

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E
TECNOLOGIA DE SÃO PAULO – IFSP

FABIO BRUCI WU

**ESCALONAMENTO DE GANHOS EM CONTROLADORES
PID/PI COM A UTILIZAÇÃO DE REDES NEURAIAS
ARTIFICIAIS APLICADOS EM MÁQUINAS DE
EMPACOTAMENTO**

São Paulo

2017

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E
TECNOLOGIA DE SÃO PAULO – IFSP

FABIO BRUCI WU

**ESCALONAMENTO DE GANHOS EM CONTROLADORES
PID/PI COM A UTILIZAÇÃO DE REDES NEURAIAS
ARTIFICIAIS APLICADOS EM MÁQUINAS DE
EMPACOTAMENTO**

Dissertação de Mestrado apresentada ao Instituto Federal de Educação, Ciência e Tecnologia de São Paulo – IFSP como parte dos requisitos para obtenção do título de Mestre em Automação e Controle de Processos.

Orientador: Prof. Dr. Paulo Roberto Barbosa

São Paulo

2017

Catalogação na fonte
Biblioteca Francisco Montojos - IFSP Campus São Paulo
Dados fornecidos pelo(a) autor(a)

W959e	<p>Wu, Fabio Bruci Escalonamento de ganhos em controladores pid/pi com a utilização de redes neurais artificiais aplicados em máquinas de empacotamento / Fabio Bruci Wu. São Paulo: [s.n.], 2017. 90 f. il.</p> <p style="text-align: center;">Orientador: Dr. Paulo Roberto Barbosa</p> <p style="text-align: center;">Dissertação (Mestrado Profissional em Automação e Controle de Processos) - Instituto Federal de Educação, Ciência e Tecnologia de São Paulo, IFSP, 2017.</p> <p style="text-align: center;">1. Neural Gain Scheduling. 2. Gain Scheduling. 3. Controle de Servo Motor. 4. Rede Neural Artificial. I. Instituto Federal de Educação, Ciência e Tecnologia de São Paulo II. Título.</p> <p>CDD 629.8</p>
-------	---



ATA DE EXAME DE DEFESA DE DISSERTAÇÃO

Nome do Programa: **Mestrado Profissional em Automação e Controle de Processos**

Nome do(a) Aluno(a): Fábio Bruci Wu

Nome do(a) Orientador(a): Prof. Dr. Paulo Roberto Barbosa

Nome do(a) Coorientador(a):

Título do Trabalho: "Escalonamento de ganhos em controladores PID/PI com a utilização de redes neurais artificiais aplicados em máquinas de empacotamento"

Abaixo o resultado de cada participante da Banca Examinadora

Nome completo dos Participantes Titulares da Banca	Sigla da Instituição	Aprovado / Não Aprovado
Prof. Dr. Paulo Roberto Barbosa – Orientador	IFSP – SJC	Aprovado
Prof. Dr. Eduardo Alves da Costa – Membro Interno	IFSP – SPO	APROVADO
Prof. Dr. Luis Fernando de Almeida - Membro Externo	UNITAU	APROVADO
Nome completo dos Participantes Suplentes da Banca	Sigla da Instituição	Aprovado / Não Aprovado
Prof. Dr. Alexandre Simião Caporali – Membro Interno	IFSP – SPO	
Prof. Dr. Graziela Marchi Tiago – Membro Externo	IFSP – SJC	

Considerando-o: APROVADO
 NÃO APROVADO

Assinaturas

São Paulo, 03 de julho de 2017



Presidente da Banca



Membro Interno



Membro Externo

Observações:

Realizar ajustes no texto apontado pela banca nos exemplares de defesa.

AGRADECIMENTOS

À empresa Masipack, que me deu diversas oportunidades ao longo da minha carreira e forneceu todo o suporte em materiais e equipamentos para que este trabalho fosse realizado.

À minha mãe, Anésia de Fátima de Moraes, que sempre ofereceu todo o apoio necessário para eu me dedicar aos estudos e nunca deixou de me incentivar em todos os momentos da minha vida.

À minha irmã, Vanessa Ikebe Yamai, que mesmo estando muito distante, jamais deixou de me apoiar emocional e financeiramente, e com quem sei que poderei contar em qualquer situação.

Ao meu pai, Wu Kuang Cheng, que infelizmente não está mais conosco nesta vida para poder participar deste momento, mas que tenho certeza de que estaria muito orgulhoso deste trabalho.

Ao meu filho, Pedro Henrique, meu eterno melhor amigo, que sempre compreendeu quando precisei subtrair as poucas horas de convívio juntos para poder me dedicar aos estudos.

À Regiane Elias, meu grande amor, que me apoiou e me incentivou desde o processo seletivo, dedicando tempos preciosos da sua vida para me ajudar.

A todos os meus amigos, dos quais reduzi momentos de convivência para que este trabalho fosse realizado.

Ao meu orientador, Paulo Roberto Barbosa, que acreditou no meu tema e sempre me atendeu da melhor forma possível, tornando-se fundamental para o sucesso deste trabalho.

A todo o corpo docente e demais funcionários do Instituto Federal de São Paulo, instituição a qual sempre fui tratado da melhor forma.

Aos professores e colegas da Universidade Paulista, instituição a qual fiz minha graduação e que me proporcionou toda a base necessária para este trabalho.

“Algo só é impossível até que alguém duvide e prove o contrário”

Albert Einstein

RESUMO

Sistemas controlados por servo motor são facilmente encontrados na indústria e, na maioria dos casos, esses sistemas não são lineares, o que pode demandar o desenvolvimento de estratégias mais elaboradas para o controle do servo motor. A planta utilizada neste estudo é um conjunto de selagem acionado por servo motor, onde o controle é feito diretamente por um servo *drive* com controle PID/PI (Proporcional Integral Derivativo/Proporcional Integral), sendo possível manipular os valores dos ganhos do controlador. O conjunto de selagem efetua o movimento de abertura e fechamento de forma cíclica, produzindo um sobressinal de posição de 2% ou mais quando ajustado pelo método de sintonização automática dos ganhos. Sabendo que a planta não é linear e que possui diferentes ganhos para distintas faixas de operação, este trabalho tem como objetivo propor um escalonamento de ganhos para essas diferentes faixas de operação, utilizando uma rede neural artificial capaz de reproduzir uma relação que irá gerar os ganhos de acordo com a faixa de operação do sistema, reduzindo o sobressinal do conjunto de 2% ou mais para um sobressinal menor que 0,5%.

Palavras-chave: *neural gain scheduling, gain scheduling*, controle de servo motor, rede neural artificial.

ABSTRACT

Servo motor controlled systems are readily available in the industry and in most cases, these systems are nonlinear which may require developing more elaborate strategies for engine control. The plant used in this study is a set of servo motor driven sealing where control is done directly by a servo drive with PID/PI control (Proportional Integral Derivative/Proportional Integral) where you can manipulate the values of the controller gains. The sealing assembly performs the opening and closing movement cyclically, producing a position overhangs of 2% or more when adjusted by the automatic gain tuning method. Knowing that the plant is not linear and that has different gains for different operating ranges, this work aims to propose a gain scheduling for the different operating ranges using an artificial neural network capable of playing a function that will generate the gains according to the system operating range, improving control performance throughout its operating range, Reducing the salience of the set of 2% or more to a salient of less than 0.5%.

Keywords: neural gain scheduling, gain scheduling, servo motor control, artificial neural network.

LISTA DE FIGURAS

Figura 1 - Máquina de empacotamento vertical.....	18
Figura 2 - Pacote almofada.....	19
Figura 3 - Posição de selagem	20
Figura 4 - Vista traseira do conjunto	21
Figura 5 - Posição de abertura	22
Figura 6 - Controlador Gain Scheduling.....	25
Figura 7 - Características da entrada e saída dos tanques	26
Figura 8 - LMS unidade de ajuste	27
Figura 9 - Janela de treinamento	28
Figura 10 - Diagrama de bloco – Gain Scheduling	28
Figura 11 - Representação em blocos do sistema nervoso	30
Figura 12 - Modelo matemático de um neurônio	32
Figura 13 - Função de limiar	34
Figura 14 - Função linear por partes.....	34
Figura 15 - Função sigmoide	35
Figura 16 - Arquitetura de uma MLP com duas camadas ocultas.....	36
Figura 17 - Regra delta de correção dos pesos sinápticos.....	37
Figura 18 - Grafo de fluxo de sinal ressaltando os detalhes do neurônio de saída j.....	38
Figura 19 - Grafo de fluxo de sinal ressaltando os detalhes do neurônio de saída k conectado ao neurônio oculto j.....	39
Figura 20 - Esquema do mordente – Posição de abertura	41
Figura 21 - Esquema do mordente – Posição de fechamento	42
Figura 22 - Diagrama de malha do servo.....	43
Figura 23 - Curva do servo motor com ajustes iniciais	46
Figura 24 - Detalhe do erro de posição na região A	48
Figura 25 - Entradas e saídas da RNA	52
Figura 26 - Rede Neural Artificial utilizada	57
Figura 27 - Desempenho do treinamento	58
Figura 28 - Performance do treinamento	59
Figura 29 - Regressão	59
Figura 30 - Estados do treinamento.....	60

Figura 31 - Neurônio – Pesos e bias	61
Figura 32 - Neurônio e seus componentes	68
Figura 33 - Declaração do número de Euler	71
Figura 34 - Atribuição de pesos e bias	71
Figura 35 - Normalização da entrada	72
Figura 36 - Neurônio N1, N2 e N3	73
Figura 37 - Normalização da saída PKP	73
Figura 38 - Ganho PKP – Simulação no CLP comparado com simulação no Matlab	74
Figura 39 - Ganho PKI – Simulação no CLP	75
Figura 40 - Ganho PKD – Simulação no CLP	75
Figura 41 - Ganho VKI – Simulação no CLP	76
Figura 42 - Ganho VKP – Simulação no CLP	76
Figura 43 - Sobressinal de posicionamento 2%.....	78
Figura 44 - Sobressinal de posicionamento menor que 0,5%.....	79
Figura 45 - Posição e corrente do servo motor sem escalonamento dos ganhos	80
Figura 46 - Posição e corrente do servo motor com escalonamento dos ganhos	81
Figura 47 - Posição e corrente pela lógica SE.....	84
Figura 48 - Comparação dos resultados.....	85

LISTA DE TABELAS

Tabela 1 - Ganhos iniciais.....	47
Tabela 2 - Ganhos ajustados para a região A.....	48
Tabela 3 - Ganhos ajustados para a região B.....	49
Tabela 4 - Ganhos ajustados para a região C.....	50
Tabela 5 - Dados para treinamento da região A.....	53
Tabela 6 - Dados para treinamento da região B	54
Tabela 7 - Dados para treinamento da região B	55
Tabela 8 - Pesos e bias da rede treinada.....	61
Tabela 9 - Valor máximo e mínimo das entradas e saídas.....	65

LISTA DE GRÁFICOS

Gráfico 1 - Deslocamento do mordente	45
Gráfico 2 - Ganho de posição PKP	62
Gráfico 3 - Ganho de posição PKI	63
Gráfico 4 - Ganho de posição PKD	63
Gráfico 5 - Ganho de velocidade VKP	64
Gráfico 6 - Ganho de velocidade VKI	64
Gráfico 7 - Ganho PKP médio	82

LISTA DE SÍMBOLOS

Θ - Parâmetros do controlador

Δt - Intervalo de tempo

$\varphi(v)$ - Função de ativação

$y_j(n)$ - Saída real do neurônio

$d_j(n)$ - Saída desejada do neurônio

$e_j(n)$ - Erro

$\delta_j(n)$ - Gradiente local

η - Taxa de aprendizagem

$\mathcal{E}(n)$ - Energia total do erro

Tansig(n) - Função de transferência tangente hiperbólica

LISTA DE ABREVIATURAS E SIGLAS

PID - Proporcional Integral Derivativo

PI - Proporcional Integral

RNA - Rede Neural Artificial

CLP - Controlador Lógico Programável

Autotuning - Sintonização automática de um controlador

Gain Scheduling - Escalonamento de ganhos

LMS - *Last Mean Square* - Mínimo erro quadrático

WP - Ganho proporcional

WI - Ganho integral

WD - Ganho derivativo

KP - Ganho proporcional

KI - Ganho integral

KD - Ganho derivativo

MLP – *Multilayer Perceptron*

PKP - Ganho proporcional de posição

PKI - Ganho integral de posição

PKD - Ganho derivativo de posição

VKP - Ganho proporcional de velocidade

VKI - Ganho integral de velocidade

$\overline{PKP(A)}$ - Ganho proporcional de posição médio – região A

$\overline{PKP(B)}$ - Ganho proporcional de posição médio – região B

$\overline{PKP(C)}$ - Ganho proporcional de posição médio – região C

SUMÁRIO

1 INTRODUÇÃO.....	18
1.1 Justificativas.....	20
1.2 Objetivo geral.....	22
1.3 Objetivos específicos.....	22
1.4 Organização da dissertação.....	23
2 REVISÃO DA LITERATURA.....	24
3 REDES NEURAIS ARTIFICIAIS.....	30
3.1 Introdução.....	30
3.2 Histórico.....	31
3.3 O neurônio - Perceptron de camada única.....	32
3.4 Funções de ativação.....	33
3.4.1 Função limiar.....	33
3.4.2 Função linear por partes.....	34
3.4.3 Função sigmoide.....	35
3.5 RNA do tipo perceptron de múltiplas camadas.....	36
3.6 Treinamento por retropropagação do erro.....	37
4 DINÂMICA DO SISTEMA.....	41
4.1 Funcionamento da planta.....	41
4.2 Controlador.....	42
5 AQUISIÇÃO DE DADOS.....	44
5.1 Determinação das regiões com ganhos diferentes.....	44
5.2 Ajuste de ganhos iniciais.....	46
5.3 Ajuste de ganhos região A.....	47
5.4 Ajuste de ganhos região B.....	49
5.5 Ajuste de ganhos região C.....	50
6 DEFINIÇÃO, TREINAMENTO E SIMULAÇÃO DA RNA.....	51

6.1 Definição de entradas e saídas	51
6.2 Organização dos dados para treinamento	52
6.3 Treinamento da RNA.....	56
6.4 Simulação da RNA.....	62
7 APLICAÇÃO DA RNA NO CLP	65
7.1 Normalização das entradas e saídas	65
7.2 Descrição matemática do neurônio	68
7.3 Aplicação dos cálculos no CLP	71
7.4 Simulação da RNA no CLP	74
8 TESTES E ANÁLISE DOS DADOS	78
8.1 Análise do escalonamento dos ganhos através da RNA.....	78
8.2 Análise da influência na corrente.....	80
8.3 Utilização do método SE	82
8.4 Comparação dos resultados	85
9 CONCLUSÃO E TRABALHOS FUTUROS.....	86
REFERÊNCIAS	88

1 INTRODUÇÃO

O segmento de máquinas de empacotamento para o setor alimentício é um campo da indústria que possui forte automação, composta por atuadores pneumáticos, servo motores, motores trifásicos, *encoder* etc. Uma máquina de empacotamento vertical, como ilustrado na Figura 1, pode fabricar pacotes do tipo almofada e, sendo assim, necessita de um controle preciso para atingir uma boa velocidade de produção e obter alta durabilidade.



Figura 1 - Máquina de empacotamento vertical

Fonte: Adaptado de www.fabrima.com/products/vffs/?lang=pt-br, 2016

Um conjunto fundamental em uma máquina de empacotamento vertical é o mordente de selagem horizontal, responsável por fazer a selagem horizontal de pacotes, controlando as variáveis de pressão, o torque de selagem, a temperatura e o tempo de selagem. O objetivo é obter uma selagem sem rugas, estrias e não deixar que o conteúdo do pacote entre em contato com o meio externo, portanto, na selagem não pode haver vazamentos.

Na Figura 2, pode-se observar um pacote do tipo almofada e sua selagem vertical e horizontal.

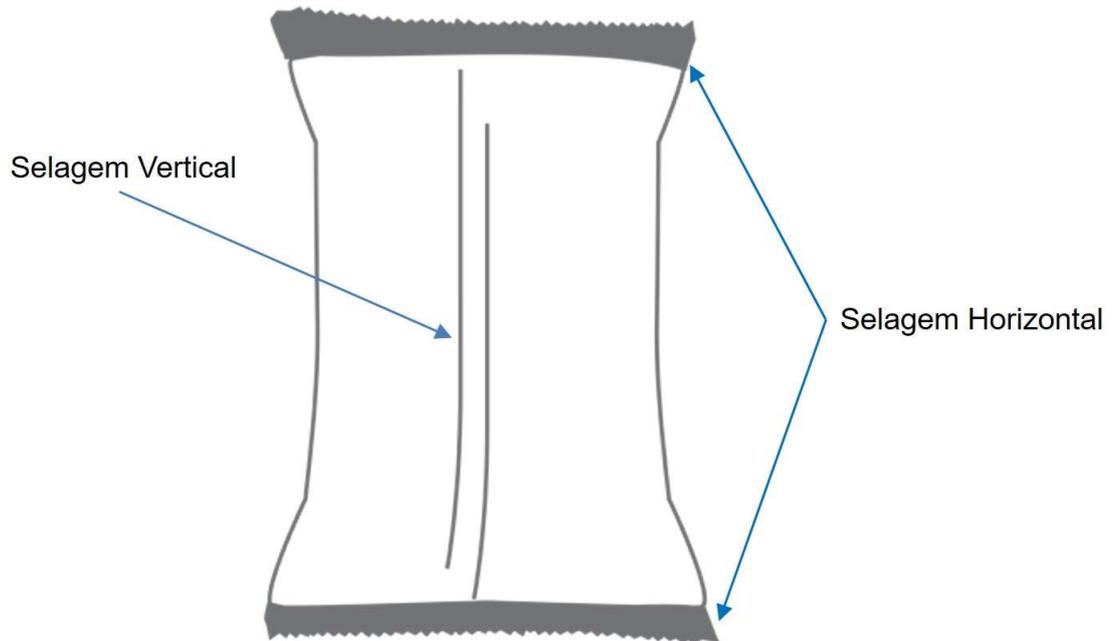


Figura 2 - Pacote almofada

Fonte: Adaptado de www.fabrima.com/products/vffs/?lang=pt-br, 2016

O conjunto de selagem analisado é acionado por um servo motor controlado por um servo *drive* e um CLP (Controlador Lógico Programável), todos de uso comum na indústria. Esse possui uma construção mecânica que torna o sistema não linear de forma a dificultar o controle do servo motor.

É comum na indústria de máquinas a utilização de controladores que utilizam um controle do tipo PID (Proporcional Integral Derivativo) para controlar servo motores e, em poucos casos, é realizada a modelagem do sistema para calcular os ganhos do controlador. Uma forma de não precisar calcular esses ganhos é fazer o procedimento de sintonização automática (*autotuning*), diretamente no *drive*. O sistema automático realiza um movimento de acordo com uma configuração feita pelo usuário, e em conformidade com a resposta do sistema, efetua o cálculo dos ganhos. Esse procedimento é muito eficaz na grande maioria dos casos, porém, em plantas onde a não linearidade é muito grande, se torna difícil encontrar um único conjunto de ganhos para toda a faixa de operação do sistema.

1.1 Justificativas

O conjunto de selagem, quando ajustado apenas pelo procedimento de sintonização automática, pode gerar um grande sobressinal de posicionamento de 2% ou mais, tanto no fechamento do conjunto de selagem, quanto na abertura.

O sobressinal na selagem implica em maior impacto entre os mordentes de selagem, o que pode acarretar maior elevação de corrente no impacto e menor durabilidade do sistema mecânico, que a cada ciclo irá absorver um grande esforço mecânico. Na Figura 3, pode-se observar a posição de fechamento do conjunto de selagem e, na Figura 4, os componentes mecânicos que podem ser avariados.

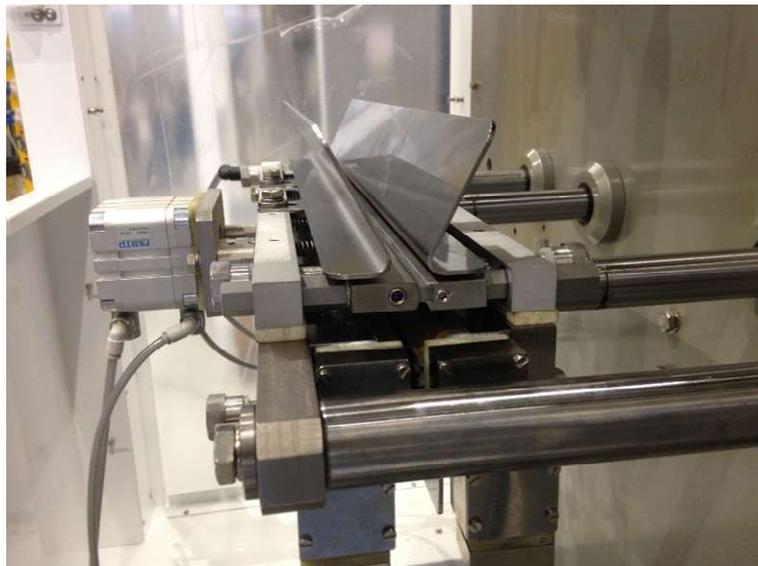


Figura 3 - Posição de selagem



Figura 4 - Vista traseira do conjunto

O sobressinal na abertura do conjunto de selagem, assim como no fechamento, impacta diretamente sobre a durabilidade das peças mecânicas, pois a parada na posição de abertura não é feita de forma suave, além de visivelmente demonstrar que o sistema possui um erro de posicionamento, o que poderá prejudicar no aspecto comercial da negociação das máquinas, quando um cliente observar tal problema. Se esse sobressinal for muito grande, o tempo de estabilização da posição também será maior, o que também poderá acarretar em um desempenho inferior da máquina, ou seja, será necessário que o mordente se estabilize na posição de abertura para poder fechar novamente. Se o mordente estabilizasse de forma mais rápida, poderia retornar à posição de fechamento antes, aumentando a produtividade da máquina. A Figura 5 ilustra a posição de abertura do sistema.



Figura 5 - Posição de abertura

1.2 Objetivo geral

O principal objetivo deste trabalho é melhorar o controle de posicionamento do servo motor utilizado no conjunto “mordente de selagem”, diminuindo o sobressinal de posicionamento na abertura do conjunto de selagem de 2% ou mais para 0,5%.

Para alcançar esse propósito, foi utilizada a técnica de escalonamento de ganhos, alterando os ganhos do controlador PID/PI em função da posição do servo motor e usada uma rede neural artificial capaz de reproduzir uma função que calcule os ganhos.

1.3 Objetivos específicos

Obter, de forma empírica, os ganhos do controlador PID/PI para diferentes faixas de operação do sistema, ou seja, para diferentes posições do servo motor, de forma que em cada faixa de operação os ganhos sejam satisfatórios para diminuir o sobressinal de posicionamento.

Obter uma rede neural artificial que seja capaz de reproduzir uma função que modifique os ganhos do controlador, configurando ganhos satisfatórios, com base em ajustes feitos para cada faixa de operação.

Implementar a rede neural artificial em um CLP para que a função possa ser gerada dentro da própria máquina, onde os ganhos poderão variar de acordo com a montagem de cada conjunto de selagem, criando uma nova função para cada máquina.

1.4 Organização da dissertação

Este trabalho apresenta um modelo de rede neural artificial com o intuito de escalonar os ganhos de um controlador do tipo PID/PI (Proporcional Integral Derivativo/Proporcional Integral), visando uma redução do sobressinal de posição na abertura do conjunto de 2% ou mais, para 0,5%.

O capítulo 2 apresenta uma revisão da literatura, com trabalhos já realizados nessa linha de pesquisa. No capítulo 3 é apresentada a teoria sobre redes neurais artificiais, iniciando com um breve histórico e focando na rede que será utilizada no trabalho: a rede multicamadas com treinamento por retropropagação do erro. O capítulo 4 apresenta o funcionamento dinâmico da planta e mostra o controlador e sua malha de controle. O capítulo 5 explica a forma como foi feita a aquisição de dados e a divisão das diferentes regiões de operação. No capítulo 6 é apresentado como a rede neural artificial foi configurada, treinada e simulada no programa computacional Matlab. O capítulo 7 demonstra como a rede neural artificial foi embarcada no CLP e a simulação da rede embarcada. O capítulo 8 trata dos resultados obtidos com a rede neural artificial implementada no CLP e é feita uma análise sobre o método SE, que utiliza expressões do tipo SE para modificar os ganhos do controlador, alternativo a rede neural artificial. No capítulo 9 é feita a conclusão e são apontadas as sugestões para trabalhos futuros. Posteriormente são especificadas as referências utilizadas.

2 REVISÃO DA LITERATURA

O servo *drive*, que controla o posicionamento do conjunto de selagem, utiliza um controlador Proporcional Integral Derivativo/Proporcional Integral (PID/PI) que tem sido estudado e utilizado na indústria por muitos anos. O PID é o tipo de controle padrão usado na maioria dos processos industriais devido a sua simplicidade, sua facilidade de ser ajustado sem a necessidade de conhecer todos os parâmetros da planta a ser controlada e sua versatilidade em ser utilizado nos mais variados processos. Para aplicações mais complexas, nas quais apenas o controle PID não consegue obter um controle satisfatório, uma série de variações tem sido estudadas para mesclar o controle PID com outros tipos de controle (VISIOLI, 2006).

Para a sintonia de um controlador PID, é necessário obter os valores dos ganhos proporcional, integral e derivativo, que podem ser calculados, caso possua o modelo matemático da planta. Caso não possua o modelo matemático da planta, ou, devido à complexidade do sistema, seja muito difícil de obtê-lo, como no conjunto de selagem, existem formas de sintonizar o controlador sem a necessidade de possuir todas as informações da planta, como o método de Ziegler-Nichols ou utilizando métodos de sintonia automática, como o *autotuning*. No *autotuning*, normalmente, é feito um movimento predeterminado e parametrizado pelo usuário do sistema, onde o controlador tenta identificar a planta e, por meio da identificação obtida, efetua os cálculos dos parâmetros do PID. Diversas formas de identificação automática têm sido estudadas (OGATA, 2003), (VISIOLI, 2006).

Segundo Bequette (2002), uma das formas mais utilizadas para se controlar processos não lineares através de um controlador PID é utilizar a técnica de *gain scheduling*. Trata-se de uma técnica de controle adaptativo, utilizada juntamente com um controlador PID, que consiste, por meio de uma variável auxiliar, em modificar os ganhos do controlador de acordo com a necessidade da planta (BEQUETTE, 2002).

Pode-se, por exemplo, utilizar uma variável que tenha relação direta com a não linearidade da planta, podendo, assim, alterar os ganhos de forma contínua ou mudá-los na medida em que a planta muda de faixa de operação. É comum que uma planta não linear tenha três ou mais faixas de operação, dependendo de sua dinâmica e, para cada faixa de operação, a variável auxiliar irá modificar os ganhos que foram previamente obtidos por *autotuning* ou outras técnicas de sintonia (BEQUETTE, 2002).

A Figura 6 ilustra o diagrama de uma planta, onde é utilizada a técnica de *gain scheduling*. Os ganhos do controlador serão alterados de acordo com a variável θ , que é obtida no bloco escalonamento de ganhos.

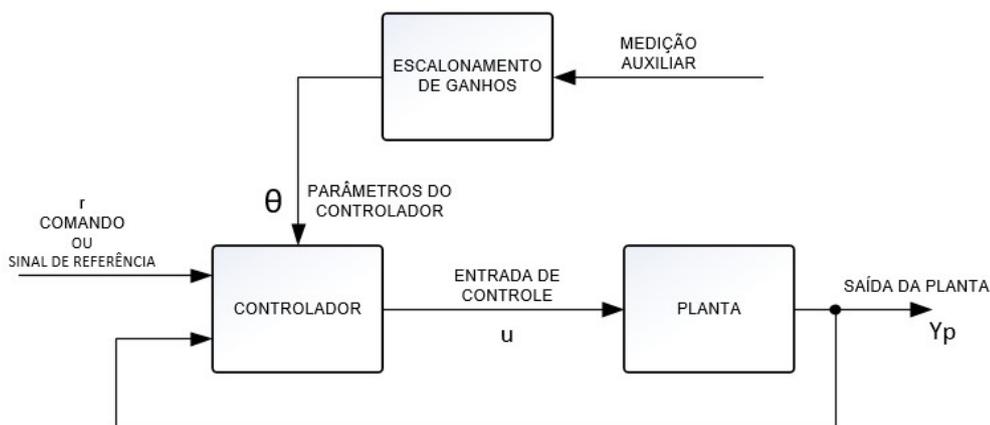


Figura 6 - Controlador Gain Scheduling

Fonte: Adaptado de BEQUETTE, 2002.

Kumar e Meenakshipriya (2012) publicaram um estudo de uma planta composta por dois tanques esféricos, controlada por um controlador PI (Proporcional Integral), que possuem comportamento não linear e que têm vantagens em comparação a tanques com comportamento linear.

Utilizando a lei da conservação de massas para fluidos e dados físicos da planta, foi feita a modelagem matemática da planta e simulada no programa computacional Simulink para constatar a não linearidade da planta, conforme a Figura 7.

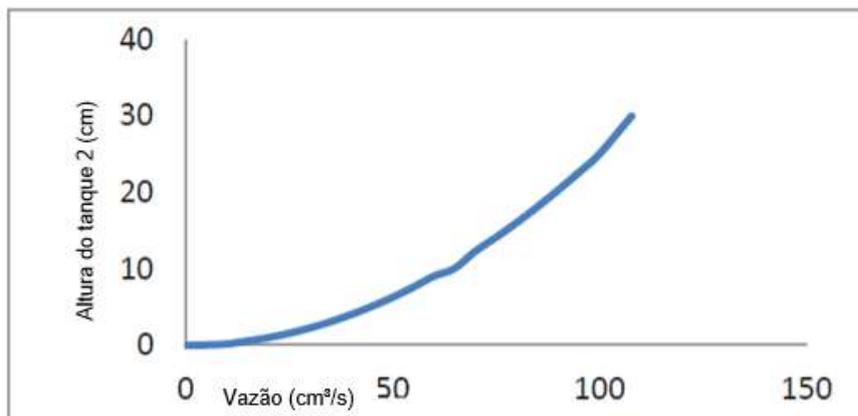


Figura 7 - Características da entrada e saída dos tanques

Fonte: Adaptado de KUMAR e MEENAKSHIPRIYA, 2012

De acordo com os dados obtidos na simulação, a planta foi dividida em três faixas de operação (5, 15 e 25 cm de altura), onde foi feito *autotuning* em cada faixa e encontrados diferentes valores de ganho para o controlador PI. Porém, os valores dos ganhos do controlador foram alterados através de uma relação direta com características da planta. Os resultados obtidos mostraram-se satisfatórios em diversos testes, em diferentes faixas de operação, evidenciando a eficácia do *gain scheduling* para plantas não lineares.

Oliveira *et al.* (1997) apresentaram um estudo experimental com uma planta não linear de tanques interativos, que possuem um acoplamento manual entre os tanques, utilizando uma técnica de *neural gain scheduling* para escalonar os diferentes controladores sintonizados para cada faixa de operação. Nesse estudo, a planta foi dividida em seis faixas de operação e para cada faixa foram ajustados ganhos pelo método experimental de Ziegler Nichols, partindo de um controlador PI.

A rede neural artificial utilizada foi do tipo *feedforward* multicamadas e foi treinada pelo algoritmo de retropropagação de erro, e cada um dos controladores sintonizados para cada faixa de operação foram utilizados no treinamento da rede. A rede mostrou-se eficaz quanto à substituição dos controladores por um único controlador neural.

Nunes *et al.* (2007) propuseram um PID LMS (*Last Mean Square*) tuning baseado em uma rede neural artificial com estrutura de *perceptron*, no qual o erro do sinal é pré-processado. A combinação linear representa os ganhos da estrutura PID e a função de ativação representa o *drive* para montar um controlador com ganhos ajustáveis através de uma rede neural artificial.

Esse método consiste na utilização de equações que minimizem o erro e correlacionem os ganhos entre si, chamados de WP, WI e WD, atualizados no controlador em um intervalo de tempo Δt , como se pode observar na Figura 8. O controle proposto foi testado com um motor DC e implementado em um CLP. Os resultados obtidos foram considerados satisfatórios.

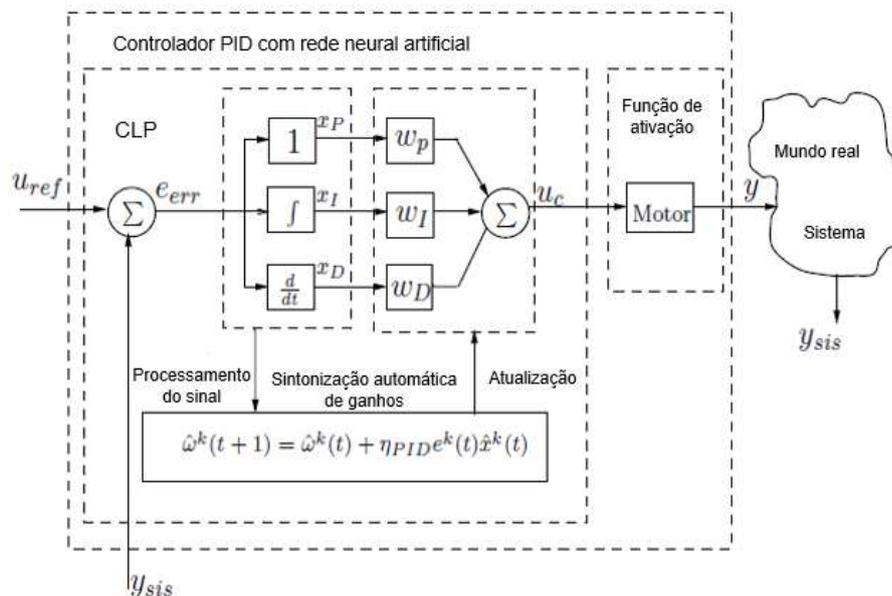


Figura 8 - LMS unidade de ajuste

Fonte: Adaptado de NUNES *et al.*, (2007)

Patel *et al.* (2015) publicaram um estudo sobre a utilização de um controlador neural PID multicamadas, no qual abordaram as diferentes formas de se utilizar, por exemplo, em série, em paralelo e com sintonização automática.

No estudo de Patel *et al.* (2015), foi utilizada uma planta não linear de levitação magnética, na qual os ganhos KP, KI e KD foram ajustados através da sintonização automática, embora também pudessem ser ajustados pelo método de Ziegler Nichols, em uma rede neural artificial com uma camada de entrada, uma camada de saída e uma camada escondida, como pode-se observar no diagrama da Figura 9.

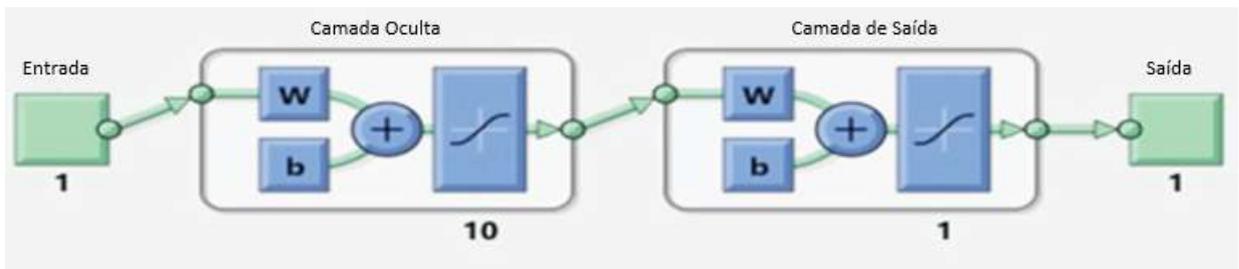


Figura 9 - Janela de treinamento

Fonte: Adaptado de PATEL *et al.* (2015)

Segundo o estudo de Patel *et al.* (2015), é sugerida uma forma de *neuro gain scheduling*, porém, não é muito aprofundada devido à dificuldade de implementação em tempo real. A Figura 10 ilustra o diagrama proposto para *neuro gain scheduling*.

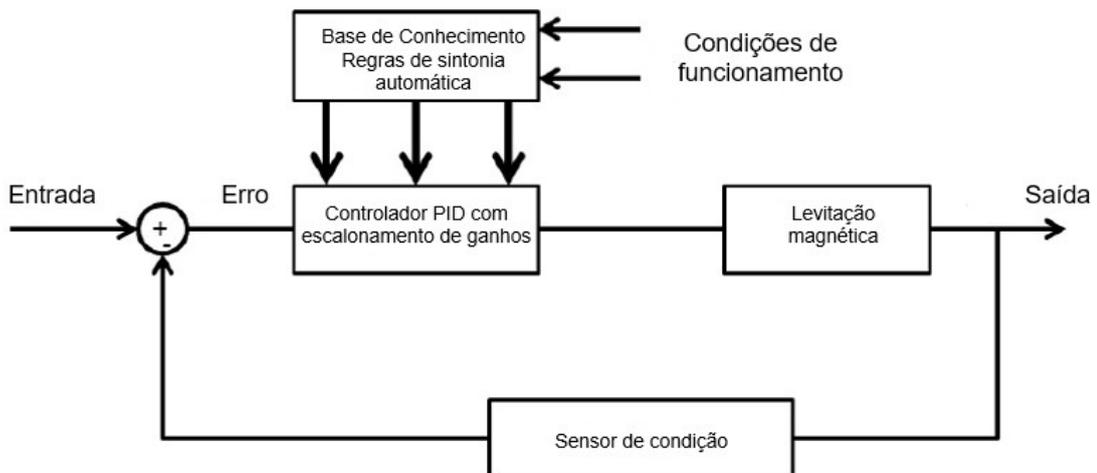


Figura 10 - Diagrama de bloco – Gain Scheduling

Fonte: Adaptado de PATEL *et al.*, (2015)

Andrade *et al.* (2015) publicaram um estudo comparativo entre um controlador PI clássico e um controlador inteligente baseado em redes neurais, aplicado em uma planta didática de sistema de vazão. No estudo, o controlador PI clássico foi sintonizado segundo o método de Chien-Hrones-Reswick para um sobressinal de 0%.

Para o sistema inteligente baseado em redes neurais, foi utilizada uma rede neural artificial com duas entradas, sendo uma o erro e a outra a vazão real, dez neurônios na camada oculta e uma saída de controle.

Andrade *et al.* (2015) utilizaram 24 curvas, obtidas através do controlador PI clássico, no treinamento e validação da RNA, sendo 18 no treinamento e seis da validação.

Para a validação prática, foram escolhidos quatro pontos de operação do sistema e feito o teste e comparação com o controlador PI clássico, onde foi demonstrado que o sistema de controle inteligente baseado em redes neurais artificiais obteve um desempenho superior ao do controlador PI clássico, obtendo um sobressinal menor nas regiões definidas, bem como um menor erro em regime, demonstrando a eficácia do controlador PI neural (ANDRADE *et al.*, 2015).

3 REDES NEURAIS ARTIFICIAIS

Neste capítulo, serão abordados os princípios teóricos sobre Redes Neurais Artificiais (RNA). Será apresentado, com mais detalhes, a RNA multicamadas com treinamento por retropropagação de erro.

3.1 Introdução

De acordo com Haykin (2001), o estudo de redes neurais artificiais (RNA) tem sido motivado pelo reconhecimento da grande capacidade de processamento de informações do cérebro humano, que é complexo, não linear e processa informações paralelamente, diferente de um computador digital convencional.

Na sua forma mais geral, uma rede neural é uma máquina que é projetada para modelar a maneira que o cérebro realiza uma tarefa particular ou função de interesse; Ou seja, pode-se dizer que uma RNA é uma tentativa de simular a forma como o cérebro realiza uma tarefa; a rede é normalmente implementada utilizando-se componentes eletrônicos ou é simulada por programação em um computador digital. (HAYKIN, 2001, p.28)

Segundo Haykin (2001), pode-se observar o sistema nervoso humano em três estágios, conforme a ilustração da Figura 11, onde o cérebro é representado pelo bloco central “rede neural”. No bloco da esquerda, “receptores” convertem os estímulos em impulsos elétricos e enviam para o cérebro. À direita, pode-se observar o bloco “atuadores”, que convertem os impulsos elétricos gerados pelo cérebro em uma forma de saída. As setas para a direita representam a transmissão da informação para o estágio seguinte e as para a esquerda representam a realimentação de cada estágio.

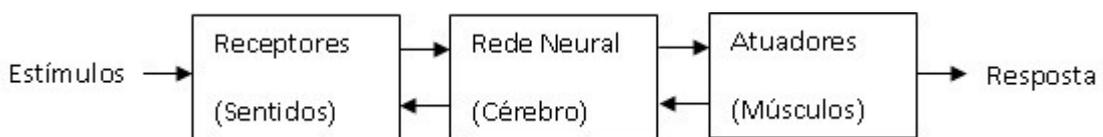


Figura 11 - Representação em blocos do sistema nervoso

Fonte: Haykin, 2001

3.2 Histórico

Segundo Hagan *et al.* (1996), a visão moderna de uma rede neural artificial começou em 1940 com o trabalho de Warren McCulloch e Walter Pitts. De acordo com o estudo, uma rede neural artificial, a princípio, poderia resolver qualquer problema aritmético ou lógico. Esse trabalho deu origem a outras pesquisas nesta área.

A primeira aplicação prática de uma rede neural artificial surgiu no final dos anos 50 com a invenção da rede *perceptron* e regras de aprendizado associativo criadas por Frank Rosenblatt. Ele e seus colegas demonstraram a eficiência da rede em reconhecer padrões e, com o sucesso obtido, estimularam a pesquisa nesta área.

Porém, o *perceptron* de Rosenblatt não era capaz de resolver problemas linearmente separáveis e sua deficiência foi evidenciada na publicação de Marvin Minsky e Seymour Papert. Muitas pessoas acabaram sendo influenciadas por Minsky e Papert, acreditando que esta linha de pesquisa não era o caminho e, tal fato, combinado com a falta de recursos computacionais capazes de auxiliar as pesquisas, fizeram com que diversos pesquisadores abandonassem as pesquisas nesse campo.

Ainda segundo Hagan *et al.* (1996), dois conceitos novos que foram apresentados nos anos 80 fizeram as pesquisas nessa área serem retomadas. O primeiro foi o uso de ferramentas estatísticas para explicar uma certa classe de redes recorrentes que podem ser usadas como memória associativa. O segundo foi o algoritmo de retropropagação do erro para o treinamento de uma RNA *perceptron* de multicamadas, que foi a resposta às limitações do neurônio *perceptron*, citadas por Minsky e Papert.

Esses conceitos mudaram o cenário das redes neurais artificiais, aumentando ainda mais o interesse nesta teoria. Desde então, diversos estudos e livros foram publicados e cada vez mais aplicações têm surgido nas mais diversas áreas. O desenvolvimento da tecnologia de redes neurais continua seu avanço, tendo em vista que ainda se sabe muito pouco sobre o funcionamento do cérebro humano.

3.3 O neurônio - *Perceptron* de camada única

Segundo Haykin (2001), um neurônio artificial é uma unidade de processamento de informações básicas e fundamental para a operação de uma rede neural artificial. No modelo apresentado na Figura 12 tem-se:

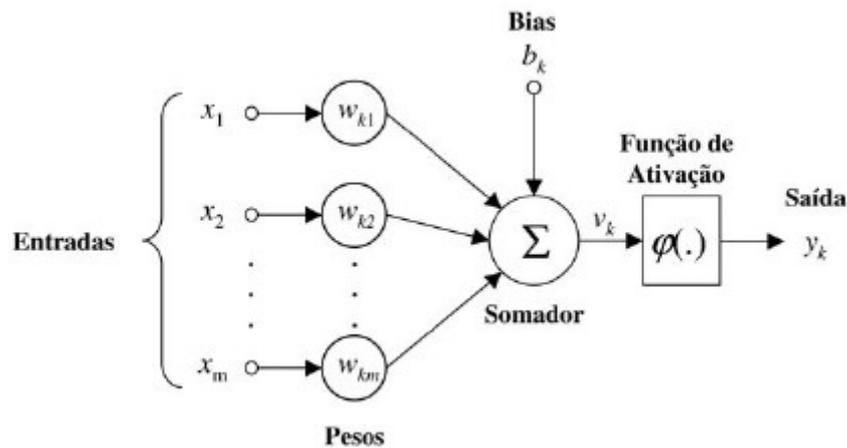


Figura 12 - Modelo matemático de um neurônio

Fonte: Haykin, 2001

- x_1, x_2, \dots, x_m são as entradas do sistema, que serão utilizadas no controle e no treinamento da rede - é por onde o neurônio recebe a informação.
- $w_{k1}, w_{k2}, \dots, w_{km}$ são os pesos sinápticos, os quais são multiplicados pelo valor da entrada, ponderando cada uma das entradas. O intuito do treinamento da rede é encontrar o valor ideal para cada peso sináptico para que a saída seja a desejada.
- O bias b_k é uma entrada externa que tem como função aumentar ou diminuir o valor no somador. Bias é um valor fixo que pode variar de acordo com a aplicação do neurônio e pode ser visto como uma entrada x_0 multiplicada por um peso w_0 .
- O somador tem a função de somar as entradas já ponderadas dos seus respectivos pesos.

- A função de ativação (φ), também referida como função restritiva, restringe a amplitude da saída de um neurônio. Existem diferentes tipos de função de ativação que irão variar diretamente com a aplicação de cada neurônio.

Pode-se modelar matematicamente um neurônio de acordo com as equações a seguir:

$$u_k = \sum_{j=1}^m W_{kj} \cdot X_j \quad (1)$$

$$v_k = (u_k + b_k) \quad (2)$$

$$Y_k = \varphi(v_k) \quad (3)$$

Onde u_k é a soma da multiplicação das entradas pelos pesos sinápticos, b_k é o bias, v_k é o campo local induzido do neurônio, (φ) é a função de ativação e Y_k é a saída do neurônio.

3.4 Funções de ativação

Existem diversos tipos de funções de ativação, que são utilizadas de acordo com a aplicação do neurônio, podendo uma rede conter um ou mais tipos diferentes de funções de ativação combinadas. A seguir, são apresentados três tipos de funções de ativação bastante utilizadas.

3.4.1 Função limiar

Este tipo de função de ativação faz com que a saída do neurônio $\varphi(\mathcal{V})$ assumo o valor 1 se o campo local induzido \mathcal{V} for não negativo e 0, caso seja negativo, conforme a equação 4 e na Figura 13.

$$\varphi(\mathcal{V}) = \begin{cases} 1 & \text{se } \mathcal{V} \geq 0 \\ 0 & \text{se } \mathcal{V} < 0 \end{cases} \quad (4)$$

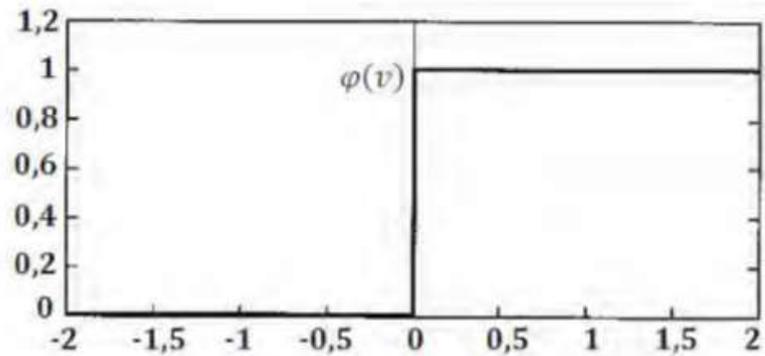


Figura 13 - Função de limiar

Fonte: Adaptado de Haykin 2001

3.4.2 Função linear por partes

Esta função de ativação pode ser vista como uma aproximação de um amplificador não linear. A Figura 14 ilustra sua forma gráfica e, na equação 5, pode-se observar seu modelo matemático.

$$\varphi(v) = \begin{cases} 1, & v \geq +0,5 \\ v, & +0,5 > v > -0,5 \\ 0, & v \leq -0,5 \end{cases} \quad (5)$$

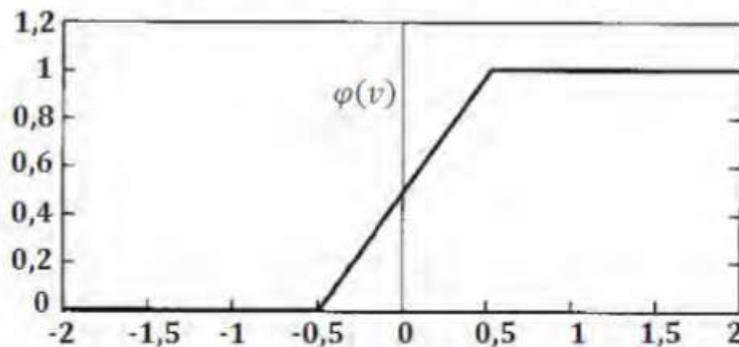


Figura 14 - Função linear por partes

Fonte: Adaptado de Haykin, 2001

3.4.3 Função sigmoide

É uma forma bastante comum na construção de redes neurais artificiais. Possui formato de “S”, como se pode observar na Figura 15. Por definição, é uma função estritamente crescente, que exibe um balanceamento adequado entre comportamento linear e não linear. Um exemplo de função sigmoide é a função logística matematicamente representada pela equação 6, onde a é o parâmetro de inclinação da função sigmoide. É possível variar a inclinação da função sigmoide variando esse parâmetro.

$$\varphi(\mathcal{V}) = \frac{1}{1 + \exp(-a\mathcal{V})} \quad (6)$$

As funções de ativação definidas nas equações 4, 5 e 6 se estendem no intervalo de 0 a +1, porém, elas também podem se estender de -1 a +1, tornando-se antissimétricas em relação à origem. Especificamente, a função de limiar da equação 4 agora é definida pela equação 7 e é chamada de função sinal.

$$\varphi(\mathcal{V}) = \begin{cases} 1 & \text{se } \mathcal{V} > 0 \\ 0 & \text{se } \mathcal{V} = 0 \\ -1 & \text{se } \mathcal{V} < 0 \end{cases} \quad (7)$$

Para a forma correspondente de uma função sigmoide, pode-se utilizar a função tangente hiperbólica, descrita pela equação 8.

$$\varphi(\mathcal{V}) = \tanh(\mathcal{V}) \quad (8)$$

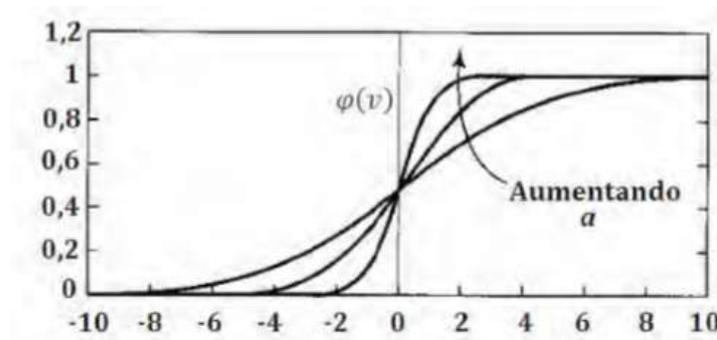


Figura 15 - Função sigmoide

Fonte: Adaptado de Haykin, 2001

3.5 RNA do tipo *perceptron* de múltiplas camadas

O *Perceptron* de Múltiplas Camadas (MLP, *Multilayer Perceptron*) tem sido aplicado com sucesso em problemas difíceis. Além da camada de entrada e de saída, possui uma ou mais camadas ocultas, que irão variar de acordo com o tipo de problema a ser solucionado.

Na Figura 16, pode-se observar uma rede neural artificial do tipo *perceptron* de múltiplas camadas, que é composta por uma camada de entrada, uma camada de saída e duas camadas ocultas. Essa rede é chamada de “totalmente conectada”, pois cada neurônio em qualquer camada está conectado a cada um dos neurônios da camada anterior. O sinal de entrada (estímulo) é propagado para frente, da esquerda para a direita, pela rede, por suas conexões sinápticas.

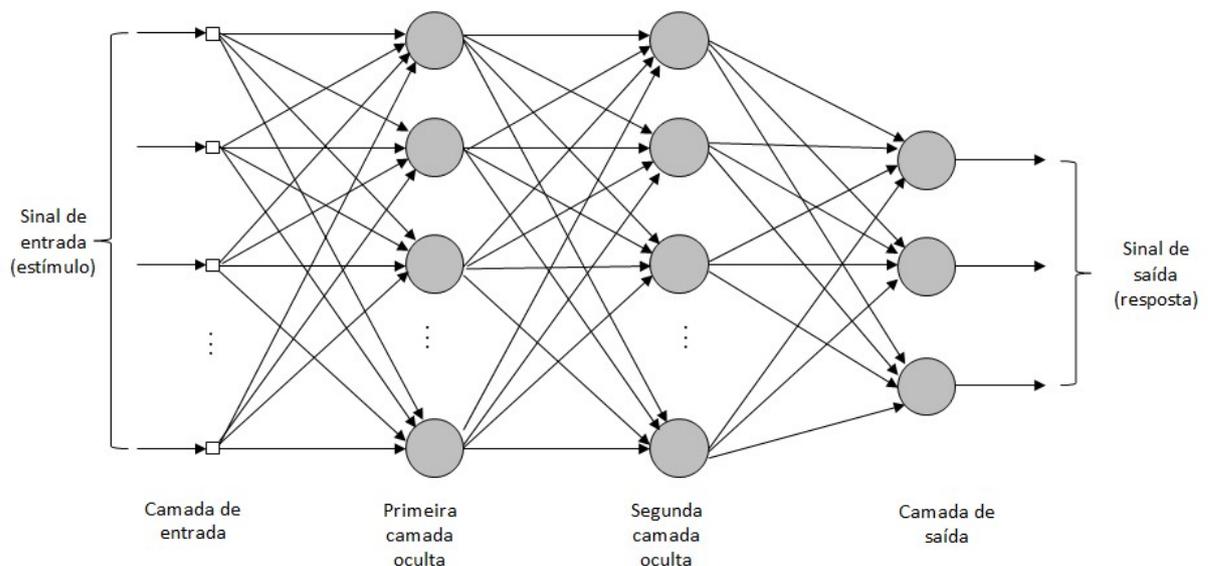


Figura 16 - Arquitetura de uma MLP com duas camadas ocultas

Fonte: Haykin, 2001

3.6 Treinamento por retropropagação do erro

O treinamento dessa RNA, ou, conforme visto anteriormente, o ajuste dos pesos sinápticos, é feito pelo algoritmo de retropropagação de erro (*error back-propagation*). Esse tipo de treinamento é composto, basicamente, por dois passos:

1 - A propagação, que consiste em aplicar um vetor de entrada à camada de entrada, sendo propagado pela rede até gerar uma saída. A saída ideal é subtraída da saída real, gerando um sinal de erro. Os pesos sinápticos se mantêm fixos nesta etapa.

2 - A retropropagação, que consiste em propagar o sinal de erro no sentido contrário das conexões sinápticas, ajustando os pesos sinápticos para que a saída real seja mais próxima da saída desejada.

O algoritmo para treinar os neurônios em uma rede multicamadas é um tanto complexo e difícil de visualizar. Para exemplificar matematicamente o algoritmo de treinamento, deve-se observar se o neurônio em questão está na camada oculta ou se está na camada de saída.

Para fazer a correção dos pesos, é necessário que a regra delta, apresentada na Figura 17, seja obedecida.

$$\begin{pmatrix} \text{Correção} \\ \text{de peso} \\ \Delta w_{ji}(n) \end{pmatrix} = \begin{pmatrix} \text{Parâmetro da} \\ \text{taxa de aprendizagem} \\ \eta \end{pmatrix} \cdot \begin{pmatrix} \text{Gradiente} \\ \text{local} \\ \delta_j(n) \end{pmatrix} \cdot \begin{pmatrix} \text{sinal de entrada} \\ \text{do neurônio } j \\ y_j(n) \end{pmatrix}$$

Figura 17 - Regra delta de correção dos pesos sinápticos

Fonte: Haykin, 2001

Verifica-se na Figura 18 um neurônio localizado na camada de saída. Neste caso, pode-se calcular o erro de forma mais fácil, utilizando a equação 9, onde $e_j(n)$ é o erro, $d_j(n)$ é a saída desejada e $y_j(n)$ é a saída real com os pesos fixos.

$$e_j(n) = d_j(n) - y_j(n) \quad (9)$$

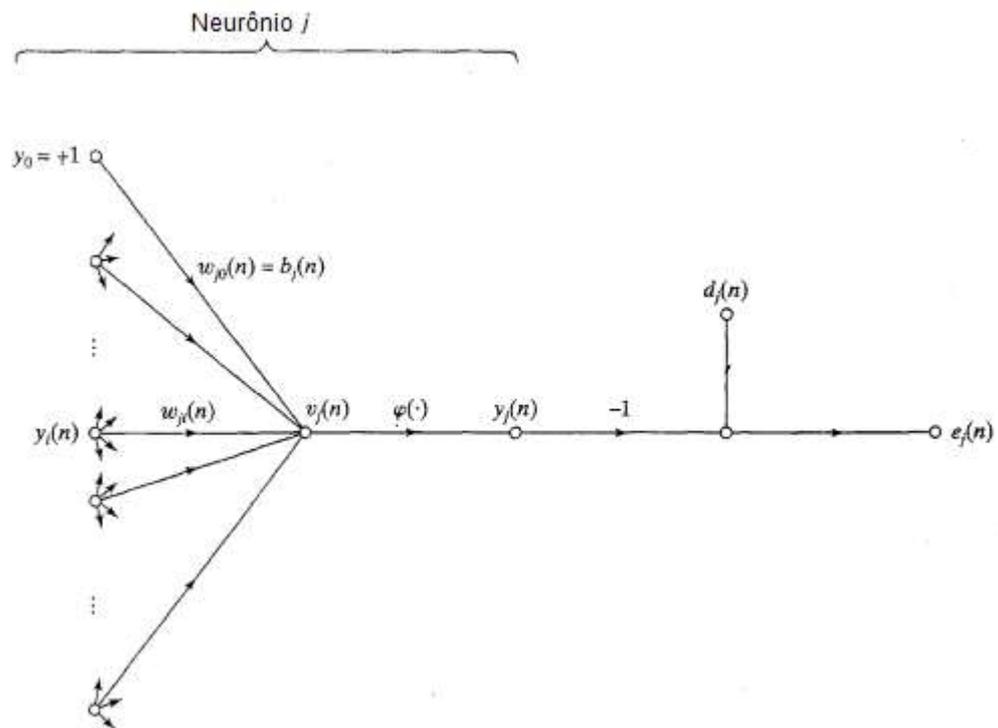


Figura 18 - Grafo de fluxo de sinal ressaltando os detalhes do neurônio de saída j

Fonte: Haykin, 2001

O gradiente local $\delta_j(n)$ aponta para as modificações necessárias nos pesos sinápticos. De acordo com a equação 10, é o produto do sinal de erro $e_j(n)$ pela derivada $\varphi'_j(v_j(n))$ da função de ativação definida para o neurônio.

$$\delta_j(n) = e_j(n)\varphi'_j(v_j(n)) \quad (10)$$

O parâmetro da taxa de aprendizagem η regula a variação dos pesos sinápticos da rede de uma interação para a outra. Quanto menor a taxa de aprendizagem, menor será a variação dos pesos, o que acarretará em uma taxa de treinamento lento. Porém, se a taxa de aprendizagem for muito grande, poderá tornar a rede oscilatória em função da grande variação entre os pesos sinápticos. A taxa de aprendizagem é escolhida de acordo com o projeto.

Caso o neurônio esteja na camada oculta, conforme o neurônio j da Figura 19, o erro deve ser calculado de forma recursiva, ou seja, em função dos sinais de erro de todos os neurônios ao qual o neurônio oculto está conectado.

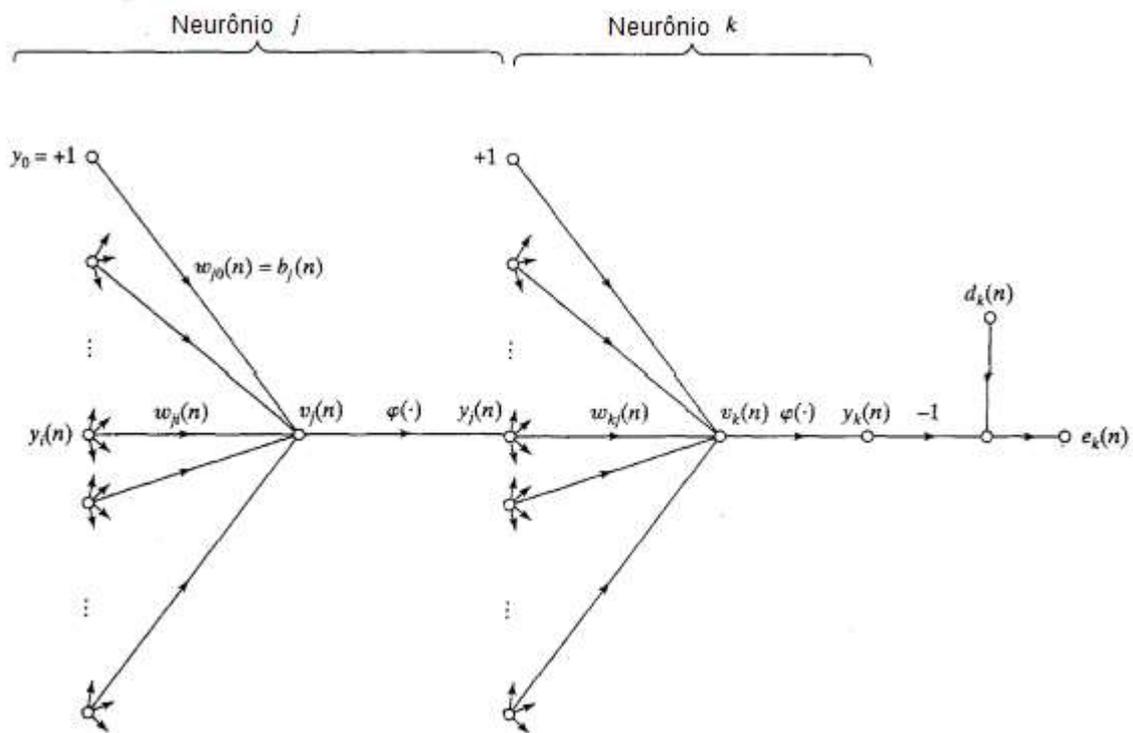


Figura 19 - Grafo de fluxo de sinal ressaltando os detalhes do neurônio de saída k conectado ao neurônio oculto j

Fonte: Haykin, 2001

Calcula-se a energia total do erro $\mathcal{E}(n)$ pela equação 11.

$$\mathcal{E}(n) = \frac{1}{2} \sum_{k \in C} e_k^2(n) \quad (11)$$

A fórmula de retropropagação para o gradiente local $\delta_j(n)$ de acordo com a equação 12, onde o gradiente local é igual ao produto da derivada associada $\varphi'_j(v_j(n))$ pela soma ponderada dos δ_s calculados para os neurônios na próxima camada, é obtida, seja ela oculta ou de saída que estejam conectadas ao neurônio j .

$$\delta_j(n) = \varphi'_j(v_j(n)) \sum_k \delta_k(n) W_{kj}(n) \quad (12)$$

Utilizando os algoritmos citados, deve-se apresentar as amostras à rede neural artificial, o que pode ser realizado de duas formas: modo sequencial, ou seja, é fornecida uma amostra à rede e, após os passos de propagação e retropropagação, os pesos são alterados e, assim, sucessivamente, a cada nova amostra de dados, ou por lote, apresentando um conjunto de dados e, somente após todos os dados serem apresentados à rede, os pesos são ajustados.

A quantidade de épocas, ou seja, o número de vezes que a amostra é apresentada à rede, é vista como um dado importante no processo de treinamento, pois irá interferir diretamente no tempo de treinamento da rede e em sua precisão. A forma como os dados são apresentados em cada época também interfere no desempenho da rede.

Não existe um critério bem definido para o momento ideal a interromper o treinamento da rede. Isso pode variar de acordo com o projeto e o tempo de treinamento disponível. Em geral, quanto mais tempo e mais amostras forem apresentadas no treinamento, mais precisa a rede se tornará.

A taxa de variação do erro quadrático médio pode ser um critério de parada. O treinamento da rede pode ser interrompido quando a taxa for considerada suficientemente pequena, o que significa que o seu valor pode estar entre 1 e 0,01, variando de acordo com a rede neural artificial. No entanto, taxas muito pequenas, por exemplo 0,01, podem resultar em um encerramento do treinamento de forma precipitada.

O gradiente também pode ser utilizado como critério de parada. Pode-se considerar que uma rede convergiu quando a norma euclidiana do vetor gradiente atingir um valor mínimo próximo de zero. Porém, esse método pode custar um tempo de treinamento maior para atingir o valor esperado.

4 DINÂMICA DO SISTEMA

Neste capítulo, será apresentado o funcionamento dinâmico da planta e do controlador responsável pelo posicionamento do servo motor.

4.1 Funcionamento da planta

O movimento do sistema é iniciado com o mordente na posição de abertura, onde é fisicamente limitada em 120 mm, como ilustrado na Figura 20.

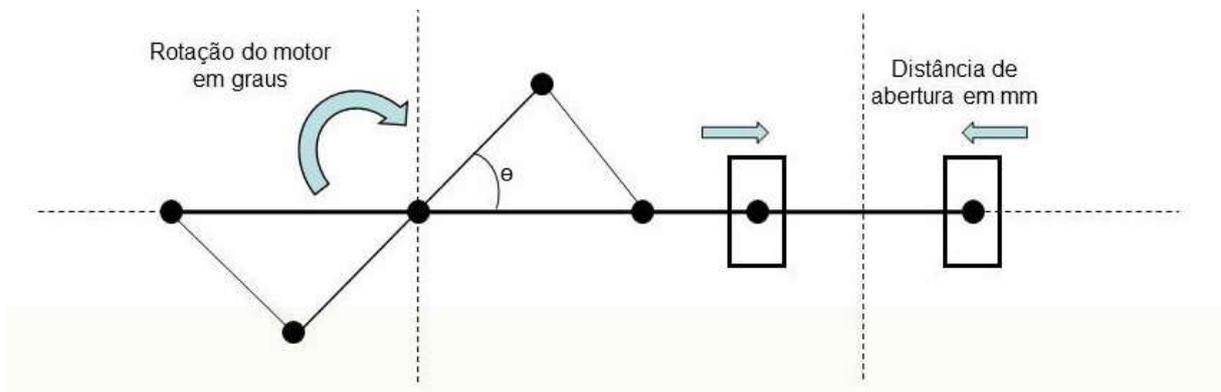


Figura 20 - Esquema do mordente – Posição de abertura

O servo motor gira no sentido horário, conforme ilustrado na Figura 20, até que os mordentes se toquem na posição de fechamento, ilustrado na Figura 21, onde é feita a selagem do pacote. Após a selagem estar completa, o servo motor gira no sentido anti-horário, conforme ilustrado na Figura 21, até a posição de abertura. A posição de fechamento é mecanicamente limitada e é denominada como posição 0 mm (origem).

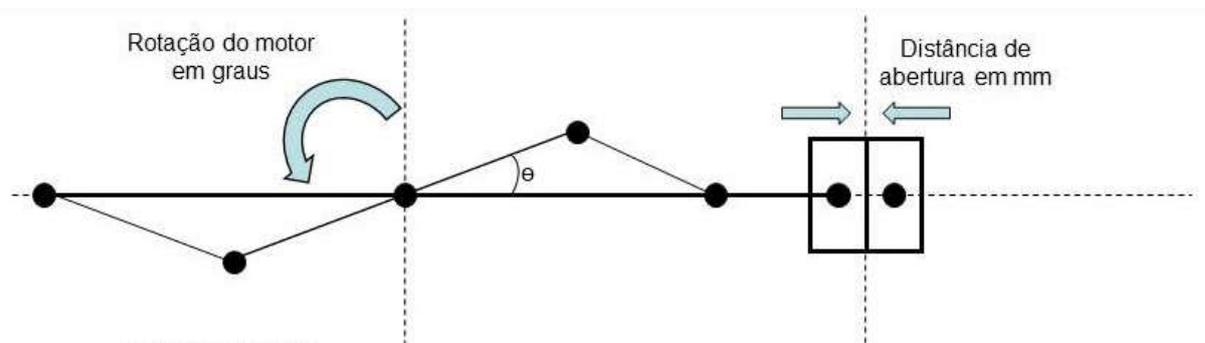


Figura 21 - Esquema do mordente – Posição de fechamento

4.2 Controlador

Para controlar o movimento do servo motor, é utilizado o servo *drive* Kinetix 300 Ethernet/IP, fabricado pela Allen Bradley, que controla o servo motor através de *index* de posicionamento. *Index* são conjuntos de parâmetros organizados de forma sequencial, como posição, velocidade e aceleração pré-determinados, que são habilitados de acordo com um comando do CLP ou pela forma que são programados no *drive*.

No caso do mordente de selagem, são utilizados três *Index* de posicionamento:

1º - *Index* de posição de abertura – Este *index* contém a posição de abertura do mordente, velocidade e aceleração do movimento. É executado de forma automática após ser executado o *index* 3, que será explicado abaixo.

2º - *Index* de posição de fechamento – Este *index* contém a posição de fechamento, velocidade e aceleração do movimento. Ele aguarda o comando do CLP para ser executado - é o início do movimento.

3º - *Index* de torque – Este *index* não contém posição, mas sim um torque e uma direção a ser aplicada que, neste caso, é o torque de selagem na direção de fechamento. Ele é executado de forma automática após ser executado o *index* 2.

O servo *drive* controla o servo motor através de uma malha de controle composta por um controle do tipo PID e um controle do tipo PI, apresentado na Figura 22.

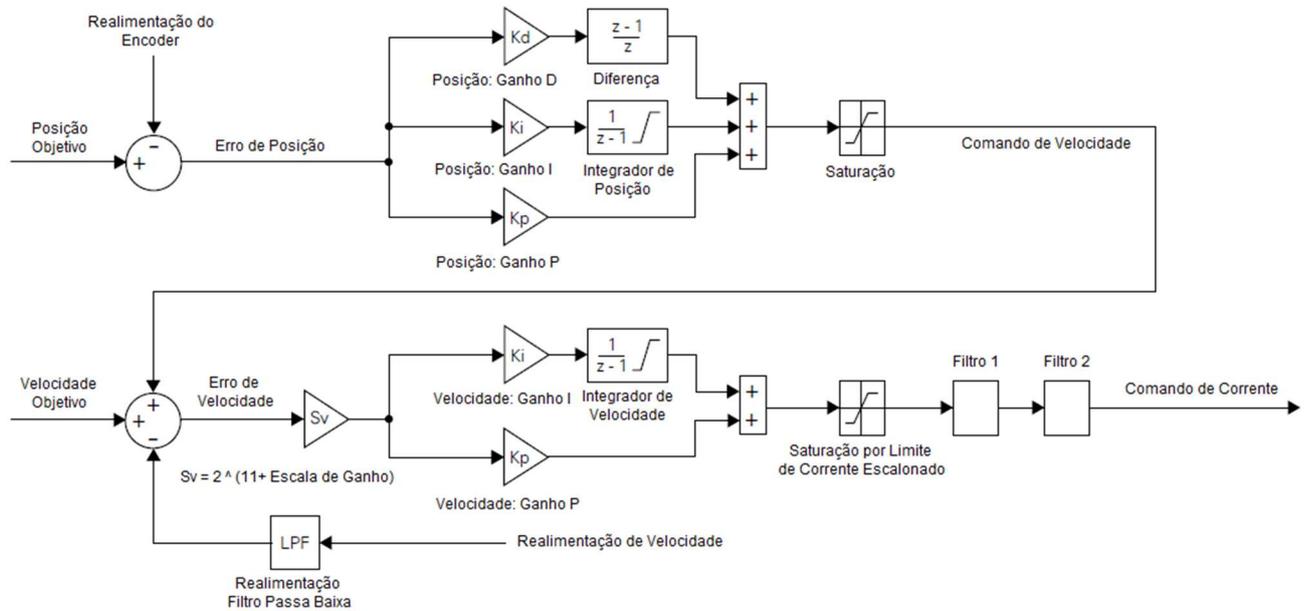


Figura 22 - Diagrama de malha do servo

Fonte: Adaptado de Allen Bradley, 2017

Para melhorar o desempenho do controle do servo motor, os ganhos do controlador (K_d , K_i e K_p) de posição e (K_i e K_p) de velocidade serão escalonados de acordo com a rede neural artificial implementada no CLP.

5 AQUISIÇÃO DE DADOS

Neste capítulo, será apresentado o procedimento de aquisição de dados para a elaboração da rede neural artificial.

5.1 Determinação das regiões com ganhos diferentes

Para que o escalonamento de ganhos seja feito em diferentes faixas de operação do sistema, é necessário identificar essas regiões. Na planta em estudo, foi observado que a variação da posição do servo motor não é linear com o deslocamento da massa do conjunto.

Para observar essa não linearidade, foi feito um experimento incrementando um valor em unidades do servo motor, que é proporcional a graus de rotação, e medido o deslocamento linear da região onde se encontra o mordente. A origem da medição foi com o mordente fechado, onde foi considerado como deslocamento zero. Foi obtido um gráfico não linear, conforme ilustrado no Gráfico 1.

Deslocamento do Mordente

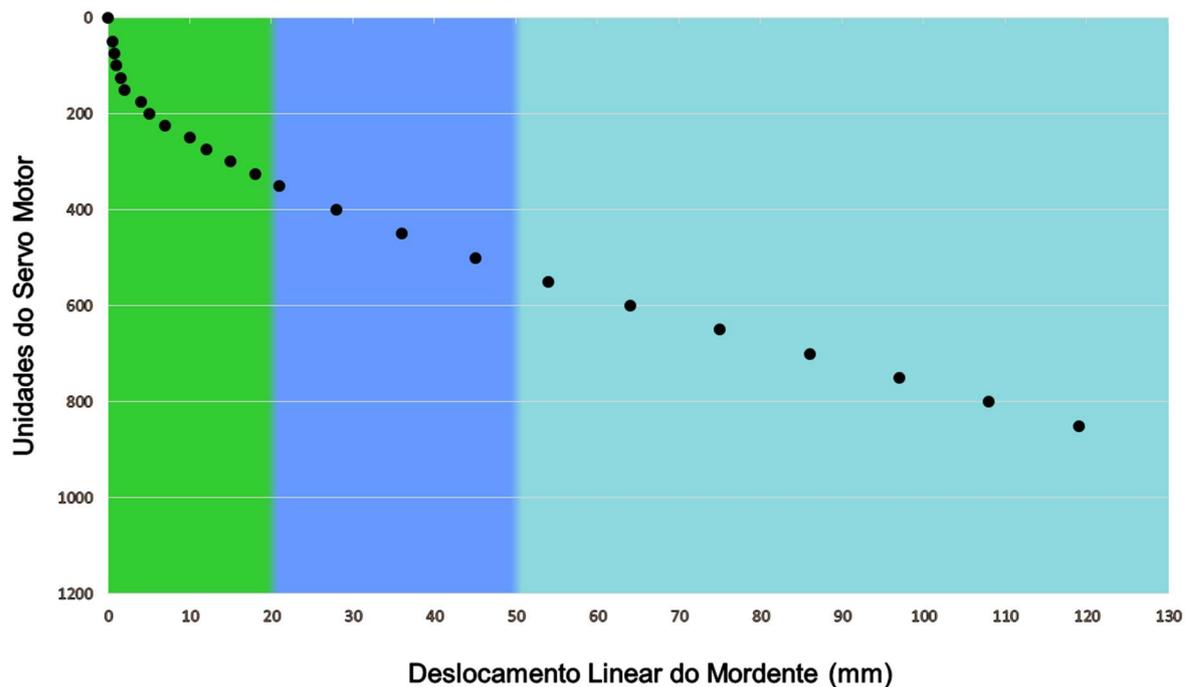


Gráfico 1 - Deslocamento do mordente

De acordo com o Gráfico 1, pode-se observar que o movimento se inicia de forma não linear e, posteriormente, se torna linear. Dessa forma, o movimento foi dividido em três regiões (A, B e C), que serão consideradas diferentes faixas de operação do controlador, ou seja, cada região terá seus ganhos ajustados para quando o posicionamento do servo motor estiver dentro deste limite, considerando a região A, quando o servo motor estiver entre a posição 0 e a posição 20 (mm), a região B, quando o servo motor estiver entre a posição 20 e 50 (mm) e a região C, quando o servo motor estiver entre a posição 50 e 120 (mm).

5.2 Ajuste de ganhos iniciais

A condição de ajuste inicial do controlador foi feita de forma empírica, partindo do ajuste automático do servo motor, até obter ganhos que demonstrassem um controle aceitável. Pode-se observar pela Figura 23 a curva de posição do servo motor durante o movimento de abertura e fechamento. Onde pode ser observado o sobressinal de posicionamento na posição de abertura de 2%, a posição de fechamento é o trecho retilíneo, localizado na parte superior da curva.

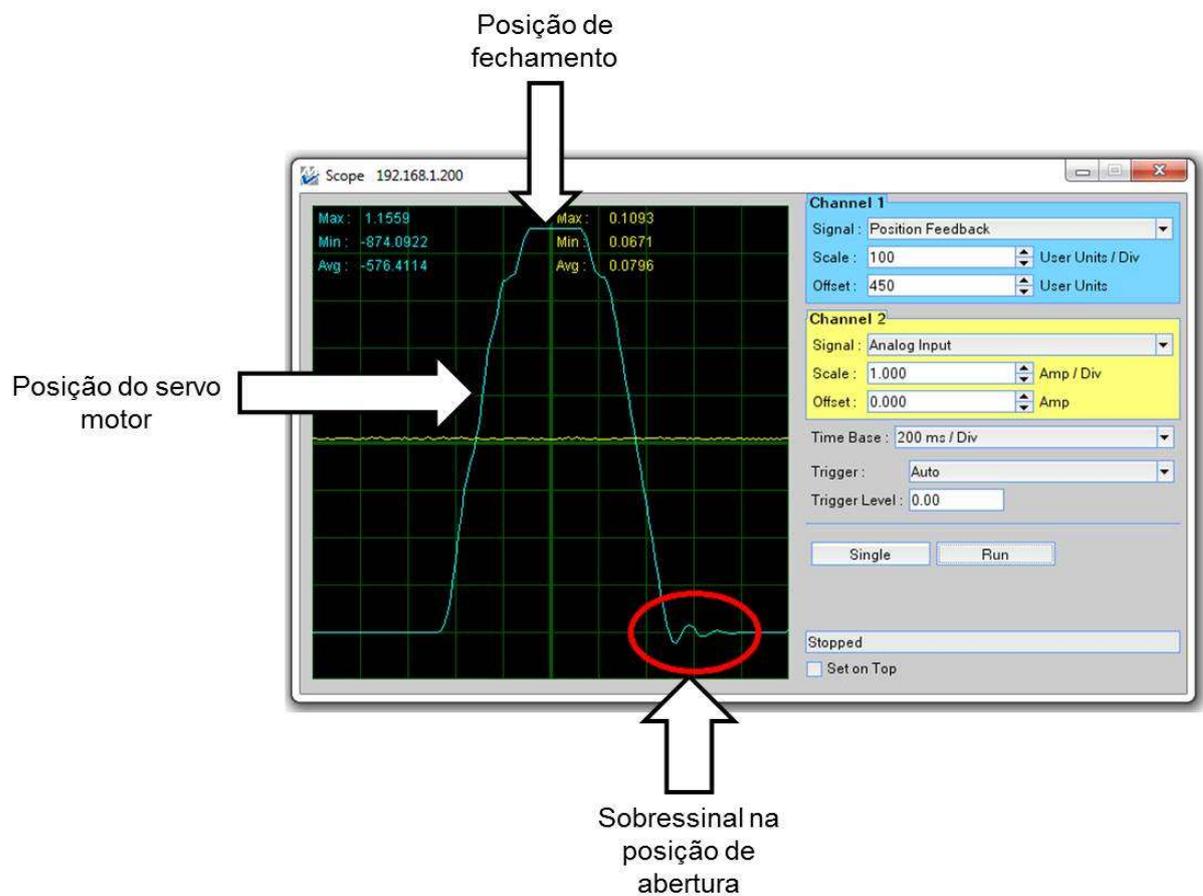


Figura 23 - Curva do servo motor com ajustes iniciais

A curva da Figura 21 foi obtida utilizando os ganhos da Tabela 1, considerando-os como único conjunto de ganhos para o movimento.

Tabela 1 - Ganhos iniciais

Ganhos	
PI (Velocidade)	
KP	11000
KI	600
PID (Posição)	
KP	600
KI	0
KD	600

5.3 Ajuste de ganhos região A

Para ajustar os ganhos da região A, foi parametrizado no servo *drive* a posição de fechamento 0mm (ponto onde os mordentes se encontram) e a de abertura 20mm, fazendo o servo motor se deslocar da posição de abertura para a posição de fechamento e vice-versa, de forma cíclica, na faixa de operação da região A. Desta forma, os ganhos foram ajustados com o conjunto em movimento, observando a redução do sobressinal na posição de abertura.

Para a região A, os ganhos foram mantidos com o objetivo de registrar o erro, que pode ser observado na Figura 24 (lado esquerdo) e, posteriormente, os ganhos foram ajustados, obtendo um conjunto de 20 ganhos diferentes para a região A, de forma que o sobressinal de posicionamento fosse menor que 0,5%, que pode ser observado na Figura 24 (lado direito).

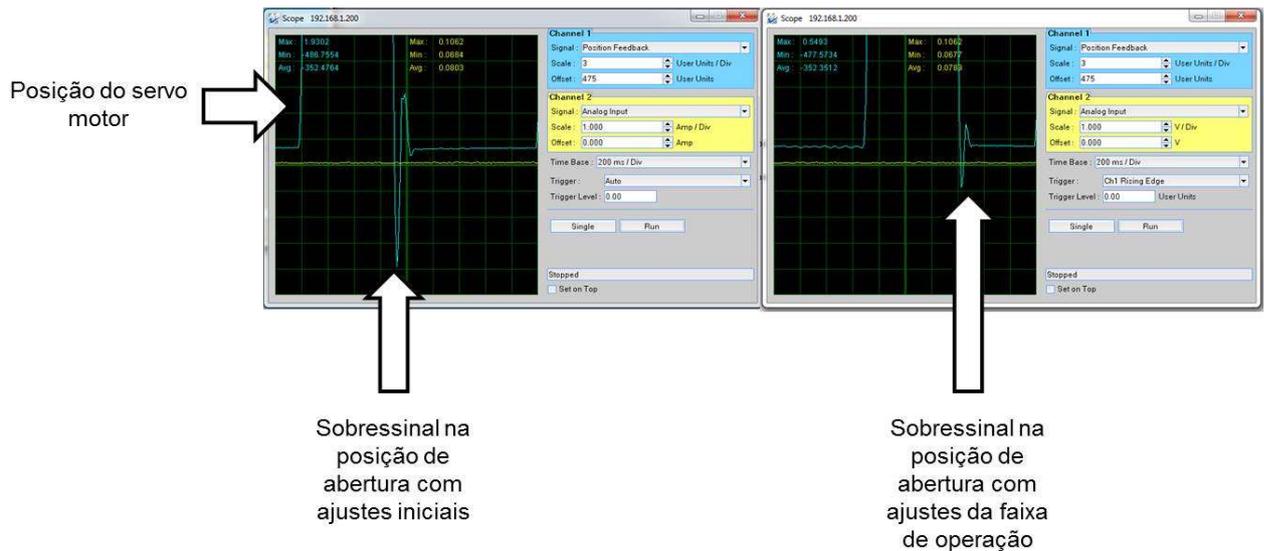


Figura 24 - Detalhe do erro de posição na região A

Na Tabela 2, é ilustrado o conjunto de 20 ganhos que foram ajustados para a região A e que serão utilizados no treinamento da rede neural artificial.

Tabela 2 - Ganhos ajustados para a região A

Ganhos de Posição			Ganhos de Velocidade	
KP	KI	KD	KP	KI
500	0	600	10000	600
515	100	580	9800	570
520	120	610	10100	610
490	200	620	10200	630
480	50	630	10300	640
600	90	615	11000	625
650	95	616	11050	626
660	98	618	11080	628
665	101	620	11100	630
670	103	523	11150	635
675	105	525	11200	640
680	110	530	11220	645
685	115	540	11250	650
690	120	545	11270	655
695	125	550	11290	700
700	130	555	11310	705
710	135	560	11330	710
720	140	565	11350	715
730	145	570	11375	720
740	150	575	11400	725

5.4 Ajuste de ganhos região B

Para ajustar os ganhos da região B, foi parametrizado no servo *drive* a posição de fechamento 20mm e a de abertura 50mm, fazendo o servo motor se deslocar da posição de abertura para a posição de fechamento e vice-versa, de forma cíclica, na faixa de operação da região B. Desta forma, os ganhos foram ajustados com o conjunto em movimento, observando a redução do sobressinal na posição de abertura.

Para a região B, os ganhos foram ajustados de forma que fosse obtido um erro de sobressinal menor que 0,5%, obtendo um conjunto satisfatório de 20 ganhos diferentes para a região B, que pode ser observado na Tabela 3 e que, posteriormente, serão utilizados no treinamento da rede neural artificial.

Tabela 3 - Ganhos ajustados para a região B

Ganhos de Posição			Ganhos de Velocidade	
KP	KI	KD	KP	KI
848	10	90	18050	10
825	60	115	17850	75
810	70	125	17800	85
818	65	128	17830	90
800	55	140	17840	95
840	100	95	18000	5
835	5	100	17950	15
830	15	105	17900	20
820	20	110	17975	25
815	25	120	17925	30
805	30	130	17875	35
795	35	135	17775	40
790	40	145	17750	45
785	45	150	17725	50
780	50	155	17700	55
775	75	160	17675	60
770	80	165	17650	65
765	85	170	17625	70
760	90	175	17600	80
755	95	180	17575	100

5.5 Ajuste de ganhos região C

Para ajustar os ganhos da região C, foi parametrizado no servo *drive* a posição de fechamento 50mm e a de abertura 120mm, fazendo o servo motor se deslocar da posição de abertura para a posição de fechamento e vice-versa, de forma cíclica, na faixa de operação da região C. Desta forma, os ganhos foram ajustados com o conjunto em movimento, observando a redução do sobressinal na posição de abertura.

Para a região C, os ganhos foram ajustados de forma que fosse obtido um erro de sobressinal menor que 0,5%, obtendo um conjunto satisfatório de 20 ganhos diferentes para a região B, que pode ser observado na Tabela 4 e que, posteriormente, serão utilizados no treinamento da rede neural artificial.

Tabela 4 - Ganhos ajustados para a região C

Ganhos de Posição			Ganhos de Velocidade	
KP	KI	KD	KP	KI
900	200	150	13500	10
890	220	140	13600	20
910	210	160	13700	30
920	220	170	13750	40
930	230	180	13800	50
950	201	155	13490	5
960	202	145	13480	15
970	203	165	13470	25
980	204	175	13460	35
990	205	185	13450	45
1000	206	152	13440	55
1010	207	142	13430	52
1020	208	162	13420	42
1030	209	172	13410	32
1040	211	182	13400	22
1050	212	147	13510	12
1060	213	157	13520	17
1070	214	167	13530	27
1080	215	177	13540	37
1090	216	187	13550	47

6 DEFINIÇÃO, TREINAMENTO E SIMULAÇÃO DA RNA

Neste capítulo, serão apresentados os dados utilizados para treinamento, teste e validação da rede neural artificial, o modelo da rede neural artificial e a simulação feita no programa computacional Matlab.

6.1 Definição de entradas e saídas

Para que a RNA seja projetada, é necessário que as entradas e saídas da rede neural artificial estejam definidas, bem como os limites que possam existir entre valores de entrada e saída.

A posição do servo motor do conjunto do mordente de selagem, conforme visto anteriormente, é limitada fisicamente no espaço amostral, que varia de 0 até 120mm de abertura. A posição do servo motor será a variável de entrada da rede neural artificial, pois é ela quem representa a mudança da faixa de operação da planta.

Conforme visto anteriormente, o controlador possui cinco ganhos, que serão escalonados KP, KI e KD de posição e KP e KI de velocidade. Estes ganhos serão as variáveis de saída da rede neural artificial e o seu limite se encontra através do valor máximo e mínimo coletados, portanto, a RNA terá uma entrada e cinco saídas, como ilustrado na Figura 25.

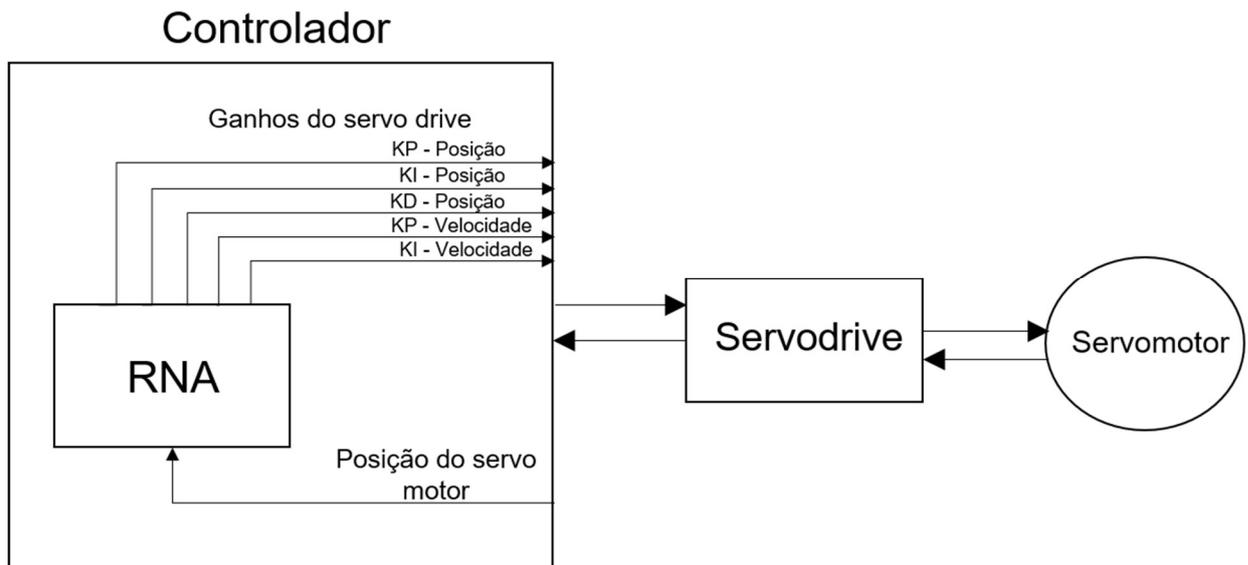


Figura 25 - Entradas e saídas da RNA

6.2 Organização dos dados para treinamento

Utilizando-se os ganhos coletados para cada região e todo o espaço amostral possível das entradas (0 – 120mm), foi criada uma tabela para cada região de operação do controlador, que será utilizada no treinamento, teste e validação da rede no programa computacional Matlab.

A região A possui 20 diferentes dados de entrada (0 – 19). Conforme visto anteriormente, foram coletados 20 diferentes conjuntos de ganhos e foi ordenado de forma aleatória um conjunto de dados para cada posição servo motor da faixa de operação A, como pode ser observado na Tabela 5.

Tabela 5 - Dados para treinamento da região A

Entradas		Saídas				
Posição do servo		Ganhos de Posição			Ganhos de Velocidade	
		KP	KI	KD	KP	KI
A	0	500	0	600	10000	600
	1	515	100	580	9800	570
	2	520	120	610	10100	610
	3	490	200	620	10200	630
	4	480	50	630	10300	640
	5	600	90	615	11000	625
	6	650	95	616	11050	626
	7	660	98	618	11080	628
	8	665	101	620	11100	630
	9	670	103	523	11150	635
	10	675	105	525	11200	640
	11	680	110	530	11220	645
	12	685	115	540	11250	650
	13	690	120	545	11270	655
	14	695	125	550	11290	700
	15	700	130	555	11310	705
	16	710	135	560	11330	710
	17	720	140	565	11350	715
	18	730	145	570	11375	720
	19	740	150	575	11400	725

A região B possui 30 diferentes dados de entrada (20 – 50). Segundo já mencionado, foram coletados 20 diferentes conjuntos de ganhos. Os conjuntos de ganhos coletados foram distribuídos de forma aleatória para as posições do servo motor da faixa de operação B, conforme pode ser observado na Tabela 6.

Tabela 6 - Dados para treinamento da região B

Entradas		Saídas				
Posição do servo		Ganhos de Posição			Ganhos de Velocidade	
		KP	KI	KD	KP	KI
B	20	675	105	525	11200	640
	21	848	10	90	18050	10
	22	825	60	115	17850	75
	23	810	70	125	17800	85
	24	818	65	128	17830	90
	25	800	55	140	17840	95
	26	840	100	95	18000	5
	27	835	5	100	17950	15
	28	830	15	105	17900	20
	29	820	20	110	17975	25
	30	815	25	120	17925	30
	31	805	30	130	17875	35
	32	795	35	135	17775	40
	33	790	40	145	17750	45
	34	785	45	150	17725	50
	35	780	50	155	17700	55
	36	775	75	160	17675	60
	37	770	80	165	17650	65
	38	765	85	170	17625	70
	39	760	90	175	17600	80
	40	755	95	180	17575	100
	41	848	10	90	18050	10
	42	825	60	115	17850	75
	43	810	70	125	17800	85
	44	818	65	128	17830	90
	45	800	55	140	17840	95
	46	840	100	95	18000	5
	47	835	5	100	17950	15
	48	830	15	105	17900	20
	49	820	20	110	17975	25
	50	815	25	120	17925	30

A região C possui 70 diferentes dados de entrada (51 – 120). Conforme já visto, foram coletados 20 diferentes conjuntos de ganhos. Os conjuntos de ganhos coletados foram distribuídos de forma aleatória para as posições do servo motor da faixa de operação C, conforme pode ser observado na Tabela 7.

Tabela 7 - Dados para treinamento da região B

Entradas		Saídas					Entradas		Saídas				
Posição do servo		Ganhos de Posição			Ganhos de Velocidade		Posição do servo		Ganhos de Posição			Ganhos de Velocidade	
		KP	KI	KD	KP	KI			KP	KI	KD	KP	KI
C	51	900	200	150	13500	10	C	86	1050	212	147	13510	12
	52	890	220	140	13600	20		87	1060	213	157	13520	17
	53	910	210	160	13700	30		88	1070	214	167	13530	27
	54	920	220	170	13750	40		89	1080	215	177	13540	37
	55	930	230	180	13800	50		90	1090	216	187	13550	47
	56	950	201	155	13490	5		91	900	200	150	13500	10
	57	960	202	145	13480	15		92	890	220	140	13600	20
	58	970	203	165	13470	25		93	910	210	160	13700	30
	59	980	204	175	13460	35		94	920	220	170	13750	40
	60	990	205	185	13450	45		95	930	230	180	13800	50
	61	1000	206	152	13440	55		96	950	201	155	13490	5
	62	1010	207	142	13430	52		97	960	202	145	13480	15
	63	1020	208	162	13420	42		98	970	203	165	13470	25
	64	1030	209	172	13410	32		99	980	204	175	13460	35
	65	1040	211	182	13400	22		100	990	205	185	13450	45
	66	1050	212	147	13510	12		101	1000	206	152	13440	55
	67	1060	213	157	13520	17		102	1010	207	142	13430	52
	68	1070	214	167	13530	27		103	1020	208	162	13420	42
	69	1080	215	177	13540	37		104	1030	209	172	13410	32
	70	1090	216	187	13550	47		105	1040	211	182	13400	22
	71	900	200	150	13500	10		106	1050	212	147	13510	12
	72	890	220	140	13600	20		107	1060	213	157	13520	17
	73	910	210	160	13700	30		108	1070	214	167	13530	27
	74	920	220	170	13750	40		109	1080	215	177	13540	37
	75	930	230	180	13800	50		110	1090	216	187	13550	47
	76	950	201	155	13490	5		111	1000	206	152	13440	55
	77	960	202	145	13480	15		112	1010	207	142	13430	52
	78	970	203	165	13470	25		113	1020	208	162	13420	42
	79	980	204	175	13460	35		114	1030	209	172	13410	32
	80	990	205	185	13450	45		115	1040	211	182	13400	22
	81	1000	206	152	13440	55		116	1050	212	147	13510	12
	82	1010	207	142	13430	52		117	1060	213	157	13520	17
	83	1020	208	162	13420	42		118	1070	214	167	13530	27
	84	1030	209	172	13410	32		119	1080	215	177	13540	37
	85	1040	211	182	13400	22		120	1090	216	187	13550	47

6.3 Treinamento da RNA

A partir dos dados definidos anteriormente, a rede foi configurada e treinada para atingir o objetivo de escalonar os ganhos do controlador nas diferentes faixas de operação.

As entradas e saídas já estão definidas, conforme visto, uma entrada e cinco saídas. Segundo Miguez (2012), não há uma regra exata para a determinação da quantidade de camadas ocultas, porém, existem estudos que fazem aproximações. De acordo com Hecht-Nielsen (1989), ao utilizar a retropropagação do erro, apenas uma camada oculta é necessária para aproximar qualquer função. Desta forma, a rede neural artificial foi configurada com apenas uma camada oculta.

Segundo Binoti *et al.* (2014), a quantidade de neurônios na camada oculta é determinada principalmente de maneira empírica, pela experiência de quem projeta a rede neural artificial, embora existam outros estudos, baseados em metodologias, propostos por Hirose *et al.* (1991), Arai (1993) e Fujita (1998). Hecht-Nielsen(1987), determinou que em uma rede neural artificial com retropropagação do erro, onde possui apenas uma camada oculta, o número de neurônios na camada oculta pode ser estimado pela equação $2n+1$, onde “n” é a quantidade de entradas da rede neural artificial. Portanto, no caso da RNA para o escalonamento dos ganhos do mordente horizontal, seriam necessários três neurônios na camada oculta.

Na determinação do número de neurônios, deve ser considerado que uma quantidade excessiva de neurônios pode acarretar na memorização dos dados de treinamento, também conhecido como *overfitting*. Porém, um número pequeno de neurônios na camada oculta pode não atingir o resultado esperado da rede neural artificial, este processo é conhecido como *underfitting* (SILVA *et al.* 2010).

Desta forma, a rede neural artificial foi inicialmente configurada com três neurônios na camada oculta, no entanto, após testes, foi constatado que com apenas dois neurônios na camada oculta, seria possível treinar a rede neural artificial de forma que obtivesse o desempenho esperado. A configuração final da RNA pode ser vista na Figura 26.

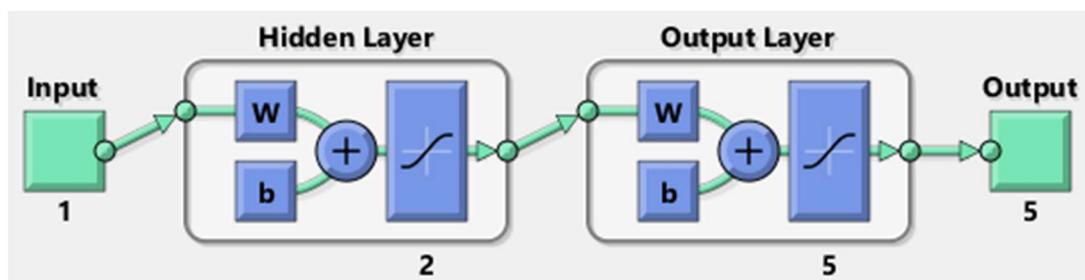


Figura 26 - Rede Neural Artificial utilizada

A função de ativação utilizada foi a função descrita no Matlab como “Tansig” que, segundo o site Mathworks, do Matlab, é a função de transferência sigmoide tangente hiperbólica. Foram testadas outras funções, entretanto, essa foi a que demonstrou melhor desempenho para o projeto.

A função de treinamento utilizada foi a função descrita no Matlab como “Trainbr” que, segundo o site Mathworks, do Matlab, é a função de regularização bayesiana, que consiste em atualizar os valores dos pesos e do bias, de acordo com a otimização de Levenberg-Marquardt. Isso minimiza a combinação do erro quadrático e dos pesos e determina a combinação correta para produzir uma rede que faça a generalização.

A função de *performance* utilizada foi a função MSE que, segundo o site Mathworks, do Matlab, é o erro médio quadrático. Essa função de *performance* deve ser utilizada quando a função de treinamento utiliza a regularização bayesiana.

Foram feitos diversos testes com funções de treinamento diferentes e distintas configurações, porém, essa foi a configuração que obteve melhor resultado e com uma quantidade menor de neurônios.

Segundo Beale (2016), no treinamento de redes multicamadas é usual que os dados sejam divididos em três conjuntos. O primeiro é o de treinamento, utilizado para computar o gradiente e atualizar os valores de peso e bias. O segundo é o de validação, onde o erro é monitorado no processo do treinamento. O terceiro conjunto é o de teste, que não é utilizado no treinamento, mas sim para comparar diferentes modelos e também para plotar o erro do conjunto de teste durante o treinamento. A

proporção da divisão dos dados utilizada no conjunto do mordente foi a padrão do Matlab, sendo 70% dos dados utilizados no treinamento, 15% na validação e 15% no teste, e a definição de quais dados fazem parte de quais conjuntos foi feita de modo aleatório, também padrão do Matlab.

Pode ser visto na Figura 27 o desempenho do treinamento com este modelo de rede neural artificial. Observa-se que o treinamento parou pelo “Mu” máximo, exibindo a mensagem “Maximum MU reached”. Segundo Abhishek (2012), “Mu” é um parâmetro que mede a taxa de adaptação/aprendizagem. A parada do treinamento pelo “Mu” máximo significa que a taxa de aprendizagem atingiu o seu máximo e que o aprendizado da rede não irá mais evoluir, por este motivo, o treinamento é encerrado.

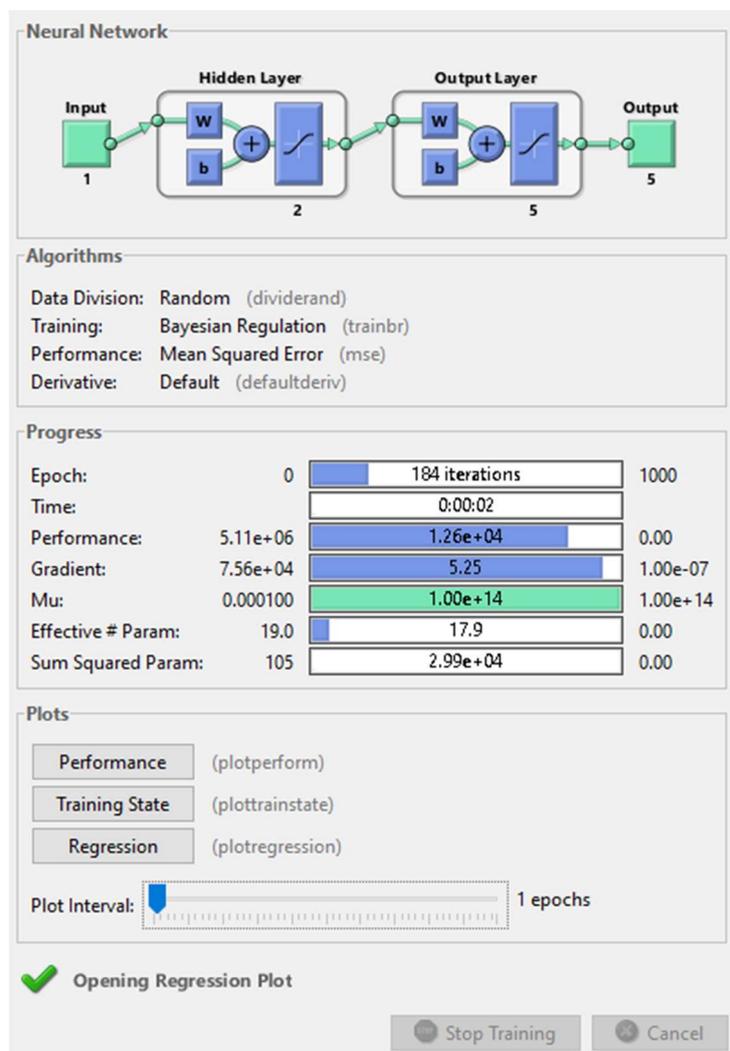


Figura 27 - Desempenho do treinamento

Pode ser visto na Figura 28 o gráfico de *performance* do treinamento.

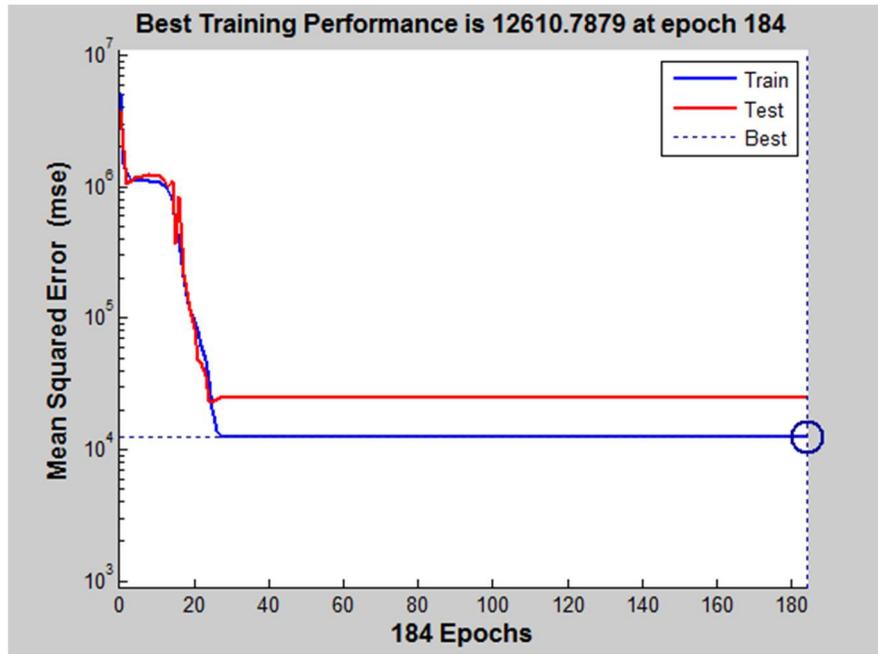


Figura 28 - Performance do treinamento

Pode ser visto na Figura 29 o gráfico de regressão do treinamento.

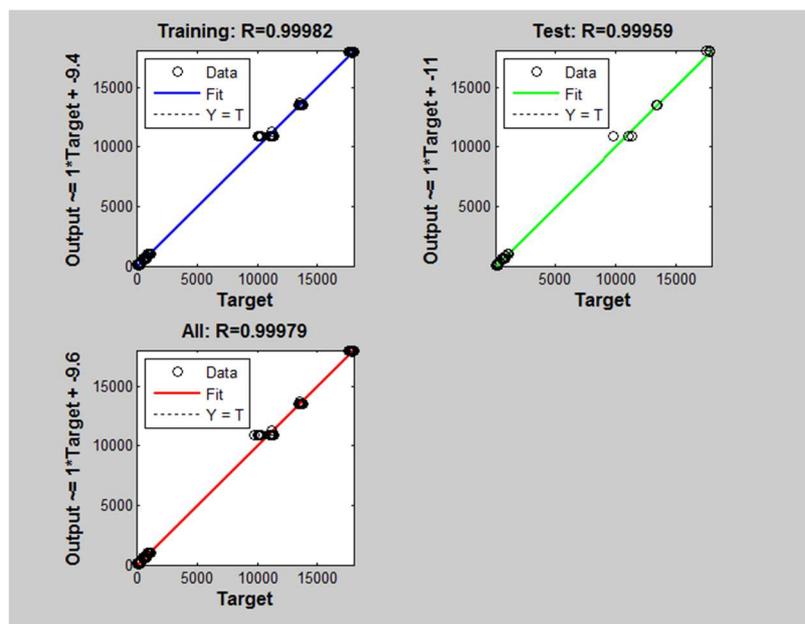


Figura 29 - Regressão

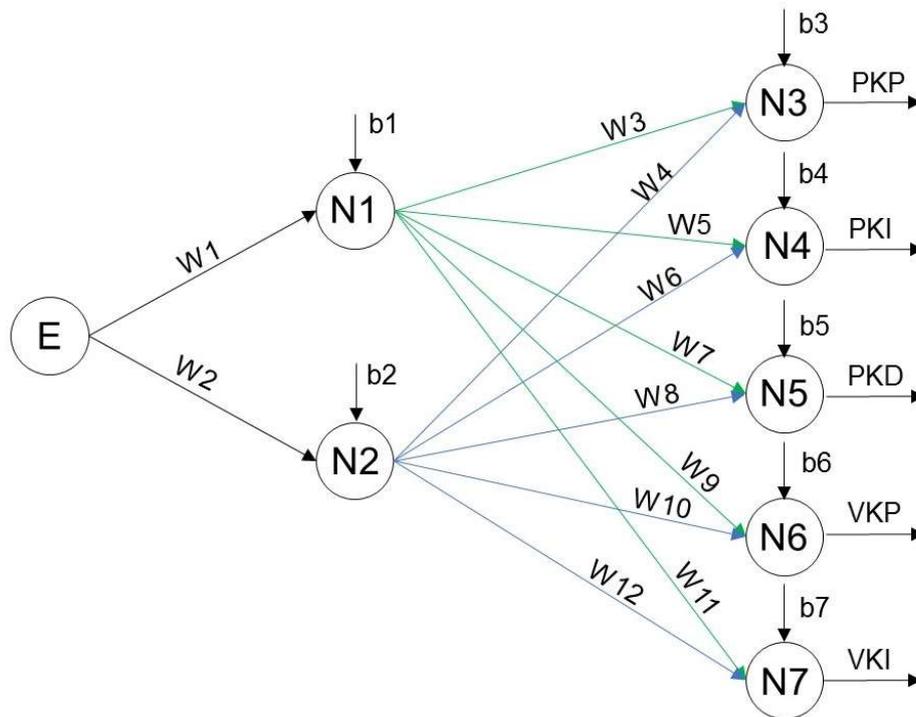


Figura 31 - Neurônio – Pesos e bias

Tabela 8 - Pesos e bias da rede treinada

Pesos		Bias	
W1=	98,6415	B1=	64
W2=	124,3306	B2=	21,4438
W3=	0,29428	B3=	0,17122
W4=	0,38982	B4=	0,50683
W5=	-0,27457	B5=	0,08267
W6=	0,87397	B6=	-0,51284
W7=	-6,6309	B7=	-0,27732
W8=	5,6281		
W9=	5,7474		
W10=	-5,3325		
W11=	-5,2594		
W12=	3,9133		

6.4 Simulação da RNA

Após treinada, a rede foi simulada no Matlab e o resultado foi ilustrado em gráficos, juntamente com os dados utilizados no treinamento, para que fosse possível visualizar o desempenho do resultado do treinamento e verificar se esse está dentro dos limites estabelecidos.

No Gráfico 2, é possível ver a curva da simulação da rede neural artificial para a saída que representa o ganho de posição PKP. Também estão plotados os valores do ganho PKP, definidos para cada posição, que foram os mesmos utilizados no treinamento da rede neural artificial.

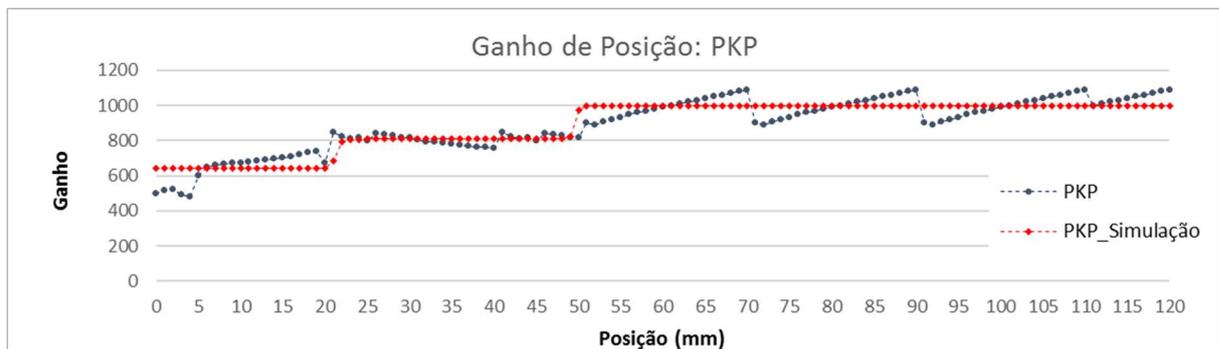


Gráfico 2 - Ganho de posição PKP

No Gráfico 3, é possível ver a curva da simulação da rede neural artificial para a saída que representa o ganho de posição PKI. Também estão plotados os valores do ganho PKI, definidos para cada posição, que foram os mesmos valores utilizados no treinamento da rede neural artificial.

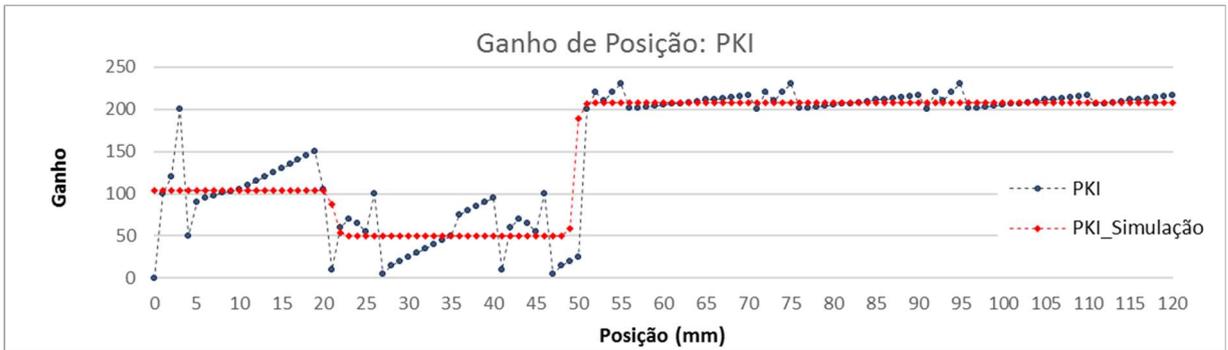


Gráfico 3 - Ganho de posição PKI

No Gráfico 4, é possível ver a curva da simulação da rede neural artificial para a saída que representa o ganho de posição PKD. Também estão plotados os valores do ganho PKD, definidos para cada posição, que foram os mesmos valores utilizados no treinamento da rede neural artificial.

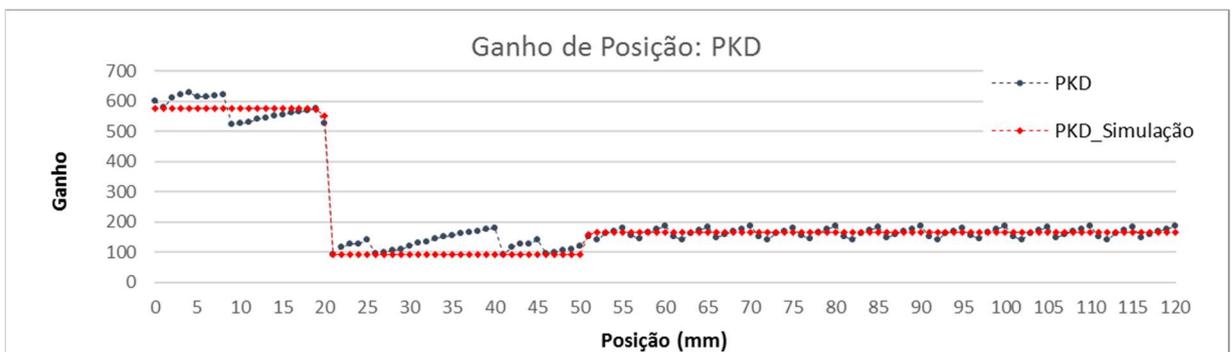


Gráfico 4 - Ganho de posição PKD

No Gráfico 5, é possível ver a curva da simulação da rede neural artificial para a saída que representa o ganho de velocidade VKP. Também estão plotados os valores do ganho VKP, definidos para cada posição, que foram os mesmos valores utilizados no treinamento da rede neural artificial.

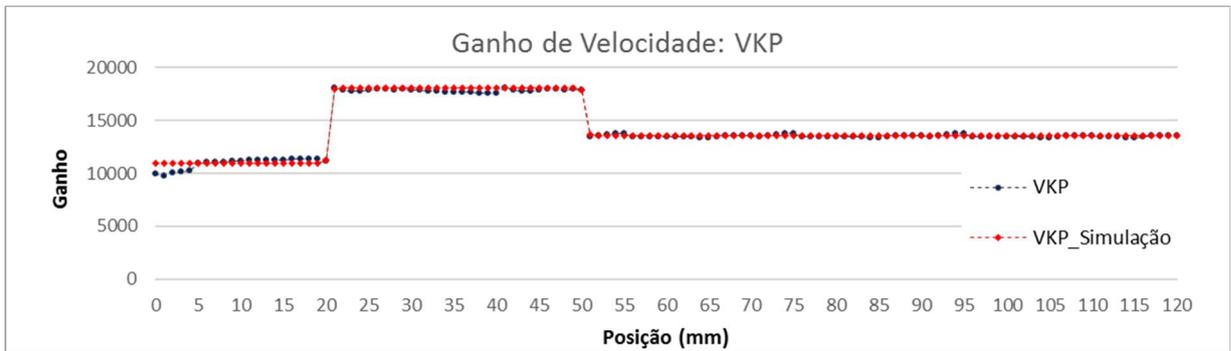


Gráfico 5 - Ganho de velocidade VKP

No Gráfico 6, é possível ver a curva da simulação da rede neural artificial para a saída que representa o ganho de velocidade VKI. Também estão plotados os valores do ganho VKI, definidos para cada posição, que foram os mesmos valores utilizados no treinamento da rede neural artificial.

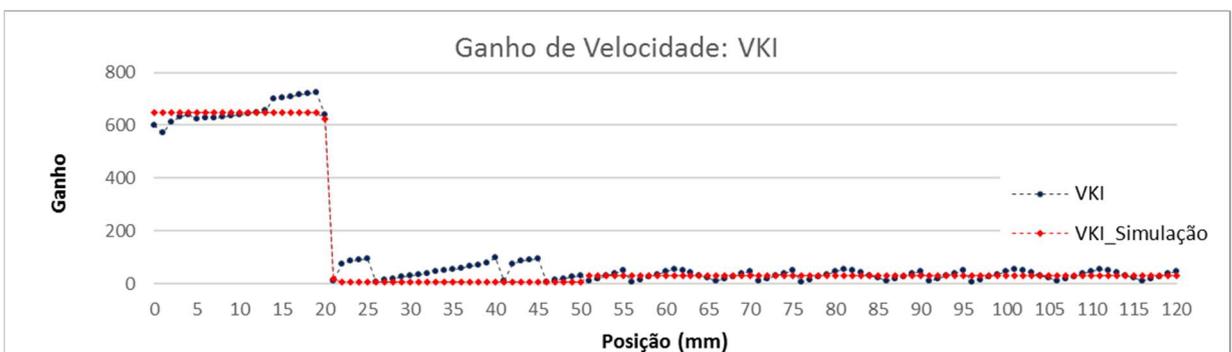


Gráfico 6 - Ganho de velocidade VKI

Os gráficos da simulação dos ganhos demonstram que a rede neural artificial obteve um bom desempenho quanto à generalização dos dados e ajuste de curva. Os dados obtidos no treinamento da rede neural artificial, como os pesos e bias, serão utilizados na implementação da rede e vistos no próximo capítulo.

7 APLICAÇÃO DA RNA NO CLP

Neste capítulo, será visto como normalizar os dados de entrada e saída, como implementar a rede neural artificial no CLP e a simulação da rede para validar a rede neural artificial embarcada.

7.1 Normalização das entradas e saídas

Antes que a rede seja embarcada no CLP, segundo Gambogi (2013), é necessário que seja feita a normalização das entradas e saídas.

Como a função de ativação é do tipo tangente hiperbólica sigmoide, é imprescindível que todos os valores de entrada e saída estejam dentro do intervalo de -1 a $+1$. Este cálculo é feito com base nos valores máximo e mínimo das entradas e saídas. Na Tabela 9, podem ser vistos os valores de máximo e mínimo das entradas e saídas.

Tabela 9 - Valor máximo e mínimo das entradas e saídas

	Posição do Servomotor	Ganho de Posição PKP	Ganho de Posição PKI	Ganho de Posição PKD	Ganho de Velocidade VKP	Ganho de Velocidade VKI
Valor Mínimo	0	480	0	90	9800	5
Valor Máximo	120	1090	230	630	18050	725

De acordo com Gambogi (2013), a normalização pode ser calculada seguindo a equação 13.

$$Vn = \frac{(V - Vmin) \cdot (Lmax - Lmin)}{Vmax - Vmin} + Lmin \quad (13)$$

Onde:

Vn : Valor normalizado

V : Valor anterior à normalização

$Vmax$: Valor máximo possível

$Vmin$: Valor mínimo possível

$Lmax$: Valor do limite máximo possível

$Lmin$: Valor do limite mínimo possível

Para que fique clara a normalização, pode-se ter como exemplo a normalização da entrada, em que o valor normalizado é o valor que será utilizado como entrada no neurônio. O valor anterior à normalização é a posição de entrada, que varia entre o valor máximo de 120 e o mínimo de 0. Os limites máximo e mínimo são referentes à função de ativação tangente hiperbólica, que varia entre o limite máximo de 1 e o limite mínimo de -1.

Substituindo os valores na equação 13, tem-se a equação 14 e, posteriormente a equação 15.

$$Vn = \frac{(V - 0) \cdot (1 - (-1))}{120 - 0} + (-1) \quad (14)$$

$$Vn = \frac{(V) \cdot (2)}{120} - 1 \quad (15)$$

Portanto, se o valor de entrada for o valor máximo de 120, o valor normalizado será 1. E se o valor de entrada for o valor mínimo 0, o valor normalizado será -1.

No caso da normalização da saída, o cálculo é o mesmo, porém, a saída do neurônio é o valor normalizado e deve ser feito o cálculo para encontrar o valor proporcional, conforme a equação 16.

$$V = \frac{(Vn - Lmin) \cdot (Vmax - Vmin)}{Lmax - Lmin} + Vmin \quad (16)$$

Pode-se ter como exemplo a saída PKP, onde o neurônio irá fornecer uma saída entre o limite máximo 1 e o limite mínimo -1, e será obtido um valor entre o valor máximo 1090 e o valor mínimo 480. Substituindo os valores na equação 16, tem-se a equação 17 e, posteriormente, a equação 18:

$$V = \frac{(Vn - (-1)) \cdot (1090 - 480)}{1 - (-1)} + 480 \quad (17)$$

$$V = \frac{(Vn + 1) \cdot (610)}{2} + 480 \quad (18)$$

Desta forma, quando o valor de saída do neurônio for o valor máximo 1, o valor de saída será o ganho máximo de 1090, e quando o valor de saída do neurônio for o valor mínimo -1, o valor de saída será o ganho mínimo 480.

7.2 Descrição matemática do neurônio

O neurônio pode ser escrito agora matematicamente para depois ser programado no CLP. Na Figura 32, pode ser visto o neurônio, bem como seus componentes.

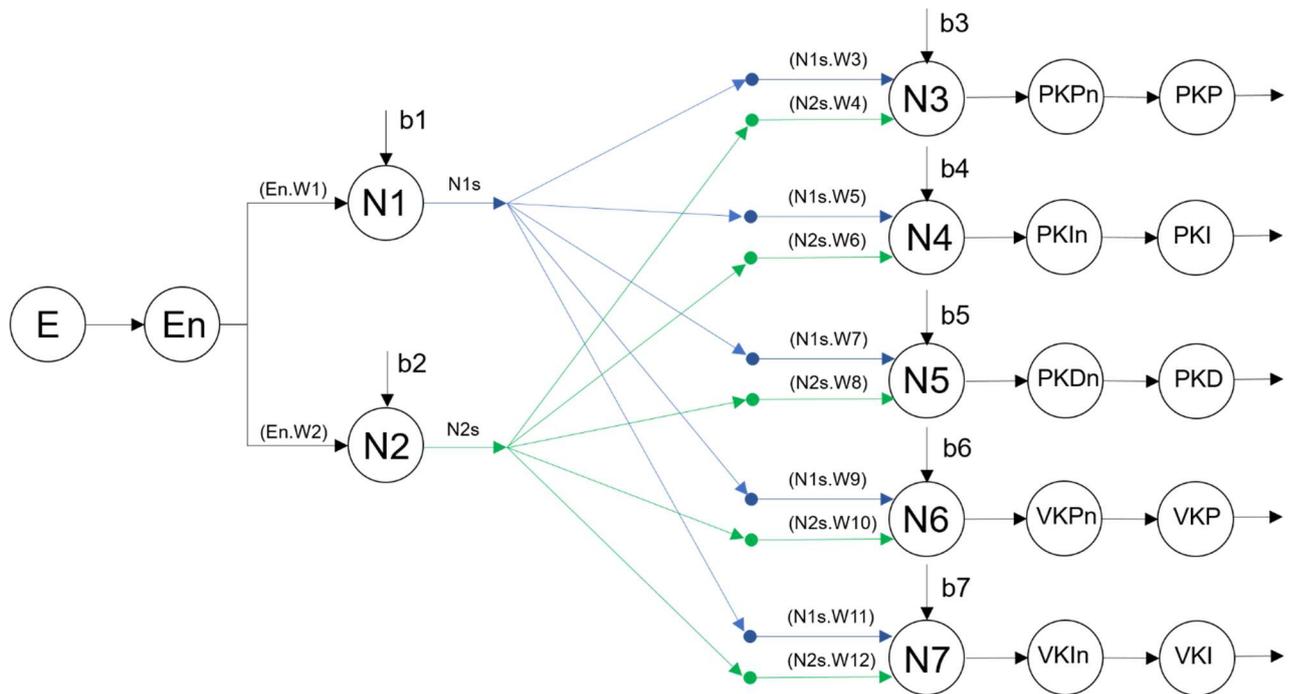


Figura 32 - Neurônio e seus componentes

Conforme visto na Figura 32:

E - Representa a entrada do neurônio, ou seja, a posição do servo motor.

En - É a entrada após o processo de normalização, conforme visto anteriormente.

$N(1-7)$ - São os neurônios da rede neural artificial.

$N(1-7)s$ - É a saída do neurônio após ter sido feita a somatória e aplicada a função de ativação.

$W(1-12)$ - São os pesos sinápticos.

$B(1-7)$ - São o bias.

PKP_n - É o valor do ganho normalizado.

PKP - É o valor do ganho a ser programado no servo *drive*.

Pode-se calcular o neurônio 1 ($N1$) pela equação 19 e o neurônio 2 ($N2$) pela equação 20:

$$N1 = (En.W1)+b1 \quad (19)$$

$$N2 = (En.W2)+b2 \quad (20)$$

Lembrando que está sendo utilizada a função de ativação tangente hiperbólica, cuja função de transferência pode ser vista na equação 21.

$$tansig(n) = \frac{2}{(1 + \exp((-2).n))} - 1 \quad (21)$$

Portanto, para se obter $N1s$ e $N2s$, deve-se substituir a equação 21 com a equação 19 e 20, resultando na equação 22 e 23.

$$N1s = \frac{2}{(1 + \exp((-2).N1))} - 1 \quad (22)$$

$$N2s = \frac{2}{(1 + \exp((-2).N2))} - 1 \quad (23)$$

Após ser obtido N1s e N2s, pode-se calcular os neurônios N3, N4, N5, N6 e N7 pelas equações 24, 25, 26, 27 e 28.

$$N3 = (N1s.W3)+(N2s.W4)+b3 \quad (24)$$

$$N4 = (N1s.W5)+(N2s.W6)+b4 \quad (25)$$

$$N5 = (N1s.W7)+(N2s.W8)+b5 \quad (26)$$

$$N6 = (N1s.W9)+(N2s.W10)+b6 \quad (27)$$

$$N7 = (N1s.W11)+(N2s.W12)+b7 \quad (28)$$

Substituindo o resultado das equações 24, 25, 26, 27 e 28 na equação 21, pode-se obter a saída do neurônio normalizada, conforme as equações 29, 30, 31, 32 e 33.

$$PKPn = \frac{2}{(1 + \exp((-2).N3))} - 1 \quad (29)$$

$$PKIn = \frac{2}{(1 + \exp((-2).N4))} - 1 \quad (30)$$

$$PKDn = \frac{2}{(1 + \exp((-2).N5))} - 1 \quad (31)$$

$$VKPn = \frac{2}{(1 + \exp((-2).N6))} - 1 \quad (32)$$

$$VKIn = \frac{2}{(1 + \exp((-2).N7))} - 1 \quad (33)$$

Os valores dos ganhos obtidos estão normalizados, ou seja, dentro do intervalo de 1 e -1. Para que os valores sejam proporcionalmente convertidos, é preciso aplicar a equação 16 em cada umas das saídas e assim obter o valor dos ganhos que serão enviados ao servo *drive*.

7.3 Aplicação dos cálculos no CLP

Utilizando-se os cálculos anteriormente vistos, uma rotina foi criada no CLP em uma periódica de 2ms e os cálculos foram implementados na linguagem *ladder* no programa computacional Studio 5000, da Allen Bradley.

O controlador não possui a função exponencial, portanto, foi necessário utilizar o valor numérico de Euler como uma constante, que foi atribuída a uma variável do tipo real, utilizando a função MOV, conforme pode ser visto na Figura 33.

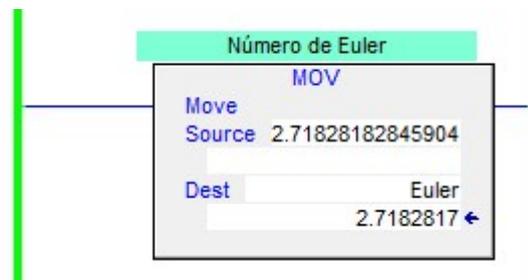


Figura 33 - Declaração do número de Euler

Todos os pesos sinápticos e bias foram atribuídos a variáveis reais, conforme pode ser visto na Figura 34.

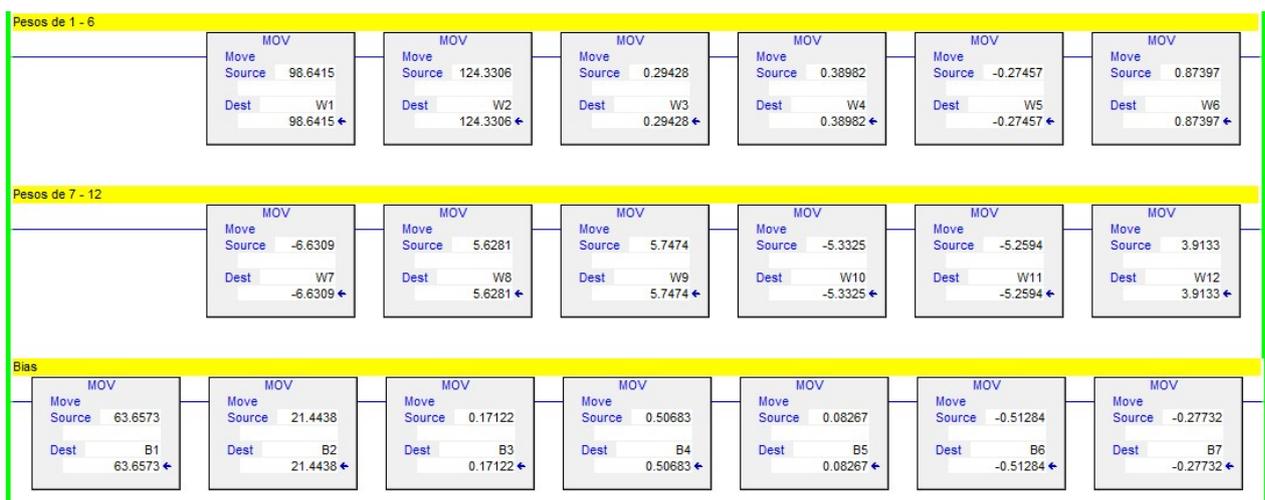


Figura 34 - Atribuição de pesos e bias

Foi feita a normalização da entrada, conforme o cálculo anteriormente demonstrado. Pode-se ver na Figura 35 a normalização da entrada, utilizando a função CPT (*compute*). Devem ser observadas as regras de cálculo do programa computacional utilizado o Studio 5000, da Allen Bradley, conforme as equações equivalentes 34 e 35.

$$X.X = X*X \quad (34)$$

$$X^2 = X**2 \quad (35)$$

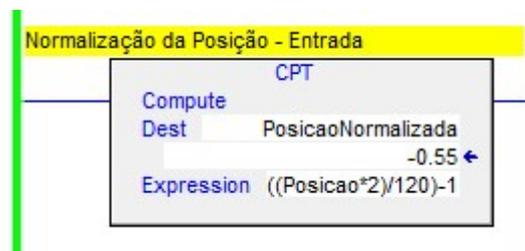


Figura 35 - Normalização da entrada

Pode ser visto na Figura 36 o cálculo do neurônio N1 e N2 da camada oculta da rede neural artificial, cuja saída será utilizada no cálculo dos neurônios da camada de saída, e o neurônio N3, cuja saída N3s é a saída do ganho PKP normalizado.

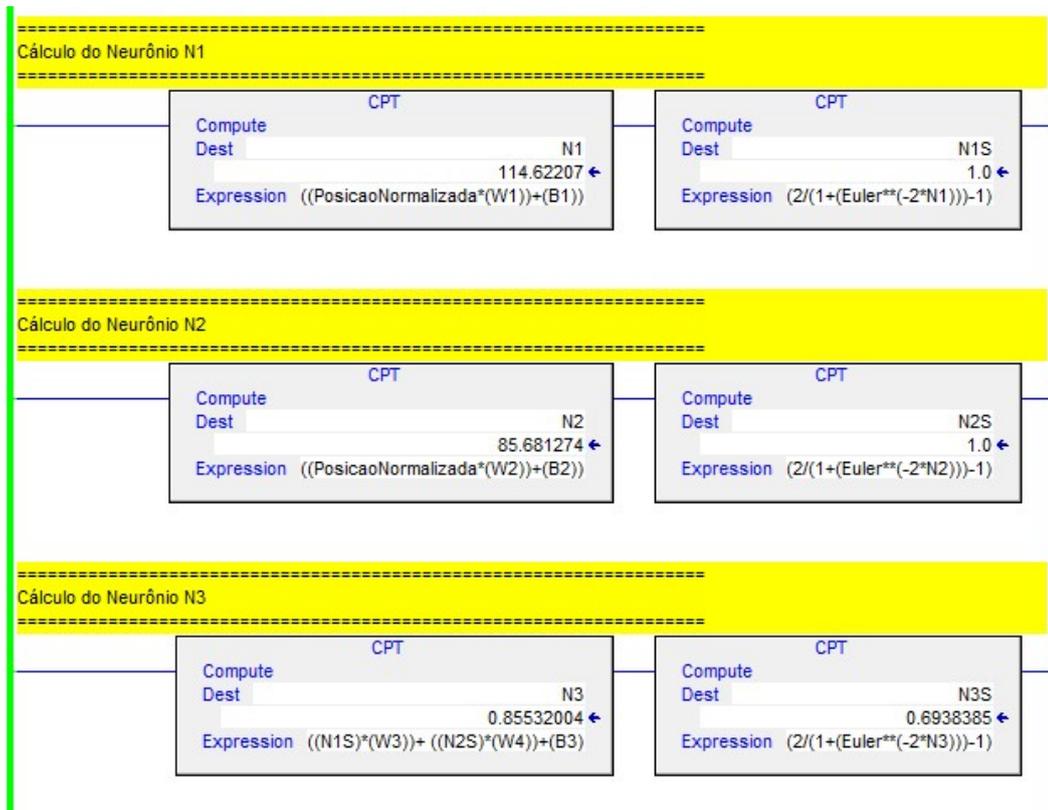


Figura 36 - Neurônio N1, N2 e N3

Utilizando-se a saída normalizada no neurônio N3s, foi gerado o ganho PKP real a ser escrito no servo *drive*, através do cálculo, que pode ser visto na Figura 37.

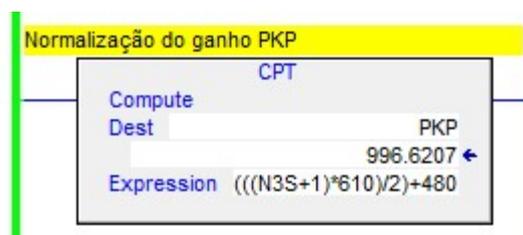


Figura 37 - Normalização da saída PKP

7.4 Simulação da RNA no CLP

Para validar o funcionamento da rede neural artificial, a mesma foi testada em simulação no CLP e os resultados foram comparados com a simulação no programa computacional Matlab.

Um eixo virtual foi criado, variando sua posição de 0 a 120mm, simulando a posição do servo motor e foi gerado um gráfico para comparar a variação dos ganhos no CLP com a dos ganhos na simulação do Matlab. A Figura 38 ilustra uma comparação entre a curva resultante da simulação no CLP e a curva resultante da simulação no Matlab. As Figuras 39, 40, 41 e 42 ilustram a curva resultante da simulação no CLP.

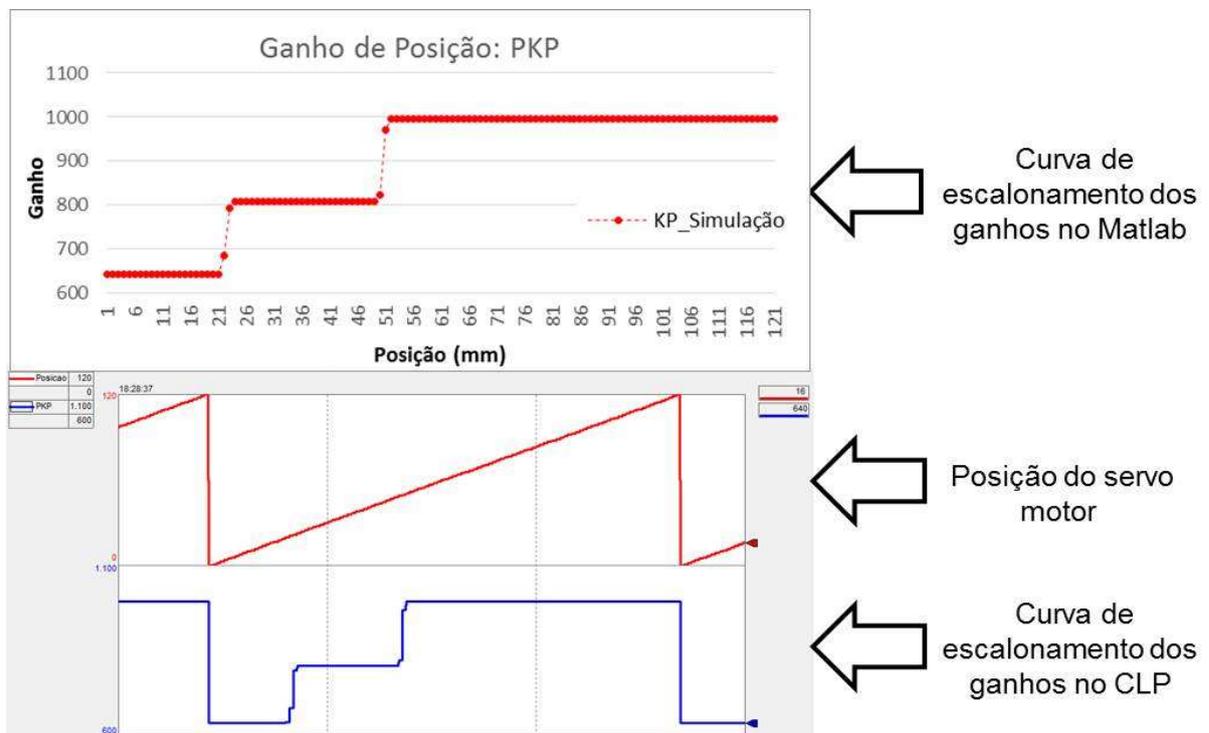


Figura 38 - Ganho PKP – Simulação no CLP comparado com simulação no Matlab

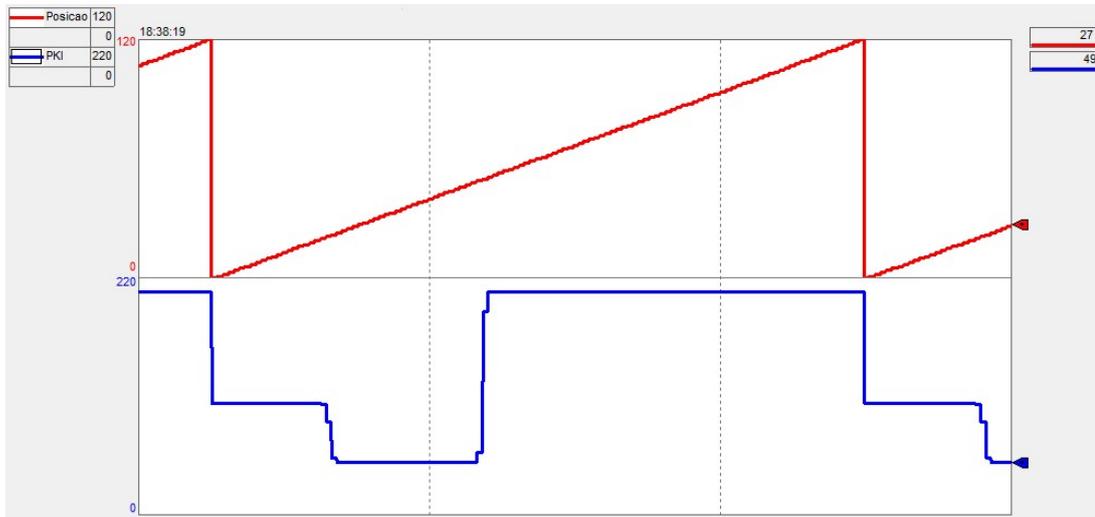


Figura 39 - Ganho PKI – Simulação no CLP

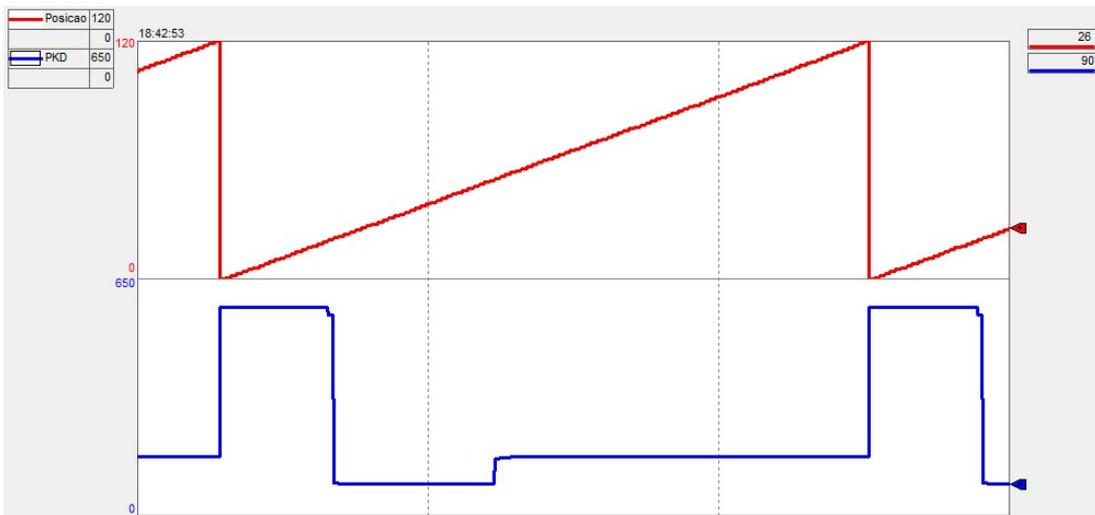


Figura 40 - Ganho PKD – Simulação no CLP

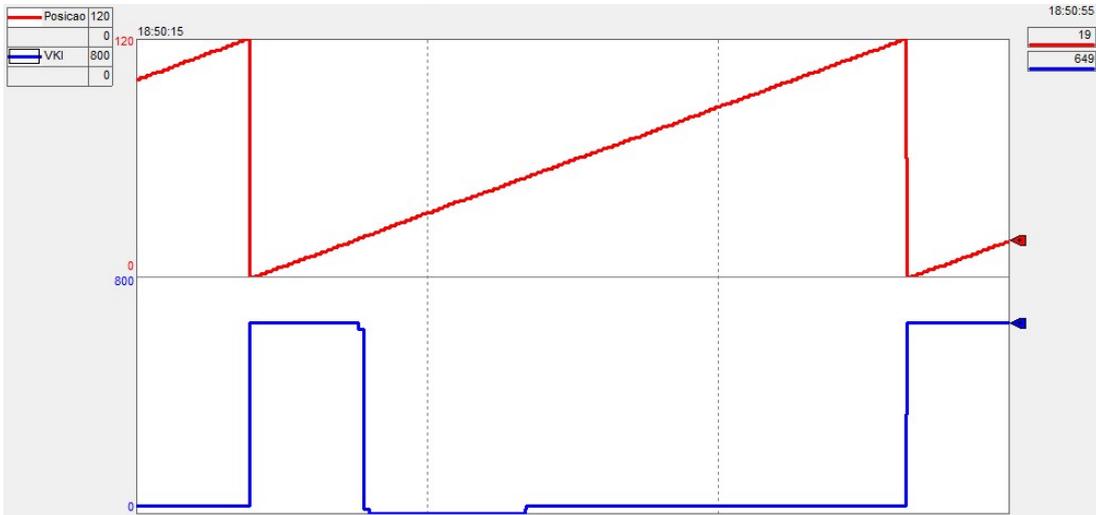


Figura 41 - Ganho VKI – Simulação no CLP

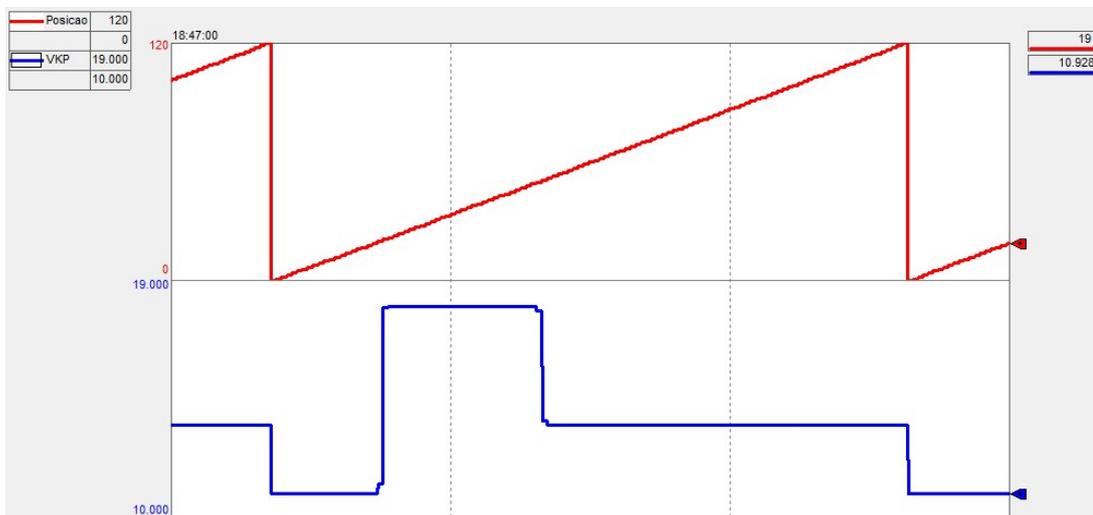


Figura 42 - Ganho VKP – Simulação no CLP

A simulação da rede neural artificial no CLP correspondeu à simulação feita no Matlab, conforme pode ser observado na comparação da Figura 38, cujos resultados se aplicam às curvas ilustradas nas Figuras 39, 40, 41 e 42, validando os cálculos e a escrita do programa em *ladder*, de forma que a rede se encontra pronta para ser utilizada no escalonamento dos ganhos do controlador, que será visto no próximo capítulo.

8 TESTES E ANÁLISE DOS DADOS

Neste capítulo, serão apresentados o teste e a análise dos resultados práticos do escalonamento dos ganhos através da rede neural artificial. Será feita uma análise da influência do escalonamento dos ganhos na corrente elétrica instantânea do servo motor e, posteriormente, estudado o método SE, alternativo à rede neural artificial. Por fim, será apresentada uma comparação entre o desempenho do modelo atual, da rede neural artificial e do método SE.

8.1 Análise do escalonamento dos ganhos através da RNA

Inicialmente, foi atribuído o conjunto de ganhos ajustado pela sintonização automática, sem que houvesse escalonamento dos ganhos, deslocando-se da posição de abertura para a posição de fechamento e vice-versa, de forma cíclica, a uma frequência de 60 ciclos por minuto. Pode ser visto na Figura 43 a curva de posição do servo motor e o sobressinal de cerca de 2% na abertura do conjunto.

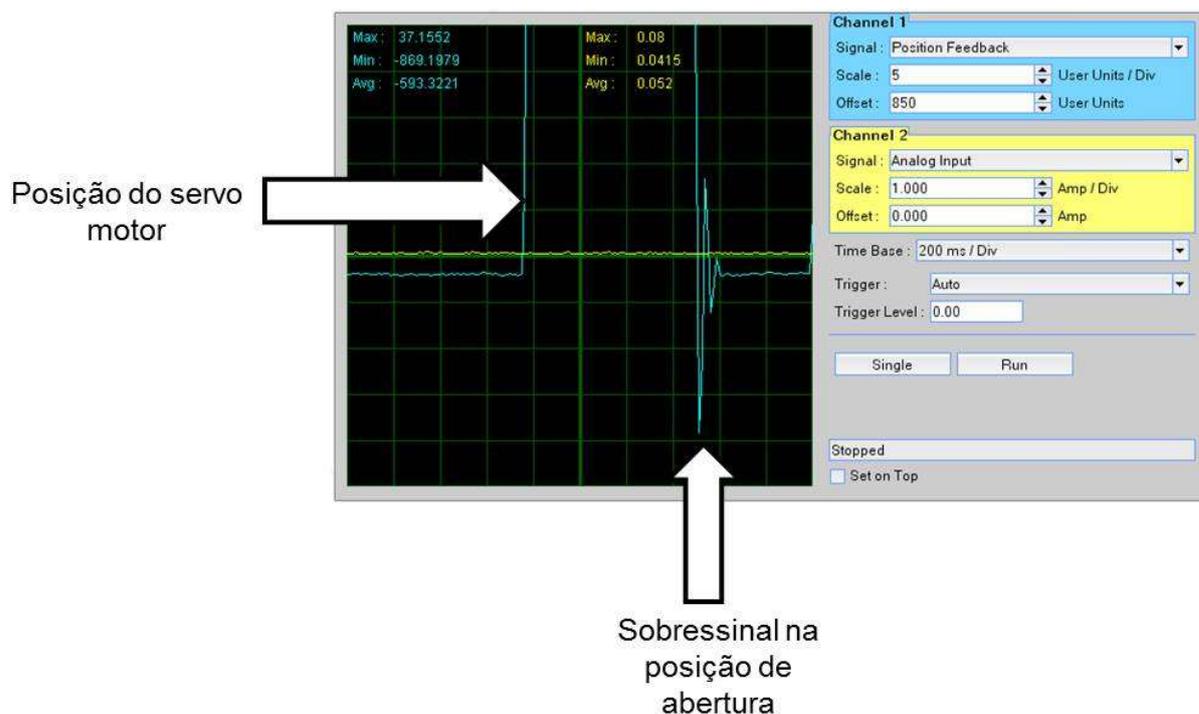


Figura 43 - Sobressinal de posicionamento 2%

Utilizando-se a rede neural artificial testada anteriormente, a variável de posição do eixo virtual foi substituída pela de posição real, fornecida diretamente do servo *drive*, e as saídas da rede neural artificial, os ganhos PKP, PKI, PKD, VKP e VKI foram direcionados para serem escritos no servo *drive*, sempre que houver uma modificação nos valores.

Desta forma, o conjunto do mordente horizontal foi acionado na frequência de trabalho nominal de 60 ciclos por minuto, deslocando-se da posição de abertura para a posição de fechamento e vice-versa, utilizando o escalonamento de ganhos através da rede neural artificial. Pode ser visto na Figura 44 o sobressinal de posicionamento após a utilização do escalonamento dos ganhos. Nota-se que o erro foi reduzido para menos que 0,5%, cumprindo o objetivo de reduzir o erro de sobressinal para menos que 0,5%.

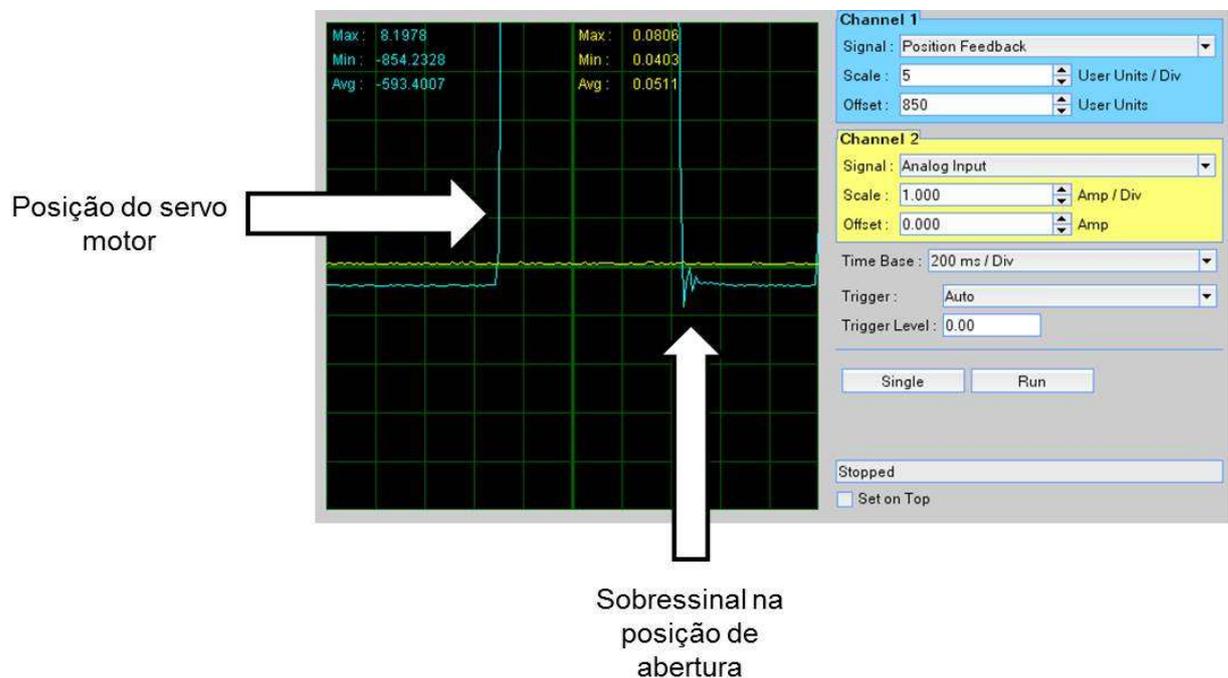


Figura 44 - Sobressinal de posicionamento menor que 0,5%

8.2 Análise da influência na corrente

Foi levantado um gráfico da corrente elétrica do servo motor em função da sua posição sem o escalonamento dos ganhos durante o deslocamento do mordente da posição de abertura para a posição de fechamento e vice-versa a 60 ciclos por minuto e, posteriormente, com a utilização do escalonamento dos ganhos, para que fosse feita uma análise da influência do escalonamento dos ganhos na corrente do servo motor.

Pode ser visto na Figura 45 a curva de posição do servo motor e a curva da corrente instantânea do servo motor sem a utilização do escalonamento dos ganhos. Percebe-se que a corrente variou entre +2,2 e -4,1 Ampere.

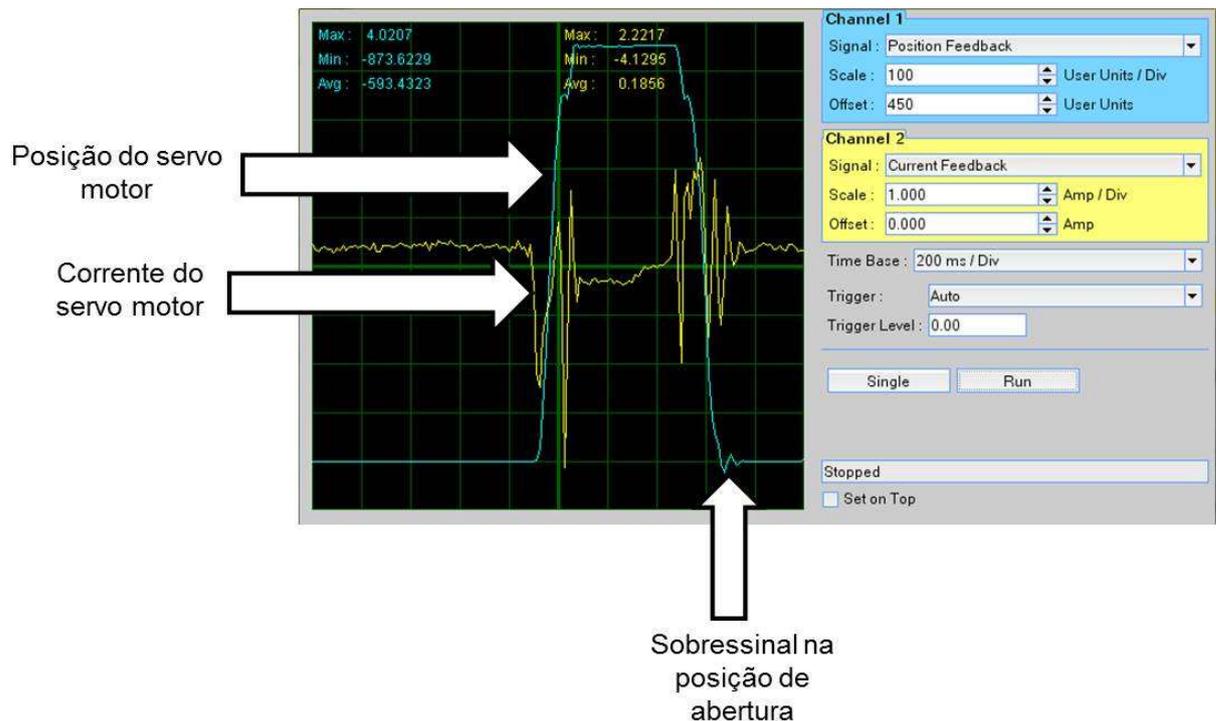


Figura 45 - Posição e corrente do servo motor sem escalonamento dos ganhos

Na Figura 46, pode ser vista a curva de posição do servo motor e sua curva de corrente instantânea. Nota-se que a corrente variou de +2,1 a -2,4 Ampere. Em comparação com o gráfico obtido sem o escalonamento dos ganhos, houve redução na amplitude da corrente de cerca de 28%. A redução na amplitude da corrente é um grande indício de diminuição do consumo elétrico da máquina, o que poderá ser estudado, de forma mais profunda, em um trabalho futuro.

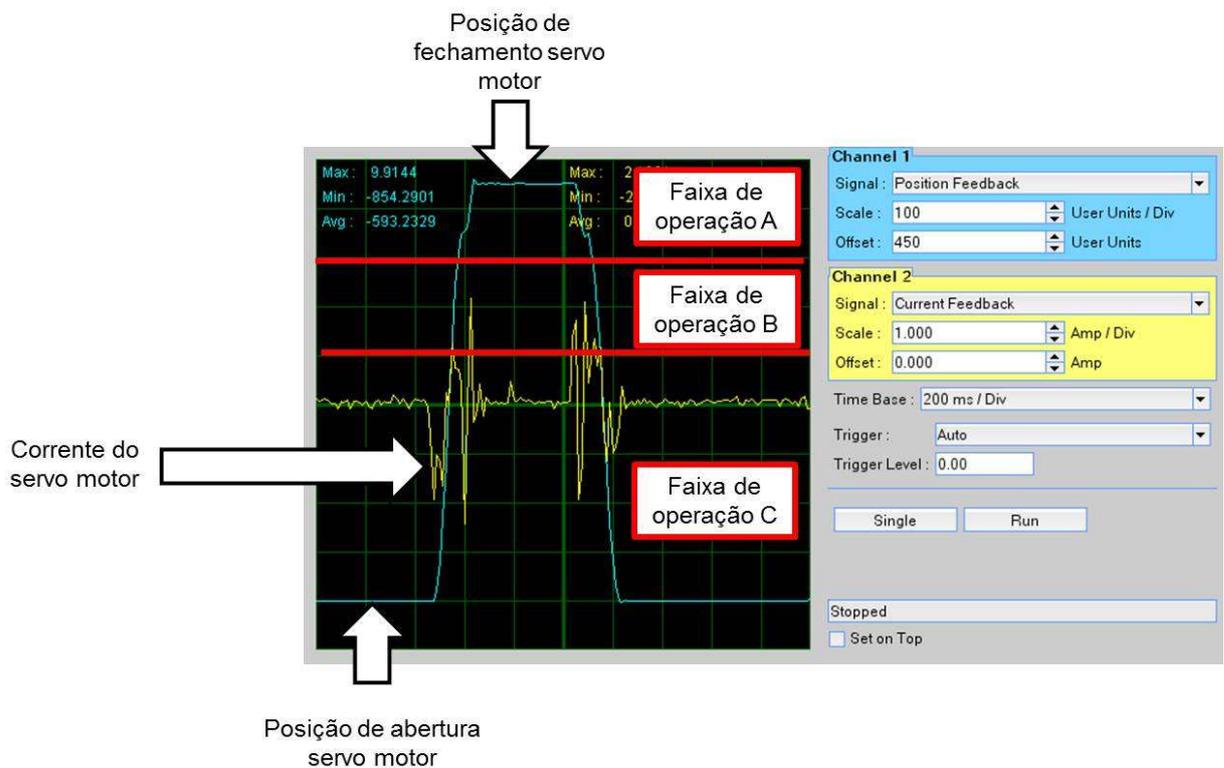


Figura 46 - Posição e corrente do servo motor com escalonamento dos ganhos

8.3 Utilização do método SE

Outra forma de escalonar os ganhos é utilizar o método SE, ou seja, se a posição do servo motor estiver na região de operação A, por exemplo, os valores dos ganhos da região A devem ser escritos no servo *drive*.

Tomando o ganho PKP para análise como exemplo, tem-se a fórmula dos ganhos das regiões A, B e C pelas equações 36, 37 e 38.

$$PKP(A) = \overline{PKP(A)} \quad (36)$$

$$PKP(B) = \overline{PKP(B)} \quad (37)$$

$$PKP(C) = \overline{PKP(C)} \quad (38)$$

Utilizando o gráfico já mostrado anteriormente com a variação dos ganhos e o resultado da simulação da rede neural artificial, foi plotada a curva do valor médio dos ganhos, como pode ser visto no Gráfico 7. Nota-se que a curva do ganho médio mostrou-se muito próxima da curva obtida pela rede neural artificial, com exceção da transição de regiões, que se demonstrou menos atenuada em relação à rede neural artificial.

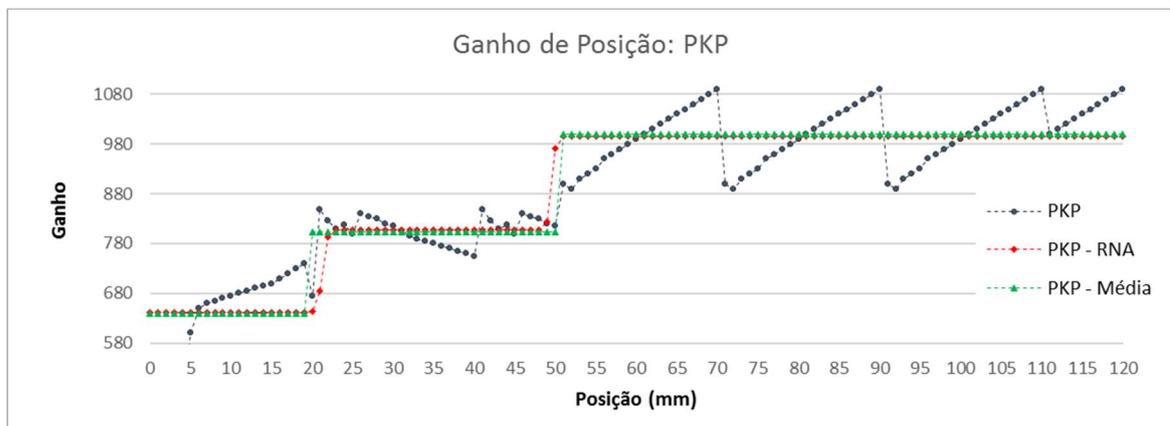


Gráfico 7 - Ganho PKP médio

Para os demais ganhos, o comportamento da curva média foi o mesmo apresentado para o ganho PKP, ou seja, a curva com o valor médio dos ganhos mostrou-se muito próxima à da rede neural artificial, nestes casos, porém, com atenuação menor na transição de regiões.

Desta forma, foi implementada uma lógica SE na linguagem *ladder* para que os ganhos fossem escalonados pelo valor médio quando a posição do servo motor estivesse dentro dos limites de cada região.

Foi aplicada a frequência nominal de trabalho do conjunto de 60 ciclos por minuto e levantado o gráfico de sobressinal de posição para comparar a lógica SE à rede neural artificial e os resultados foram os mesmos obtidos com a rede neural artificial quanto ao sobressinal de posicionamento, ou seja, o sobressinal de posição ficou abaixo de 0,5%.

Foi levantado o gráfico de posição e corrente do servo motor para comparar o comportamento da corrente com a utilização da lógica SE com o comportamento da corrente quando utilizada a rede neural artificial, que pode ser visto na Figura 47. Nota-se que a corrente variou de +2,9 a - 2,4 Ampere, ou seja, a corrente teve uma variação de amplitude maior em cerca de 18%, se comparado ao gráfico gerado utilizando a rede neural artificial, que pode ser atribuído à atenuação dos valores dos ganhos na transição das regiões de operação.

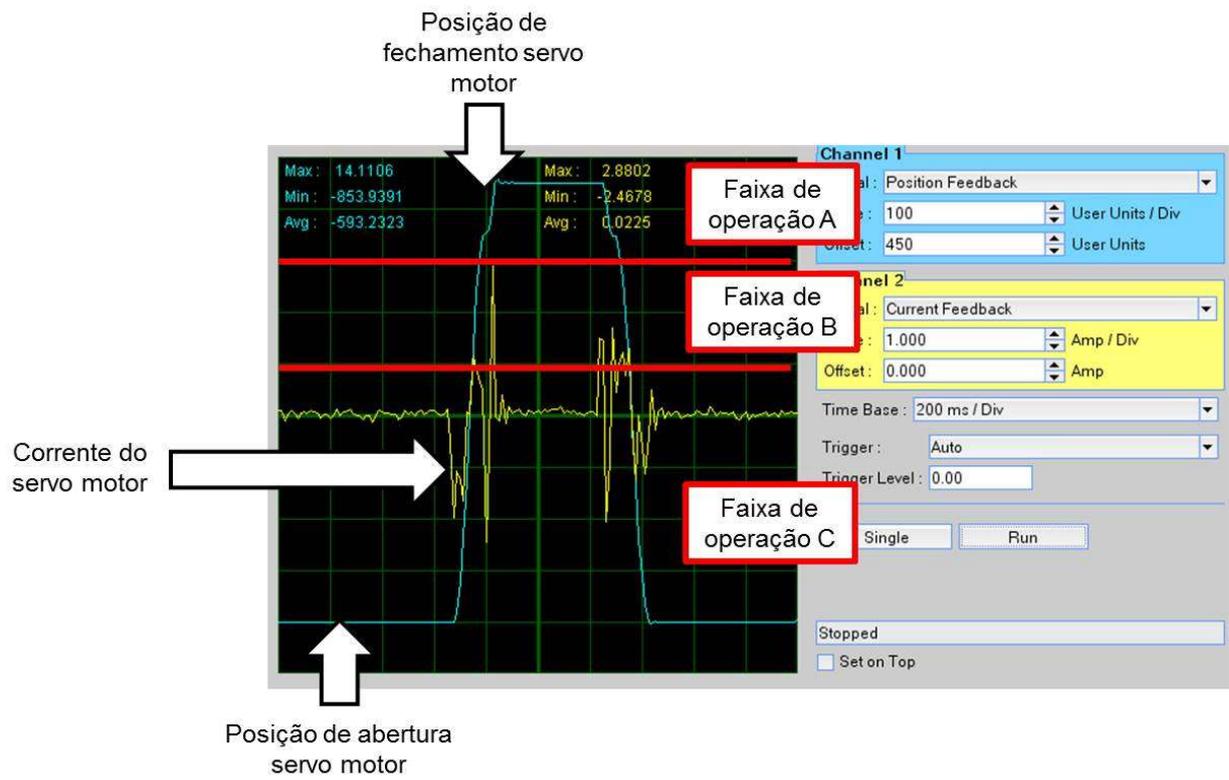


Figura 47 - Posição e corrente pela lógica SE

8.4 Comparação dos resultados

Na Figura 48 pode ser observado um comparativo dos resultados do controle do servo motor durante o movimento sem escalonamento dos ganhos, com escalonamento pela lógica SE e com escalonamento dos ganhos, utilizando a rede neural artificial.

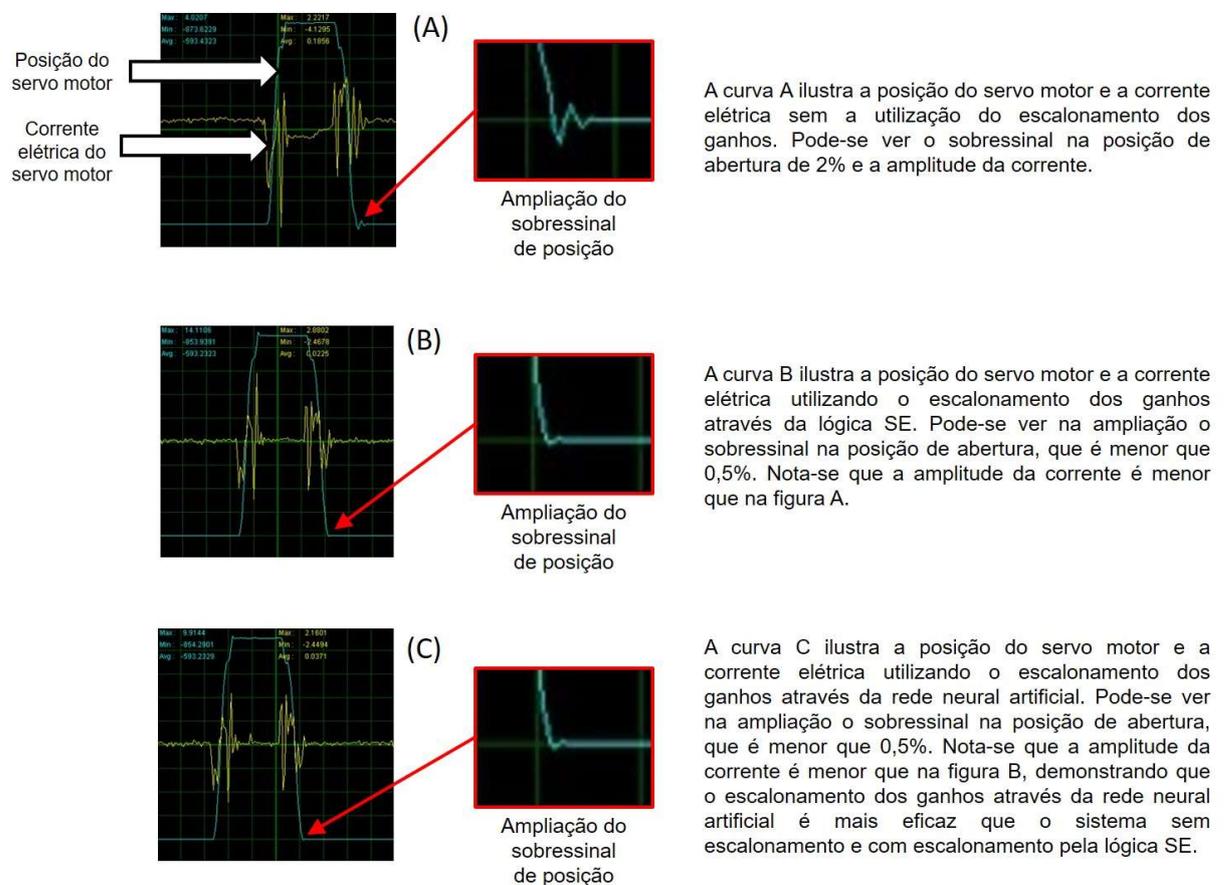


Figura 48 - Comparação dos resultados

9 CONCLUSÃO E TRABALHOS FUTUROS

Os ganhos foram ajustados para cada região com sucesso, assim como a rede neural artificial, que foi simulada no programa computacional Matlab e, posteriormente, embarcada no CLP em linguagem *ladder*, obtendo os mesmos resultados da simulação no Matlab.

A rede neural artificial demonstrou eficácia no escalonamento dos ganhos do controlador, obtendo uma redução do sobressinal de posição para um valor menor que 0,5%. Quanto à corrente elétrica do servo motor, o escalonamento dos ganhos através da rede neural artificial demonstrou uma amplitude menor de corrente, o que é um indício de redução do consumo elétrico da máquina.

Foi testado o método SE, alternativo à rede neural artificial, que demonstrou o mesmo despenho da rede neural artificial quanto ao sobressinal de posicionamento, porém, ao analisar a corrente do servo motor através deste método, constatou-se que o escalonamento dos ganhos através da rede neural artificial obteve uma amplitude menor de corrente elétrica em 18%, embora a aplicação com a lógica SE seja mais simples de ser implementada.

Em trabalhos futuros, seria ideal que o treinamento da rede neural artificial fosse embarcado no CLP de forma a não precisar da utilização do programa computacional Matlab sempre que fosse necessário treinar a rede neural artificial, pois isso tornaria o sistema mais simples de ser utilizado, sem a necessidade de um especialista em Matlab para fazer o treinamento da RNA. Em um trabalho futuro, também poderá ser estudado, de forma mais profunda, a influência da redução da amplitude da corrente elétrica do servo motor no consumo elétrico da máquina quando utilizada a rede neural artificial.

O escalonamento de ganhos do controlador PID/PI através de uma rede neural artificial para o mordente de selagem estudado, que possui diferentes faixas de operação, demonstrou ser eficaz para redução do sobressinal de posicionamento de 2% para menor que 0,5% e ainda apresentou indícios de menor consumo de corrente elétrica para o conjunto analisado, evidenciando, através dos resultados,

ser uma ferramenta que pode ser implementada no sistema atual e produzir o aperfeiçoamento estabelecido no funcionamento da máquina.

REFERÊNCIAS

ABHISHEK, Kumar *et al.* Weather forecasting model using Artificial Neural Networks. **Procedia technology** 4, Bangalore, p. 311-318, 2012. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S221201731200326X>>. Acesso em: 15 de Fevereiro de 2017.

Allen Bradley, Kinetix 300 EtherNet-IP Indexing Servo Drives. Disponível em: <http://literature.rockwellautomation.com/idc/groups/literature/documents/um/2097-um001_-en-p.pdf>. Acesso em: 10 de Janeiro de 2017.

ANDRADE, Lucas H. S. *et al.* Controle PI neural aplicado a um sistema de vazão. **VII SBAI**, Natal-RN, p. 468-473, 2015. Disponível em: <https://www.google.com.br/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&cad=rja&uact=8&ved=0ahUKEwj21tT_1LPRAhUMj5AKHd6zDHIQFggaMAA&url=http%3A%2F%2Fwww.sbai2015.dca.ufrn.br%2Fdownload%2Fartigo%2F136&usg=AFQjCNFcR FhphL4wBuYswTD7IUvgZqF1BA&sig2=T7e26GTMBT8rK7Fnw49eHQ&bvm=bv.143423383,d.Y2I>. Acesso em: 10 de Janeiro de 2017.

ARAI, Masahiko. Bounds on the number of hidden units in binary-valued three-layer neural networks. **Neural Networks**, v.6, n.6, p. 855-860, 1993. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0893608005801303>>. Acesso em: 10 de Janeiro de 2017.

BEALE, Mark Hudson *et al.* **Neural Network Toolbox: User Guide**. V. R2016b, 2016. Disponível em: <https://www.mathworks.com/help/pdf_doc/nnet/nnet_ug.pdf>. Acesso em: 10 de Janeiro de 2017.

BEQUETTE, B. Wayne. **Process Control: Modeling, Design and Simulation**. New Jersey: Prentice Hall, 2002.

BINOTI, Daniel Henrique Breda. Configuração de redes neurais artificiais para estimação do volume de árvores. **Brazil jornal wood science** v.5, p. 58-67, 2014. Disponível em: <<https://periodicos.ufpel.edu.br/ojs2/index.php/cienciadamadeira/article/view/4067>>. Acesso em: 10 de Janeiro de 2017.

FUJITA, Osamu. Statistical estimation of the number of hidden units for feedforward neural networks. **Neural Networks**, v.11, n.5, p. 851-859, 1998. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0893608098000434>>. Acesso em: 10 de Janeiro de 2017.

GAMBOGI, Jarbas Aquiles. **Aplicação de redes neurais na tomada de decisão no mercado de ações**. Dissertação (Mestrado em Engenharia Elétrica) – Universidade de São Paulo, 2013.

HAGAN, Martin T. *et al.* **Neural Network Design**. Boston: PWS Publishing Company, 1996.

HAYKIN, Simon. **Redes neurais: Princípios e práticas**. 2.ed. Porto Alegre: Bookman, 2001.

HECHT-NIELSEN, Robert. Kolmogorov's mapping neural network existence theorem. **IEEE First international conference on neural networks**, California, p. III-11 – III-14, 1987.

HECHT-NIELSEN, Robert. Theory of the backpropagation neural network. **International joint Conference**, Washington, p. 593-605, 1989.

HIROSE, Yoshio *et al.* Back-propagation algorithm which varies the number of hidden units. **Neural Networks**, v.4, n.1, p. 61-66, 1991. Disponível em: <<http://www.sciencedirect.com/science/article/pii/089360809190032Z>>. Acesso em 10 de Janeiro de 2017.

KUMAR, D.Dinesh; Meenakshipriya, B. Design and implementation of non linear system using gain scheduled PI Controller. **Procedia Engineering** 38, p. 3105-3112, 2012. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S1877705812022746>>. Acesso em: 10 de Janeiro de 2017.

MathWorks, Erro médio quadrado. Disponível em: <<https://www.mathworks.com/help/images/ref/immse.html>>. Acesso em: 10 de Janeiro de 2017.

MathWorks, Regularização Bayesiana. Disponível em: <<https://www.mathworks.com/help/nnet/ref/trainbr.html>>. Acesso em: 10 de Janeiro de 2017.

MathWorks, Tangente Hiperbólica Sigmoide. Disponível em: <<https://www.mathworks.com/help/nnet/ref/tansig.html>>. Acesso em: 10 de Janeiro de 2017.

MIGUEZ, Geraldo Azar. **Otimização do algoritmo de backpropagation pelo uso da função de ativação bi-hiperbólica**. Tese (Doutorado em Engenharia de Sistemas e Computação) – Universidade Federal do Rio de Janeiro, 2012.

NUNES, Vitor P. *et al.* Adaptive PID controllers tuning: LMS gain scheduling training and industrial programmable logic controllers. **Congresso brasileiro de redes neurais**, 2007. Disponível em: <<http://www.sba.org.br/rsv/SBAI/SBAI2007/docs/50100064.pdf>>. Acesso em: 10 de Janeiro de 2017.

OGATA, Katsuhiko. **Engenharia de controle moderno**. 4.ed. São Paulo: Pearson, 2003.

OLIVEIRA, Julio C. P. *et al.* Controle neural gain scheduling de tanques interativos. **III SBAI**, Vitória-ES, p. 224-228, 1997. Disponível em: <http://fei.edu.br/sbai/SBAI1997/ARTIGOS/III_SBAI_35.pdf>. Acesso em: 10 de Janeiro de 2017.

PATEL, Raul *et al.* Multilayer neuro PID controller based on back propagation algorithm. **Procedia Computer Science** **54**, p. 207-214, 2015. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S1877050915013472>>. Acesso em: 10 de Janeiro de 2017.

SILVA, Ivan Nunes *et al.* **Redes Neurais Artificiais**: para engenharia e ciências aplicadas. São Paulo: Artliber, 2010.

VISIOLI, Antonio. **Practical PID control**. Londres: Springer, 2006.