Unisanta Science and Technology, 2020, 1, December
Published Online 2020 Vol.9 Nº2 http://periodicos.unisanta.br/index.php/sat)

UNISANTA
Science and Technology
ISSN 2317-1316

Tratamento de Incertezas com Sinais de Funções obtidos por configurações de Células Neurais Artificiais Paraconsistentes

Arnaldo de Carvalho Jr[1,2], Hyghor M. Côrtes[1], Clovis M. da Cruz[1,3], Dorotéa Vilanova Garcia[1], Maurício C. Mario[1] e João Inácio da Silva Filho[1]

[1]**UNISANTA** – Universidade Santa Cecília. Departamento de Pós-Graduação em Engenharia Mecânica
Rua Oswaldo Cruz, 288 CEP 11045-000, Santos – SP – Brasil.

[2]**IFSP**- Instituto Federal de Educação, Ciência e Tecnologia de São Paulo
Rua Maria Cristina, 50 - Jardim Casqueiro - Cubatão - São Paulo

[3]**UNIMONTE** Centro Universitário Monte Serrat
Rua Comendador Martins, 52 - Vila Matias, Santos – SP-Brasil,

E-mail: adecarvalhojr@yahoo.com.br
Received january, *2020*

Resumo: Tratamentos de incertezas mostram melhores resultados quando se utiliza lógicas não clássicas. A Lógica Paraconsistente Anotada com anotação de dois valores (LPA2v) é um tipo de lógica não clássica que, difere da lógica clássica (binária) por permitir o processamento de sinais de informações contraditórias. Uma Célula Neural Artificial Paraconsistente (CNAP) é um algoritmo que utiliza LPA2v para simular o comportamento de neurônios. Aplicando os graus de evidência favorável (μ) e desfavorável (λ) em suas duas entradas uma CNAP apresenta em suas saídas o grau de evidência de análise resultante μ_E (ou evidência de análise resultante real (μ_{ER})) e o intervalo de evidência resultante (φ_E). Uma Célula Neural Artificial Paraconsistente de Aprendizagem (CNAPap) usa a saída como uma realimentação para a entrada do grau de evidência desfavorável e pode ser treinada para responder (aprender) qualquer valor dentro de um intervalo [0,1] pertencente ao conjunto dos números reais. Configurações de (CNAPap) podem apresentar o comportamento de funções matemáticas utilizadas na análise e tratamento de sinais de dados analógicos. Neste trabalho apresentam-se os resultados de simulações de CNAPap e suas variações para comportamento de um integrador e diferenciador, o que pode vir a ser uma alternativa viável para reproduzir sinais de funções utilizados em sistemas de controle de automação.
Palavras-chave: lógica paraconsistente anotada, redes neurais, sistemas inteligentes, inteligência artificial.

Treatment of Uncertainties with Function Signals obtained by configurations of Paraconsistent Artificial Neural Cells

Abstract: Treatments of uncertainties show better results when non-classical logic is used. The Paraconsistent Annotated Logic with annotation of two values (PAL2v) is one type of non-classical logics that differs from classical logic (binary) by allowing the processing of signals of contradictory information. A Paraconsistent Artificial Neural Cell (PANcel) is an algorithm that uses PAL2v to simulate neuron behavior. Applying the degrees of favorable evidence (μ) and unfavorable evidence (λ) on its two inputs, it presents at its outputs the degree of resulting analysis evidence (μ_E) or real analysis evidence (μ_{RE}) and the resulting evidence interval (φ_E). A Paraconsistent Artificial Neural Cell of Learning (LPANcel) and its variations is a type of PANcel that uses the output as a feedback to the input of the unfavorable degree of evidence in time, it can learn any real value within a closed range (interval values [0,1]). This cell can be used in signal analysis and treatment. In this work, the results of simulations of LPANcel are presented and its variations as integrator and differentiator, what may be a viable alternative to reproducing signals of functions used in automation control systems.

Keywords: paraconsistent annotated logic, neural network, intelligent systems, artificial intelligence.

1. Introdução

Representações de situações inconsistentes e de informações contraditórias não são permitidas quando os sistemas são convencionais e fundamentados em Lógica Clássica. Isto porque a Lógica Clássica é estritamente binária e aceita apenas 2 estados possíveis para uma proposição *P*. Apesar disso a Lógica Clássica, justamente por ter a sua propriedade binária, é a base para os

Copyright © 2020 ISESC.

modernos sistemas eletrônicos digitais e de computadores. Com os recentes avanços da tecnologia, os limites entre o que é Falso e o que é Verdadeiro ficaram mais estreitos e verificou-se que, no mundo real, as contradições, as ambiguidades e as inconsistências podem aparecer ao se descrever situações que se deseja analisar, deixando a Lógica Clássica inoperante [1][2]. Devido a estas constatações surgiram as Lógicas não Clássicas, que têm como principal característica se opor aos princípios binários da Lógica Clássica [3][4].

As Lógicas Paraconsistentes (LP) são não Clássicas e surgiram da necessidade de se desenvolver ferramentas e metodologias para tratar situações contraditórias ou inconsistentes e apresentam resultados que permitem considerar as ambiguidades e contradições em sua estrutura de maneira não trivial. A LP e suas extensões têm se mostrado adequadas no tratamento de problemas ocasionados por situações em que surgem contradições ao se trabalhar com descrições do mundo real [2][5][6].

Este artigo utiliza um tipo particular de LP, a Lógica Paraconsistente Anotada com anotação de 2 valores, para o tratamento de dados, em um algoritmo denominado Célula Neural Artificial Paraconsistente de Aprendizagem (CNAPap) para produzir sinais que simulam as funções matemáticas da integração e da derivação [7][8][9][10][11] proporcionando condições de serem utilizados em sistemas de tratamento de incertezas. A seguir serão descritas as principais propriedades da Lógica Paraconsistente e os algoritmos que compõem uma CNAPap.

2. Lógica Paraconsistente Anotada - LPA

A Lógica Paraconsistente Anotada – LPA pertence a uma família de lógicas paraconsistentes que possui a propriedade de ser representada por um diagrama de Hasse de quatro vértices. Em cada um dos quatro vértices do Reticulado são representados os estados lógicos paraconsistentes extremos de uma proposição (*P*). Dessa forma, conforme a formalização da LPA2v uma proposição *P* é assim acompanhada por anotação que denota estados lógicos pertencentes a um reticulado finito [3] [4]. Portanto, cada anotação com valores máximos está relacionada a um estado lógico extremo que, por sua vez está localizado no vértice do reticulado, tal que:
T= Inconsistente, V = Verdadeiro, F = Falso e ⊥= Paracompleto ou Indeterminado.

2.1. Lógica Paraconsistente Anotada Com Anotação De dois Valores –LPA2v

Na Lógica Paraconsistente Anotada com Anotação de dois valores (LPA2v) a representação de evidências em um reticulado no plano real é formada por pares de graus de evidência que compõem uma anotação (μ,λ), permitindo assim atingir um maior poder de representação. Neste caso a proposição *P* é suportada pelo grau de evidência favorável (μ) e o grau da evidência desfavorável que nega ou rejeita a proposição *P*, representada pelo elemento (λ) [3].

Um operador é introduzido e definido como segue:

$\sim:|\tau|\rightarrow|\tau|$, onde $\tau=\{(\mu,\lambda)\}\,|\,\mu,\lambda\in[0,1]\subset\Re$ e ~ tem o significado de negação [1][2][4][5].

A associação de um par (μ, λ) para uma proposição *P* significa que o grau de evidência favorável em *P* é μ, e o grau de evidência desfavorável é λ, conforme as anotações no reticulado [4] [5]:

(1,0) → indica existência de evidência favorável total e evidência desfavorável igual a zero, assinalando uma conotação lógica verdadeira para a proposição *P*.

(0,1) → indica a existência de evidência favorável igual a zero e evidência desfavorável total, dando uma conotação de falsidade lógica para a proposição *P*.

(1,1) → indica a existência de ambos, evidência favorável e desfavorável totais, atribuindo uma conotação lógica de inconsistência à proposição *P*.

(0,0) → indica a existência de ambos, evidência favorável e desfavorável zero, atribuindo uma conotação lógica de indeterminação para a proposição *P*.

A figura 1 apresenta um reticulado representativo de quatro vértices associado com a LPA2v [6].

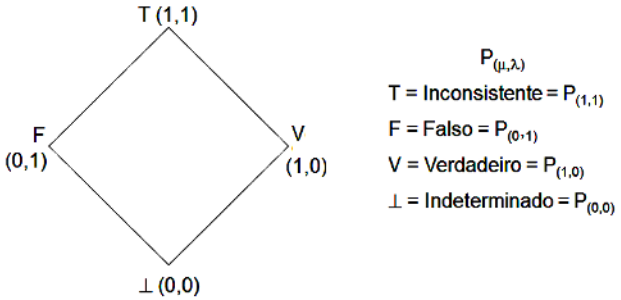


Figura 1. Reticulado finito de Hasse associado à LPA2v.

Através de transformações lineares entre um Quadrado Unitário no Plano Cartesiano (QUPC) e o reticulado associado à LPA é possível se chegar às seguintes equações matemáticas relacionadas à LPA2v [2][3] apresentadas a seguir.

Complemento do Grau de evidencia favorável será o Grau de Evidência Desfavorável.

$$\lambda = 1-\mu_2 \qquad (1)$$

O Grau de Certeza pode ser encontrado por:

Copyright © 2020 ISESC

$$Gc = \mu_1 - \lambda \quad (2)$$

Onde: μ_1 é o grau de evidência favorável, proveniente da primeira fonte de informação; λ é o grau de evidência desfavorável, como o complemento da segunda do graus evidência (μ_2) pertencente à segunda fonte de informação.

O Grau de Contradição pode ser encontrado por:

$$Gct = \mu_1 + \lambda \quad -1 \quad (3)$$

A análise paraconsistente gera valores de Gc e Gct entre +1 e -1. Para ser consistente com as entradas (μ,λ), em aplicações práticas eles precisam ser normalizados e seus valores limitados novamente entre 0 e 1 para respeitar os limites da lógica paraconsistente. Portanto, o μ_E é o grau de evidência resultante, sendo a saída normalizada de G_C, enquanto que μ_{CTR} é a saída normalizada de G_{CT}. O Grau de Evidência resultante será:

$$\mu_E = \frac{Gc+1}{2} \Rightarrow \mu_E = \frac{\mu_1 - \lambda + 1}{2} \quad (4)$$

Onde: G_C é o grau de certeza.

O Grau de contradição normalizado será calculado por:

$$\mu_{CTR} = \frac{\mu_1 + \lambda}{2} \quad (5)$$

Conforme visto em [1][8] pode-se obter um resultado mais preciso extraindo-se os efeitos da contradição em sucessivas análises de modo a se obter o grau de certeza resultante real (G_{CR}), como indicado na figura 2, e apresentado nas equações matemáticas a seguir. Sendo que D é a distância entre o estado lógico paraconsistente - $\varepsilon\tau(G_C,G_{CT})$ e o vértice V (se $G_C>0$) ou F (se $G_C<0$); então G_{CR} é o grau de certeza real; φ é o intervalo de certeza; μ_{ER} é o grau de evidência resultante real (G_{CR} normalizado com valores entre (0,1)); μ_{CTR} é o grau de contradição normalizado e φ_E é o intervalo de evidência resultante [7][8][9][10][11].

Cálculo da distância *D*:

$$D = \sqrt{\left(1 - |Gc|\right)^2 + Gct^2} \quad (6)$$

Cálculo do Grau de Certeza Real:

Para $Gc > 0$

$$G_{CR} = 1 - D \quad (7a)$$

Para $Gc < 0$

$$G_{CR} = D - 1 \quad (7b)$$

O Intervalo de Certeza será:

$$\phi = 1 - |Gct| \quad (8)$$

O Calculo do Grau de Evidência Real:

$$\mu_{ER} = \frac{G_{CR}+1}{2} \quad (9)$$

Na equação (7a,b), G_{CR} é obtido após os efeitos da contradição removidos, portanto sendo o resultado considerado o de valor puro.

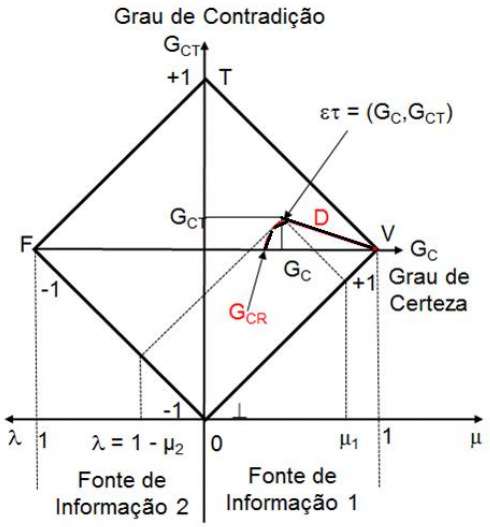


Figura 2. Reticulado LPA2v com Grau de certeza (Gc), Grau de Contradição (Gct) e Grau de Certeza real (G_{CR}).

Em uma aplicação o sinal de saída de um sistema paraconsistente de tratamento de incertezas fornecerá o resultado de Grau de Certeza Resultante Real (G_{CR}) sendo representado pelos valores (G_{CR} , $\varphi(\pm)$) [8].

Devido a retirada dos efeitos da contradição, o grau de evidência resultante real (μ_{ER}) da equação (9) é um valor mais consistente do que o grau de evidência resultante (μ_E) da equação (4) que não recebeu esta ação. Se o Gct tende a inconsistente, φ é marcado como "+"; e se Gct tende a indeterminado, φ é marcado como "-".

2.2. A Célula Neural Artificial Paraconsistente Padrão - CNAPp

Conforme apresentado em [1], pode-se escrever o conjunto de equações matemáticas da Lógica Paraconsistente Anotada e suas interpretações para formar um algoritmo denominado de Nó de Análise Paraconsistente (NAP) [6][7][8][11], sendo este o bloco básico de uma Célula Neural Artificial Paraconsistente (CNAP).

A figura 3 mostra a representação do algoritmo de análise paraconsistente – LPA2v.

A figura 4 apresenta o símbolo de uma CNAP padrão [10][10][11].

Copyright © 2020 ISESC

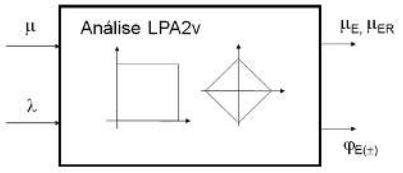


Figura 3. Representação do algoritmo de Análise Paraconsistente LPA2v.

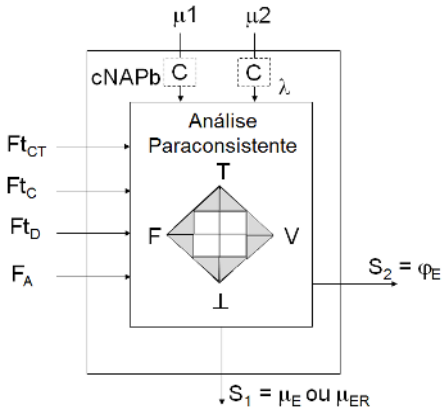


Figura 4. Símbolo da CNAP padrão.

As entradas (Ft_{CT} – Fator de Tolerância à Contradição, Ft_C – Fator de Tolerância à Certeza, Ft_D – Fator de Tolerância à Decisão, F_A – Fator de Aprendizagem) são valores externos de ajuste entre (0,1), permitindo-se atribuir outros valores de limites superior e inferior para G_C, G_{CT}, φ e taxa de aprendizagem. Estes fatores podem ou não ser utilizados, dependendo de como se aplica a CNAP e estão bem detalhados em [6]. O valor "1" é máximo e "0" é mínimo quando aplicado a cada fator [7][8][9][10][11].

As CNAPs podem ser combinadas para formar uma rede neural artificial paraconsistente - RNAP. As saídas de uma CNAP ou de uma RNAP podem ser aplicadas a um sistema de tratamento de sinais incertos ou de tomada de decisão, conforme proposto em [10]. A partir da CNAP básica várias células neurais paraconsistentes foram propostas em [6][11] e podem ser utilizadas em arranjos de interconexão para formar Redes Neurais Artificiais Paraconsistentes – RNAPs como elementos chave de sistemas especialistas [7][8][9] [11].

2.3. Célula Neural Artificial Paraconsistente Analítica – CNAPa

A CNAPa faz a análise paraconsistente e apresenta em sua saída o valor do grau de evidência resultante - μ_E, cuja saída poderá variar entre os limites superior e inferior estabelecidos pelo fator de tolerância à certeza, Ft_C.

Uma variante desta CNAP é a Célula Neural Artificial Paraconsistente Analítica Real - CNAPar que na saída apresenta μ_{ER}, ao invés de μ_E. A forma como o grau de evidência resultante e grau de evidência resultante real são calculados é descrito nas equações (1) a (9). A figura 5a apresenta o símbolo da CNAPa enquanto o símbolo da CNAPar é apresentado na figura 5b [10][11].

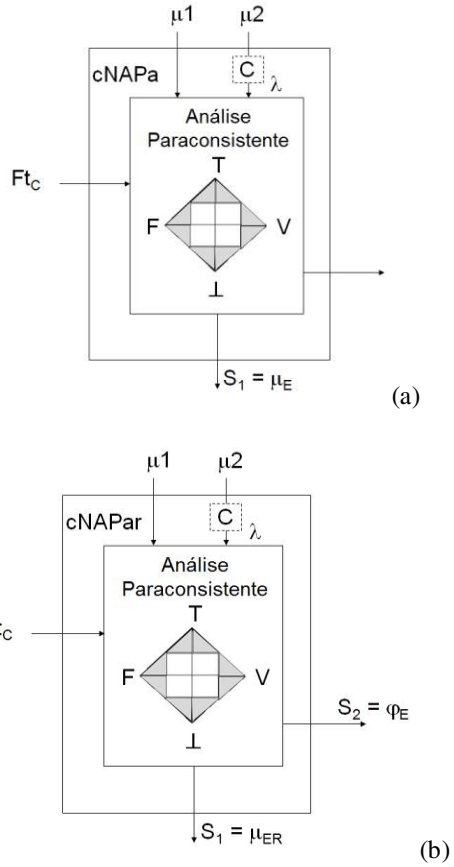


Figura 5. Representação de dois tipos de Célula Neural Artificial Paraconsistente Analítica: CNAPa (a) e CNAPar (b).

2.4. Célula Neural Artificial Paraconsistente De Aprendizagem - CNAPap

A célula Neural Artificial Paraconsistente de Aprendizagem – CNAPap, consiste basicamente de uma CNAPa cuja saída μ_E é aplicada à entrada do grau de evidência desfavorável - λ [5][9]. Da mesma forma, a Célula Neural Artificial Paraconsistente de Aprendizagem Real – CNAPapr consiste basicamente de uma CNAPar.

A figura 6a apresenta a CNAPap enquanto a CNAPapr é apresentada na figura 6b [10][11].

Copyright © 2020 ISESC

Como a célula está sempre realizando a análise entre o grau de evidência favorável (μ_1) atual com o grau de evidência resultante (μ_E) (ou grau de evidência resultante real (μ_{ER})) anterior, há iterações em um sentido de relógio e a célula levará um tempo para apresentar o resultado final, que dependerá da precisão (casas decimais) desejada.

A CNAPapr recebe o nome de célula de aprendizagem porque a saída tende a responder, ou aprender, qualquer valor real no intervalo fechado [0,1] aplicado à entrada μ_1. O fator de aprendizagem F_A funciona como um ajuste (ou ganho) do sinal realimentado, permitindo que a célula tenha uma aprendizagem mais rápida ou mais lenta, conforme equações (10) a (13) [10][11].

$$\mu_{Ek} = \frac{\{\mu_{(\kappa)} - (\lambda_{(\kappa)}) * F_A\} + 1}{2} \Rightarrow$$

$$\mu_{Ek} = \frac{\{\mu_{(\kappa)} - (1 - \mu_{E(\kappa-1)}) * F_A\} + 1}{2} \quad (10)$$

$$\lambda_{(\kappa)} = 1 - \mu_{E(k-1)} \quad (11)$$

$$\mu_{ERk} = \frac{\{\mu_{(\kappa)} - (\lambda_{(\kappa)}) * F_A\} + 1}{2} \Rightarrow$$

$$\mu_{ERk} = \frac{\{\mu_{(\kappa)} - (1 - \mu_{ER(\kappa-1)}) * F_A\} + 1}{2} \quad (12)$$

$$\lambda_{(\kappa)} = 1 - \mu_{ER\,k(\quad -} \quad (13)$$

Onde k refere-se ao valor atual e k-1 a amostra imediatamente anterior.
F_A o Fator de aprendizado (valor entre 0 e 1)

As figuras 7 e 8 apresentam um exemplo de comportamento de uma CNAPap e CNAPapr para um sinal quadrado aplicado à sua entrada, com F_A igual a 1. Também é visto que a CNAPap leva uma quantidade de amostras menor que a CNAPapr para "aprender o valor de entrada"[10][11].

Nota-se ainda que as saídas μ_E e μ_{ER} apresentam características de integral enquanto que a saída μ_{CTR} apresenta características de diferencial normalizado [0,1].

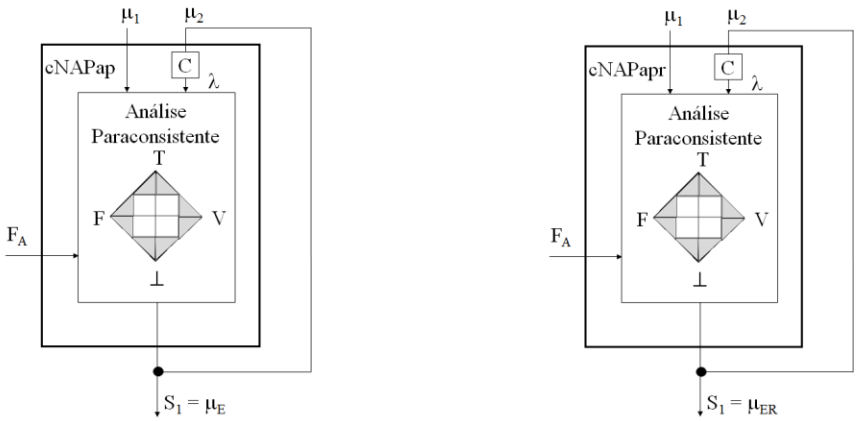


Figura 6. CNAPap (a) e CNAPapr (b).

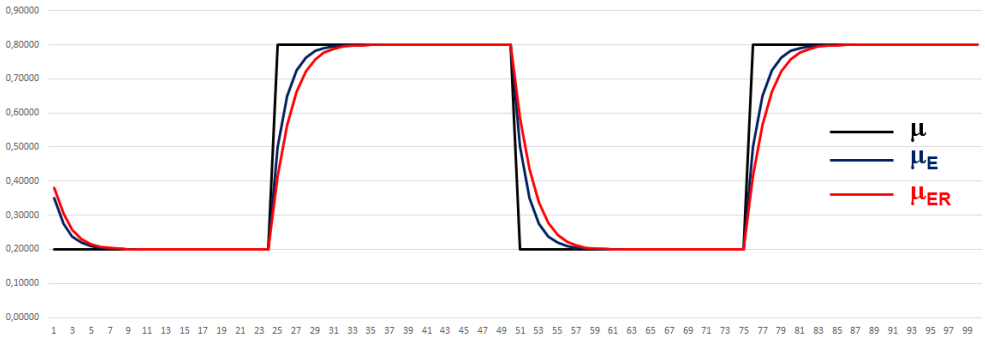


Figura 7. Saídas μ_E da CNAPap e μ_{ER} da CNAPapr como respostas à um sinal de onda quadrada aplicado à entrada de grau de evidência favorável μ.

Copyright © 2020 ISESC

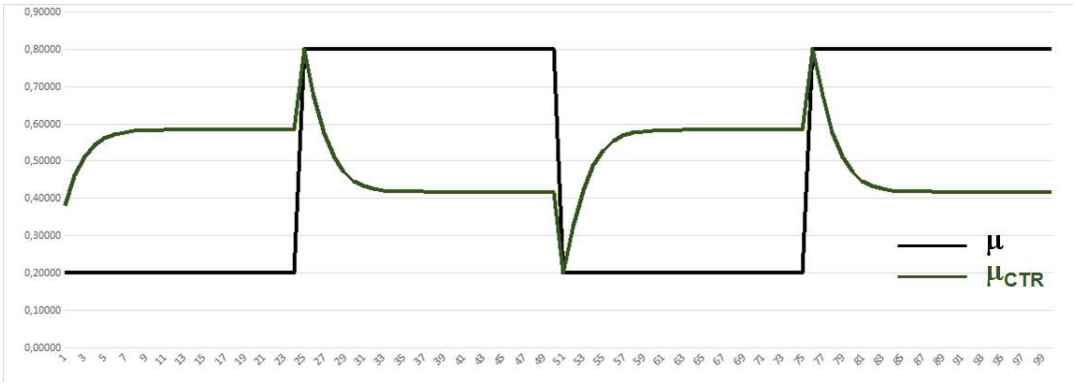


Figura 8. Saída μ_{CTR} da CNAPapr como resposta à um sinal de onda quadrada aplicado à entrada μ.

Pode-se verificar que a CNAPap necessita de 20 amostras ou iterações para "aprender" reproduzindo assim o valor exato do sinal de entrada na saída. A CNAPapr necessita de 21 iterações para obter estas condição de repetir o valor da entrada na saída [10][11].

2.5. A CNAP Funcionando por Extração do Efeito da Contradição

A CNAPap responde de modo esperado quando o Fator de aprendizagem – F_A tem valor unitário, ou quando apenas o F_A é variável e o valor de μ aplicado na entrada é constante. Entretanto, conforme apresentado em [11], quando se varia simultaneamente o valor do grau de evidência favorável - μ e o fator de aprendizagem - F_A, a saída converge para um valor diferente do esperado a ser obtido na entrada [10][11].

Para corrigir esta condição em [11] foi proposta uma alteração na equação da CNAPap, sendo chamada de célula Neural Artificial Paraconsistente por Extração do Efeito da Contradição - CNAPapxct, cujo valor do grau de evidência resultante atual, é obtido pela equação apresentada a seguir:

$$\mu_{Ek} = \mu_{E(k-1)} + G_{CT} * F_A \qquad (14)$$

Onde F_A = Fator de aprendizagem com valores entre 0 e 1.

A figura 9 apresenta a resposta a um sinal de onda quadrada aplicada na entrada de uma CNAPapxct para diferentes valores de F_A.

Nota-se que com F_A nulo não há aprendizagem, por outro lado, com F_A de valor unitário a saída reflete diretamente o valor apresentado na entrada [10][11].

A CNAPapxct utilizando F_A com valor de 0,5 apresenta comportamento idêntico à cNAPap com F_A de valor unitário.

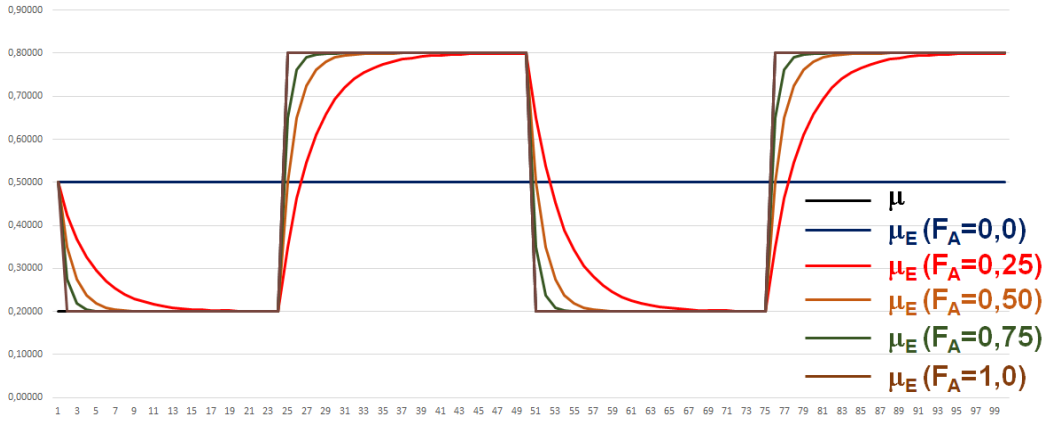


Figura 9. Saídas μ_E da CNAPapxct à um sinal de onda quadrada aplicado à entrada μ para diferentes valores de F_A.

Copyright © 2020 ISESC

3. Materiais e Métodos

Neste trabalho, para demostrar, através de simulações as diferentes formas de suas aplicações no tratamento de sinais de dados analógicos foram utilizadas funções do *software* MATLAB® R2008a com aplicações do SIMULINK na configuração das células e suas estruturações. As figuras 10, 11 e 12 apresentam respectivamente modelos de CNAPap, CNAPapr e CNAPapxct criados no SIMULINK para as simulações utilizadas.

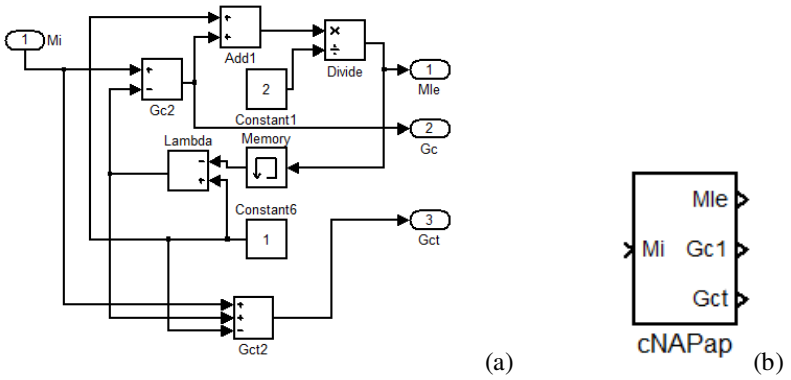


Figura 10. Unidade de CNAPap criada no MATLAB® SIMULINK (a) e em bloco (b).

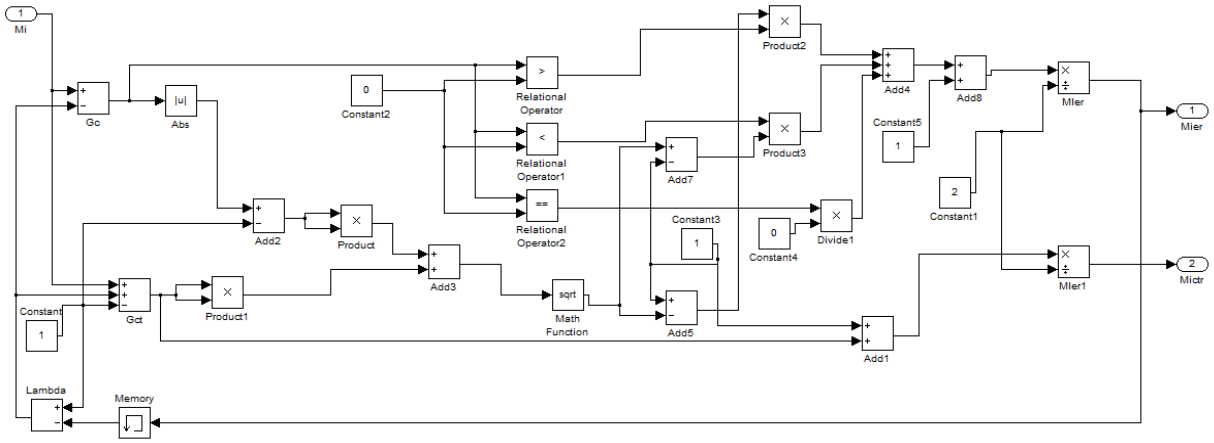


Figura 11. Unidade de CNAPapr criada no SIMULINK.

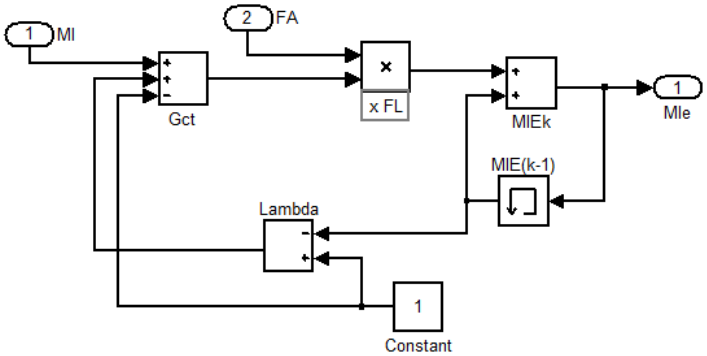


Figura 12. CNAPapxct criada no SIMULINK.

Copyright © 2020 ISESC

4. Resultados e Discussões

A seguir são mostrados os resultados de simulações que estão relacionados as funções de integração e de derivação simuladas pelas configurações das Células. Os resultados mostram a aplicação de configurações de CNAPs na geração de Sinais de integração e de derivação típicos de utilização em sistemas de controle e automação.

4.1. Integrador

A figura 13 apresenta uma célula CNAPapxct, funcionando como Integrador, para um sinal de onda quadrada com diferentes valores de Fator de Aprendizagem – F_A. Nesta aplicação foi utilizada uma onda quadrada com período de 5 segundos (0,20 Hz) e tempo de amostragem de 0,1 no simulador (10 amostras por segundo).

Nota-se que para um fator de aprendizagem muito baixo, F_A=0,05, o sinal de saída é praticamente uma onda triangular.

A figura 14 apresenta uma simulação da CNAPapxct para um sinal rampa aplicado à entrada.

Observa-se nesta figura que apenas para valores de F_A muito baixos é que o efeito exponencial do integrador se apresenta na saída.

Na figura 15 apresenta-se o resultado de uma simulação semelhante, porém com um sinal senoidal aplicado à entrada de grau de evidência favorável, ao invés de uma onda quadrada.

Percebe-se neste caso que a CNAPapxct funciona mais como um módulo de atraso (ou defasagem) do que propriamente um integrador. Verifica-se que quanto menor o F_A, maior o atraso e menor o valor pico a pico na saída.

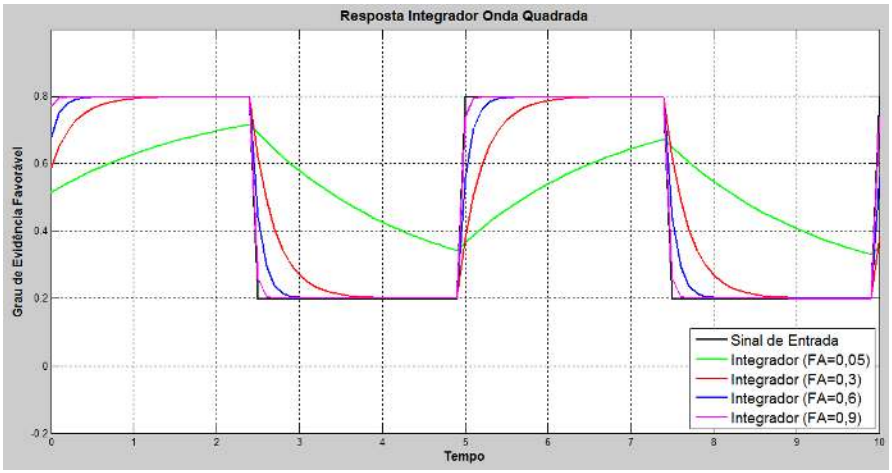


Figura 13. Onda Quadrada aplicada à uma CNAPapxct com diferentes valores de F_A.

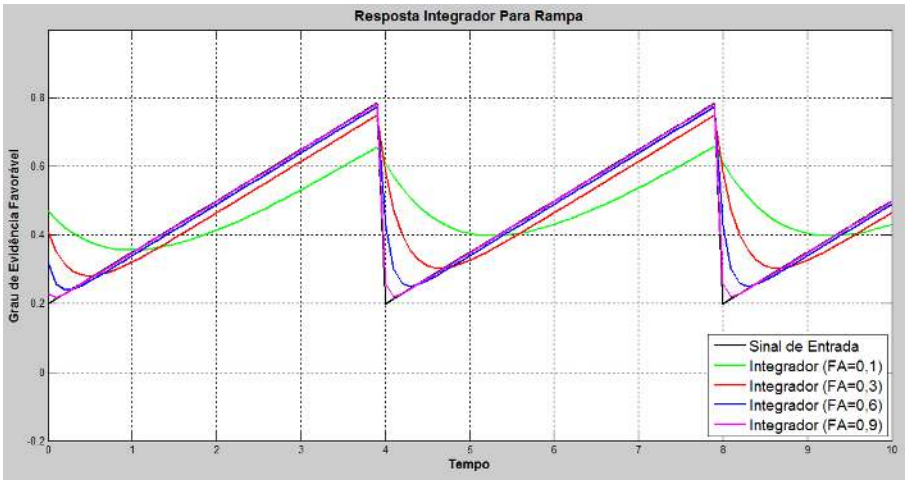


Figura 14. Sinal Rampa aplicado à uma CNAPapxct com diferentes valores de F_A.

Copyright © 2020 ISESC

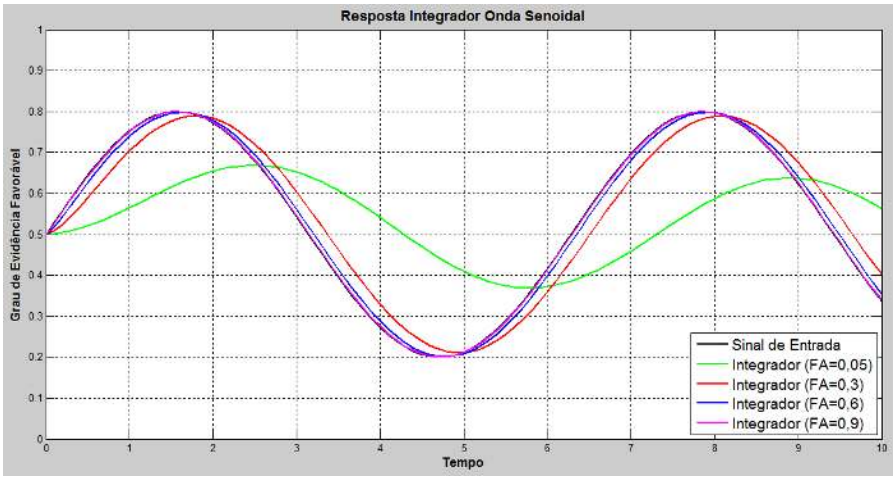


Figura 15. Onda Senoidal aplicada à uma CNAPapxct com diferentes valores de F_A.

4.2. Diferenciador

Conforme apresentado anteriormente, o sinal proveniente de G_C, G_{CR}, μ_E e μ_{ER} tem efeito de integração. Por outro lado, o sinal proveniente de G_{CT} e μ_{CTR} tem efeito diferencial. Assim sendo, foram repetidas as simulações da CNAPapxct apresentadas para o INTEGRADOR, porém com os valores de saída do G_{CT} para diferentes valores de F_A, conforme mostram os resultados das figuras 16, 17 e 18. Percebe-se que para a Onda Quadrada da figura 16, quanto maior o F_A, mais próximo de um Diferenciador Ideal a CNAPapxct se comporta, isto porque o aprendizado da célula é mais rápido. De acordo com a figura 17, para o sinal de rampa, nota-se que para valores de F_A baixo, a saída μ_{CTR} tende a um valor constante diferente de 0,5 apesar de ainda próximo do mesmo.

Assim como para o Integrador, conforme a figura 18, no Diferenciador nota-se também uma característica mais de defasagem do que função Derivada propriamente dita para sinais senoidais, sendo que quanto menor o F_A, menor a amplitude na saída.

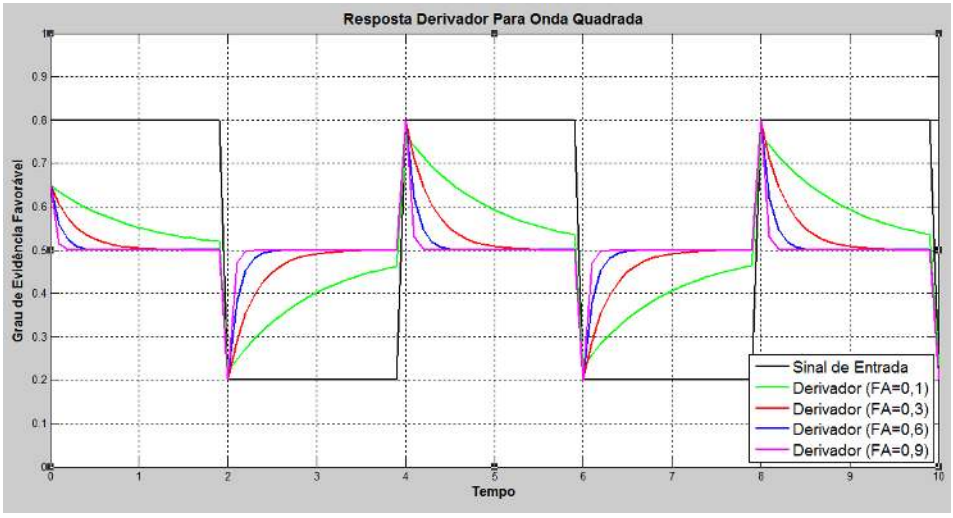


Figura 16. Onda Quadrada aplicada à uma CNAPapxct e saída μ_{CTR} com diferentes valores de F_A.

Copyright © 2020 ISESC

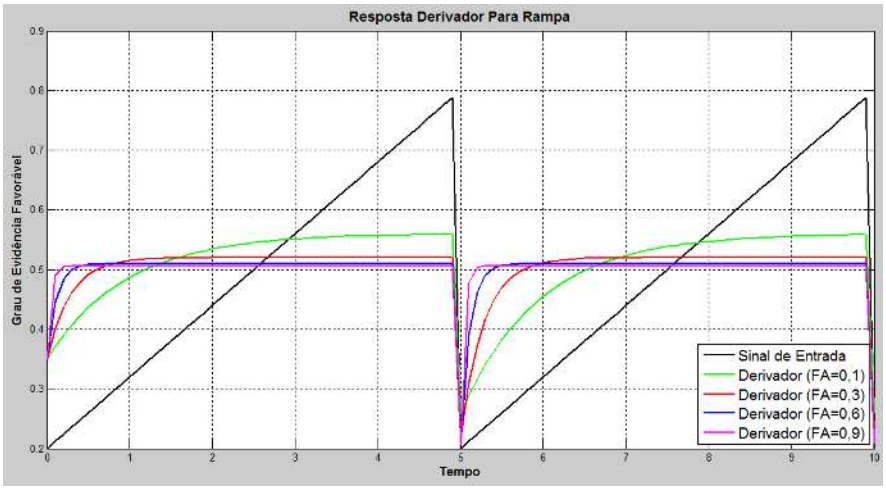


Figura 17. Sinal Rampa aplicado à uma CNAPapxct e saída μ_{CTR} com diferentes valores de F_A.

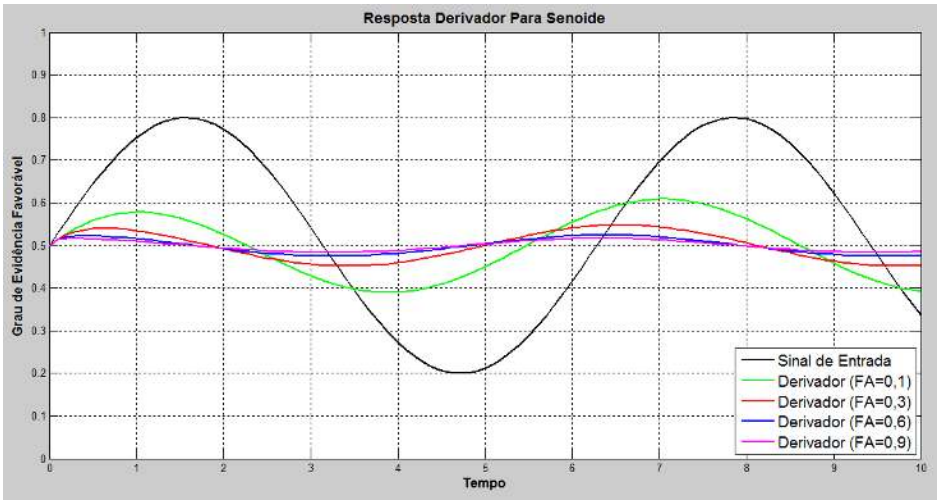


Figura 18. Onda Senoidal aplicada à uma CNAPapxct e sinais de saída μ_{CTR} para diferentes valores de F_A.

5. Conclusões

O presente estudo apresentou duas diferentes aplicações potenciais de CNAP, onde é possível a aplicação das variantes de células neurais artificiais paraconsistentes de aprendizagem no tratamento de sinais de informação para sistemas de automação. Para as configurações, as equações da LPA2v são relativamente simples e por esta razão consomem poucos recursos computacionais, sendo muito úteis para sistemas com limitações de processamento e memória, tais como sensores de baixo consumo de energia e sistemas embarcados. As células neurais artificiais paraconsistentes configuradas para tratar incertezas e gerar sinais de funções conhecidas no controle da Automação apresentam resultados promissores e podem, dentro de certos limites, ser aplicadas no tratamento de sinais de sistemas industriais, desde que novas investigações sejam feitas para se estudar com mais detalhes o comportamento destes sinais, bem como as suas características elétricas e de ruído que estejam submetidas e seus limites nas diferentes configurações propostas pela LPA2v.

Copyright © 2020 ISESC

Referências

[1] Da Silva Filho, J. I.. **Métodos de Aplicações da Lógica Paraconsistente Anotada de anotação com dois valores-LPA2v**. Revista Seleção Documental, ISSN 1809-0648, Santos, SP, BR Número 1, Ano 1, p. 18 – 25, Março 2006.

[2] Da Silva Filho, J. I.. **Análises de Sinais de Informações em Lógica Paraconsistente Anotada**. Revista Seleção Documental, ISSN 1809-0648, Santos, São Paulo – Brasil, Número 14, Ano 4, p. 22 – 26, Junho 2009.

[3] Da Costa, N. C. A.; Abe, J. M.; Subrahmanian, V.S. . **Remarks on annotated logic, Zeitschrift f. math. Logik und Grundlagen d. Math**.. 37, Vol.37, pp 561-570, 1991.

[4] Subrahmanian, V. S. .**On the semantics of quantitative logic programs**. Proc. 4th. IEEE Symposium on Logic Programming", Computer Society Press, Washington D.C., 1987.

[5] Da Cruz, C. M.. .**Application of Paraconsistent Artificial Neural network in Statistical Process Control acting on voltage level monitoring in Electrical Power Systems** - Intelligent System Application to Power Systems (ISAP), pp.(1-6), Porto–PT, September, 2015.

[6] Da Silva Filho, J. I.. **Introdução às Células Neurais Artificiais Paraconsistentes**. , Revista Seleção Documental, SP. Brazil. ISBN 18090648. No 8, Dezembro, 2007.

[7] Da Silva Filho, J. I.. Da Cruz, C. M.. Rocco, A.. Garcia, D. V.. Ferrara, L.F. P. Onuki, A. S.. Mario, . C.. Abe, J. M.. M. A. .**Paraconsistent Artificial Neural Network for structuring Statistical Process Control in Electrical Engineering**. Springer International Publishing, ISBN 978-3-319-40418-9, 2016.

[8] Da Silva Filho, J. .I. .**Treatment of Uncertainties with Algorithms of the Paraconsistent Annotated Logic**., Journal of Intelligent Learning Systems and Applications, 2012, 4, 144-15. Published Online May 2012 at http://dx.doi.org/10.4236/jilsa.2012.42014.

[9] Mario, M. C.; Ferrara, L. F. P.; Da Silva Filho, J. I..**Treinamento de uma Célula Neural Artificial Paraconsistente de Aprendizagem**. Seleção Documental Magazine, No 6, pp 11 – 16, 2007, SP. Brazil. ISBN 18090648.

[10] Minicz, M. de F.; Matuck, G. R.; Tasinaffo, P. M.; Da Silva Filho, J. I.. .**Célula Neural Artificial Paraconsistente de Aprendizagem por Extração do Efeito da Contradição**.. Seleção Documental Magazine, No 33, pp 3 – 9, Março, 2014, SP. Brasil. ISBN 18090648.

[11] De Carvalho Junior A., Mario M. C., Cortês, H. M. e Da Silva Filho J. I. **Célula Neural Artificial Paraconsistente utilizada como um Módulo Redutor de Ruídos em Sinais com Amplitudes oscilantes**. *Unisanta Science and Technology* (52-60) p.p. ISSN 2317-1316, 2017.

Copyright © 2020 ISESC