**Quadro 2 - Tabela de endereçamento do CP com as suas aplicações.**

		<b>Aplicação</b>
<b>Entradas Digitais</b>	%I0.0	Reset - Reinicia o Bloco FB_CNAP
	%I0.1	Refresh – Atualização o padrão de entrada u1
	%I0.2	Sensor magnético do atuador
	%I0.4	Botão para o avanço do cilindro do eixo X
	%I0.5	Botão para o recuo do cilindro do eixo X
	%I0.6	Botão para o Avanço do cilindro do eixo Z
	%I0.7	Botão para ligar o vácuo
	%I1.0	Chave – Modo Imitação
<b>Saídas Digitais</b>	%Q0.0	Válvula solenoide para o avanço do cilindro do eixo X
	%Q0.1	Válvula solenoide para o recuo do cilindro do eixo X
	%Q0.2	Válvula solenoide para o Avanço do cilindro do eixo Z
	%Q0.3	Válvula solenoide para ligar o vácuo
	%Q0.4	Sinalizador de Célula Treinada
	%Q0.5	Sinalizador modo Imitação
<b>Entrada Analógica</b>	%ID4	Entrada Analógica proveniente do transdutor de posição

O módulo de entrada analógica (AI) utilizado disponibiliza a leitura de diferentes tipos de grandezas e faixas de valores, o que torna possível adequar o CP aos diversos tipos de equipamentos e instrumentos instalados em uma planta industrial. Neste trabalho, foi ajustado o módulo AI para realizar a leitura dos valores de 0 a 10V, devido o transdutor de posição linear, fornecer somente este tipo de sinal.

## 2.2 Características do Software utilizado

Para a elaboração do programa, utilizou-se o *software* TIA PORTAL da SIEMENS, que segue os padrões da norma IEC 61131-3. Este *software* integra também pacotes de configuração de Redes e de *Interface* Homem-Máquina (IHM). A sua escolha ocorreu pelo fato de que o seu fabricante detém grande fatia do mercado mundial na área de automação e, portanto, os resultados deste trabalho, poderiam vir a atingir uma vasta gama de aplicações e usuários.

De acordo com a IEC 61131-3, o TIA PORTAL possibilita desenvolver programas estruturados, permitindo que a estrutura da lógica de programação seja organizada em Blocos Funcionais, Programas e Funções. Tal estrutura traz vantagens como, modularidade e reutilização dos códigos no desenvolvimento de aplicações para área de automação industrial.

## 2.3 Configuração Inicial do Software TIA PORTAL

O processo de construção do Bloco Funcional FB\_CNAP, iniciou-se pela configuração de Hardware no *software* TIA PORTAL, onde foi necessário inserir de forma virtual os módulos correspondentes aos que estão instalados fisicamente e que compõem o conjunto Controlador Programável (CP).

Para configurar o *Hardware*, foi necessário acessar a “árvore de projetos” e clicar em *Device Configuration*. Nesta tela (figura 5) é apresentado o Rack (*Rail*), cuja função é integrar todos os módulos do CP.

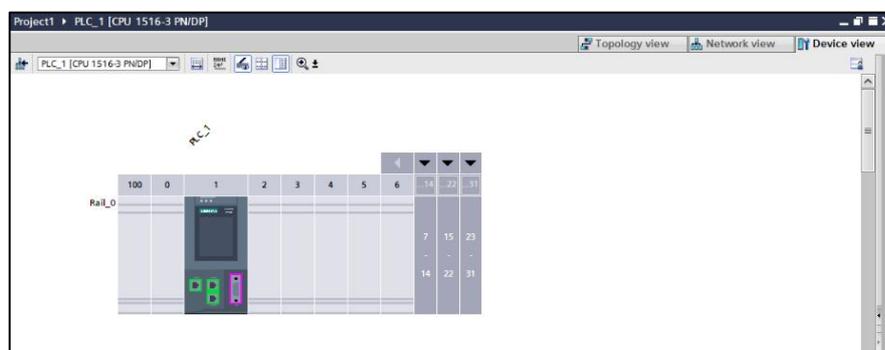


Figura 5 - Rack da configuração de hardware.

O Rack é composto por Slots e em cada um destes é alocado um módulo específico, previamente selecionado no catálogo virtual. A sequência de instalação

destes módulos obedeceu a seguinte ordem:

- a) **Slot 0** - Fonte de Alimentação (*PS – Power Supply*).
- b) **Slot 1** – Unidade Central de Processamento (*CPU – Central Processing Unit*).
- c) **Slot 2** – Módulo de Entrada Digital (*DI – Digital Input*).
- d) **Slot 3** – Módulo de Saída Digital (*DQ – Digital Output*).
- e) **Slot 4** – Módulo de Entrada Analógica (*AI – Analog Input*).
- f) **Slot 5** – Módulo de Saída Analógica (*AQ – Analog Output*).

Os *Slots* preenchidos pelos módulos citados anteriormente podem ser vistos na figura 6.

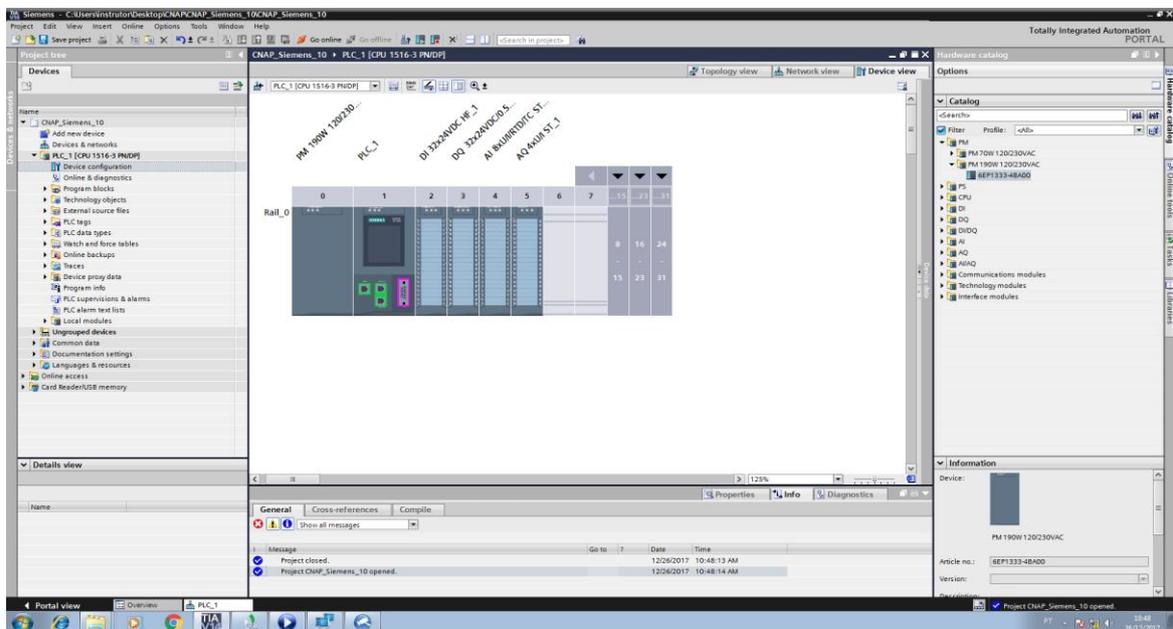


Figura 6 - Configuração de *Hardware*.

## 2.4 Elaboração do bloco FB\_CNAP

De acordo com a programação estruturada de um CP, configurou-se um programa principal, aqui denominado *Organization Block* (OB1) com a função de “chamar” outros blocos, bem como executar toda a rotina de programação. Já o Bloco funcional deve armazenar suas informações em um Bloco de Dados denominado (DB – *Date Block*). A tela que apresenta a hierarquia de estrutura da programação no *software* TIA PORTAL é demonstrada na figura 7:

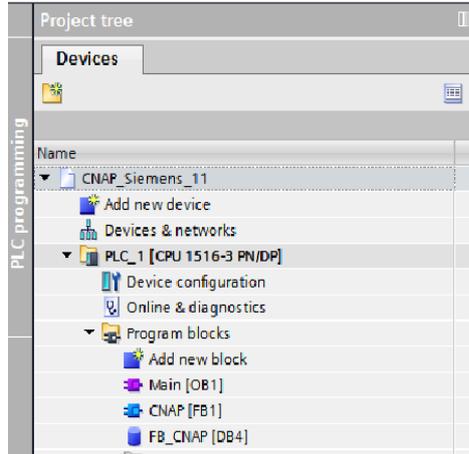


Figura 7 – Estrutura da programação no **Software TIA Portal**.

Na figura 8 é possível visualizar a tela de configuração (*Add new block*), onde foi selecionado o tipo Bloco Funcional (FB - *Function Block*). Após a escolha sua escolha foi atribuído o nome FB\_CNAP e selecionado um Bloco de Dados (DB – *Date Block*), com a função de armazenar os dados internos do bloco. A programação interna do bloco fora realizada com base no algoritmo completo da Célula Neural Artificial Paraconsistente de Aprendizagem (CNAPap) no qual foi escolhida a linguagem *Ladder*, devido a sua ampla utilização pelos profissionais da área de automação industrial.

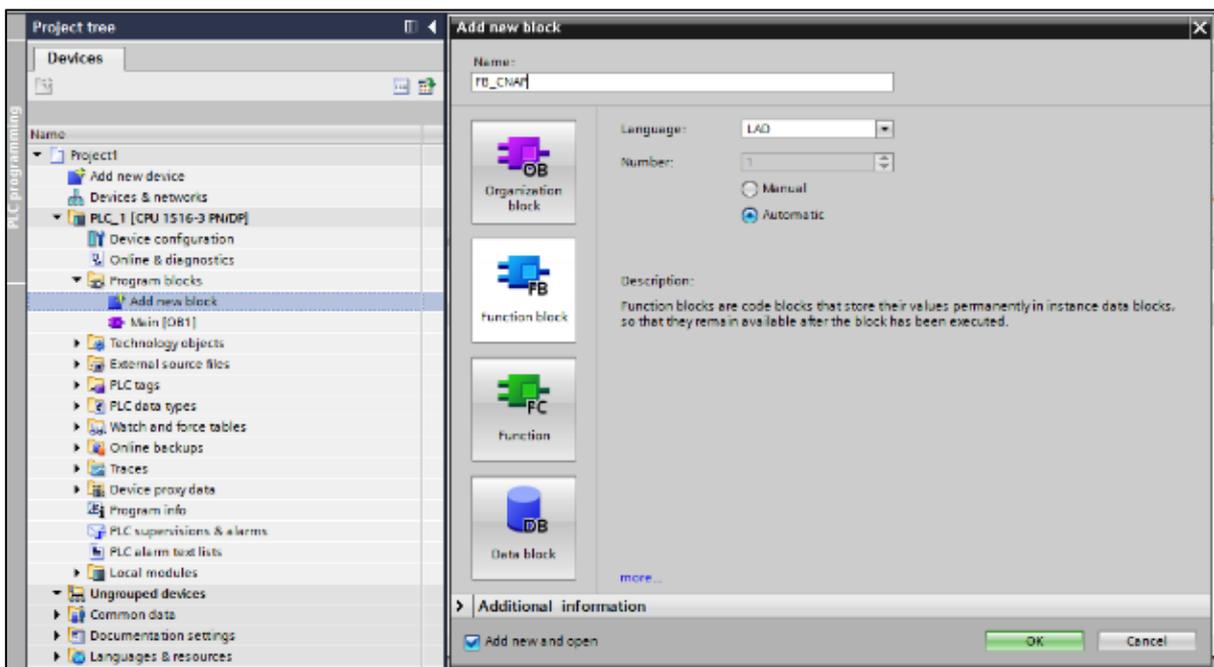


Figura 8 – Criação do FB\_CNAP.

Após a criação do bloco FB\_CNAP, foi dado início a sua configuração interna, através da declaração das variáveis de entradas, saídas e as estáticas. Esta configuração é feita utilizando símbolos e, para tal, devemos declará-los na Tabela de Declaração de Variáveis do *software* TIA PORTAL.

Inicialmente foram declaradas as variáveis de entrada (*input*) do FB\_CNAP (*u1*, *Fa*, *Fda*, *Reset* e *Refresh*). A figura 9 apresenta a declaração das variáveis de entrada.

CNAP							
	Name	Data type	Default value	Retain	Accessible f...	Writa...	Visible in ...
1	Input				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
2	u1	Real	0.0	Non-ret...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
3	Fa/Fda	Real	0.0	Non-retain	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
4	Reset	Bool	false	Non-retain	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
5	Refresh	Bool	false	Non-retain	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
6	Output				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
7	uE	Real	0.0	Non-retain	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
8	uE_Sinalizada	Real	0.0	Non-retain	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Figura 9 - Declaração das variáveis de entrada.

A figura 10, apresenta a declaração das variáveis de saída (*Output*) do FB\_CNAP que terão a função de gerar os resultados do bloco.

6	Output				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
7	uE	Real	0.0	Non-retain	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
8	uE_Sinalizada	Real	0.0	Non-retain	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Figura 10 - Declaração das variáveis de saída.

As variáveis estáticas (*STATIC*), são dados locais de um bloco funcional (FB) armazenado em um bloco de dados *instance* (DB), o qual armazena os parâmetros formais e dados estáticos do bloco funcional e permanecem armazenados até a próxima vez que o bloco funcional for processado. A figura 11, apresenta a declaração das variáveis estáticas utilizadas na construção do algoritmo do FB\_CNAP (*Pdi*, *uEi*, *u2*, *C4*, *u2C*, *Gc*, *Gct* e *Refresh\_aux*), com os seus respectivos tipos, Real e Booleana.

11	Static				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
12	Pdi	Real	0.0	Non-retain	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
13	uEi	Real	0.0	Non-retain	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
14	u2	Real	0.0	Non-retain	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
15	C4	Real	0.0	Non-retain	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
16	u2C	Real	0.0	Non-retain	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
17	Gc	Real	0.0	Non-retain	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
18	Gct	Real	0.0	Non-retain	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
19	refresh_aux	Bool	false	Non-retain	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	

Figura 11 - Declaração das variáveis estáticas.

Com o objetivo de facilitar a compreensão dos resultados nos testes do bloco FB\_CNAP, os valores utilizados na escrita do algoritmo foram normalizados entre 0 a 100, sendo que o algoritmo original do CNAP possui valores que variam de 0 a 1. Para isso foi utilizado um bloco funcional padrão do programa TIA Portal denominado *SCALE*, o qual é possível visualizar na figura 12.

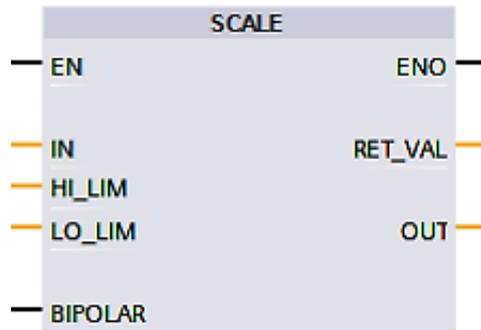


Figura 12 – Bloco de Conversão SCALE.

A comparação dos valores de conversão pode ser observada através do gráfico da figura 13.

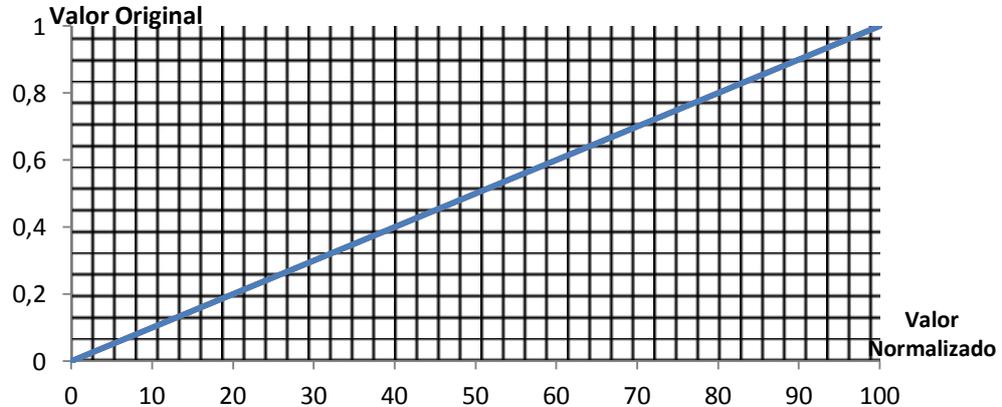


Figura 13 - Normalização dos valores do algoritmo original para 0 a 100.

#### 2.4.1 Programação do bloco FB\_CNAP

A escrita da programação do bloco FB\_CNAP seguiu as características do algoritmo completo de uma da Célula Neural Artificial Paraconsistente de aprendizagem e desaprendizagem, Para isso foi utilizado a linguagem *Ladder* e para realizar operações matemáticas foram feitos incrementos com a linguagem de Blocos denominada FBD. O algoritmo do FB\_CNAP é apresentado a na figura 14.

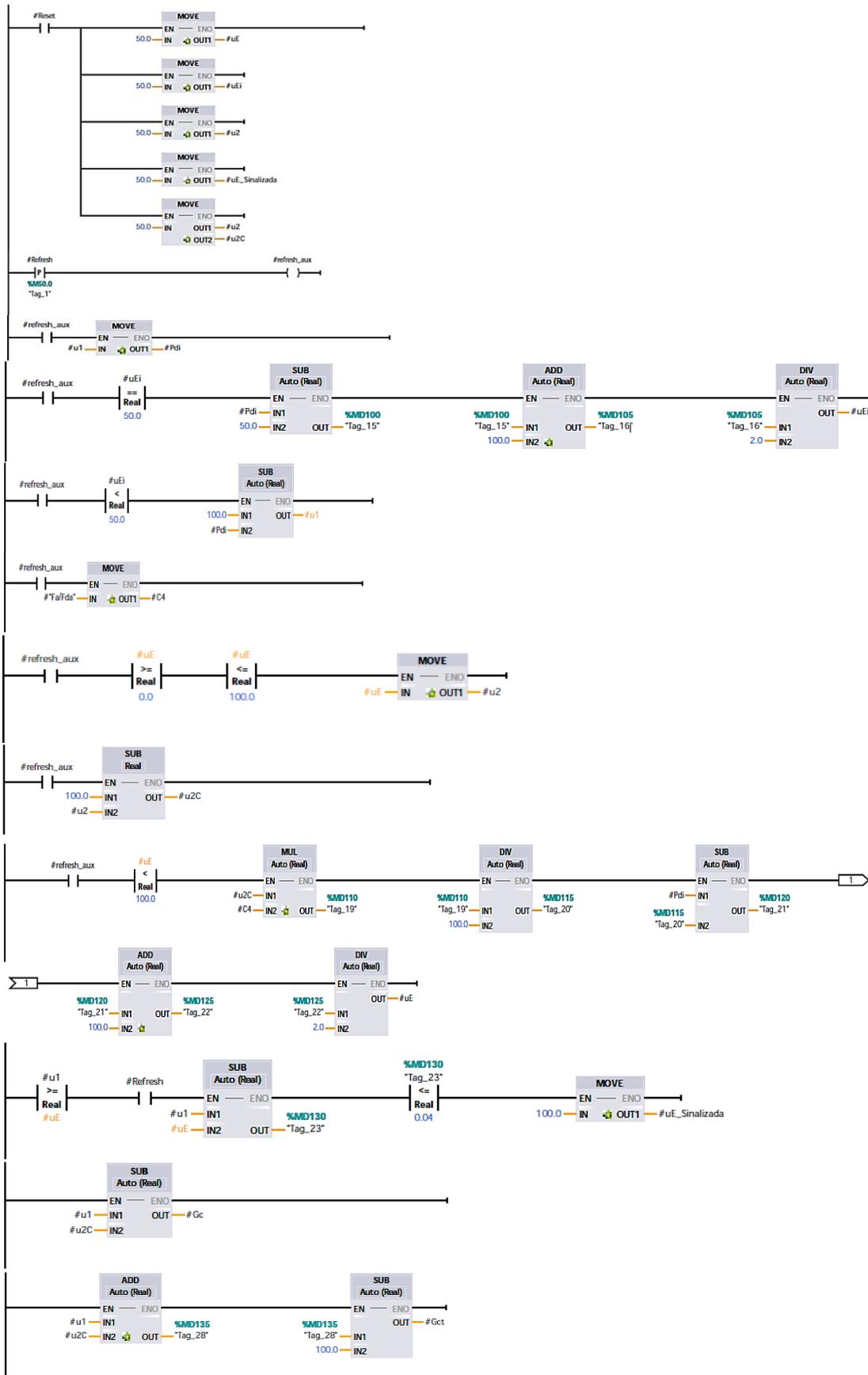


Figura 14 – Representação do algoritmo do bloco FB\_CNAP.

Após a conclusão da escrita do algoritmo, o bloco funcional FB\_CNAP foi importado na biblioteca virtual, o qual pode ser verificado por completo na Fig. 15. A esquerda do bloco estão localizadas as entradas: EN (Enable), o padrão de entrada u1, Fa/Fda, o Reset e o Refresh. As variáveis de saída estão localizadas a direita do bloco, que são: ENO (Enable Output), o grau de evidência uE e o uE\_Sinalizada. A fim de proporcionar uma melhor análise do comportamento do bloco FB\_CNAP, foi normalizado os valores de Fa, Fda, u1 e uE no intervalo entre 0 e 100, sendo que em uma CNAPap os valores correspondentes de FA, FDA  $\mu_1$  e  $\mu_E$  se situam entre 0 e 1.

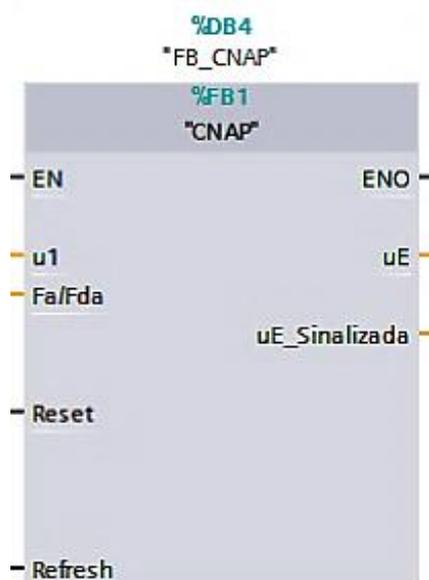


Figura 15 – Bloco Funcional FB\_CNAP.

A nomenclatura utilizada no bloco FB\_CNAP (u1, Fa/Fda e uE) correspondem respectivamente as variáveis ( $\mu_1$ ,  $\mu_E$  e  $F_A/F_{DA}$ ) utilizados em uma CNAPap, sendo estas as adotadas no decorrer deste trabalho com o propósito de facilitar a compreensão dos dados. A função das variáveis de entrada e saída do FB\_CNAP são descritas a seguir:

- a) **EN**: Possui a função de habilitar o bloco quando o seu nível lógico for 1.
- b)  $\mu_1$ : Recebe o padrão de entrada a ser aprendido pelo FB\_CNAP no intervalo de 0 a 100.
- c) **FA/FDA**: Correspondem a rapidez que a célula irá aprender o padrão de entrada. Recebe valores inteiros no intervalo de 0 a 100.
- d) **Reset**: Ao receber nível lógico 1, reinicia o bloco, atribuindo o valor 50 a saída  $\mu_E$ .
- e) **Refresh**: Atualiza o padrão de entrada após receber um pulso em nível lógico 1.

- f) **ENO**: Fornece nível lógico 1 quando o bloco está ativo.
- g)  $\mu_E$ : Fornece o grau de evidência, que corresponde a saída do bloco no intervalo de 0 a 100.
- h) **uE\_Sinalizada**: fornece nível lógico 1 quando o bloco finaliza a aprendizagem ou desaprendizagem.

## 2.5 Implementação do Bloco Funcional FB\_CNAP para a “Aprendizagem Por Demonstração- APD

A automação tem o seu conceito ligado a um conjunto de técnicas capazes de construir sistemas ativos, que atuam em sistemas produtivos com qualidade e eficiência através de informações recebidas do meio em que atuam (REZENDE, 2003). Desta forma um sistema de automação de posse de informações recebidas, calcula a ação corretiva mais adequada, ou seja, é possível dizer que um sistema de automação possui características do comportamento do ser humano, o qual age após tratar as informações sensoriais recebidas.

O bloco construído FB\_CNAP seguiu a norma IEC 61131-3 e possui características modulares, ou seja, pode ser utilizado quantas vezes forem necessárias, ou até mesmo possibilitar que implementações sejam feitas de maneira rápida, graças a disponibilidade da célula neural (FB\_CNAP) armazenada em uma biblioteca virtual dentro do *software* do CP.

Dessa forma, nesta parte do trabalho, será apresentada uma aplicação do bloco funcional FB\_CNAP, dentro de cenário de Aprendizagem por Demonstração (ApD) a fim de observar as suas características e comportamentos diante de um sistema de automação industrial. Para esta aplicação fora utilizado um manipulador de peças (figura 16).

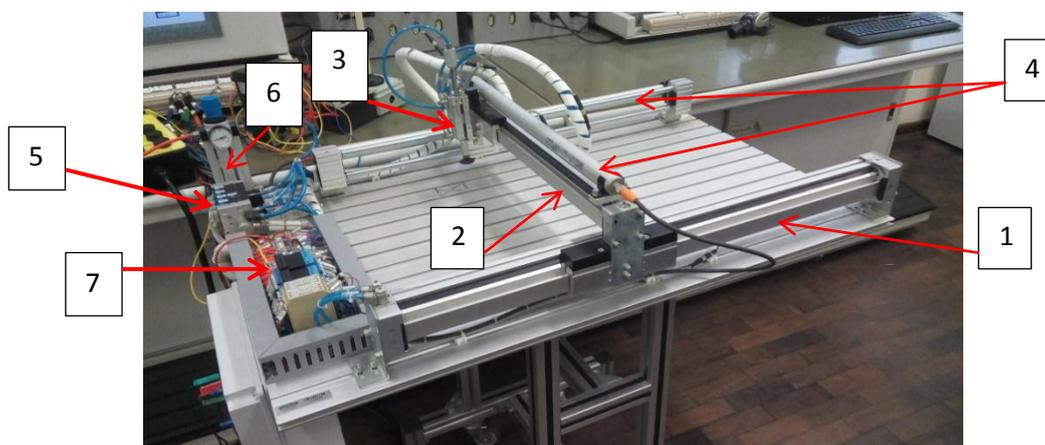


Figura 16 – Manipulador utilizado para os testes.

O bloco FB\_CNAP terá a função de aprender um padrão de movimento durante a realização de passos executados no modo manual. Após o bloco sinalizar que está treinado, o sistema deverá por imitação repetir de forma contínua os movimentos antes realizados manualmente.

Para a realização dos ensaios de aplicação do bloco FB\_CNAP, fora utilizado um manipulador de três eixos eletropneumático do fabricante Festo. O Manipulador (Figura 17) é composto por um cilindro de dupla ação sem haste (1) com montagem em pórtico, que realiza movimentos no eixo X, um cilindro de dupla ação sem haste fixado no cilindro anterior (2), o qual tem a função de se movimentar no eixo Y, um cilindro de haste passante vazada com ventosa (3), cuja função é movimentar no eixo Z, dois transdutores lineares (4), um bloco de válvulas eletropneumáticas (5), uma unidade de conservação de ar comprimido (6) e um conjunto de bornes (7) para a *interface* do controle elétrico do manipulador.

O conjunto possui arruela de pressão nos pontos de fixação, sensores magnéticos que são fixados nos atuadores e possuem conectores para a troca rápida e diodo *led* para a sinalização e todas as conexões pneumáticas possuem engate rápido anti-vazamento.

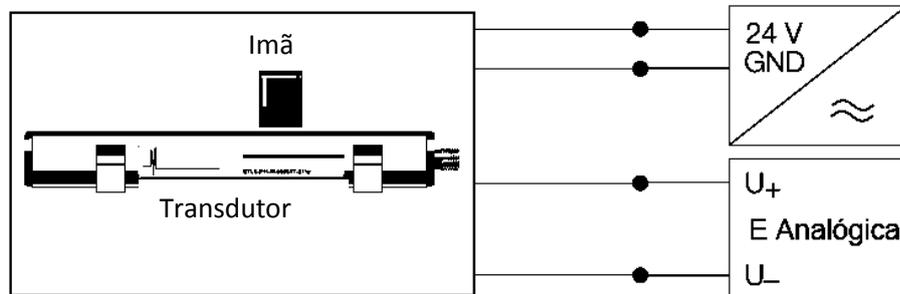


**Figura 17 – Detalhes do Manipulador eletropneumático de três eixos.**

O sinal inserido como padrão de entrada ( $\mu_1$ ) do bloco FB\_CNAP é proveniente de um transdutor linear de posicionamento modelo BTL6 do fabricante Balluf. O seu princípio de funcionamento baseia-se no conceito de magnetorestrição, que é uma propriedade na qual alguns tipos de materiais se deformam na presença de um campo magnético polarizado, como o de um ímã.

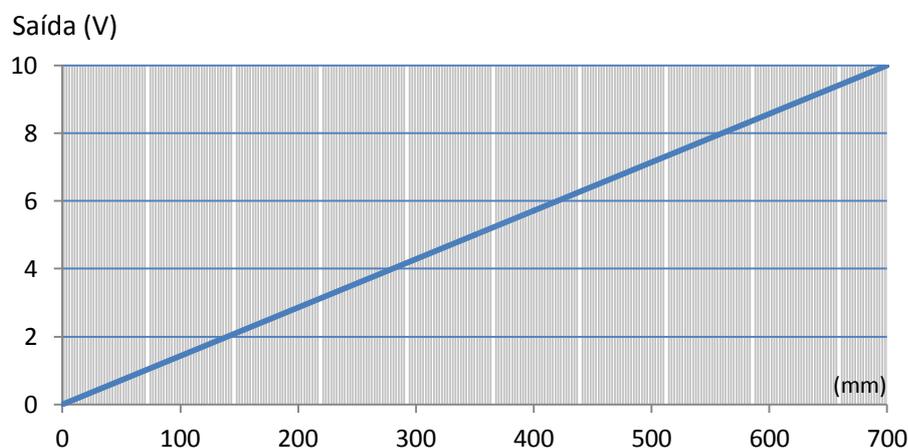
O transdutor possui um curso nominal de 700 mm, tensão de alimentação de

24Vcc e fornece uma tensão de saída em uma faixa analógica de 0 a 10V de acordo com o movimento do imã que está acoplado ao atuador do eixo X (figura 17).



**Figura 18– Alimentação do transdutor Linear.**

Para o controle do bloco FB\_CNAP, o cilindro deve ser movimentado pelo operador numa faixa de 0 a 700mm no eixo X, assim o transdutor interpreta esse posicionamento fornecendo um sinal o sinal analógico de 0 a 10V. Este sinal é recebido pela entrada u1 do bloco FB\_CNAP convertido em uma faixa de 0 a 100. O gráfico da figura 18 apresenta a correspondência entre o deslocamento do atuador do eixo X e a tensão de saída analógica do transdutor.

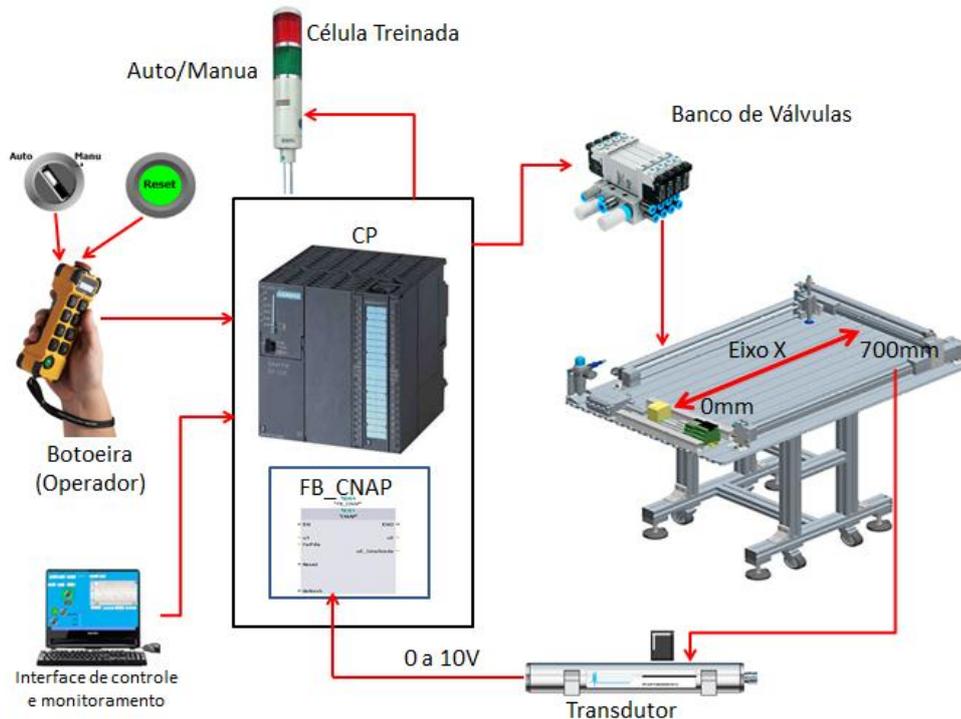


**Figura 19 - Correspondência deslocamento do cilindro e a tensão do transdutor.**

A implementação do Bloco Funcional FB\_CNAP que fora fundamentado no algoritmo de uma Célula Neural Artificial Paraconsistente de Aprendizagem tem como objetivo demonstrar a funcionalidade do bloco em sistemas de automação industrial, de modo que este sistema consiga aprender um determinado movimento com um operador e imitá-lo em ciclo contínuo sem nenhuma intervenção. Para os testes foi escolhido um sistema de manipulação de peças, que possui movimentos

nos eixos XYZ, no qual sua movimentação pode ocorrer de forma manual por meio de um controle remoto ou de forma automática no qual um programa deve ser inserido no Controlador Programável (CP) que por sua vez terá a função de ler as entradas, compará-las com o programa do usuário e fornecer sinais de saída, de modo a controlar uma sequência pré-definida de movimentos do manipulador.

A estrutura de montagem do sistema de movimentação de peças (Figura 20) é composta por um manipulador, um controlador programável (CP), um transdutor de deslocamento, um banco de válvulas, um sinalizador, uma botoeira e um computador. Esse sistema tem como elemento controle principal o CP associado ao bloco funcional FB\_CNAP, que por sua vez terá as funções de uma Célula Neural Artificial Paraconsistente de Aprendizagem. O bloco funcional FB\_CNAP possui todas as características solicitadas pela norma IEC 61131-3, principalmente no que se refere à função modularidade, no qual novos blocos podem facilmente ser reinseridos, da maneira que aplicação exigir.

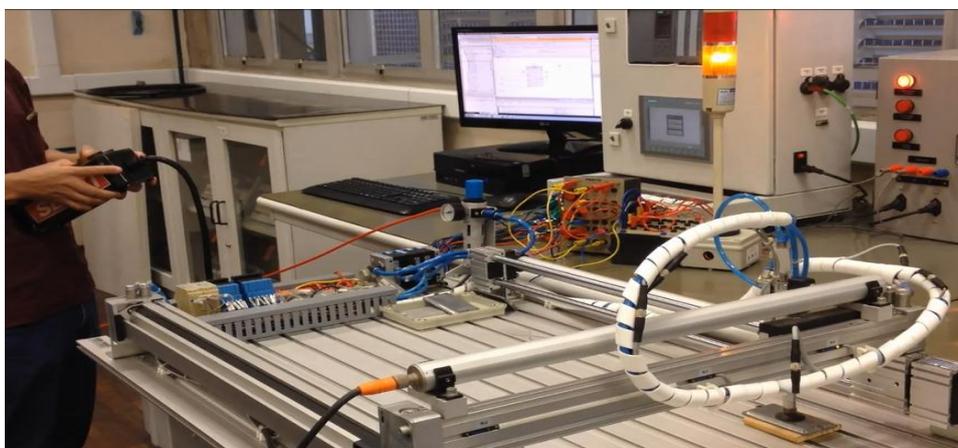


**Figura 20 - Sistema de movimentação de peças.**

A seguir é apresentado a sequência e o procedimento para o funcionamento do sistema de movimentação de peças.

- 1) Inicialmente deve ser inserido na tela de *interface*, o Fator de Aprendizagem requerido.
- 2) Acionar o botão *Reset*, que irá inserir o valor inicial 50 do grau de evidência.
- 3) Colocar a chave na posição Manual.
- 4) Acionar a botoeira de modo a deslocar o cilindro do eixo X até a posição desejada pelo operador.
- 5) Acionar a botoeira para ligar a ventosa e o cilindro do eixo Z a fim de pegar a peça.
- 6) Com a botoeira, levar a peça até a posição 0 mm e soltá-la no ponto de descarga.
- 7) Repetir os itens 4, 5 e 6 até o sinalizador indicar Célula Treinada, o que demonstra que o bloco FB\_CNAP aprendeu o movimento com o operador.
- 8) Com a célula treinada, deslocar o cilindro do eixo X até a posição 0 mm.
- 9) Passar a chave da botoeira para a posição Imitação
- 10) Soltar a botoeira e acompanhar o processo que será a partir daí realizado em ciclo contínuo.
- 11) Caso queira interromper o processo, basta passar a chave para manual ou pressionar o *Reset*.
- 12) Para reiniciar uma nova posição de aprendizagem, basta seguir o procedimento, dos itens 1 ao 10.

A figura 21 mostra uma foto da implementação realizada:



**Figura 21 – Sistema de movimentação de peças.**

O sistema possui um *software* supervisor denominado *Blue Plant* do fabricante Altus, que se comunica com o CP por meio do protocolo *Profinet*. Sua função é realizar a *interface* entre o sistema de movimentação de peças e o operador, de modo a facilitar a sua interação. A tela foi construída com o intuito de proporcionar para o operador uma alternativa ao controle convencional pela botoeira, bem como permitir que ajustes sejam realizados de maneira dinâmica, como na possibilidade de inserir o fator de aprendizagem e acompanhar o padrão de entrada em conjunto com grau de evidência por meio de gráficos ou dados numéricos. Ao fim dos passos de aprendizado, o supervisor indica a ocorrência do aprendizado do bloco FB\_CNAP. A tela do supervisor construída é apresentada na figura 22.

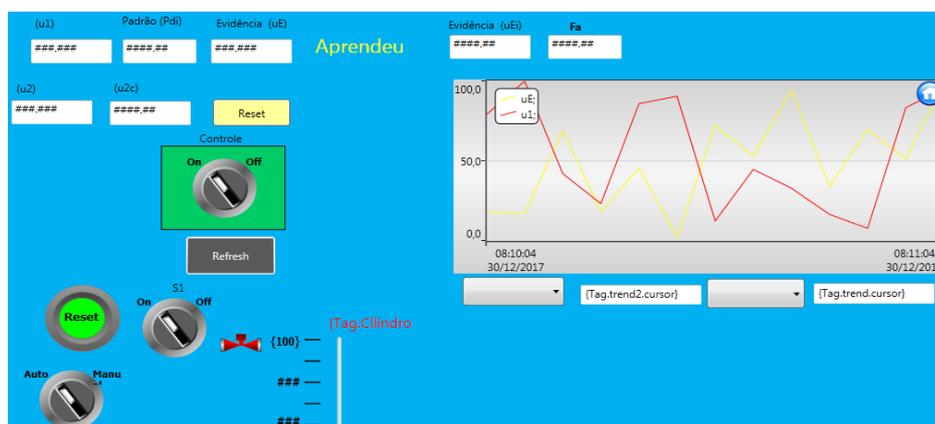


Figura 22 – Tela do Sistema de *interface* Supervisor.

### 2.5.1. Funcionamento do Bloco FB\_CNAP no controle do sistema

O Bloco FB\_CNAP após ser construído tornou-se um bloco modular, ou seja, para utilizá-lo, bastou buscá-lo na biblioteca e o inserir na *Network 1* do programa. A *Network* corresponde a uma linha de programação na linguagem *Ladder*, linguagem esta, fundamentada na norma IEC 61131-3. Nas suas entradas u1, Fa/Fda e nas saídas uE e uE\_sinalizada, foram atribuídos endereços de memórias com a função de armazenar os dados gerados durante o controle do sistema. Nas entradas e saídas *Reset*, *Refresh* e ENO foram inseridos endereços digitais, de maneira a receber e enviar sinais para o controle do bloco e do manipulador. Os endereçamentos do bloco podem ser visualizados na figura 23.

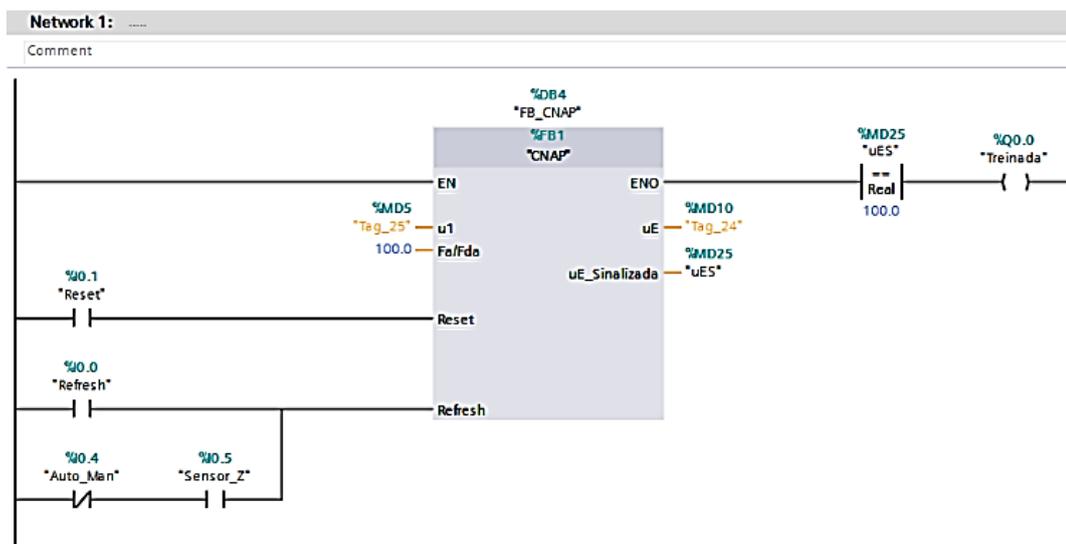


Figura 23 – Endereçamentos do bloco FB\_CNAP.

O funcionamento do Bloco FB\_CNAP no controle do sistema será apresentado a seguir:

- 1) O bloco ao receber um pulso na entrada Reset gera na sua saída  $\mu_E$  o valor 50 que corresponde ao valor 0,5 no algoritmo original e, portanto está pronto para o início da aprendizagem ou desaprendizagem.
- 2) Na entrada FA/FDA é colocado valores no intervalo de 0 a 100, que irá interferir na rapidez de aprendizagem.
- 3) Ao colocar o sistema em operação manual, o bloco entrará no modo aprendido.
- 4) O operador ao movimentar o cilindro do eixo X, o sinal proveniente do transdutor é enviado a uma memória do CP que por sua vez insere os valores convencionados de 0 a 100 no padrão de entrada  $\mu_1$ .
- 5) Ao retirar uma peça da mesa do manipulador, um passo é dado e o bloco FB\_CNAP, recebe uma atualização do padrão de entrada  $\mu_1$ , através do Refresh.
- 6) Ao atualizar  $\mu_1$ , o grau de evidência  $\mu_E$ , é recalculado de acordo com o algoritmo e o seu valor é armazenado em uma memória do CP.
- 7) Ao repetir um passo,  $\mu_E$  é novamente calculado de forma a diminuir a contradição e se aproximar cada vez mais do padrão definido pelo posição do cilindro do eixo X.

- 8) Após repetições de passos, o grau de evidência  $\mu_E$  se iguala ao padrão  $\mu_1$  e o bloco FB\_CNAP sinaliza célula treinada e o aprendizado foi concluído através da saída uE\_sinalizada.
- 9) O cilindro do eixo X ao ser colocado na posição 0mm com chave na posição imitação, o bloco FB\_CNAP definiu os parâmetros para que o sistema movimentasse da mesma maneira como foi demonstrado.

A seguir serão apresentados os resultados gerados nos testes realizados com o bloco FB\_CNAP no controle do sistema inteligente de controle de peças.

### 3. RESULTADOS E DISCUSSÃO

Nesta parte do trabalho será apresentado e discutido os resultados dos testes do bloco FB\_CNAP atuando no controle do sistema de movimentação de peças. O objetivo é efetuar treinamentos no bloco com valores distintos de padrões ( $\mu_1$ ), bem como fatores de aprendizagem e desaprendizagem ( $F_A$  e  $F_{DA}$ ) de forma a observar o seu comportamento na atuação do grau de evidência ( $\mu_E$ ) e por consequência no controle do sistema. Foram observados também os valores de grau de certeza ( $G_c$ ) e grau de contradição ( $G_{ct}$ ) a fim de verificar a correlação destes com resultados de saída do bloco.

Os testes apresentados a seguir demonstram o treinamento do bloco FB\_CNAP e para isso os testes obedeceram a seguinte sequência:

- 1) Colocado a chave na posição manual;
- 2) Acionado o botão reset do bloco;
- 3) Movimentado eixo X até a posição desejada, retirado uma peça e levada até a posição de descarga.
- 4) Neste instante foi coletado os dados de  $\mu_1$ ,  $F_A$ ,  $F_{DA}$ ,  $G_c$ ,  $G_{ct}$  e  $\mu_E$ .
- 5) Repetiu-se passos com os itens 3 e 4 até a célula sinalizar que houve a aprendizagem.

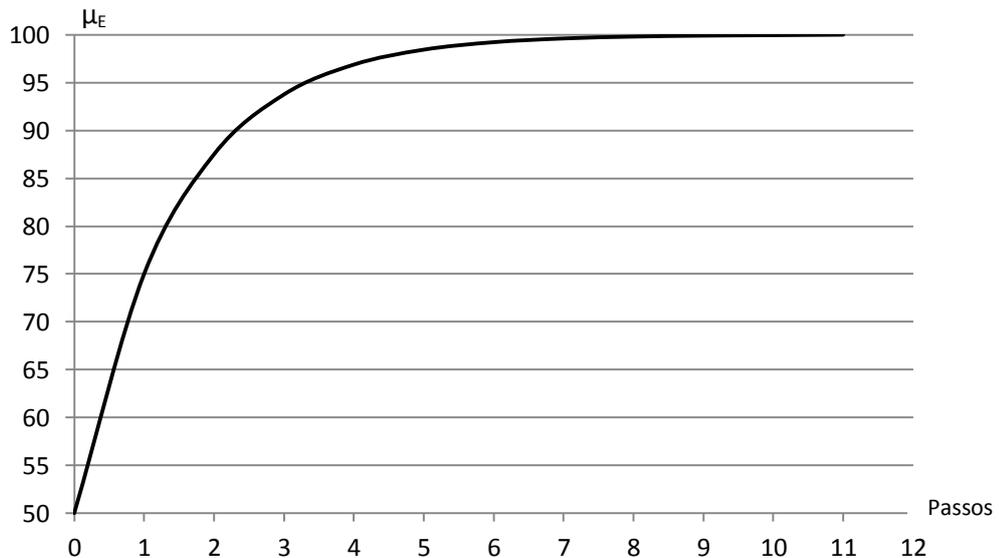
A tabela 2 apresenta os dados coletados com  $\mu_1 = 100$ , no qual equivale a posição de 700 mm do cilindro no eixo X.

Notou-se que após 10 passos realizados, ou seja, após 10 peças coletadas,  $\mu_E$  se igualou a  $\mu_1$  e o bloco sinalizou célula treinada.

**Tabela 2 – Aplicação do FB\_CNAP com  $\mu_1 = 100$  e  $F_A = 100$ .**

Passos	$F_A$	$\mu_1$	$G_C$	$G_{ct}$	$\mu_E$
0			0	-100	50
1	100	100	50	50	75
2	100	100	75	25	87,5
3	100	100	87,5	12,5	93,75
4	100	100	93,75	6,25	96,875
5	100	100	96,875	3,125	98,4375
6	100	100	98,4375	1,5625	99,21875
7	100	100	99,21875	0,78125	99,60938
8	100	100	99,60938	0,390625	99,80469
9	100	100	99,80469	0,1953125	99,90234
10	100	100	99,90234	0,09765625	99,95117
11	100	100	100	0	100

Os dados visualizados na tabela 2 podem também ser compreendidos por meio do gráfico da figura 24.



**Figura 24 - Valores de aprendizagem do bloco FB\_CNAP com  $u_1 = 100$  e  $F_A = 100$ .**

Os resultados apresentados na tabela 2 e no gráfico da figura 24 foram comparados com os valores correspondentes aos do algoritmo original (DA SILVA FILHO et al.,2008) e pode ser constatado que ocorre equivalência entre os dados obtidos, o que demonstra a fidelidade do bloco FB\_CNAP com a Célula Neural Artificial Paraconsistente CNAP original.

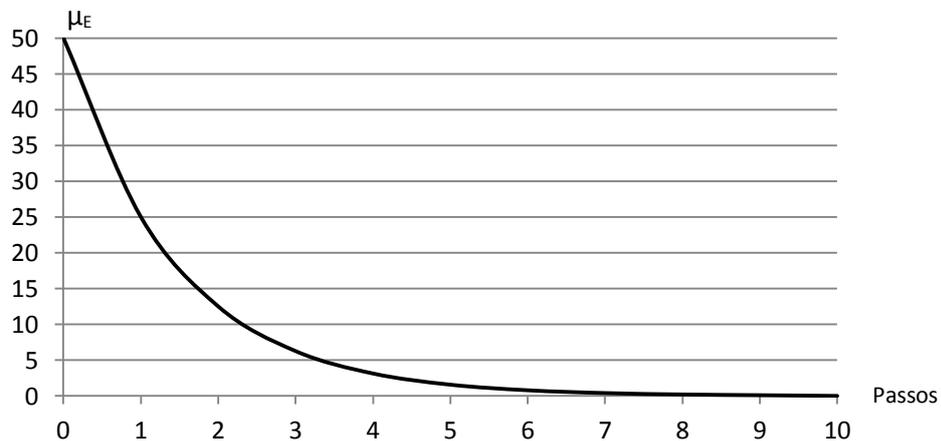
Com o objetivo de demonstrar a funcionalidade do FB\_CNAP em realizar a desaprendizagem, no próximo teste buscou-se utilizar o padrão  $u_1 = 0$ , porém

devido a característica do manipulador pneumático implementado, chegou-se a um valor mínimo de  $u_1 = 0,0055$ , o que não prejudicou os testes. A tabela 3 e o gráfico da figura 25 demonstram os resultados obtidos.

**Tabela 3 – Aplicação do FB\_CNAP com  $\mu_1 = 0,0055$  e  $F_{DA} = 100$ .**

Passos	$F_{DA}$	$\mu_1$	$G_c$	$G_{ct}$	$\mu_E$
0			50	50	50
1	100	0,0055	-49,9945	-49,9945	25,00275
2	100	0,0055	-74,99175	-24,99725	12,50412
3	100	0,0055	-87,49038	-12,49862	6,25841
4	100	0,0055	-93,73969	-6,249306	3,130154
5	100	0,0055	-96,86434	-3,124657	1,567829
6	100	0,0055	-98,42667	-1,562325	0,786631
7	100	0,0055	-99,20783	-0,7811661	0,3960838
8	100	0,0055	-99,59841	-0,3905869	0,2007942
9	100	0,0055	-99,79371	-0,1952896	0,1003
10	100	0,0055	-100	0	0

Assim como no teste anterior, neste último, os resultados também se mostraram equivalentes ao observado no algoritmo original, encontrado em (DA SILVA FILHO et al.,2008).



**Figura 25 - Valores de aprendizagem do bloco FB\_CNAP com  $\mu_1 = 0,0055$  e  $F_{DA} = 100$ .**

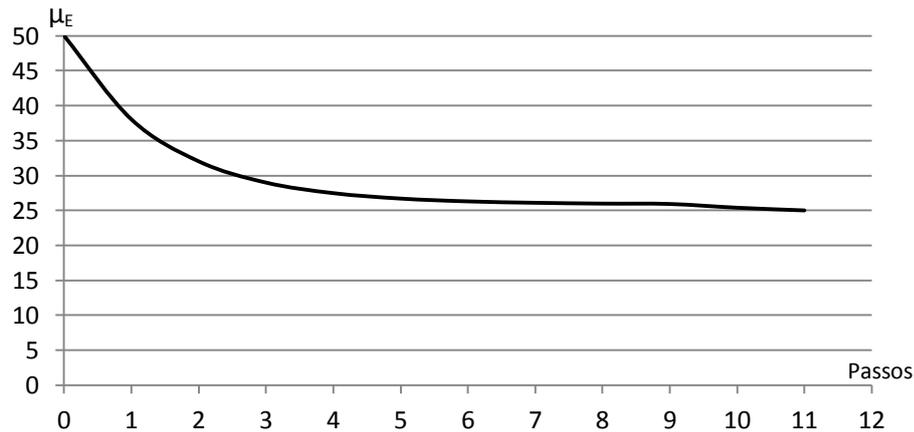
A fim de demonstrar a capacidade do FB\_CNAP em reconhecer padrões significativos entre 0 e 100, foi realizado testes utilizando o padrão  $\mu_1 = 25$ , porém o valor mais próximo conseguido no ajuste do sistema foi de  $\mu_1 = 25,275$ .

Observa-se na Tabela 4 e no gráfico da figura 26, que foram necessários 11 passos até ocorrer a aprendizagem do bloco FB\_CNAP e dessa forma o mesmo

demonstra que é capaz de aprender ou desaprender valores intermediários a 0 e 100.

**Tabela 4 – Aplicação do FB\_CNAP com  $\mu_1 = 25,275$  e  $F_{DA} = 100$ .**

Passos	$F_{DA}$	$\mu_1$	$G_c$	$G_{ct}$	$\mu_E$
0			50	50	50
1	100	25,275	-24,65712	-24,66084	38,03094
2	100	25,275	-36,65598	-12,71786	32,0222
3	100	25,275	-42,67962	-6,727753	28,9843
4	100	25,275	-45,75105	-3,723373	27,47094
5	100	25,275	-47,29422	-2,22400	26,69376
6	100	25,275	-48,03041	-1,417938	26,29214
7	100	25,275	-48,4283	-1,012589	26,09319
8	100	25,275	-48,63098	-0,8136444	25,97509
9	100	25,275	-48,74536	-0,6992645	25,91976
10	100	25,275	-48,80441	-0,6439362	25,88465
11	100	25,275	-50	0	25



**Figura 26 - Valores de aprendizagem do bloco FB\_CNAP com  $\mu_1 = 25,275$  e  $F_A = 100$**

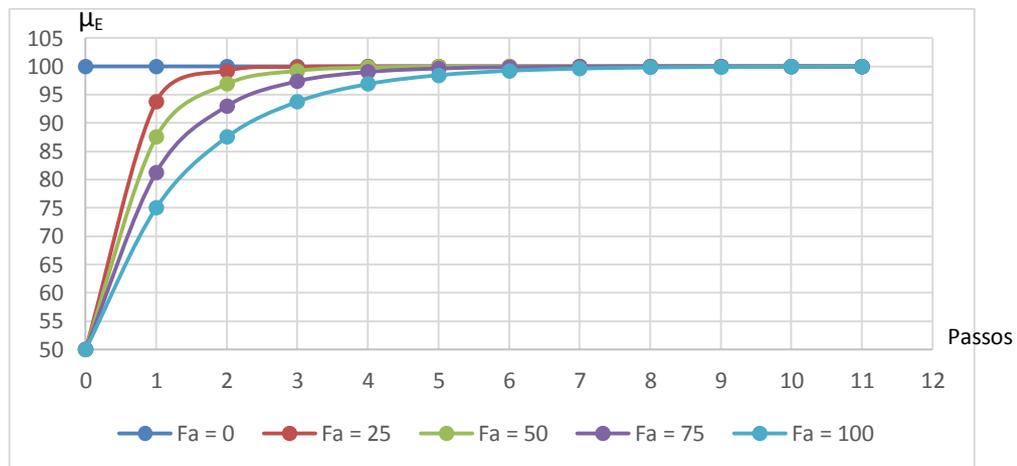
A tabela 5 e o gráfico da figura 27 demonstram o comportamento do FB\_CNAP diante de diferentes fatores de aprendizagem  $F_A$ , ou seja, será observada a rapidez e a precisão de aprendizado do bloco. Para isso foram utilizados os seguintes valores de Fatores de aprendizagem: 0, 25, 50, 75 e 100 com um padrão  $\mu_1 = 100$ .

Assim como o algoritmo original (DA SILVA FILHO et al., 2008), o bloco FB\_CNAP demonstrou equivalência com os valores obtidos. Observa-se no gráfico da figura 27, que para um  $F_A = 0$  o grau de evidência  $\mu_E$  atinge 100 após um passo apenas, caracterizando um sinal na forma de degrau. Nota-se que quanto maior o valor de  $F_A$ , limitado a 100, mais passos serão necessários para o aprendizado ou

desaprendizado do FB\_CNAP. Dessa forma, é possível observar que ao utilizar o valor máximo de  $F_A$ , obtêm-se uma quantidade maior de valores intermediário de  $\mu_E$ , caracterizando em uma convergência de maior precisão.

**Tabela 5 - Comportamento do FB\_CNAP com diferentes Fatores de aprendizagem  $F_A$ .**

Passos	$\mu_1$	$\mu_E$				
		$F_A = 0$	$F_A = 25$	$F_A = 50$	$F_A = 75$	$F_A = 100$
0		50	50	50	50	50
1	100	100	93,75	87,5	81,25	75
2	100	100	99,21875	96,875	92,96875	87,5
3	100	100	99,90234	99,21875	97,36328	93,75
4	100	100	100	99,80469	99,01123	96,875
5	100	100	100	99,95117	99,62921	98,4375
6	100	100	100	100	99,86095	99,21875
7	100	100	100	100	99,94786	99,60938
8	100	100	100	100	100	99,80469
9	100	100	100	100	100	99,90234
10	100	100	100	100	100	99,95117
11	100	100	100	100	100	100



**Figura 27 - Resultados de diferentes Fatores de Aprendizagem  $F_A$ .**

Durante o deslocamento do cilindro do eixo X, o operador do sistema de movimentação de peças, poderá eventualmente errar os pontos de parada do cilindro por motivos diversos, como desatenção ou mau posicionamento da peça. Diante deste contexto foi realizado testes em que foi inserido desvios nos passos 3 e 4, movimentando o cilindro para 90,455 ao invés de 100, que era o valor padrão  $\mu_1$  pré-definido. Observa-se na tabela 6 e no gráfico da figura 28, que o FB\_CNAP conseguiu se ajustar quando  $\mu_1$  foi corrigido para 100.

Tabela 6 – Aprendizado do FB\_CNAP com desvio de 90,455.

Passos	$F_A$	$\mu_1$	$G_c$	$G_{ct}$	$\mu_E$
0			50	50	50
1	100	100	50	50	75
2	100	100	75	25	87,5
3	100	90,455	77,9669	2,959236	89,26275
4	100	90,455	79,72198	1,192764	90,144
5	100	100	90,14412	9,855881	95,07206
6	100	100	95,07206	4,92794	97,53603
7	100	100	97,53603	2,463974	98,76801
8	100	100	98,76801	1,231987	99,384
9	100	100	99,384	0,61	99,692
10	100	100	99,692	0,3079987	99,846
11	100	100	99,846	0,1539993	99,923
12	100	100	100	0	100

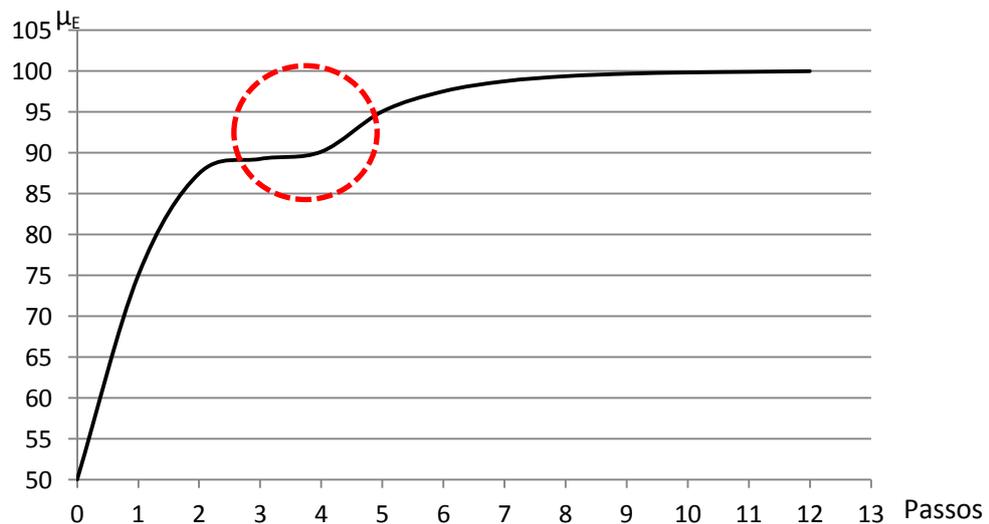


Figura 28 - Aprendizado do FB\_CNAP com desvio de 90,455.

A fim de comprovar a confiabilidade de aprendizado do bloco diante de desvios diversos, foi atribuído um novo valor de  $\mu_1$ . Neste teste foi desviado o movimento do cilindro para 70,1297. A tabela 7 e o gráfico da figura 29 demonstram um comportamento de correção semelhante ao teste anterior, com a diferença que neste caso foi necessário realizar 14 passos até a sinalização do aprendizado, isto demonstra que quanto maior for desvio do padrão  $\mu_1$ , mais passos serão necessários para o FB\_CNAP concluir o aprendizado. Portanto o bloco FB\_CNAP demonstra com esses testes a sua confiabilidade na ocorrência de desvios de padrão  $\mu_1$ .

Tabela 7 – Aprendizado do FB\_CNAP com desvio de 70,1297.

Passos	$F_A$	$\mu_1$	$G_c$	$G_{ct}$	$\mu_E$
0			50	50	50
1	100	100	50	50	75
2	100	100	75	25	87,5
3	100	70,1297	57,6148	-17,3852	79,16132
4	100	70,1297	49,27239	-9,05024	74,98639
5	100	100	74,98639	25,01361	87,49319
6	100	100	87,49319	12,50681	93,7466
7	100	100	93,7466	6,253403	96,8733
8	100	100	96,8733	3,126701	98,43665
9	100	100	98,43665	1,563354	99,21832
10	100	100	99,21832	0,7816772	99,60916
11	100	100	99,60916	0,3908386	99,80458
12	100	100	99,80458	0,1954193	99,90229
13	100	100	99,90229	0,09770966	99,95114
14	100	100	100	0	100

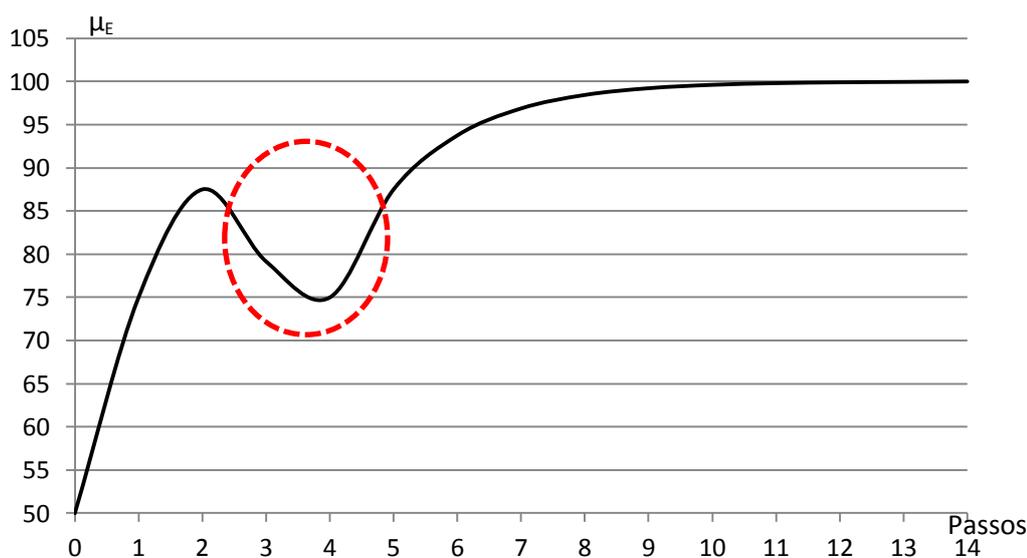


Figura 29 - Gráfico de aprendizado do FB\_CNAP com desvio de 70,1297.

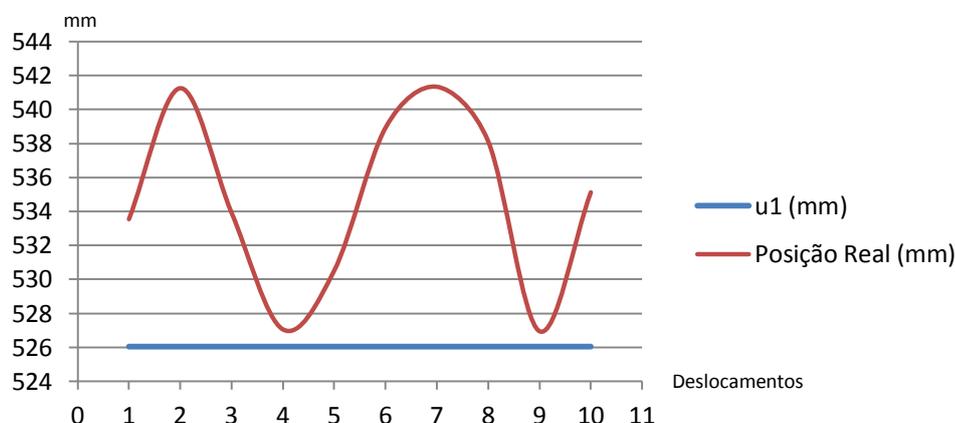
Vale ressaltar que durante os testes também foram observados os valores dos graus de certeza ( $G_c$ ) e graus de contradição ( $G_{ct}$ ) e após a análise das tabelas nota-se que no passo onde  $G_{ct}$  é igual a 0, o grau de evidência  $\mu_E$  se iguala a  $\mu_1$ , ou seja, neste momento não existe mais contradição e portanto ocorre o aprendizado da célula. Desta forma, essa informação pode ser utilizada como mais uma referência em determinadas aplicações. Outro item a ser destacado, é referente a utilização do Fator de Aprendizagem igual a 100, propiciando com este valor maior precisão na correção do desvio inserido.

Finalizados os testes de treinamento do Bloco FB\_CNAP, verificou-se a necessidade de analisar se o sistema estava executando no modo aimita o movimento antes realizado em manual. Para isso foram feitos quatro testes em posição de aprendizado diferentes, levantando os valores do posicionamento da parada do cilindro do eixo X no instante de retirada da peça da mesa. Para cada teste foram observados 10 deslocamentos, e para uma melhor compreensão dos resultados, foi inserido nas tabelas os valores de posição real correspondentes em milímetros (mm) de 0 a 700mm que é o curso do cilindro. A tabela 8 mostra o teste com padrão  $\mu_1$  de 75,15 que é equivalente a 526,05mm.

**Tabela 8 – Teste de deslocamento do sistema na posição 75,15 (526,05 mm).**

Deslocamento	$F_A$	$\mu_1$	Posição	$\mu_1$ (mm)	Posição Real (mm)
1	100	75,15	76,2212	526,05	533,5484
2	100	75,15	77,3213	526,05	541,2491
3	100	75,15	76,2745	526,05	533,9215
4	100	75,15	75,2947	526,05	527,0629
5	100	75,15	75,7882	526,05	530,5174
6	100	75,15	76,9892	526,05	538,9244
7	100	75,15	77,33334	526,05	541,33338
8	100	75,15	76,87572	526,05	538,13004
9	100	75,15	75,27924	526,05	526,95468
10	100	75,15	76,44592	526,05	535,12144

No gráfico da figura 30, é possível observar com mais detalhes a posição real de parada do cilindro em relação ao padrão de  $\mu_1$  com 526,05 mm, no qual notou-se um desvio máximo de 15,28338 mm no passo 7.



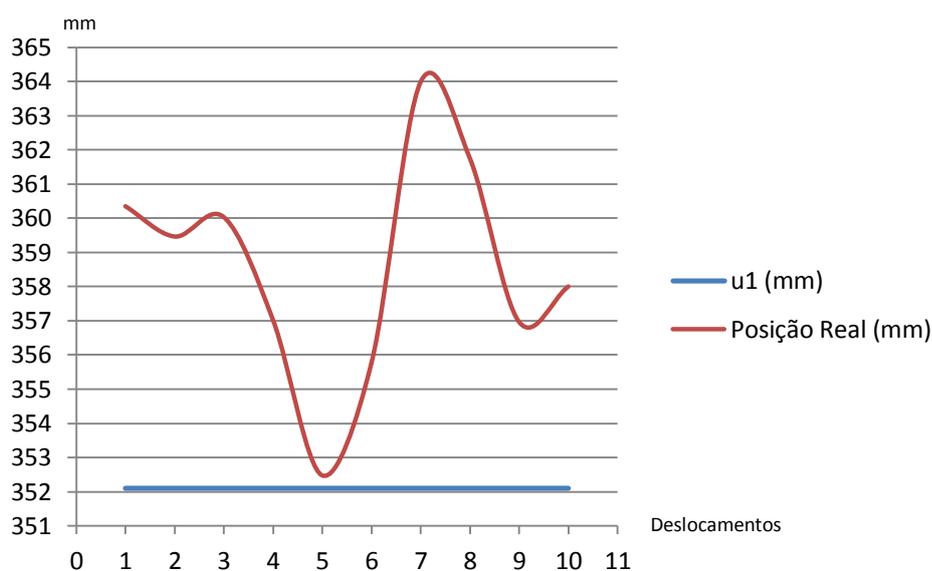
**Figura 30 - Deslocamento do cilindro para a posição 75,15 (526,05 mm).**

Seguindo o teste anterior, serão apresentados a seguir mais três testes com padrão de aprendizado distintos, a fim de verificar a ocorrência de oscilações da posição real do cilindro do eixo X. A tabela 9 demonstra os resultados com padrão de 50,3 correspondentes a 352,1 mm.

**Tabela 9 – Testes de deslocamento do sistema na posição 50,13 (352,1 mm).**

Deslocamento	$F_A$	$\mu_1$	Posição	$\mu_1$ (mm)	Posição Real (mm)
1	100	50,3	51,47892	352,1	360,35244
2	100	50,3	51,35232	352,1	359,46624
3	100	50,3	51,43333	352,1	360,03331
4	100	50,3	51,00332	352,1	357,02324
5	100	50,3	50,35372	352,1	352,47604
6	100	50,3	50,82424	352,1	355,76968
7	100	50,3	51,99876	352,1	363,99132
8	100	50,3	51,67892	352,1	361,75244
9	100	50,3	50,99533	352,1	356,96731
10	100	50,3	51,14332	352,1	358,00324

No gráfico da figura 31, é possível verificar que a posição do cilindro chegou ao valor máximo de 361,75244, provocando um desvio de 9,65244 mm em relação ao padrão  $\mu_1$  (mm) de 352,1mm.



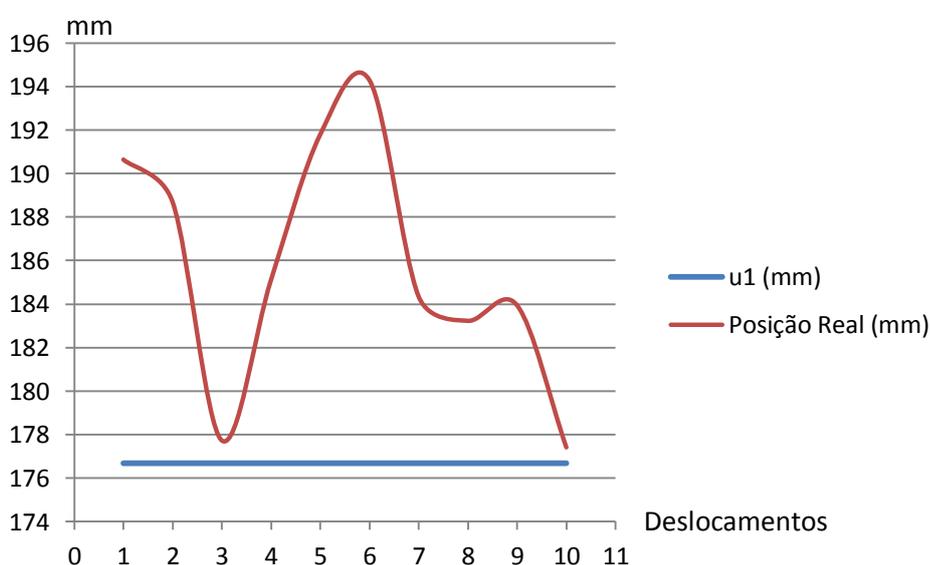
**Figura 31 - Deslocamento do cilindro para a posição 50,13 (352,1mm).**

Na tabela 10 é apresentado um teste utilizando o padrão  $\mu_1$  de 25,24 correspondente a 176,68 mm.

**Tabela 10 – Teste de deslocamento do sistema na posição 25,24 (176,68 mm).**

Deslocamento	F <sub>A</sub>	$\mu_1$	Posição	$\mu_1$ (mm)	Posição Real (mm)
1	100	25,24	27,23425	176,68	190,63975
2	100	25,24	26,95524	176,68	188,68668
3	100	25,24	25,38931	176,68	177,72517
4	100	25,24	26,44443	176,68	185,11101
5	100	25,24	27,39927	176,68	191,79489
6	100	25,24	27,75582	176,68	194,29074
7	100	25,24	26,33395	176,68	184,33765
8	100	25,24	26,17552	176,68	183,22864
9	100	25,24	26,27722	176,68	183,94054
10	100	25,24	25,34452	176,68	177,41164

Verifica-se no gráfico da figura 32 que o cilindro atingiu o valor máximo 194,29074 mm o que corresponde a 17,61074 mm de desvio em relação ao padrão  $\mu_1$  (mm) de 176,68mm.



**Figura 32 - Deslocamento do cilindro para a posição 25,24 (176,68 mm).**

Por fim, a tabela 11 apresenta um teste com o cilindro no curso seu curso máximo, o equivale a um padrão  $\mu_1$  igual a 100, o que corresponde a 700 mm de deslocamento.

Tabela 11 – Testes de deslocamento do sistema na posição 100 (700 mm).

Deslocamento	F <sub>A</sub>	$\mu_1$	Posição	$\mu_1$ (mm)	Posição Real (mm)
1	100	100	100	700	700
2	100	100	100	700	700
3	100	100	100	700	700
4	100	100	100	700	700
5	100	100	100	700	700
6	100	100	100	700	700
7	100	100	100	700	700
8	100	100	100	700	700
9	100	100	100	700	700
10	100	100	100	700	700

É possível observar no gráfico da figura 33, que ao utilizar o padrão máximo  $\mu_1$  de 100 (700 mm) o cilindro do eixo X movimentou exatamente para a posição desejada, ou seja não houve desvio.

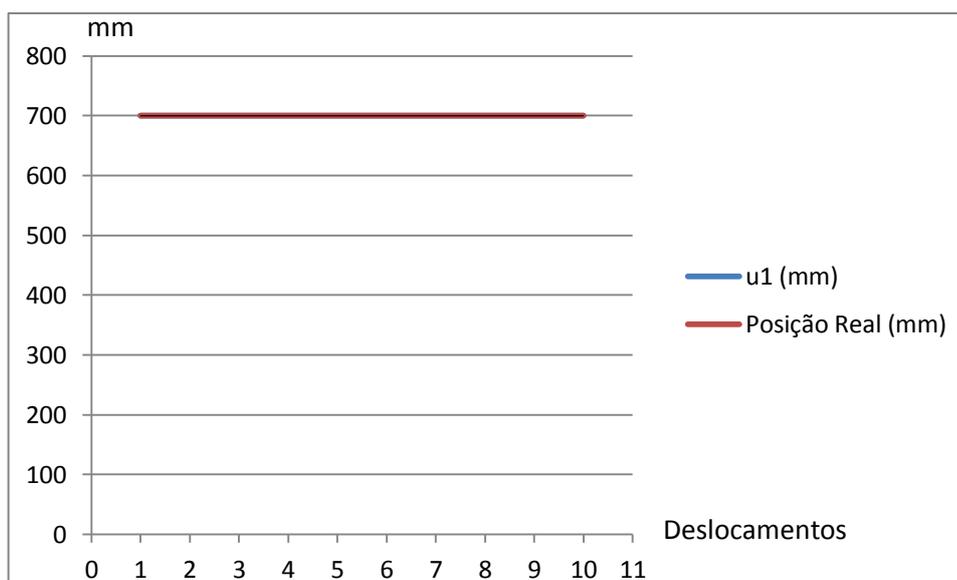


Figura 33 - Deslocamento do cilindro para a posição 100 (700 mm).

Analisando os resultados, pode-se verificar que o desvio em relação dos padrões  $u_1$  (mm), oscilou de 0 quando o padrão foi 700mm a até 17,61074 mm quando o padrão foi de 176,68 mm. Diante da discrepância dos resultados de parada do cilindro, foi dado início a uma investigação a fim de encontrar o motivo desta oscilação.

Após diversas verificações, constatou-se dois motivos para a sua causa: a primeira é que devido o sistema trabalhar com um fluido pneumático e este ser

compressível, fez com que o cilindro não conseguisse ter a precisão necessária para o controle eficiente de parada. O outro motivo diz respeito à válvula e a sua atuação no instante da interrupção do envio do ar comprimido para o cilindro, fazendo com que isso não aconteça de forma imediata devido as características do sistema eletropneumático. Foi observado que no teste com padrão  $\mu_1$  (mm) em 700 mm, houve certa precisão, e isso aconteceu devido este ser o valor limite de deslocamento do cilindro, ou seja, devido existir um batente físico, não houve a possibilidade de ocorrer desvios.

Desta forma acredita-se que se for utilizado outra forma de energia para o deslocamento do sistema, como a hidráulica, poderá proporcionar maior precisão, garantindo assim mais eficiência no controle da parada automática em aplicações que envolverem o bloco FB\_CNAP.

#### 4. CONCLUSÕES

Esta pesquisa apresentou a construção de uma Célula Neural Artificial Paraconsistente de Aprendizagem CNAPap na forma de um bloco funcional de acordo com a norma IEC 61131-3. O bloco funcional, denominado como FB\_CNAP foi construído para um Controlador Programável constituindo-se em um Sistema Paraconsistente de Aprendizagem Por Demonstração - SPAPD-LPA2v. Este sistema foi testado em um manipulador pneumático com o objetivo de se observar o comportamento e a real aplicabilidade do bloco FB\_CNAP.

Este trabalho objetiva opções eficientes e confiáveis para o controle de processos industriais, pois utiliza uma nova abordagem para movimentação de máquinas, contribuindo também com uma maior facilidade em novas implementações, garantindo que mesmo em diferentes linguagens de programação e fabricantes de CP diversos, o bloco funcional FB\_CNAP apresenta-se como opção modular de aplicação em projetos industriais.

Os ensaios com o bloco FB\_CNAP geraram resultados equivalentes a uma Célula Neural Artificial Paraconsistente de Aprendizagem, o que demonstra a fidelidade com o algoritmo original, garantindo assim confiabilidade na utilização do bloco funcional em diversas aplicações relacionadas à automação industrial.

A implementação realizada em um sistema eletropneumático de movimentação de peças, foi de grande importância na validação do funcionamento do bloco FB\_CNAP. Nos testes iniciais foi adotado os padrões 0 e 1, que neste trabalho foi convencionado para 0 e 100 a fim de proporcionar uma melhor análise do comportamento do sistema Paraconsistente de Aprendizagem Por Demonstração - SPAPD-LPA2v. Apesar de não ter conseguido chegar ao valor zero para o teste de desaprendizagem, o comportamento do bloco gerou valores corretos na sua saída, permitindo o controle do automatismo do deslocamento das peças. Os testes com valores intermediários entre 0 e 100 também se mostraram satisfatórios para o treinamento do bloco FB\_CNAP, porém durante o controle de imitação ocorreram desvios durante a parada do cilindro para retirada das peças da mesa. Foi verificado que a ocorrência destes desvios não foi ocasionada pelo bloco, mas sim pela característica da falta de precisão do sistema eletropneumático e dos dispositivos que o compõem.

Foram inseridos desvios de padrão durante um processo de treinamento do

bloco, e este se demonstrou apto a realizar as correções desde que retorne ao padrão definido anteriormente. O que garante a confiabilidade na aplicação do bloco em um processo operacional, pois naturalmente poderão ocorrer desvios de forma involuntária por parte do operador.

Enfim, verifica-se com este trabalho que o Sistema Paraconsistente de Aprendizagem Por Demonstração - SPAPD-LPA2v na forma do bloco funcional FB\_CNAP pode realmente ser implementado em aplicações industriais que envolvam as características da LPA2v e Inteligência Artificial. A modularidade do bloco garante que este seja aplicado na formação de Redes Neurais Artificiais Paraconsistentes de forma simplificada, pois o bloco estará pronto e disponível em uma biblioteca virtual. Portanto, este trabalho contribui de forma significativa em aplicações da Inteligência Artificial em áreas de sistemas de automação industrial e robótica, proporcionando uma maior abertura para que pesquisas futuras sejam exploradas.

O trabalho apresentado demonstra que existe um vasto campo para ser explorado e que trabalhos futuros podem ser realizados de modo a aproveitar as implementações realizadas com o bloco funcional FB\_CNAP. Serão descritos a seguir algumas sugestões de trabalhos futuros em aplicações com Controladores Programáveis (CP):

- Realizar estudos com a inserção de aprendizagem de mais movimentos em um manipulador.
- Aplicar o sistema neste estudo desenvolvido em sistema robótico.
- Com base no conceito de modularidade da norma IEC 61131-3, um trabalho futuro pode ser realizado a fim de construir uma família de Células Neurais Artificiais Paraconsistentes em CP.
- A partir da família de Células Neurais, será possível construir Redes Neurais Artificiais Paraconsistentes e aplicá-las em novos sistemas automáticos aplicados com CP.
- Com base no bloco funcional FB\_CNAP, poderá este ser implementado em novas aplicações de automação industrial.
- Novos testes podem ser feitos utilizando o FB\_CNAP, aplicando-o em um sistema de atuadores com maior precisão, como o hidráulico ou servomotores.

Enfim, nesta parte do trabalho foram feitas apenas algumas sugestões, o que não impede que aplicações diferentes sejam implementadas, o que propiciará novos estudos no que tange as aplicações em Sistema Paraconsistente de Aprendizagem Por Demonstração - SPAPD-LPA2v.

## 5. VERSÃO COMPACTA

### Desenvolvimento de Célula Neural Artificial Paraconsistente em Controlador Programável para Aplicação em Técnicas De Aprendizagem por Demonstração Aplicadas à Automação

Rodrigo Silvério da Silveira<sup>1</sup>

<sup>1</sup> Universidade Santa Cecília – Programa de Pós Graduação em Engenharia Mecânica - Rua Oswaldo Cruz, 288, Boqueirão Santos-SP CEP- 11045-000

Email: [rodrisilverio@gmail.com](mailto:rodrisilverio@gmail.com)

**Resumo:** Em Inteligência Artificial a “Aprendizagem Por Demonstração-*ApD* (*learning from demonstration - LfD*) é o processo no qual se baseia em ensinar novas tarefas às máquinas por meio de manipulação repetitiva, sem a necessidade de programação. Com o objetivo de proporcionar maior facilidade na atribuição de movimentos em máquinas industriais, foi construído uma Célula Neural Artificial Paraconsistente de aprendizagem CNAPap sendo implementada em um Controlador Programável (CP) constituindo-se em um Sistema Paraconsistente de Aprendizagem Por Demonstração - SPAPD-LPA2v. Para testes de validação o SPAPD-LPA2v foi aplicado em uma máquina com processo de movimentação de peças eletropneumático, composto por atuadores lineares, ventosa e sensores analógicos de posicionamento linear. Os resultados obtidos neste trabalho demonstram que o SPAPD-LPA2v se apresenta fundamentalmente como uma potencial alternativa para novas implementações de controles em processos industriais.

**Palavras – Chave:** Lógica Paraconsistente Anotada. Célula Neural Artificial Paraconsistente. Controlador Programável. Aprendizagem por demonstração. Automação.

#### Development of an Paraconsistent Artificial Neural Cell in Programmable Controller for application in Learning for Demonstration techniques applied to automation

**Abstract:** In Artificial Intelligence "Learning from Demonstration - *Apd* (*LdD*) is the process in which it relies on teaching new tasks to machines by means of repetitive manipulation, without the need for programming. In order to facilitate the assignment of movements in industrial machines, a Paraconsistent Artificial Neural Cell (*PAN<sub>Cell</sub>*) was implemented in a Programmable Controller (CP), constituting a Paraconsistent Demonstration Learning System (*SPAPD-PAL2v*). For validation tests the *SPAPD-PAL2v* was applied in a machine with electropneumatic parts movement process, composed of linear actuators, suction cup and linear positioning analog sensors. The results obtained in this work demonstrate that the *SPAPD-PAL2v* presents fundamentally as a potential alternative for new implementations of controls in industrial processes.

**Keywords:** Paraconsistent Annotated Logic. Paraconsistent Artificial Neural Cell. Programmable Controller. Demonstration learning. Automation.

#### Introdução

A competitividade industrial cresce a cada dia e com isso as exigências relacionadas a produção se elevam na mesma proporção. Dessa forma a indústria a fim de garantir maior

eficiência, maior qualidade com custos reduzidos, busca tornar os seus processos cada vez mais automatizados [1].

Dentro deste contexto está a robótica e o controle de máquinas, no qual estão cada vez mais sofisticadas proporcionando assim crescentes ganhos produtivos as empresas. No entanto, o alto custo de programação dos equipamentos e o tempo gasto com novas implementações, consomem recursos importantes das empresas e de certa forma causam restrições quando se pensa em modificações no processo fabril [2]. Desta forma pesquisas avançam a fim de proporcionar que a programação dos robôs possa ser feita de maneira mais rápida, flexível e intuitiva, utilizando para isso conceitos como o de aprendizagem por demonstração. A técnica conhecida como “Aprendizagem por Demonstração- ApD (Learning from Demonstration - LfD) é uma das mais difundidas atualmente e propõe que um robô aprenda com as demonstrações feitas por um “professor” [3]. Com base nesta técnica, novos estudos surgem a fim de contribuir para a melhora dos resultados, como no caso deste trabalho, onde se utiliza a Lógica Paraconsistente Anotada como base para criar um processo de aprendizagem por demonstrações repetitivas [4]. Para isto utiliza-se a Célula Neural Artificial Paraconsistente de aprendizagem CNAPap como elemento base para a construção deste sistema denominado Sistema Paraconsistente de Aprendizagem Por Demonstração - SPApD-LPA2v. O sistema SPApD-LPA2v pode ser implementado em diversas plataformas, porém neste trabalho, optou-se pelo uso do Controlador Programável (CP).

Diante do contexto apresentado, será demonstrado neste trabalho a construção de um Bloco Funcional fundamentado no Sistema Paraconsistente de Aprendizagem Por Demonstração - SPApD-LPA2v. O seu Bloco Funcional é denominado FB\_CNAP e tem a sua construção baseada na norma IEC 61131-3 e implementação em um Controlador Programável (CP). Para demonstrar a sua aplicação, será utilizado um sistema de movimentação de peças, no qual o bloco FB\_CNAP irá sofrer um processo de treinamento em modo manual através do conceito de Teleoperação [3] e depois de treinado deverá controlar o sistema sem intervenção de operador.

## **Objetivos**

Com o objetivo tornar mais flexíveis novas implementações no que diz respeito à programação de máquinas na área de automação industrial, este trabalho busca novas alternativas no controle da automação, de modo a agregar maior eficiência para a produção industrial. Desta forma este trabalho propõe a inclusão de Células Neurais Artificiais em sistemas de Aprendizagem por Demonstração (ApD) de maneira modular em Controladores Programáveis (CP).

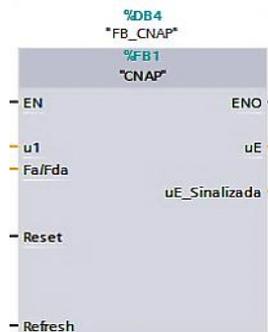
## Material e Métodos

Neste trabalho foi utilizado o Controlador Programável (CP) do fabricante SIEMENS modelo S7-1500, composto por uma Fonte de Alimentação Unidade Central de Processamento, um Módulo de Entrada Digital, um Módulo de Saída Digital, além dos Módulos de Entrada e Saída Analógica.

Para a elaboração do programa, utilizou-se o *software* TIA PORTAL da SIEMENS. Para a configuração foi selecionado um Bloco Funcional (FB - *Function Block*) sendo atribuído o nome FB\_CNAP. A programação interna do bloco fora realizada com base no algoritmo completo da Célula Neural Artificial Paraconsistente de Aprendizagem (CNAPap) no qual foi escolhida a linguagem *Ladder*, devido a sua ampla utilização pelos profissionais da área de automação industrial.

Com o objetivo de facilitar a compreensão dos resultados nos testes do bloco FB\_CNAP, os valores utilizados na escrita do algoritmo foram normalizados entre 0 a 100, sendo que o algoritmo original do CNAP possui valores que variam de 0 a 1.

O bloco finalizado pode ser verificado por completo na figura 1.



**Figura 1 – Bloco Funcional FB\_CNAP.**

Para os testes de aprendizagem fora utilizado um manipulador eletropneumático, composto por um cilindro de dupla ação sem haste que realiza movimentos no eixo X, um cilindro de dupla ação sem haste fixado no cilindro anterior, no qual tem a função de se movimentar no eixo Y, um cilindro de haste passante vazada com ventosa, cuja função é movimentar no eixo Z, dois transdutores lineares, um bloco de válvulas eletropneumáticas. Neste manipulador foi escolhido o eixo X para receber o processo de aprendizagem de movimentação.

## Resultados e Discussão

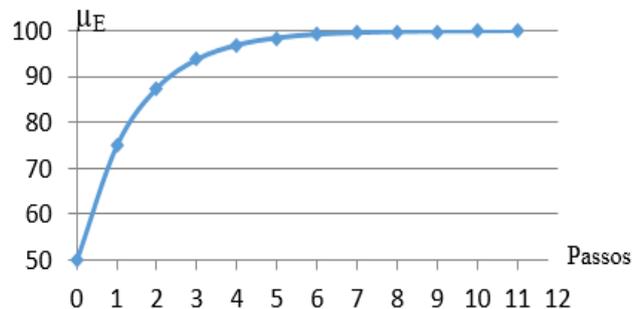
Foram realizados treinamentos no bloco FB\_CNAP com valor de padrões ( $\mu_1$ ), e fator de aprendizagem ( $F_A$ ) iguais a 100 de forma a observar o seu comportamento na atuação do grau de evidência ( $\mu_E$ ) e por consequência no controle do sistema. A tabela 1 apresenta os dados coletados com  $\mu_1 = 100$ , no qual equivale a posição de 700 mm do cilindro no eixo X.

Notou-se que após 10 passos realizados, ou seja, após 10 movimentos do eixo X,  $\mu_E$  se igualou a  $\mu_1$  e o bloco sinalizou célula treinada.

**Tabela 1 – Aplicação do FB\_CNAP com  $\mu_1 = 100$  e  $F_A = 100$ .**

Passos	$F_A$	$\mu_1$	$G_c$	$G_{ct}$	$\mu_E$
0			0	-100	50
1	100	100	50	50	75
2	100	100	75	25	87,5
3	100	100	87,5	12,5	93,75
4	100	100	93,75	6,25	96,875
5	100	100	96,875	3,125	98,4375
6	100	100	98,4375	1,5625	99,21875
7	100	100	99,21875	0,78125	99,60938
8	100	100	99,60938	0,390625	99,80469
9	100	100	99,80469	0,1953125	99,90234
10	100	100	99,90234	0,09765625	99,95117
11	100	100	100	0	100

Os dados visualizados na tabela 1 podem também ser compreendidos por meio do gráfico da figura 2.



**Figura 2 - Valores de aprendizagem do FB\_CNAP com  $\mu_1 = 100$  e  $F_A = 100$ .**

## Conclusões

Neste trabalho verificou-se que o Sistema Paraconsistente de Aprendizagem Por Demonstração - SPAPD-LPA2v na forma do bloco funcional FB\_CNAP pode realmente ser implementado em aplicações industriais que envolvam as características da LPA2v e Inteligência Artificial. A modularidade do bloco garante que este seja aplicado na formação

de Redes Neurais Artificiais Paraconsistentes de forma simplificada, pois o bloco estará pronto e disponível em uma biblioteca virtual. Portanto, este trabalho contribui de forma significativa em aplicações da Inteligência Artificial em áreas de sistemas de automação industrial e robótica, proporcionando uma maior abertura para que pesquisas futuras sejam exploradas, visto que estas ainda são poucas e restritas a área acadêmica.

### Referências bibliográficas

1. FERNANDES, C.L.M., MARIO C. M., DA SILVA FILHO, J. I. “**Study for inclusion of Paraconsistente Annotated logic in specific Standards for use in Programable Controllers**”, Science and Technology, ISSN: 2317-1316, Vol 1 2, p.p 49-53, 2012.
2. KARLSSON, M., Robertsson, A., & Johansson, R. **Autonomous Interpretation of Demonstrations for Modification of Dynamical Movement Primitives**. In IEEE International Conference on Robotics and Automation (ICRA), 2017 (pp. 316-321). Singapore: Institute of Electrical and Electronics Engineers Inc. DOI: 10.1109/ICRA.2017.7989040, 2017.
3. ARGALL, B.; CHERNOVA, S.; VELOSO, M.; BROWNING, B.: **A survey of robot learning from demonstration**. En: **Robotics and Autonomous Systems** 57. p. 469–483, 2009.
4. DA SILVA FILHO, J. I.; ABE, J. M.; TORRES, G. L., **Inteligência Artificial com as Redes de Análises Paraconsistentes** Editora LTC, 1ª. Ed., Rio de Janeiro, pp.4-5, pp.40-84, pp.101-185. 2008.
5. DEMIRIS, Y. & Khadhour, B. (2006). **Hierarchical attentive multiple models for execution and recognition of actions**. Robotics and autonomous systems, 54(5), 361–369.
6. DA SILVA FILHO J. I, G. Lambert-Torres and J. M. Abe “**Uncertainty Treatment Using Paraconsistent Logic—In- troducing Paraconsistent Artificial Neural Networks,**” Vol. 21, IOS Press, Amsterdam, 2010.
7. D’ARPINO, C. Pérez e Shah J.A., **C-LEARN: Learning geometric constraints from demonstrations for multi-step manipulation in shared autonomy** - Robotics and Automation (ICRA), IEEE, 2017.

## REFERÊNCIAS

- ABRAMCZUK, A. A. **A Prática da tomada de decisão**. São Paulo: Atlas. 2009.
- ALISSANDRAKIS, A., Nehaniv, C. L., & Dautenhahn, K. **Imitation with alice: Learning to imitate corresponding actions across dissimilar embodiments**. Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on, 32(4), 482–496, 2002.
- ARGALL, Brenna D. CHERNOVA, Sonia; VELOSO, Manuela; BROWNING, Brett: **A survey of robot learning from demonstration**. En: **Robotics and Autonomous Systems** 57. p. 469–483, 2009.
- BAPTISTA, R., DA SILVA FILHO, J. I. MORILLA, J. C. **Proposal of Automatic Control three-way Valve through algorithms of Paraconsistent Annotated logic**, Science and Technology, ISSN: 2317-1316, Vol 2, No 2, p.p.69-73, 2013.
- BECKER, S. **“Unsupervised learning procedures for neural networks,”** International Journal of Neural Systems, vol. 2, pp. 17–33., 1991.
- BREAZEAL, C. & Scassellati, B., **A context-dependent attention system for a social robot**. (pp. 1146–1151), 1999.
- BRENDEN M. Lake, Tomer D. Ullman, Joshua B. Tenenbaum, Samuel J. Gershman, **Building Machines That Learn and Think Like People**. Cornell University Library, 2016.
- D'ARPINO, C. Pérez e Shah J.A., **C-LEARN: Learning geometric constraints from demonstrations for multi-step manipulation in shared autonomy** - Robotics and Automation (ICRA), IEEE, 2017
- DEMIRIS, Y. & Khadhour, B. (2006). **Hierarchical attentive multiple models for execution and recognition of actions**. Robotics and autonomous systems, 54(5), 361–369.
- DEMIRIS, Y and G. Hayes. **Imitation as a dual-route process featuring predictive and learning components: a biologically-plausible computational model**. In Dautenhahn and Nehaniv , chapter 13, 2002.
- COSTA, N. C. A., ABE, J. M., DA SILVA FILHO, J. I., MUROLO, A. C. **Lógica Paraconsistente Aplicada**, 2ª. Ed., Editora Atlas, 214 p., 1999.
- DA COSTA MOTTA. **Aprendizagem por demonstração baseada em redes neurais artificiais aplicada à robótica móvel**. Dissertação de Graduação. Universidade de Brasília UnB, Brasília/ DF, 2015.
- DA SILVA FILHO, J. I. & ABE, J. M., **Fundamentos das Redes Neurais Paraconsistentes – Destacando Aplicações em Neurocomputação**, (in Portuguese) Editora Arte & Ciência, ISBN 85-7473-045-9, 247 pp., 2001.

DA SILVA FILHO, J. I.; ABE, J. M.; TORRES, G. L., **Inteligência Artificial com as Redes de Análises Paraconsistentes** Editora LTC, 1ª. Ed., Rio de Janeiro, pp.4-5, pp.40-84, pp.101-185. 2008.

DA SILVA FILHO, J. I., “**Introdução às Células Neurais Artificiais Paraconsistentes**”. Seleção Documental, N.8 Ano2, 12/2007.

DA SILVA FILHO, J. I., “**Metodos de Aplicacoes da Logica Paraconsistente Anotada de anotação com dois valores-LPA2v**”. Seleção Documental, N.1 Ano1, 3/2006.

DA SILVA FILHO, J. I., “**Treatment of Uncertainties with Algorithms of the Paraconsistent Annotated Logic,**” **Journal of Intelligent Learning Systems and Applications**, Vol. 4 No. 2, 2012, pp. 144-153. 10.4236/jilsa.2012.42014.

FERNANDES, C.L.M.; Leite, A. G. **Instalação de sistemas eletroeletrônicos industriais**. Departamento Nacional, Departamento Regional de São Paulo. Editora: SENAI SP. 2013.

FERNANDES, C. L. M. “**Lógica Paraconsistente aplicada em Sistemas de Automação e Controle**”. Dissertação de Mestrado. Universidade Santa Cecília Unisanta, Santos/SP, 2012.

FERNANDES, C.L.M., Mario C. M., Da Silva Filho, J. I. “**Study for inclusion of Paraconsistente Annotated logic in specific Standards for use in Programable Controllers**”, **Science and Technology**, ISSN: 2317-1316, Vol 1 2, p.p 49-53, 2012.

FONOONI B. **Robot Learning and Reproduction of High-Level Behaviors**. Department of Computing Science Umeå University. ISBN: 978-91-7459-712-7, Umeå, Sweden 2013.

GÄRDENFORS, P. **Conceptual Spaces**. MIT Press., 2000.

GIENGER, M., Muhlig, M., & Steil, J. J.. **Imitating object movement skills with robots - a task-level approach exploiting generalization and invariance**. In **Intelligent Robots and Systems (IROS)**, 2010 IEEE/RSJ International Conference on (pp. 1262–1269).: IEEE., 2010.

GUIMARÃES, H.C.F. “**NORMA IEC 61131-3 Para Programação De Controladores Programáveis: Estudo e Aplicação**”. Dissertação de Graduação. Universidade Federal do Espírito Santo, Vitória ES, 2005.

HAYKIN S. **Redes Neurais – Princípios e Práticas**. 2ed.Porto Alegre: Bookman; 2001.

HAYKIN S.. **Neural Networks and Learning Machines** - 3 Ed, Bookman, pages 902. 2008.

IEC. **IEC61131-3 INTERNATIONAL ELECTROTECHNICAL COMISSION Programmable Controllers - Part 3. Programming Languages**, IEC. 1993.

SWEENEY J. D. and R. A. Grupen. **A model of shared grasp affordances from demonstration**. In Proceedings of the IEEE-RAS International Conference on Humanoids Robots, 2007.

CHEN J. and A. Zelinsky. **Programing by demonstration: Coping with suboptimal teaching actions**. The International Journal of Robotics Research, 22(5):299–319, 2003.

JOHN, K. H, Tiegelkamp M. “IEC61131-3: **Programing Industrial Automation Systems**”. Concepts and Prohraming Languages, Requiriments for Programing Systems, Decision – Making Aids. 2ed. New York, Springer, 390p., 2010.

KARLSSON, M., Robertsson, A., & Johansson, R. **Autonomous Interpretation of Demonstrations for Modification of Dynamical Movement Primitives**. In IEEE International Conference on Robotics and Automation (ICRA), 2017 (pp. 316-321). Singapore: Institute of Electrical and Electronics Engineers Inc.. DOI: 10.1109/ICRA.2017.7989040, 2017.

LANGLEY, P., Laird, J. E., & Rogers, S. (2009). **Cognitive architectures: Research issues and challenges**. Cognitive Systems Research, 10(2), 141–160.

MARIO, M. C. “**Modelo de análise de variáveis craniométricas através de redes neurais artificiais paraconsistentes**”. Tese de Doutorado. Universidade de São Paulo, 2006.

MÁRIO, M.C.; Ferrara, L.F.P.; Da Silva Filho, J. I. “**Treinamento de uma Célula Neural Artificial Paraconsistente de Aprendizagem**”. Seleção Documental, N.6 Ano2, 6/2007.

NEHANIV, C. L. & Dautenhahn, K. . **The correspondence problem. Imitation in animals and artifacts**, (pp. 41–61), 2002.

NETTO, D. A. C., MOLARINO, I., MÁRIO, M. C., ROCCO A. e DA SILVA FILHO, J. I. **Ensaio em Sistemas Microcontrolados com Algoritmos da Lógica Paraconsistente Anotada** - Revista Seleção Documental n 30 ISSN 1809-0648, pp 21-24 - Santos SP-Brasil., 2013.

NG, A. Coates, M. Diel, V. Ganapathi, J. Schulte, B. Tse, E. Berger, and E. Liang. **Inverted autonomous helicopter flight via reinforcement learning**. In International Symposium on Experimental Robotics, 2004.

M. NICOLESCU and M. J. Mataric. **Learning and interacting in human-robot domains**. Special Issue of IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans, 31(5):419–430, 2001a.

MORAES; CASTRUCCI. **Engenharia de Automação Industrial**. [S.l.]: Editora LTC, 2007.

OLIVEIRA, M.; Seixas, C.; Bottura Filho, J., **Aplicando a norma IEC 61131 na Automação de Processos**, 2007.

P. K. POOK and D. H. Ballard. **Recognizing teleoperated manipulations**. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA'93), 1993.

P. E. Rybski, K. Yoon, J. Stolarz, and M. M. Veloso. **Interactive robot task training through dialog and demonstration**. In Proceedings of the 2nd ACM/IEEE International Conference on Human-Robot Interactions (HRI'07), 2007.

REZENDE S.O. **Sistemas inteligentes – Fundamentos e Aplicações**. Editora Manole Ltda, 142, Barueri SP, 2003.

ROSÁRIO, João Mauricio. **Automação Industrial**. São Paulo: Editora: Baraúna. 2009.

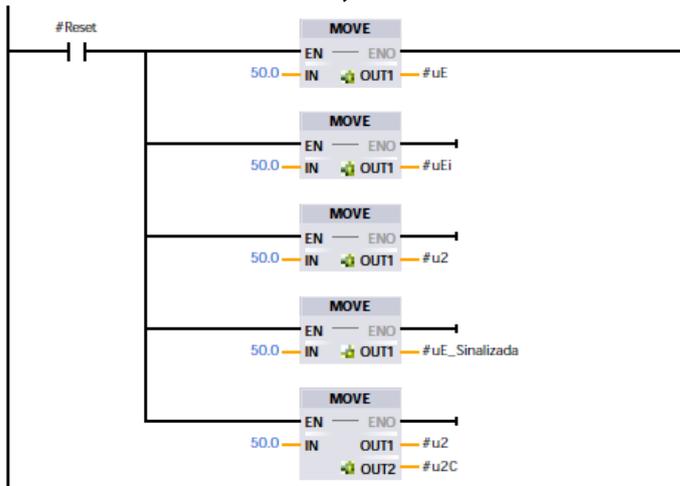
SILVEIRA, S.S.; Fernandes, C.L.M.; Reis, S., **Projeto de Sistemas Eletroeletrônicos Industriais**. Departamento Nacional, Departamento Regional de São Paulo. Editora: SENAI SP, 2013.

SILVEIRA, S.S.; Leite, A. G. **Manutenção de Sistemas Eletroeletrônicos Industriais**. Departamento Nacional, Departamento Regional de São Paulo. Editora: SENAI SP. 2013.

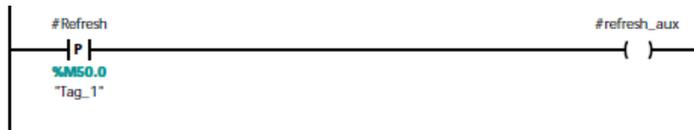
TORRES, C. R., **Sistema Inteligente Baseado na Lógica Paraconsistente Anotada Evidencial Et para Controle e Navegação de Robôs Móveis Autônomos em um Ambiente não Estruturado**, Tese de doutorado em Ciências em Energia Elétrica – Universidade Federal de Itajubá. 193 p. Itajubá- MG, agosto de 2010.

APÊNDICE A - Programação do bloco FB\_CNAP

Network 1: Inicia com 50, as variáveis do CNAP



Network 2: Atualiza o padrão de entrada apenas na borda de subida



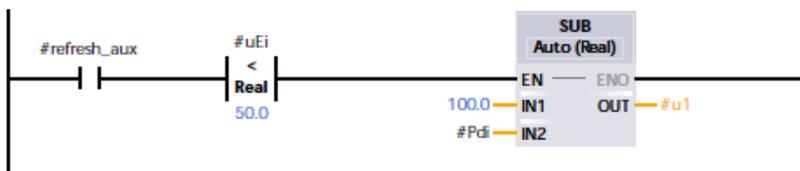
Network 3: Entre com o padrão inicial



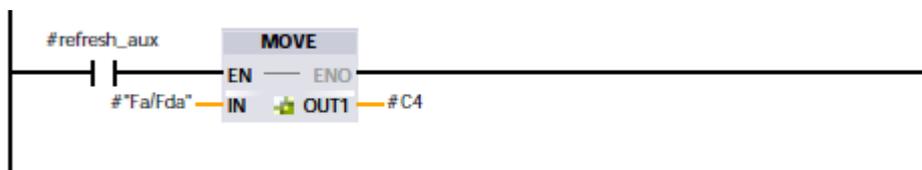
Network 4: Início e Calcule do grau de evidência inicial



Network 5: Determine o padrão a ser aprendido e complemente a entrada do grau de evidência

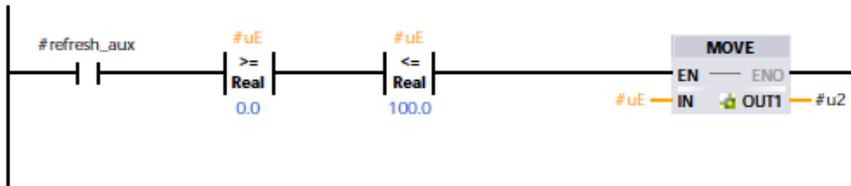


Network 6: Entre com o Fator de aprendizagem ou Desaprendizagem



APÊNDICE A - Programação bloco FB\_CNAP (continuação)

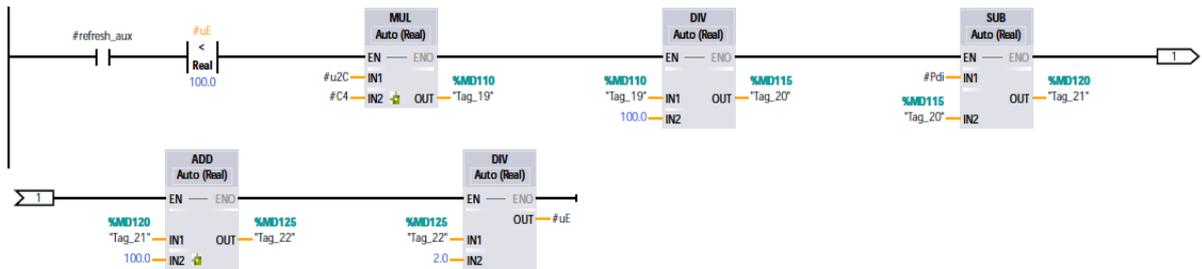
**Network 7: Conecte a saída da célula na entrada do grau de evidência**



**Network 8: Aplique o operador complemento na entrada do grau de evidência**



**Network 9: Calcule o grau de evidência**



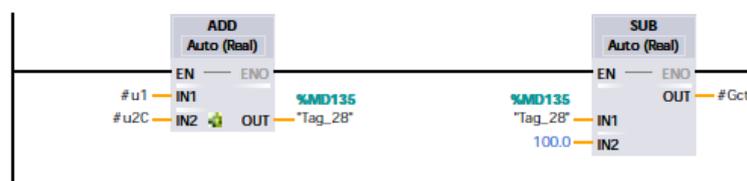
**Network 10: Sinalize Treinada**



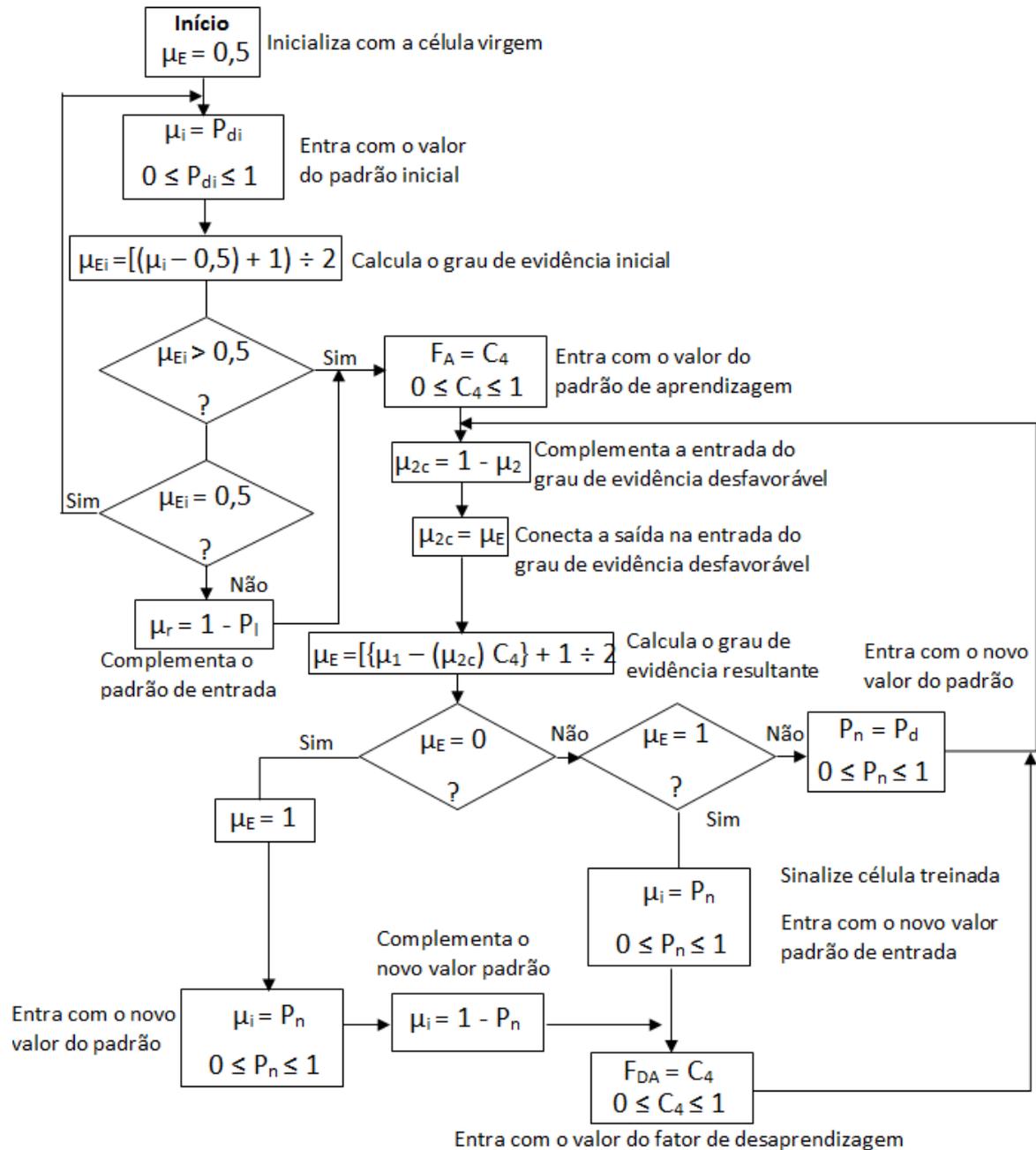
**Network 11: Cálculo do Grau de certeza (Gc)**



**Network 12: Cálculo do Grau de contradição (Gct)**



## ANEXO A - Fluxograma do processo de Aprendizagem e Desaprendizagem da CNAPap



Fonte: (DA SILVA FILHO et al.,2008)