

**UNIVERSIDADE DE SOROCABA
PRÓ-REITORIA DE PÓS GRADUAÇÃO, PESQUISA, EXTENSÃO E INOVAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO PROFISSIONAL EM PROCESSOS
TECNOLÓGICOS E AMBIENTAIS**

Diego de Assis Santos

**MÉTODOS PARA CONTROLE DE SISTEMAS ELETROMECAÑICOS POR REDES
NEURAS ARTIFICIAIS**

**Sorocaba/SP
2019**

PRÓ-REITORIA DE PÓS GRADUAÇÃO, PESQUISA, EXTENSÃO E INOVAÇÃO

DIEGO DE ASSIS SANTOS

**MÉTODOS PARA CONTROLE DE SISTEMAS ELETROMECAÂNICOS POR REDES
NEURAS ARTIFICIAIS**

Dissertação de Mestrado, apresentada à Banca Examinadora do Programa de Pós-Graduação Profissional em Processos Tecnológicos e Ambientais, da Universidade de Sorocaba, como exigência parcial para obtenção do título de Mestre em Processos Tecnológicos e Ambientais.

Orientador: Prof. Dr. Waldemar Bonventi Jr.

**Sorocaba/SP
2019**

Ficha Catalográfica

Santos, Diego de Assis
S234m Métodos para controle de sistemas eletromecânicos por redes neurais artificiais / Diego de Assis Santos. – 2019.
147 f. : il.

Orientador: Prof. Dr. Waldemar Bonventi Junior
Dissertação (Mestrado em Processos Tecnológicos e Ambientais)
– Universidade de Sorocaba, Sorocaba, SP, 2019.

1. Redes neurais (Computação). 2. Inteligência artificial. 3. Sistemas dinâmicos. I. Bonventi Junior, Waldemar, orient. II. Universidade de Sorocaba. III. Título.

Diego de Assis Santos

**MÉTODOS PARA CONTROLE DE SISTEMAS ELETROMECAÑICOS POR REDES
NEURAS ARTIFICIAIS**

Dissertação de Mestrado, apresentada à Banca Examinadora do Programa de Pós-Graduação Profissional em Processos Tecnológicos e Ambientais, da Universidade de Sorocaba, como exigência parcial para obtenção do título de Mestre em Processos Tecnológicos e Ambientais.

Aprovado em 22/02/2019

BANCA EXAMINADORA:



Prof. Dr. Waldemar Bonventi Jr.
Universidade de Sorocaba



Prof. Dr. Antonio César Germano Martins
Universidade Estadual Paulista "Júlio de Mesquita Filho"
Câmpus de Sorocaba



Prof. Dr. Eduardo Paciência Godoy
Universidade Estadual Paulista "Júlio de Mesquita Filho"
Câmpus de Sorocaba

AGRADECIMENTOS

Agradeço primeiramente à Deus pelo amparo e auxílio em minha jornada profissional. Agradeço ao corpo docente do Programa de Mestrado em Processos Tecnológicos e Ambientais da Universidade de Sorocaba, pela oportunidade de participar do programa de Mestrado.

Também quero agradecer ao meu orientador, Waldemar Bonventi Jr., da Universidade de Sorocaba. Sempre que eu precisava sanar algumas dúvidas sobre o trabalho, o professor Bonventi estava de portas abertas para me orientar.

Quero agradecer aos meus pais, que são o alicerce da minha educação. Pelos conselhos de meu pai, e pelo amor de minha mãe, meus passos foram bem conduzidos na minha infância.

Finalmente, gostaria de agradecer aos meus colegas do curso de Mestrado em Processos Tecnológicos e Ambientais pela companhia adorável e pelas experiências compartilhadas. Em especial, devo agradecer ao meu colega Willian Patrick pela contribuição na apresentação do trabalho no evento científico da Amostras de Atividades Acadêmicas da Universidade de Sorocaba e à Neusa Valentim, pela orientação na parte de controle de sistemas dinâmicos.

“Seu trabalho vai preencher uma parte grande da sua vida, e a única maneira de ficar realmente satisfeito é fazer o que você acredita ser um ótimo trabalho. É a única maneira de fazer um excelente trabalho é amar o que você faz. “

(Steve Jobs)

RESUMO

Este trabalho apresenta alguns exemplos de sistemas dinâmicos controlados por redes neurais. Grande parte dos sistemas dinâmicos apresentam um grau de complexidade elevado, sendo necessária a implementação de ferramentas computacionais para controlar esses sistemas. As redes neurais artificiais (RNAs) são uma das ferramentas bioinspiradas desenvolvidas pela Inteligência Artificial, que podem ser aplicadas na solução de problemas não lineares. Elas podem ser empregadas na área de controle de processos e devolver resultados satisfatórios, assim como os controladores PID tradicionais. O objetivo deste trabalho é mostrar a aplicação de duas abordagens de controle neural em três sistemas eletromecânicos com diferentes complexidades e dinâmicas conhecidas. Na primeira abordagem, uma rede neural aproxima a função de controle de um controlador PID sintonizado. Essa estratégia se apresentou eficaz, uma vez que a rede reproduz uma ação de controle semelhante ao do PID. Na segunda abordagem, duas classes de controladores neurais são aplicados no controle dos sistemas. Esses controladores apresentaram uma resposta mais lenta em relação ao PID e em alguns casos podem desestabilizar o sinal de saída. Os sistemas adotados diferem nas características de transitório e na ordem: O primeiro é um sistema de terceira ordem auto controlado que necessita de um controle mais rápido, o segundo é um sistema de segunda ordem com baixo amortecimento e o terceiro é um sistema de quarta ordem com baixo amortecimento e longa acomodação. Estes sistemas são adotados com a finalidade de melhorar sua resposta em função das especificações de projeto. Os resultados são obtidos por meio de simulações virtuais analisando-se os índices de desempenho do sinal controlado e comparando com o desempenho de um controlador PID. É demonstrado neste trabalho o processo de treinamento e execução de controle dessas redes.

Palavras-chave: Redes neurais artificiais, Controle neural, Sistema dinâmico.

ABSTRACT

This work presents some examples of dynamic systems controlled by artificial neural networks. Most of these systems show high degree of complexity, so it is necessary to implement adequate computational tools to control these systems. Artificial neural networks (ANNs) are one of the bioinspired tools developed by Artificial Intelligence, which can be applied in solving nonlinear problems. They can be employed in processes control and return satisfactory results, as well as the traditional PID controllers. The objective of this work is to show the application of two approaches of neural control in three electromechanical systems with different complexities and known dynamics. In the first approach, a neural network approaches the control function of a tuned PID controller. This strategy was efficient, since the network reproduce a control action similar to the PID. In the second, two classes of neural controllers are applied in the control of the systems. These controllers may respond more slowly to the PID and in some cases may destabilize the output signal. The systems adopted differ in transient and order: The first is a self-controlled third order system that needs a faster response, the second is a second order system with a low damping and the third is a fourth order system with a low damping and long accommodation. These systems are adopted with the purpose of improve their response according to the design specifications. The results are obtained through virtual simulations analyzing the performance indices of the controlled signal and comparing it with the performance of a PID controller. In this works it is demonstrated the training process and the control execution of these networks.

Keywords: Neural Networks, Neural control, Dynamic systems

LISTA DE ILUSTRAÇÕES

| | |
|---|----|
| Figura 1 - Modelo para a tomada de decisões..... | 19 |
| Figura 2 - Estrutura de um Sistema Especialista | 20 |
| Figura 3 – Diagrama de blocos do modelo de identificação da planta..... | 25 |
| Figura 4 - Arquiteturas de controle usando redes neurais artificiais. | 28 |
| Figura 5 - Modelo de aproximação. | 29 |
| Figura 6 – Sistema MIMO – Ato de dirigir o carro..... | 32 |
| Figura 7 - Sistema de controle..... | 33 |
| Figura 8 - índices de desempenho da resposta transitória | 36 |
| Figura 9 - Sistema estável com resposta natural tendendo a um. | 37 |
| Figura 10 – Sistema instável com resposta natural tendendo a infinito | 38 |
| Figura 11 – Sistema marginalmente estável com resposta natural oscilatória | 38 |
| Figura 12 – erro de estado estacionário com entrada em degrau | 40 |
| Figura 13 – erro de estado estacionário com entrada em rampa | 41 |
| Figura 14 – representação do erro estacionário em diagrama de blocos em malha fechada | 42 |
| Figura 15 – Modelo não-linear de neurônio | 46 |
| Figura 16 – Função de ativação limiar | 47 |
| Figura 17 – Função de ativação linear por partes..... | 48 |
| Figura 18 – Função de ativação sigmóide | 49 |
| Figura 19 - Estrutura primária de um controlador PID neural. | 50 |
| Figura 20 – Rede tipo <i>multilayer perceptron</i> com uma camada oculta e uma camada de saída | 51 |
| Figura 21 - Configuração de controle modelo de aproximação | 52 |
| Figura 22 – Configuração <i>Model Reference Adaptive Control</i> (MRAC)..... | 53 |
| Figura 23 - Detalhes das RNAs no modelo de referência..... | 53 |
| Figura 24 - Rede NARMA-L2 configurada para a etapa de identificação do sistema. | 54 |
| Figura 25 - Rede NARMA-L2 configurada para a etapa de controle. | 55 |
| Figura 26 - Decaimento do erro quadrático médio durante o treinamento da rede..... | 59 |
| Figura 27 – Verificação de estabilidade da antena de azimute..... | 62 |
| Figura 28 – Verificação de estabilidade do motor de tração..... | 63 |
| Figura 29 – Verificação de estabilidade da suspensão ativa | 63 |
| Figura 30 - componentes da antena de azimute..... | 64 |
| Figura 31 - sistema antena de azimute em blocos funcionais | 65 |
| Figura 32 – Diagrama de blocos da antena de azimute com as funções de transferência.... | 65 |

| | |
|--|----|
| Figura 33 – Diagrama de blocos do sistema antena de azimute com as constantes inseridas | 66 |
| Figura 34 – Esboço do motor CC | 67 |
| Figura 35 – Diagrama de blocos do acionamento de um motor elétrico para veículo de ferrovia | 68 |
| Figura 36 – Controle de velocidade de um motor elétrico de tração..... | 69 |
| Figura 37 – Diagrama de blocos do sistema de controle de velocidade do motor CC..... | 69 |
| Figura 38 - Modelo de suspensão ativa (1/4)..... | 70 |
| Figura 39 – Diagrama de blocos completo da suspensão ativa..... | 71 |
| Figura 40 – Diagrama de blocos do sistema suspensão ativa ¼..... | 72 |
| Figura 41 - curva de resposta transitória para uma entrada em degrau unitário do sistema antena de azimute | 76 |
| Figura 42 - Janela de configuração do controlador PID para o sistema antena de azimute.. | 77 |
| Figura 43 - Janela de ajuste interativo do sistema antena de azimute | 77 |
| Figura 44 - resposta transitória da antena sob ação do controlador PID | 78 |
| Figura 45 - Decaimento do erro quadrático médio durante o treinamento da rede MLP <i>feed-forward</i> | 79 |
| Figura 46 - curva de resposta transitória da antena sob ação | 79 |
| Figura 47 - tela de configuração da rede NARMA-L2 com os parâmetros de simulação do sistema antena de azimute..... | 81 |
| Figura 48 - Gráfico do treinamento da rede NARMA-L2..... | 81 |
| Figura 49 - Decaimento do erro quadrático médio durante o treinamento da rede..... | 82 |
| Figura 50 - Curva de resposta da antena controlada pelo controlador neural NARMA-L2 | 82 |
| Figura 51 - Janela de configuração da rede de modelo..... | 84 |
| Figura 52 - Janela de configuração da rede controladora | 85 |
| Figura 53 – Dados de treinamento gerados por degrau aleatório..... | 85 |
| Figura 54 – Desempenho do treinamento da rede de modelo..... | 86 |
| Figura 55 – Conjunto de dados gerados para o treinamento da rede controladora..... | 87 |
| Figura 56 – Desempenho do treinamento da rede controladora..... | 87 |
| Figura 57 – Atuação da rede neural comparada a um modelo de sistema controlado | 87 |
| Figura 58 – Resposta da antena sob ação do controlador neural modelo de referência | 88 |
| Figura 59 – comportamento dinâmico do motor CC de veículo de ferrovia | 89 |
| Figura 60 - Janela de configuração do controlador PID para o sistema motor de tração | 90 |
| Figura 61 - Janela de ajuste interativo do sistema motor de tração..... | 90 |
| Figura 62 – resposta do motor de tração sob ação do controlador PID..... | 91 |
| Figura 63 - Decaimento do erro quadrático médio durante o treinamento da rede..... | 92 |

| | |
|--|-----|
| Figura 64 – resposta do motor de tração sob ação do controlador neural pelo método de aproximação | 93 |
| Figura 65 - tela de configuração da rede NARMA-L2 com os parâmetros de simulação do sistema motor de tração. | 94 |
| Figura 66 - gráfico do treinamento da rede NARMA-L2 para o motor de tração..... | 95 |
| Figura 67 - Decaimento do erro quadrático médio durante o treinamento da rede..... | 95 |
| Figura 68 – resposta do motor de tração sob ação do controlador neural NARMA-L2..... | 96 |
| Figura 69 - Janela de configuração da rede de modelo..... | 97 |
| Figura 70 - Janela de configuração da rede controladora | 98 |
| Figura 71 – Dados de treinamento gerados por degrau aleatório..... | 99 |
| Figura 72 – Desempenho do treinamento da rede de modelo..... | 99 |
| Figura 73 – Conjunto de dados gerados para o treinamento da rede controladora..... | 100 |
| Figura 74 – Desempenho do treinamento da rede controladora..... | 100 |
| Figura 75 - Atuação da rede neural comparada a um modelo de sistema controlado..... | 100 |
| Figura 76 – Resposta do motor sob ação do controlador neural modelo de referência..... | 101 |
| Figura 77 – resposta dinâmica do sistema suspensão ativa $\frac{1}{4}$ | 102 |
| Figura 78 - Janela de configuração do controlador PID para o sistema suspensão ativa $\frac{1}{4}$ | 103 |
| Figura 79 – resposta da suspensão ativa $\frac{1}{4}$ sob ação do controlador PID..... | 104 |
| Figura 80 - Decaimento do erro quadrático médio durante o treinamento da rede..... | 105 |
| Figura 81 – resposta da suspensão ativa $\frac{1}{4}$ sob ação do controlador neural pelo método de aproximação | 106 |
| Figura 82 - tela de configuração da rede NARMA-L2 com os parâmetros para o sistema suspensão ativa $\frac{1}{4}$ | 108 |
| Figura 83 – gráfico de treinamento da rede NARMA-L2 para a suspensão ativa. | 108 |
| Figura 84 - Decaimento do erro quadrático médio durante o treinamento da rede..... | 109 |
| Figura 85 – suspensão ativa $\frac{1}{4}$ sob ação do controlador neural NARMA-L2..... | 109 |
| Figura 86 - Janela de configuração da rede de modelo..... | 111 |
| Figura 87 - Janela de configuração da rede controladora | 112 |
| Figura 88 – Dados de treinamento gerados por degrau aleatório..... | 112 |
| Figura 89 – Desempenho do treinamento da rede de modelo..... | 113 |
| Figura 90 – Conjunto de dados gerados para o treinamento da rede controladora..... | 113 |
| Figura 91 – Desempenho do treinamento da rede controladora..... | 114 |
| Figura 92 - Atuação da rede neural comparada a um modelo de sistema controlado..... | 114 |
| Figura 93 – Resposta do motor sob ação do controlador neural modelo de referência..... | 115 |
| Figura 94 – Desempenho dos controladores para uma entrada em degrau do sistema antena de azimute. | 117 |

| | |
|--|-----|
| Figura 95 - Desempenho de todos os controladores para uma entrada de degraus aleatórios do sistema antena de azimute. | 118 |
| Figura 96 - Desempenho dos controladores para uma entrada em degrau do sistema motor de tração. | 119 |
| Figura 97 - Desempenho de todos os controladores para uma entrada de degraus aleatórios do sistema motor de tração. | 120 |
| Figura 98 - Desempenho dos controladores para uma entrada em degrau do sistema suspensão ativa $\frac{1}{4}$ | 121 |
| Figura 99 – Desempenho de todos os controladores para o sistema suspensão ativa $\frac{1}{4}$ | 122 |
| Figura 100 - Caixa de diálogo do aplicativo Neural Net Fitting. | 130 |
| Figura 101 - Coleta de dados do PID para treinamento das redes MLP pelo método de aproximação. | 131 |
| Figura 102 – caixa de diálogo do conjunto de dados de entrada e saída a serem utilizados no treinamento. | 131 |
| Figura 103 – Determinação da divisão dos dados em grupos de treinamento, validação e teste. | 132 |
| Figura 104 – Configuração do número de neurônios da camada oculta da rede. | 132 |
| Figura 105 - Inicialização do treinamento da rede MLP. | 133 |
| Figura 106 – Processo de treinamento da rede MLP | 134 |
| Figura 107 - Gráfico de decaimento do Erro Médio Quadrático. | 134 |
| Figura 108 - Tela de avaliação da rede MLP. | 135 |
| Figura 109 – Tela de implantação de solução | 135 |
| Figura 110 – Tela de configuração do controlador neural NARMA-L2. | 136 |
| Figura 111 – Geração de dados de treinamento para a rede NARMA-L2. | 137 |
| Figura 112 – Tela de configuração da rede NARMA-L2, iniciando treinamento | 138 |
| Figura 113 – Desempenho do treinamento da rede NARMA-L2. | 138 |
| Figura 114 - Desempenho da rede em relação ao grupo de dados de treinamento. | 139 |
| Figura 115 - Decaimento do erro médio quadrático após o treinamento da rede. | 139 |
| Figura 116 - Janela de configuração da rede controladora | 141 |
| Figura 117 - Janela de configuração da rede de modelo. | 141 |
| Figura 118 – Processo de geração de dados para o treinamento da rede de modelo | 142 |
| Figura 119 – Desempenho do treinamento da rede controladora. | 142 |
| Figura 120 - Janela de configuração do controlador PID do software matlab. | 144 |
| Figura 121 - Caixa de diálogo do bloco PID com o desempenho inicial | 145 |
| Figura 122 - Conjunto de medições de desempenho e robustez. | 145 |
| Figura 123 - Resposta de malha fechada nova configuração. | 146 |

Figura 124 - Caixa de diálogo do controlador PID com parâmetros P, I, D e N atualizados.

..... 146

LISTA DE TABELAS

| | |
|--|-----|
| Tabela 1 - Principais entradas de teste | 34 |
| Tabela 2 – estrutura da tabela de Routh | 39 |
| Tabela 3 - resposta do sistema controlado em função dos parâmetros do PID..... | 44 |
| Tabela 4 - variáveis do sistema antena de azimute..... | 65 |
| Tabela 5 - contém os parâmetros das constantes do sistema antena de azimute | 66 |
| Tabela 6 - valores dos parâmetros do sistema de suspensão ativa | 70 |
| Tabela 7 – Parâmetros de desempenho transitório | 76 |
| Tabela 8 - Parâmetros de desempenho transitório da antena | 78 |
| Tabela 9 - Parâmetros de desempenho transitório da antena sob ação do controlador neural pelo método de aproximação..... | 80 |
| Tabela 10 - Parâmetros de desempenho transitório da antena controlada pelo controlador neural NARMA-L2..... | 83 |
| Tabela 11 - Parâmetros de desempenho transitório | 88 |
| Tabela 12 - Parâmetros de desempenho transitório | 89 |
| Tabela 13 - parâmetros de desempenho transitório do motor de tração sob ação do controlador PID..... | 91 |
| Tabela 14 – parâmetros de desempenho transitório do motor de tração sob ação do controlador neural pelo método de aproximação..... | 93 |
| Tabela 15 – parâmetros de desempenho transitório do motor sob ação do controlador neural | 96 |
| Tabela 16 - Parâmetros de desempenho transitório..... | 101 |
| Tabela 17 - Parâmetros de desempenho transitório do sistema suspensão ativa 1/4 de veículo | 102 |
| Tabela 18 – parâmetros de desempenho transitório da suspensão ativa $\frac{1}{4}$ sob ação do controlador PID..... | 104 |
| Tabela 19 – parâmetros de desempenho transitório da suspensão ativa 1/4 sob ação do controlador neural pelo método de aproximação..... | 106 |
| Tabela 20 – parâmetros de desempenho transitório da suspensão ativa 1/4 sob ação do controlador neural modelo de referência | 110 |
| Tabela 21 - Parâmetros de desempenho transitório..... | 115 |
| Tabela 22 - Desempenho dos controladores do sistema antena de azimute | 117 |
| Tabela 23 - Desempenho dos controladores do sistema motor de tração..... | 119 |
| Tabela 24 - Desempenho dos controladores do sistema suspensão ativa..... | 121 |

LISTA DE SIGLAS E ABREVIATURAS

| | |
|-----|--------------------------|
| AO | Amplificador operacional |
| CC | Corrente contínua |
| DB | Diagrama de blocos |
| FT | Função de transferência |
| FA | Função de ativação |
| RNA | Rede Neural Artificial |

SUMÁRIO

| | | |
|----------|---|-----------|
| 1 | INTRODUÇÃO | 18 |
| 1.1 | Importância do uso de redes neurais artificiais em controle de sistemas | 21 |
| 1.2 | Histórico do uso de redes neurais artificiais em controle de sistemas | 22 |
| 1.3 | Aplicações das redes neurais em controle | 24 |
| 1.3.1 | Redes neurais para a Identificação da planta | 24 |
| 1.3.2 | Configurações de redes neurais utilizadas para controle de planta | 26 |
| 1.4 | Limitações e precauções | 27 |
| 1.5 | Objetivo | 29 |
| 1.6 | Sistemas e configurações utilizadas neste trabalho | 30 |
| 2 | REFERÊNCIAS CONCEITUAIS | 31 |
| 2.1 | Controle de sistemas | 31 |
| 2.1.1 | Funções de transferência em controle | 31 |
| 2.1.2 | Entradas de teste | 33 |
| 2.1.3 | Índices de desempenho transitório | 35 |
| 2.1.4 | Estabilidade dos sistemas | 37 |
| 2.1.5 | Erro em estado estacionário | 40 |
| 2.1.6 | Controlador PID e sintonia | 43 |
| 2.2 | Redes neurais artificiais | 44 |
| 2.2.1 | Tipos de funções de ativação | 47 |
| 2.2.2 | Topologias (ou arquiteturas) de RNAs | 50 |
| 2.2.3 | Aprendizado das redes | 56 |
| 2.2.4 | Treinamento, teste, validação – generalização | 57 |
| | Conclusão do capítulo | 59 |
| 3 | MODELAGEM DOS SISTEMAS DINÂMICOS ESCOLHIDOS | 61 |
| 3.1 | Análise de estabilidade dos sistemas | 61 |
| 3.2 | Antena de azimute | 64 |
| 3.3 | Motor elétrico CC | 67 |
| 3.4 | Suspensão ativa ¼ de roda | 69 |
| 4 | SIMULAÇÕES REALIZADAS E RESULTADOS | 74 |
| 4.1 | Algoritmos de otimização utilizados para a simulação | 74 |
| 4.2 | Modelos de planta utilizados para o treinamento das redes NARMA-L2 e Modelo de Referência | 74 |
| 4.3 | Especificações de projeto | 75 |
| 4.4 | Antena de azimute, sistema sem controle | 75 |
| 4.4.1 | Controle por PID sintonizado | 76 |
| 4.4.2 | Controle por RNA substituindo PID | 78 |
| 4.4.3 | Rede Neural de controle NARMA-L2 | 80 |

| | | |
|------------|--|------------|
| 4.4.4 | Controle de modelo de referência | 83 |
| 4.5 | Motor elétrico de tração de veículo de ferrovia | 88 |
| 4.5.1 | Controle por PID sintonizado | 89 |
| 4.5.2 | Controle por RNA substituindo PID | 91 |
| 4.5.3 | Rede Neural de controle NARMA-L2 | 93 |
| 4.5.4 | Controle de modelo de referência | 96 |
| 4.6 | Suspensão ativa | 102 |
| 4.6.1 | Controle por PID sintonizado | 103 |
| 4.6.2 | Controle por RNA substituindo PID | 104 |
| 4.6.3 | Rede Neural de controle NARMA-L2 | 106 |
| 4.6.4 | Controle de modelo de referência | 110 |
| 5 | DISCUSSÃO DOS RESULTADOS..... | 116 |
| 6 | CONCLUSÕES | 123 |
| | REFERÊNCIAS | 126 |
| | APÊNDICE A – TREINAMENTO DAS REDES MULTICAMADAS <i>FEED-FOWARD</i> VIA SIMULINK | 130 |
| | APÊNDICE B – TREINAMENTO DO CONTROLADOR NARMA-L2 VIA SIMULINK | 136 |
| | APÊNDICE C – TREINAMENTO DO CONTROLADOR NEURAL MODELO DE REFERÊNCIA | 140 |
| | APÊNCICE D – SINTONIA DO CONTROLADOR PID NO SIMULINK..... | 143 |

1 INTRODUÇÃO

Os métodos baseados em Inteligência Artificial (frequentemente denominados “métodos ou técnicas inteligentes”) tem sido cada vez mais aplicadas na área de Controle e Automação, atingindo estágios crescentes em maturidade. As novas contribuições são embasadas em fundamentos cada vez mais sólidos. Por outro lado, se tem observado que as técnicas baseadas em conceitos inteligentes, aplicadas em controle e automação, não substituem as técnicas clássicas (controle ótimo, robusto, adaptativo, estocástico e outros), mas são empregadas para complementá-las, criando alternativas de abordagem. Muitas vezes não é vantajoso substituir um controle PID (proporcional-integrativo-derivativo) bem sintonizado por um sistema baseados em conhecimentos, mas podem ser utilizados para supervisionar grupos de controladores em sistemas mais complexos, reprogramação e reconfiguração, ou outras tarefas de alto nível, como gerência de contingências ou em recuperação de falhas (NASCIMENTO Jr., YONEYAMA, 2000).

Um sistema que utiliza Inteligência Artificial (IA) atua no sentido de imitar e integrar diversas tarefas e comportamentos considerados inteligentes. Por exemplo, sistemas dotados de sensores, percepção do ambiente, bancos de dados e de conhecimentos, de modo que possa realizar inferências e tomar decisões de modo autônomo. Um conceito fundamental em IA é o *aprendizado*, em que conceitos são adquiridos e ocorre a estruturação do conhecimento. O aprendizado em máquinas deu origem a uma subárea da IA (*machine learning*) com o objetivo de compreender, implementar e automatizar o processo de aprendizado. O aprendizado pode ocorrer por meio de *descoberta, exemplos* (aprendizado indutivo) ou por *programação* (RUSSELL, NORVIG, 2003).

A conjunção das potencialidades oferecidas pela IA com os métodos de controle abre um vasto horizonte para lidar com sistemas muito complexos, de descrição matemática muito extensa e, por vezes, desconhecida, como também ampliar a capacidade adaptativa e decisória de tais sistemas.

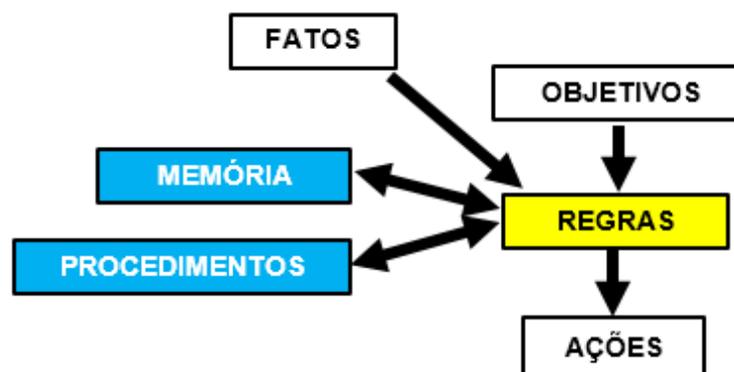
Para a solução de problemas complexos e funções não lineares, foi desenvolvida a Inteligência Artificial (IA), que é uma ferramenta computacional que, conforme Costa (2009, p. 390), ela “compreende os métodos, ferramentas e sistemas para solucionar problemas que normalmente requer o uso da inteligência humana”. Há vários sistemas que são considerados como sistemas inteligentes, como por exemplo:

- ✓ Sistemas especialistas (SE);
- ✓ Sistemas baseados na lógica fuzzy;
- ✓ Sistemas baseados em redes neurais (RN);

- ✓ Sistemas baseados em algoritmos genéticos (AG).

Os sistemas especialistas são também chamados de *sistemas baseados em conhecimento*. Nesses sistemas há um motor de inferência responsável por usar os dados disponíveis e os conhecimentos armazenados na sua base para gerar novos dados de forma interativa até obter a solução (CAMPOS; SAITO, 2004). Para este sistema o motor é o “coração dinâmico” do SE, porque explora a base de conhecimentos através dos dados recebidos e, por fim, emite os resultados. Este sistema é muito parecido com o modelo para a tomada de decisões do ser humano.

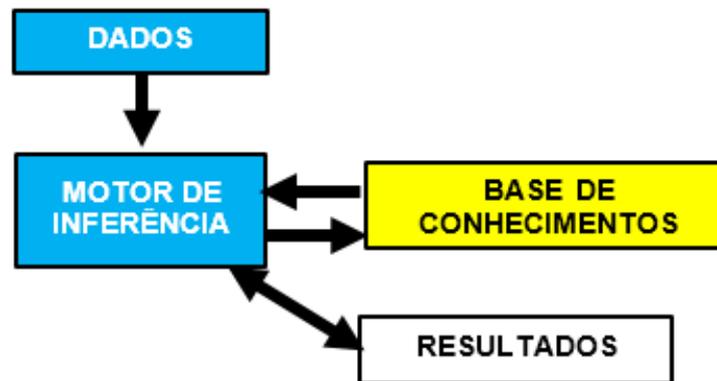
Figura 1 - Modelo para a tomada de decisões



Fonte: CAMPOS M.; SAITO K. **Sistemas Inteligentes em Controle e Automação de Processos**. Editora Moderna Ltda., 2004. Rio de Janeiro. Adaptado pelo próprio autor.

A Figura 1 mostra um esquema do modelo de tomada de decisões baseado no raciocínio e decisão do ser humano, pois é necessário ter fatos e objetivos do domínio do problema, processá-los por meio de regras que estão a todo tempo na memória e seguindo procedimentos, fazendo uma busca seletiva do conhecimento necessário à ação, o que leva o nome de “regras de inferência”. Assim, são obtidos novos fatos por meio dos sentidos levando esses às memórias declarativa (memória que contém todos os conhecimentos de fatos do ser humano) e procedural (memória que contém registradas todas as competências de um ser humano, chamadas também de capacidade de se andar, dirigir, jogar futebol etc.), relativas a conhecimentos declarativos associados ao problema e conhecimentos procedurais (CAMPOS; SAITO, 2004). A Figura 2 mostra um sistema parecido construído para a estrutura de um SE.

Figura 2 - Estrutura de um Sistema Especialista



Fonte: CAMPOS M.; SAITO K. *Sistemas Inteligentes em Controle e Automação de Processos*. Editora Moderna Ltda., 2004. Rio de Janeiro. Adaptado pelo próprio autor.

Na Figura 2 é visto como os dados agem como os fatos e objetivos que são inseridos no núcleo dinâmico do sistema (Motor de Inferência), enviando informações para a base de conhecimentos e recebendo ao mesmo tempo novas informações gerando assim os resultados (ações).

Para este trabalho, o sistema inteligente abordado foi o baseado em redes neurais artificiais (RNAs). As RNAs “são modelos matemáticos que tentam simular o funcionamento do cérebro humano com o objetivo de resolver problemas da vida real. Elas são utilizadas nas mais diversas áreas de aplicação como previsão, classificação, controle entre outros” (GONÇALVES; SILVA; ALMEIDA, 2015, p. 46).

RNAs são “também conhecidas como conexionismo ou sistemas de processamento paralelo e distribuído. Esta forma de computação não-algorítmica é caracterizada por sistemas que, em algum nível, relembram a estrutura do cérebro humano. Por não ser baseada em regras ou programas, a computação neural se constitui em uma alternativa à computação algorítmica convencional. [...] O funcionamento destas redes é inspirado em uma estrutura física conhecida pela natureza: o cérebro humano.” (BRAGA; LUDEMIR; CARVALHO, 2000).

As estruturas das RNAs são baseadas no sistema nervoso humano, composta por *nós* interconectados. Os *nós* representam os neurônios humanos, capazes de processar os sinais de entrada e transmiti-los para outros neurônios conectados a ele. As redes neurais são compostas de camadas de neurônios artificiais que processam os sinais transmitidos da camada de entrada até a de saída. Os neurônios artificiais que compõem a rede “são as

unidades básicas de processamento da informação, projetadas para simular o comportamento de neurônios biológicos.” (CERQUEIRA, ANDRADE, POPPI, 2001, p. 865).

1.1 Importância do uso de redes neurais artificiais em controle de sistemas

A rede neural não requer nenhum conhecimento prévio dos relacionamentos funcionais entre as variáveis de entrada e saída. Ao invés disso, a rede neural simplesmente gera um mapeamento da entrada-saída a partir dos exemplos utilizados no treinamento.

Portanto, a primeira vantagem da abordagem utilizando redes neurais, comparada com a identificação clássica de sistemas dinâmicos, é que evitamos a especificação da estrutura do modelo, a qual pode ser muito difícil. Isto fica evidente quando a estrutura do sistema real não pertence ao conjunto de modelos que podem ser gerados pelo método selecionado para identificação. Os modelos aqui utilizados requerem experimentos cuidadosamente controlados sobre o intervalo de operação do processo. Conforme explica Lima (2000):

“Para sistemas não-lineares a serem identificados usando esses modelos lineares, o conjunto de dados deve ser separado em subconjuntos onde o processo é aproximadamente linear. Por outro lado, redes neurais permitem que o mapeamento de entrada-saída seja detectado sem qualquer entendimento do relacionamento funcional entre entrada e saída. A desvantagem desta técnica é a necessidade de especificar dados de entrada para a rede neural que cubra toda a região de interesse: sem conhecimento da estrutura do sistema real, os resultados dos experimentos não podem ser extrapolados para fora da região onde o experimento foi conduzido. Portanto, identificação utilizando redes neurais requer pouca intervenção do usuário na definição dos relacionamentos estruturais entre as entradas e saídas do processo. Entretanto, experimentos de identificação utilizando redes neurais podem necessitar de um conjunto de dados (pares de entrada-saída) maior - talvez em ordens de magnitude – em relação ao experimento de identificação clássica.”

Conclui-se que o emprego de redes neurais artificiais em controle de sistemas é um campo ainda muito vasto de investigação para a proposta de modelos e práticas experimentais.

Utilizar uma rede neural para as situações que envolvem sistemas complexos pode ser vantajoso, visto que elas possuem capacidade adaptativa (COSTA, 2009, p. 390). Essa propriedade permite que a rede ajuste seus valores de ponderação das ligações (sinapses ou conexões entre os neurônios) após receber o conjunto de entrada a ser aprendido várias vezes. Esse processo é caracterizado como aprendizado da RNA.

As RNAs são muito atrativas no que demanda soluções de problemas, pois essas redes criam a possibilidade de desempenho maior comparados aos modelos convencionais (BRAGA; LUDEMIR; CARVALHO, 2000). Porém, antes de solucionar o problema, essas redes passam por um processo de aprendizagem, isto é, um conjunto de exemplos são apresentados para a rede, a qual automaticamente detecta e absorve as características necessárias para representar o dado fornecido. Estas características são mais tarde interpretadas e usadas para dar a solução ao problema. Além disso, as RNAs são capazes de atuar como mapeadores universais de funções de muitas variáveis, são capazes de fazer auto-organização e desenvolvem processamento temporal o que as tornam ainda mais atrativas no âmbito de solucionar problemas complexos.

As redes neurais possuem outra característica relevante que é a capacidade de atualizar de maneira incremental os seus parâmetros, à medida que novos dados de treinamento estejam disponíveis (TEIXEIRA, 2016). Isso se torna particularmente interessante para sistemas que alteram suas configurações devido a mudanças climáticas, por exemplo. Para a área de controle, essa característica pode ser muito útil, uma vez que a rede neural se torna um controlador de auto-aprendizagem que fornece uma resposta mais rápida e melhor em comparação a outros controladores e fornece a saída de sobressinal de zero (YADAV, CHANDRA, 2015).

1.2 Histórico do uso de redes neurais artificiais em controle de sistemas

O primeiro estudo sobre neurocontroladores foi discutido por Widrow e Smith em 1963, em que foi utilizada uma rede neural do tipo ADALINE para estabilizar e controlar o equilíbrio de um pêndulo invertido. (OMATU, 2010)

De acordo com Omatu et al. (2010), “essa área de pesquisa se desenvolveu em torno 1987, quando a primeira conferência IEEE sobre redes neurais foi realizada em San Diego”. Os trabalhos apresentados nesta conferência “demonstraram que os métodos de neurocontroladores podem ser aplicados com sucesso para controlar sistemas não lineares desconhecidos”, enquanto as “abordagens convencionais de controle baseado na teoria do sistema dinâmico linear não poderiam resolver esses problemas de controle”.

Muitas estruturas de neurocontroladores foram desenvolvidas. Os métodos mais comuns de controle são: aprendizado por realimentação do erro; modelo de controle neurointerno; controle neuropreditivo; Modelagem direta e inversa; aprendizado generalizado e especializado; neurocontrolador autossintonizado.

Outra estrutura que foi desenvolvida ao longo dos anos para controle de processos é mencionada por BALAKRISHNAN e WEIL (1996), conhecido como redes “CMAC” (*Cerebellar Model Articulation Controller*) desenvolvida por um pesquisador chamado Albus em 1975. Um exemplo de aplicação dessa estrutura de rede é citado por Campos e Saito (2004), que apresentam uma aplicação dessa rede em um processo petroquímico no controle de composição de topo e de fundo de uma coluna de destilação.

O uso de redes neurais se tornou mais abrangente a medida que mais estudos eram realizados com o uso delas, aplicando-se em tecnologias mais recentes, como no estudo de caso de Guo, Song e Cai (2007). Nesse trabalho foi proposta uma estratégia de controle que substituía o uso de controladores PID em sistemas de aquecimento centralizados com unidades de tratamento de ar (ventilação e ar-condicionado).

Propomos uma estratégia de controle em cascata para o controle de temperatura da unidade de tratamento de ar. Em vez de um controlador PID fixo no esquema clássico de controle em cascata, um controlador de rede neural (NN) é usado no circuito de controle externo. Esta abordagem não apenas supera o tedioso procedimento de ajuste para os parâmetros PID de loop interno e externo de um sistema clássico de controle em cascata, mas também faz com que todo o sistema de controle seja adaptativo e robusto. (Guo, Song e Cai, 2007)

Outro trabalho que tem o mesmo objetivo é o de Yadav e Chandra (2015). Nesse trabalho, os autores afirmam que os controladores tradicionais de ar-condicionado como os controladores *on-off* e PID não fornecem resposta suficiente e apresentam elevado consumo.

Projetamos esse sistema de controle de rede neural que especula sua própria lei de controle. A vantagem de usar redes neurais é que este controlador é um sistema de autoaprendizagem e fornece uma resposta mais rápida e melhor em a comparação com outros controladores e fornece a saída de sobressinal zero.

Mas a aplicação das redes neurais não se restringiu somente a controlar sistemas de aquecimento com unidades de tratamento de ar. Há trabalhos que foram desenvolvidos para controlar motores elétricos, como é o caso de Omatu et. al. (2010). Nesse estudo foi utilizado uma estrutura do tipo Neuro-PID para o controle de torque e velocidade de um veículo elétrico.

No Japão, cerca de oitenta e quatro por cento das indústrias de processo usaram o controle PID. Depois de derivar o esquema de controle PID de auto-sintonia (neuro-PID) usando a capacidade de aprendizagem da rede neural, mostraremos os resultados do controle usando o controle de velocidade e torque de um veículo elétrico.

O estudo demonstrou que o controle neuro-PID é robusto com mudanças de parâmetros e ruído, sendo aplicável a muitos tipos de problemas de controle, como por exemplo em controle de processo ou problema de regulador.

No trabalho aqui desenvolvido, são abordadas duas classes de redes neurais: as multicamadas do tipo *feedforward* e as recorrentes. Essas duas classes de redes foram amplamente utilizadas em identificação e controle de sistemas dinâmicos.

Duas classes de redes neurais que receberam considerável atenção na área de redes neurais artificiais nos últimos anos são: 1) redes neurais multicamadas e 2) redes recorrentes. Redes multicamadas provaram ser extremamente bem-sucedidas em problemas de reconhecimento de padrões, enquanto as redes recorrentes foram usadas em memórias associativas, bem como na solução de problemas de otimização. Do ponto de vista da teoria dos sistemas, as redes multicamadas representam mapas estáticos não-lineares, enquanto as redes recorrentes são representadas por sistemas de realimentação dinâmico não-lineares. (NARENDRA, PARTHASARATHY, 1990)

1.3 Aplicações das redes neurais em controle

As aplicações se estendem a diversas áreas. Neste trabalho são abordadas duas aplicações específicas: para identificação e controle de plantas de sistemas. A seguir são explicadas estas duas aplicações.

1.3.1 Redes neurais para a Identificação da planta

De acordo com Ogata (2010), as plantas são definidas como qualquer sistema físico a ser controlado, como um componente mecânico, um forno, um reator químico ou uma espaçonave. Elas geralmente estão associadas em série com o controlador no diagrama de blocos e apresentam uma dinâmica que pode ser representada por uma *função de transferência* (FT). No entanto, essa dinâmica nem sempre é conhecida. Para isso, pode-se utilizar as RNAs para a identificação da dinâmica dessas plantas. Esse processo é conhecido como *modelagem da resposta dinâmica de um processo*. (CAMPOS, SAITO; 2004)

O processo de modelagem tem como objetivo configurar uma rede neural que consiga prever a resposta dinâmica deste processo. Dessa forma, a rede neural irá aprender a função temporal que irá representar o modelo do processo.

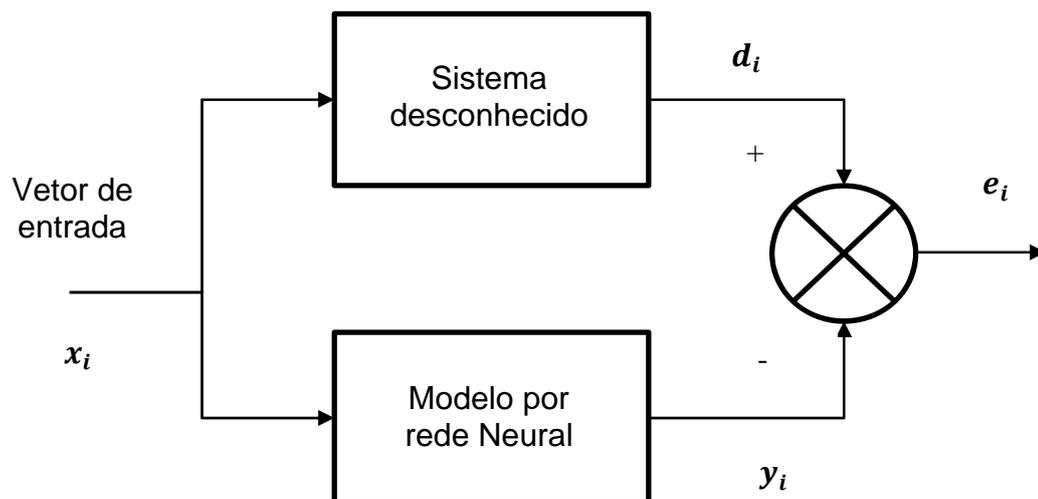
Contudo, antes de iniciar o processo de modelagem pela rede é necessário ajustar seus parâmetros, que segundo Campos e Saito (2004), consiste em 4 etapas:

- Escolha de um conjunto de variáveis de entrada iniciais;
- Escolha da arquitetura (número de camadas, número de neurônios por camada etc.)
- Treinamento da rede;
- Caso o desempenho não seja satisfatório, então retomar os passos anteriores;

Realizados os passos citados anteriormente, deve-se em seguida determinar uma configuração no diagrama de blocos em que a RNA irá aprender a dinâmica da planta. Em geral, a rede costuma situar em paralelo com a planta do sistema para que a rede possa gerar um erro entre a saída do sistema e a saída da rede identificadora.

Observe, por exemplo, a figura 3. Nesta configuração, a rede está disposta em paralelo com a planta e um sinal de erro é gerado entre a saída da planta e a da rede para que esses dados possam ser utilizados pelo algoritmo de aprendizagem da rede.

Figura 3 – Diagrama de blocos do modelo de identificação da planta.



Nota: Elaboração própria.

Na configuração da figura 3, a rede neural é utilizada como um aproximador de função. Uma entrada é aplicada à rede e ao sistema de dinâmica desconhecida. Em seguida um erro é gerado pela a saída da planta e da rede. Os valores de erro produzidos são utilizados pela própria rede no processo de aprendizagem realizado pelo algoritmo de retropropagação. Pode-se apresentar essa configuração de uma forma matemática, onde o sinal de entrada aplicado ao sistema e à rede é representado por u_k , e a saída do sistema de dinâmica

desconhecida é dado por y_p , e a saída da rede y_r (NARENDRA, PARTHASARATHY, 1990). Dessa forma, um erro é gerado (e_i) entre a saída do sistema e da rede identificadora (equação 1).

$$y_r - y_p < e_i \quad (1)$$

O erro produzido é minimizado pelo algoritmo de aprendizagem, ou seja $e_i \rightarrow 0$. Ao final do processo de aprendizagem, a RNA consegue reproduzir o comportamento dinâmico da planta.

1.3.2 Configurações de redes neurais utilizadas para controle de planta

Ao se pensar em projetar controladores para um sistema, deve-se enfatizar que “o objetivo de um sistema de controle é influenciar o comportamento dinâmico de um processo, de maneira a manter as variáveis de saída nos seus respectivos valores desejados (*set-points*), ou forçá-los a seguirem determinadas trajetórias”. (CAMPOS e SAITO, 2004)

No entanto, esses sistemas podem apresentar algumas dificuldades, como: grau elevado de complexidade devido ao número de variáveis que interagem no sistema; presença de muita não-linearidade; incertezas nas medições e nos parâmetros internos da planta.

Uma das soluções adotadas atualmente para as não-linearidades foi o uso de redes neurais artificiais multicamadas do tipo *perceptron* em controle de processos dinâmicos. Campos e Saito (2004) afirmam que por elas terem uma capacidade “de modelar e aproximar sistemas complexos e não-lineares” elas sempre foram vistas como uma solução para o controle de processos.

Atualmente, existem diversas formas de controle utilizando redes neurais. Os métodos de neurocontrole que foram desenvolvidos no período entre 1950 a 1980, foram aplicadas inicialmente em sistema dinâmicos não-lineares, visto que as não-linearidades eram problemas complexos de se resolverem pela teoria tradicional linear.

Os métodos de neurocontrole podem ser aplicados com sucesso para controlar sistemas não-lineares desconhecidos, enquanto abordagens de controle convencionais baseadas na teoria do sistema dinâmico linear não poderiam resolver tais problemas de controle (Omatu et al, 2010).

ANTOLINES, QUIROGA e PACHÓN, (2012) descrevem algumas arquiteturas de controle utilizadas (fig. 4). Na primeira arquitetura, uma rede neural ligada em série com a planta irá minimizar o erro dado pela saída desejada $y_d(t)$ e a saída da planta $y(t)$. A arquitetura de número 2 é denominada PID neural. Nesta configuração, os ganhos K_p , K_i e K_d do controlador são substituídas por variáveis que dependem do erro gerado entre a saída

desejada $y_d(t)$ e a saída da planta $y(t)$. Já na arquitetura de número 3, tem-se um controlador neural, mas o controle da planta é compartilhado com um PID.

Campos e Saito (2004) também apresentam algumas aplicações de redes neurais artificiais em controle de processos. Esses autores estabelecem que uma forma de controle ideal seria pela obtenção de um modelo inverso da dinâmica do processo. Dessa forma, a variável de saída “y” seria sempre igual ao valor desejado, visto que o produto da função de transferência (FT) do controlador pela do processo seria igual à matriz identidade.

No entanto, essa abordagem de obter a inversa do processo pode não representar a melhor solução visto que grande parte das plantas encontradas não possuem FT invertíveis. Outras configurações também são utilizadas atualmente, com uma ou duas RNAs. Uma delas escolhida para aplicação aqui é o Modelo de Aproximação (figura 5).

Aplicações de controle neural também são apresentadas por Hagan e Demuth (1999). Outra das arquiteturas que despertou o interesse no desenvolvimento do trabalho aqui realizado é o controlador neural NARMA-L2 (*Feedback Linearization*)

1.4 Limitações e precauções

Cada uma das arquiteturas de neurocontroladores tem suas vantagens e desvantagens. Por exemplo, a técnica de linearização de feedback só pode ser aplicada a uma classe específica de sistemas (BREEMEN e VEELTURF, 1996). A técnica de controle adaptativo de modelo de referência não tem garantia de estabilidade. A técnica de controle adaptativo inverso requer a existência de uma planta estável inversa.

Em geral, essas técnicas que garantem estabilidade aplicam-se a uma classe restrita de sistemas. A medida que a área de neurocontroladores continua a progredir, métodos estáveis de neurocontroladores serão desenvolvidos para uma ampla classe de sistemas.

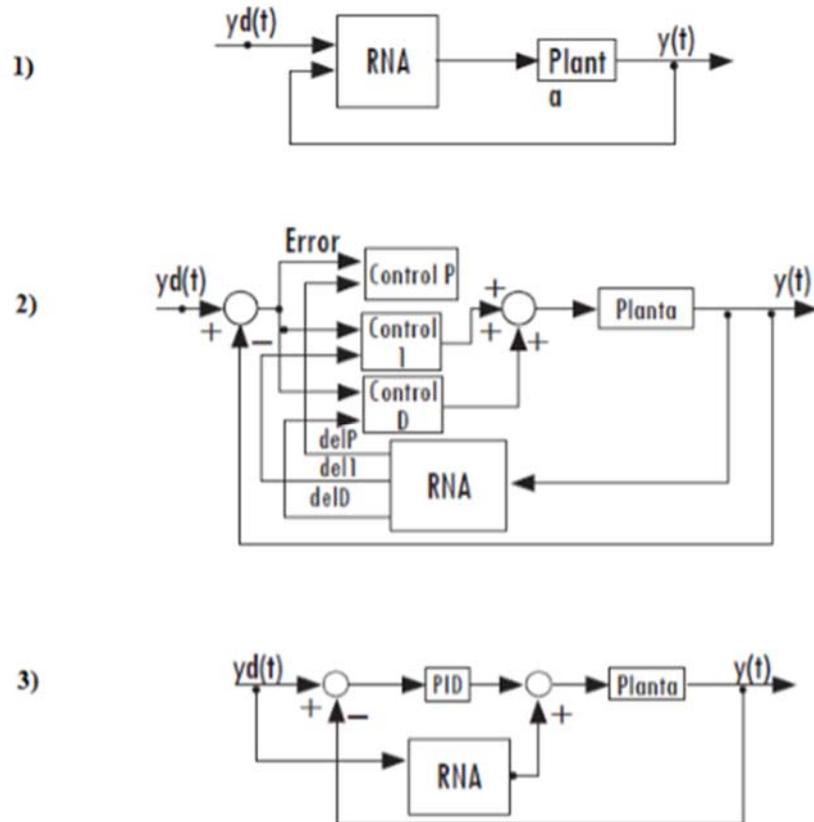
Um dos principais problemas práticos para muitos dos sistemas com neurocontroladores é a questão da generalização – a capacidade de uma rede para executar bem em novas situações. Por exemplo, a arquitetura de controle preditivo requer que uma rede neural de modelo identifique a planta do sistema.

Esta planta modelo é um mapeamento de entradas anteriores da planta e saídas para futuras saídas da planta. Para modelar precisamente a planta, a rede precisa ser treinada com dados que abrangem toda a gama de possíveis entradas da rede. Pode ser difícil de obter esses dados, uma vez que nós não temos controle direto sobre saídas anteriores da planta.

Às vezes, podemos ter controle independente sobre as entradas da planta, mas apenas controle indireto sobre as saídas da planta (que então se tornam entradas para a rede). Para

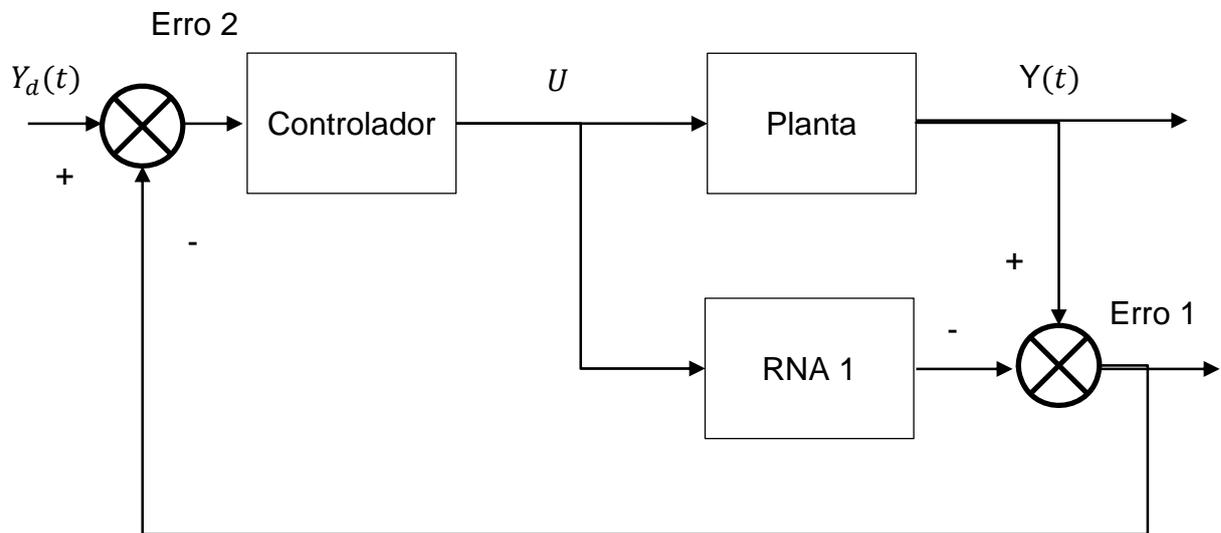
sistemas de ordem elevada, pode ser difícil obter dados em que a resposta da planta cubra todas as partes utilizáveis do espaço de estados. Nessas situações, será importante que a rede seja capaz de detectar situações nas quais as entradas estão fora das regiões onde a rede recebeu dados de treinamento.

Figura 4 - Arquiteturas de controle usando redes neurais artificiais.



Fonte: ANTOLINES, José Danilo Rairán, QUIROGA, Diego Fernando Chiquiza, PACHÓN, Miguel Ángel Parra. Implementação de neurocontroladores em linha: Três configurações, três plantas. Ing. Univ. Bogotá. Junho, 2012. Disponível em: <<http://www.scielo.org.co/pdf/inun/v16n1/v16n1a10.pdf>>. Acesso em: 17/05/2018.

Figura 5 - Modelo de aproximação.



Fonte: CAMPOS, Mario Massa de, SAITO, Kaku. *Sistemas inteligentes em controle e automação de processos*. Rio de Janeiro: Editora Ciência Moderna Ltda., 2004.

1.5 Objetivo

Conforme relatado na seção 1.3, várias estratégias de controle de sistemas por RNAs podem ser adotados, com as limitações e requisitos comentados na seção 1.4. Decidiu-se escolher três dessas estratégias e aplicar em sistemas com características diferentes, para fins de análise do comportamento, viabilidade do controle e sua parametrização.

Com isto, o objetivo principal deste trabalho é estudar e simular três sistemas dinâmicos controlados por redes neurais artificiais e comparar o desempenho dos controladores neurais com o desempenho de um controlador PID tradicional.

Como consequência, medir parâmetros de desempenho e comparar os controladores neurais com o controlador PID para cada sistema e, decorrente disto, discutir recomendações e limitações dos controladores propostos para cada sistema.

1.6 Sistemas e configurações utilizadas neste trabalho

Neste trabalho, duas abordagens são utilizadas para o controle de três sistemas previamente modelados. Estes sistemas foram escolhidos na literatura com o modelo de planta já estabelecido e respectivas funções de transferência obtidas, sendo:

- Posicionamento azimutal de uma antena parabólica
- Motor elétrico de tração
- Suspensão ativa em uma das rodas de um veículo

Para o controle desses sistemas foram escolhidas as seguintes estratégias:

- Emprego da RNA tipo *perceptron* multicamadas para reproduzir a função de controle de um PID – aproximador de funções – (figura 4, configuração 1)
- Emprego de neurocontrolador do tipo NARMA-L2 com duas RNAs perceptron multicamadas internas para controlar a planta.
- Emprego de neurocontrolador do tipo Modelo de Referência

Assim, o presente trabalho está organizado da seguinte maneira: no capítulo 2 são apresentados os referenciais teóricos sobre controle de sistemas e redes neurais artificiais aplicadas a esse propósito, no capítulo 3 mostram-se os modelos físicos escolhidos para os experimentos executados e seus modelos de controle e no capítulo 4 os resultados e análises desses experimentos, com discussões apresentadas no capítulo 5.

2 REFERÊNCIAS CONCEITUAIS

Como o objetivo deste trabalho é explorar o controle de sistemas eletromecânicos por redes neurais artificiais, apresentam-se neste capítulo os fundamentos conceituais relacionados a controle de sistemas e redes neurais.

2.1 Controle de sistemas

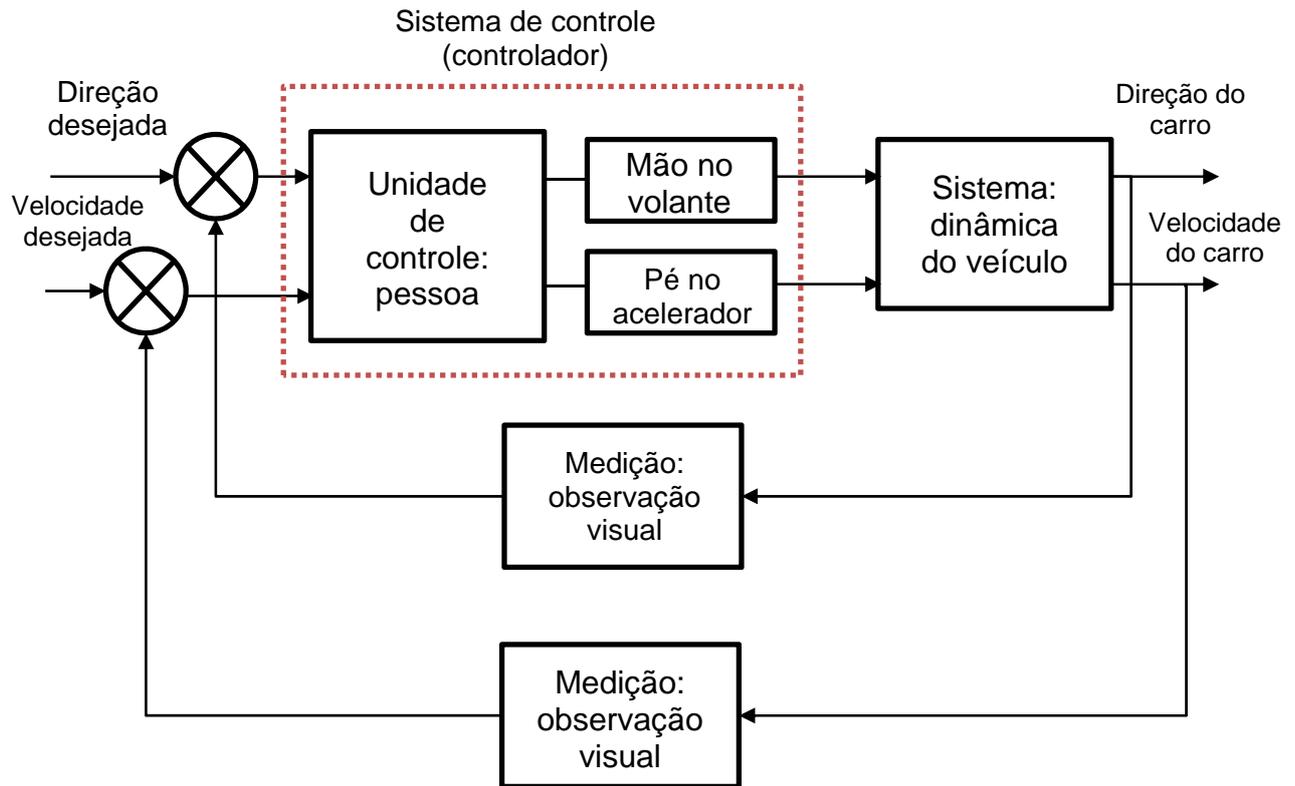
Do ponto de vista do Controle, um sistema é constituído de componentes que interagem entre si de acordo com leis naturais. O restante (que não integra o sistema) é considerado o meio ambiente. A interação deste com o sistema altera características de ambos, na forma de transferência de energia, massa, calor, eletricidade etc., até mesmo informação. As grandezas mensuráveis envolvidas nas interações são classificadas como *entradas* ou *saídas*. As entradas não ajustáveis são classificadas como *perturbações* ou *ruído*. Automatizar então refere-se a prover aos sistemas mecanismos que os levem a um comportamento desejado ou esperado, com o mínimo de intervenção humana (NASCIMENTO Jr., YONEYAMA, 2000).

2.1.1 Funções de transferência em controle

Um sistema de controle, segundo Nise (2000), “consiste em subsistemas e processos (ou plantas) reunidos com o propósito de controlar as saídas dos processos”. Em sua forma simples, ele apresenta pelo menos uma entrada e uma saída. Ao se aplicar um estímulo em sua entrada, gera-se uma saída ou uma resposta a esse estímulo. Pode-se representar um sistema de controle por meio de diagrama de blocos como o da figura 36. Usa-se esta representação quando se deseja visualizar os diversos subsistemas componentes de um sistema maior e as interações entre eles.

O sistema de controle pode ser classificado em: SISO (Simple Input, Simple Output), MISO (Multiinput, Simple Output) e MIMO (Multi Input, Multi Output). Um exemplo de representação de um sistema MIMO dentro da área de automação é o ato de dirigir o carro, como mostra a Figura 6.

Figura 6 – Sistema MIMO – Ato de dirigir o carro

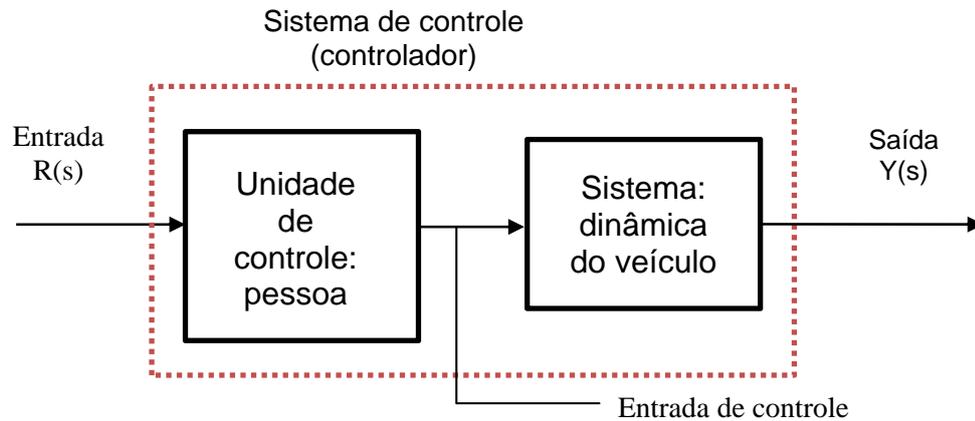


Nota: Elaboração própria.

É importante salientar que um sistema dinâmico pode ser matematicamente representado no domínio do tempo por um conjunto de equações diferenciais. O sistema de equações diferenciais relaciona como serão as saídas do sistema em função de suas entradas. Dessa forma, ao sistema de equações diferenciais, é aplicada o operador Laplace, o sistema passa a ser representado por um conjunto de equações algébricas no domínio s de Laplace, complexo ou da frequência. O sistema fica bem identificado por uma função de transferência (FT), que caracteriza a saída a partir das entradas e das condições iniciais do sistema (OGATA, 2010), podendo ser visualizado de uma forma algébrica. A FT de um sistema de controle é definida “como a razão entre a transformada de Laplace da variável de saída e a transformada de Laplace da variável de entrada” (DORF, BISHOP, 2013, pg. 47), conforme a equação 2 e Figura 7:

$$G(s) = \frac{Y(s)}{R(s)} \quad (2)$$

Figura 7 - Sistema de controle



Nota: Elaboração própria.

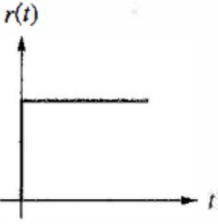
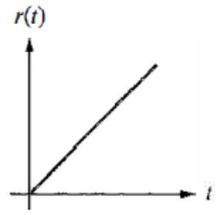
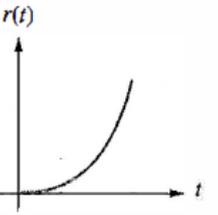
Outro fator importante para sistema de controle é a realimentação, ou *feedback*, que é parte do sistema de controle em sistema de malha fechada ao contrário da malha aberta que possui somente o avanço (*feedforward*). A realimentação é representada pelo elo que liga a saída do sistema até o ponto somador da entrada do mesmo, como mostrado na Figura 35.

2.1.2 Entradas de teste

Os sistemas dinâmicos são avaliados por meio de índices de desempenho transitório, que serão explicados na sessão 4.2. Entretanto, é necessário antes verificar sua estabilidade (sessão 4.3). “Se ele for estável, a resposta para um sinal de entrada específico irá fornecer várias medidas de desempenho”. (DORF e BISHOP, 2013). Porém, sabe-se que o sinal de entrada real do sistema é geralmente desconhecido. Para isso, um sinal de entrada de teste padrão é utilizado. Dorf e Bishop afirmam que os sinais de entrada de teste padrão mais utilizados são “a entrada em degrau, a entrada em rampa e a entrada parabólica”. As entradas de teste também são utilizadas “para análise e projeto de erro de estado estacionário” (NISE, 2000), uma das especificações de projetos abordadas na sessão 4.4.

A Tabela 1 resume as principais entradas de teste utilizadas em projetos de controle, bem como sua descrição física, sua função no domínio do tempo e sua respectiva função de Laplace.

Tabela 1 - Principais entradas de teste

| Forma de onda | Nome | Interpretação física | Função no domínio do tempo | Função de Laplace |
|---|----------|----------------------|----------------------------|-------------------|
|  | Degrau | Posição constante | 1 | $\frac{1}{s}$ |
|  | Rampa | Velocidade constante | t | $\frac{1}{s^2}$ |
|  | Parábola | Aceleração constante | $\frac{1}{2}t^2$ | $\frac{1}{s^3}$ |

Fonte: NISE, Norman S. Engenharia de sistemas de controle. 3 ed. Rio de Janeiro: LTC, 2000.

Estes sinais de testes são usados tipicamente da seguinte maneira: as entradas em degrau representam posição constante, por conseguinte, são úteis para determinar a capacidade do sistema de controle em se posicionar com relação a um alvo estacionário, como um satélite em órbita geoestacionária. Um sistema de controle de posicionamento de antena constitui exemplo de sistema cuja exatidão pode ser testada usando entradas em degrau. Entradas em rampa representam entradas com velocidade constante em um sistema de controle de posição com a amplitude da entrada crescendo linearmente com o tempo. Podem ser usadas para testar a capacidade do sistema para seguir uma entrada linearmente crescente, ou, em outras palavras, de rastrear um alvo com velocidade constante. Para concluir, as parábolas, cujas segundas derivadas são constantes, representam entradas de aceleração constante em sistemas de controle de posição e podem ser usadas para representar alvos sendo acelerados, como mísseis. (NISE, 2000)

Como complemento às entradas em degrau, pode-se dizer que são bastante úteis para observar a resposta do sistema a uma mudança brusca de um nível constante para outro, seu comportamento transitório, à luz de cinco parâmetros apresentados a seguir.

2.1.3 Índices de desempenho transitório

Na teoria de sistemas de controle, é comum utilizar índices que caracterizam o desempenho da resposta transitória de um sistema controlado.

Os índices de desempenho podem ser obtidos matematicamente por meio de equações algébricas (em termos do fator de amortecimento e frequência natural de um sistema de segunda ordem) ou de forma gráfica quando se tem uma curva de resposta simulada computacionalmente. Os desempenhos da resposta do sistema sob ação dos controladores PID e controlador neural são comparados nos tópicos seguintes.

Esses índices de desempenho transitório serão utilizados como parâmetros na comparação entre as respostas transitórias do sistema controlado com PID e com controlador neural. Nesse trabalho, as condições iniciais dos sistemas são as mencionadas por Ogata (2010):

- Inicialmente em repouso;
- Valor da variável de saída igual a zero;
- Todas as derivadas das equações do sistema em função do tempo iguais a zero;

Conforme mencionado anteriormente, para análise de resposta transitória dos sistemas são geradas curvas de simulação. Para obtenção dos índices de desempenho serão coletados alguns pontos da curva de simulação. Cada parâmetro é descrito por Ogata (2010) como:

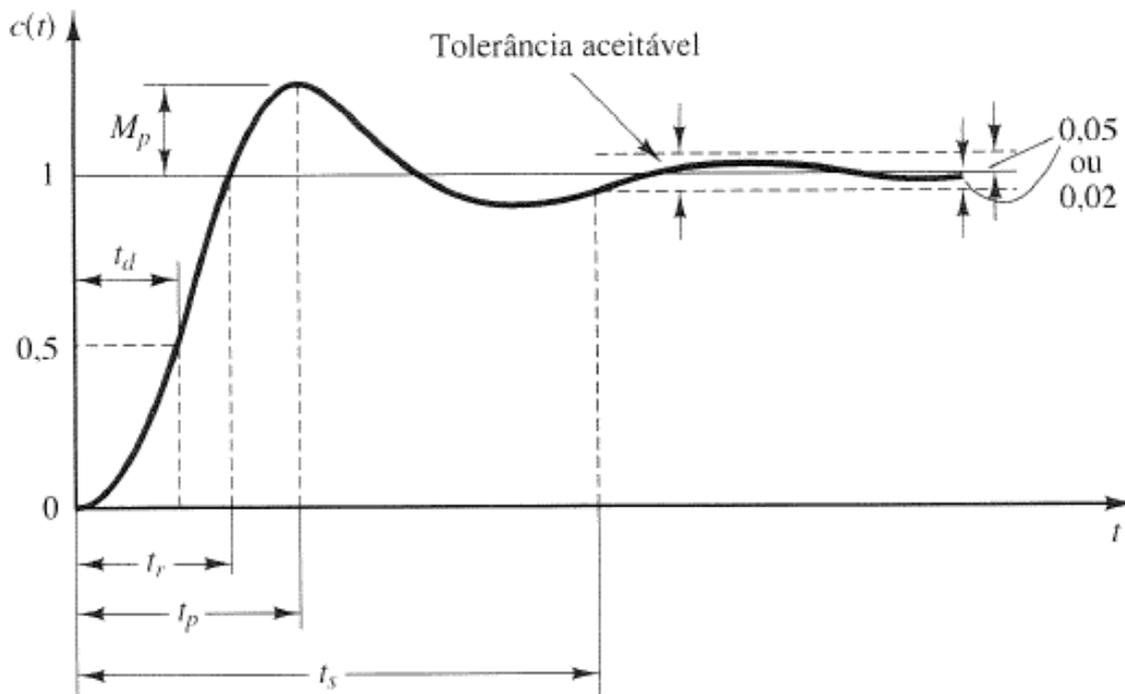
1. Tempo de atraso, t_d : tempo requerido para que a resposta alcance metade de $c(t)$, valor final pela primeira vez.
2. Tempo de subida, t_r : tempo requerido para que a resposta do sistema passe de 10 a 90%, ou de 5 a 95%, ou de 0 a 100% do valor final. Para sistemas de segunda ordem subamortecidos, o tempo de subida de 0 a 100% é normalmente utilizado. Para sistemas superamortecidos, o tempo de 10 a 90% é o mais comumente utilizado.
3. Tempo de pico, t_p : é o tempo para que a resposta atinja o primeiro pico de sobressinal.
4. Máximo sobressinal (em porcentagem), M_p : é o valor máximo de pico da curva de resposta, medido a partir da unidade. Se o valor final da resposta em regime permanente diferir da unidade, então é comum utilizar porcentagem máxima de sobressinal, definida por:

$$M_p = \frac{c(t_p) - c(\infty)}{c(\infty)} * 100\% \quad (3)$$

O valor máximo (em porcentagem) do sobressinal indica diretamente a estabilidade relativa do sistema.

5. Tempo de acomodação, t_s : tempo necessário para que a curva de resposta alcance valores em uma faixa (geralmente de 2% ou 5%) em torno do valor final, aí permanecendo indefinidamente. O tempo de acomodação está relacionado à maior constante de tempo do sistema de controle. Pode-se determinar qual porcentagem deve ser utilizada no critério de erro a partir dos objetivos do projeto do sistema em questão.

Figura 8 - índices de desempenho da resposta transitória



Fonte: OGATA, Katsuhiko. Engenharia de controle moderno. 5 ed. São Paulo: Pearson Prentice Hall, 2010.

As especificações dadas no domínio do tempo são importantes porque a maioria dos sistemas de controle está no domínio do tempo. Dessa forma, o sistema de controle deve ser modificado até que satisfaça os objetivos do projeto. A figura 8 apresenta os índices de desempenho obtidos a partir de um gráfico da resposta transitória.

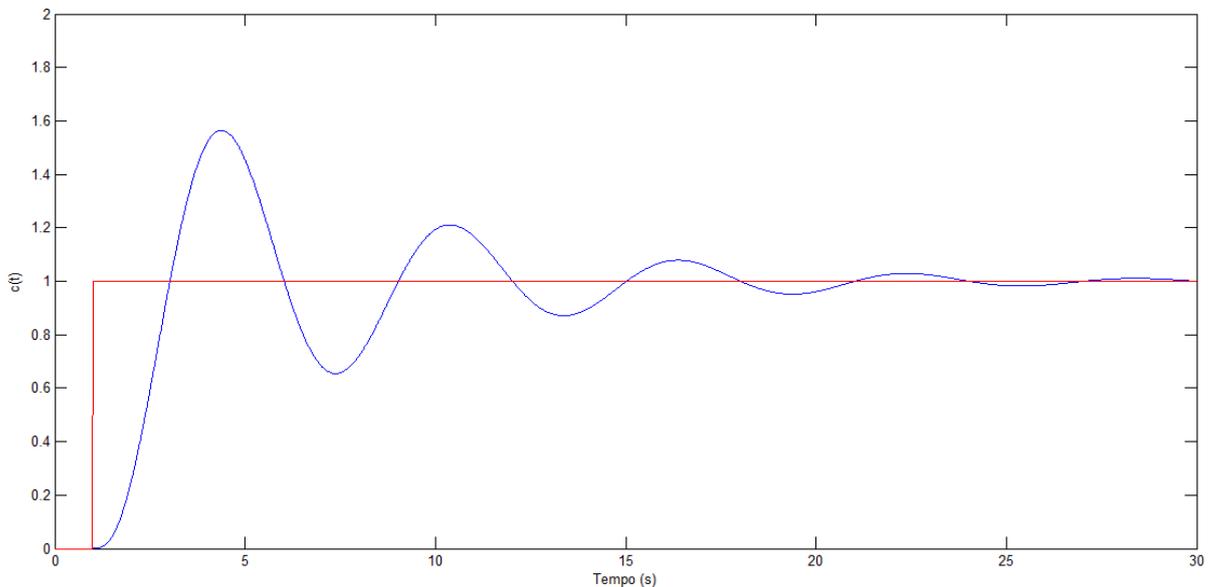
O gráfico da curva de resposta transitória é gerado pelo simulador, composta por um conjunto de pontos discretos que formam uma curva de uma função relacionando a grandeza de saída em relação ao tempo. Assim, os índices de desempenho podem ser obtidos pela

curva de simulação delimitando alguns pontos da curva que definem esses parâmetros. Essas curvas são apresentadas no tópico de sistemas estudados.

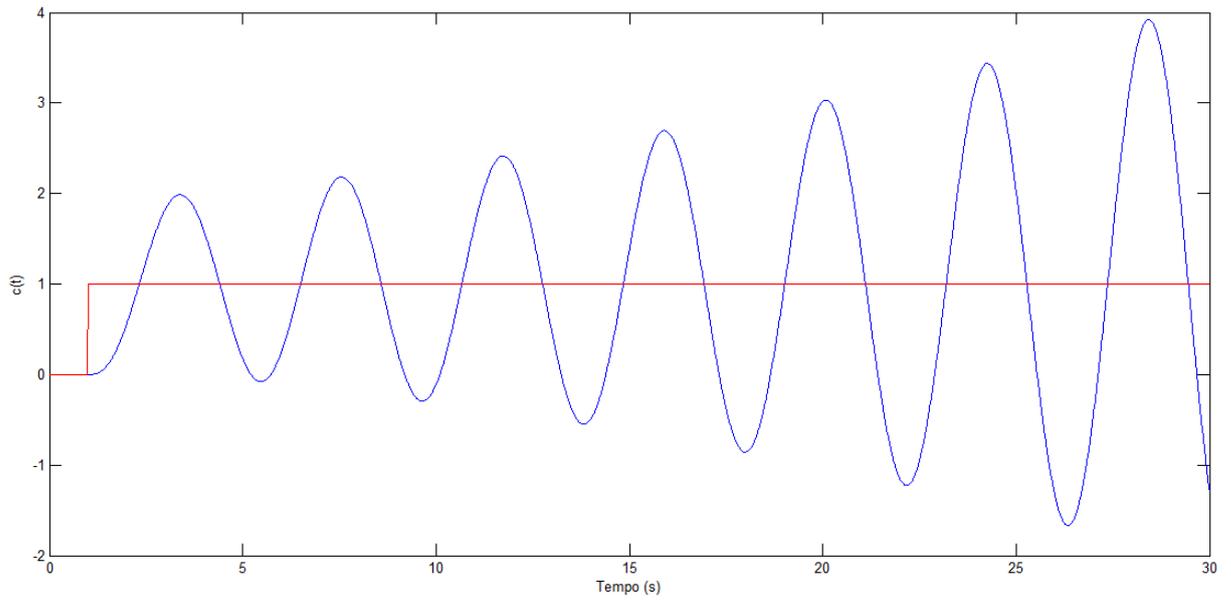
2.1.4 Estabilidade dos sistemas

De acordo com Nise (2000), “a estabilidade é a especificação mais importante do sistema. Se um sistema é instável, a resposta transitória e os erros de estado estacionário deixam de ter significado”. As figuras 9, 10 e 11 apresentam um sistema estável, um sistema instável e um marginalmente estável com resposta no domínio do tempo.

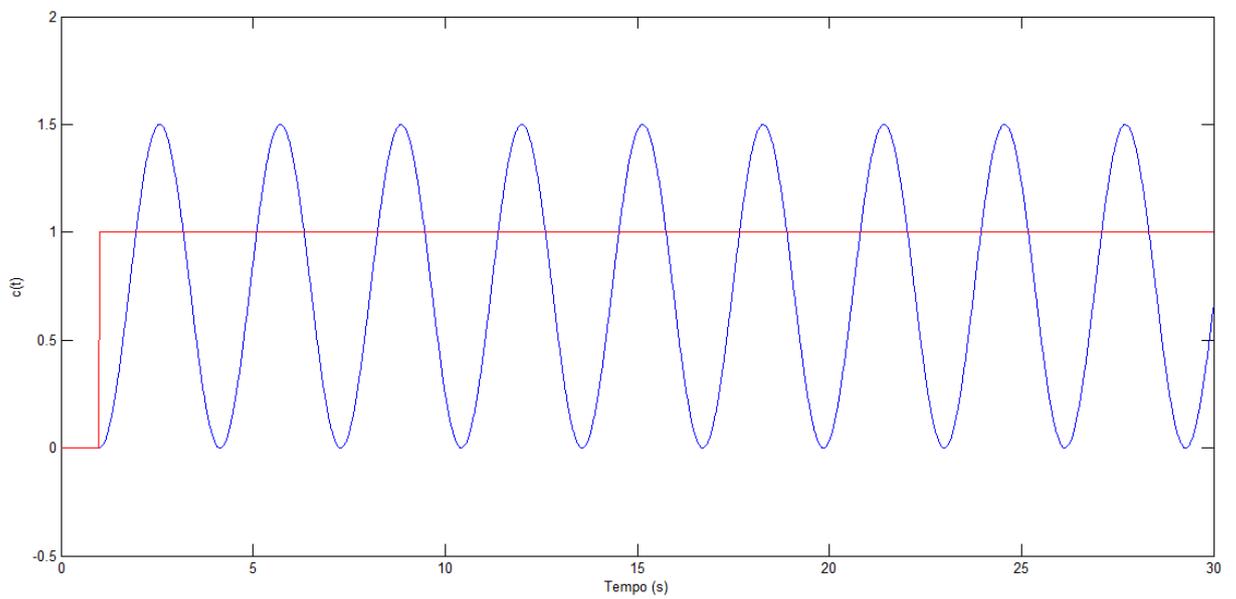
Figura 9 - Sistema estável com resposta natural tendendo a um.



Nota: Elaboração própria.

Figura 10 – Sistema instável com resposta natural tendendo a infinito

Nota: Elaboração própria.

Figura 11 – Sistema marginalmente estável com resposta natural oscilatória

Nota: Elaboração própria.

2.1.4.1 Critério de Routh-Hurwitz

Um dos critérios amplamente utilizados por engenheiros para averiguar a estabilidade de sistemas lineares é o método de Routh-Hurwitz. (DORF e BISHOP, 2013). Partindo-se da FT de um sistema dinâmico, que possui a seguinte forma:

$$T(s) = \frac{p(s)}{q(s)} \quad (4)$$

chama-se o polinômio do denominador $q(s)$ de equação característica, que corresponde à seguinte forma:

$$q(s) = a_n s^n + a_{n-1} s^{n-1} + a_{n-2} s^{n-2} \dots + a_1 s^1 + a_0 \quad (5)$$

Para averiguar a estabilidade do sistema é necessário determinar se alguma das raízes da equação característica se situa no semiplano direito do plano s . Para isso, é necessário primeiro elaborar uma tabela ou lista com os coeficientes do polinômio da equação característica $q(s)$, conforme a seguir:

Tabela 2 – estrutura da tabela de Routh

$$\begin{pmatrix} s^n & a_n & a_{n-2} & a_{n-4} \\ s^{n-1} & a_{n-1} & a_{n-3} & a_{n-5} \\ s^{n-2} & b_{n-1} & b_{n-3} & b_{n-5} \\ s^{n-3} & c_{n-1} & c_{n-3} & c_{n-5} \\ \dots & \dots & \dots & \dots \\ s^0 & h_{n-1} & h_{n-3} & h_{n-5} \end{pmatrix}$$

Nota: elaboração própria.

Em que

$$b_{n-1} = \frac{a_{n-1}a_{n-2} - a_n a_{n-3}}{a_{n-1}} = -\frac{1}{a_{n-1}} \begin{vmatrix} a_n & a_{n-2} \\ a_{n-1} & a_{n-3} \end{vmatrix} \quad (6)$$

$$b_{n-3} = -\frac{1}{a_{n-1}} \begin{vmatrix} a_n & a_{n-4} \\ a_{n-1} & a_{n-5} \end{vmatrix} \quad (7)$$

$$c_{n-1} = -\frac{1}{b_{n-1}} \begin{vmatrix} a_{n-1} & a_{n-3} \\ b_{n-1} & b_{n-3} \end{vmatrix} \quad (8)$$

Segundo Dorf e Bishop (2013), “o critério de Routh-Hurwitz declara que o número de raízes de $q(s)$ com parte real positiva é igual ao número de trocas de sinal da primeira coluna

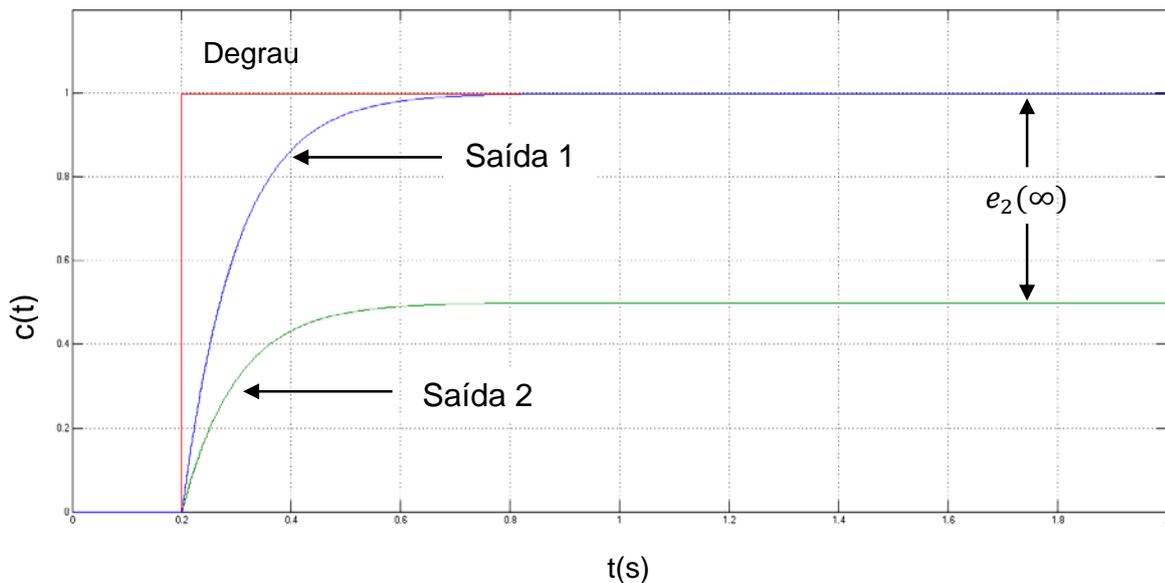
da tabela de Routh”. Esse critério requer que não haja troca de sinal na primeira coluna para um sistema estável.

2.1.5 Erro em estado estacionário

A análise e o projeto de sistemas de controle enfocam três especificações: (1) resposta transitória, (2) estabilidade e (3) erro de estado estacionário. Antes de efetuar a análise e o projeto do erro de estado estacionário, deve-se antes, verificar a estabilidade do sistema. Caso o sistema seja instável, as expressões deduzidas para calcular o erro de estado estacionário podem ser aplicadas erroneamente aos sistemas instáveis. (NISE, 2000)

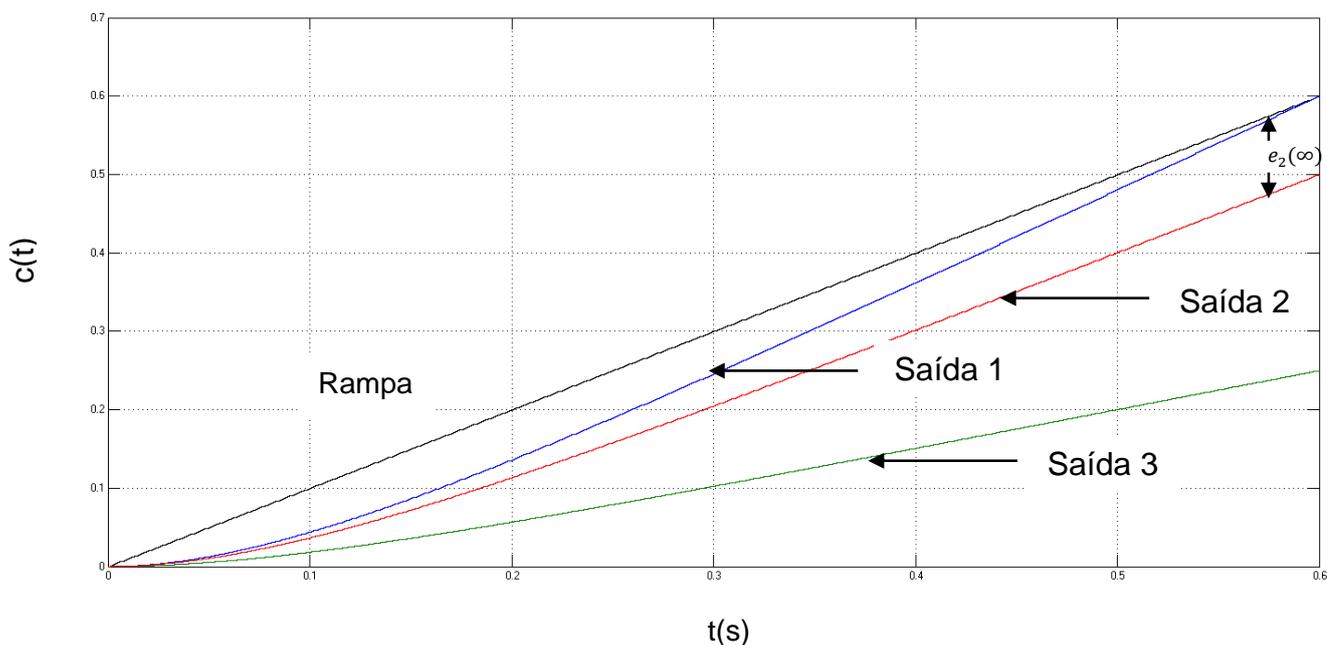
Ao medir o erro permanente de um sistema, é necessário submetê-lo a uma das entradas de teste (degrau, rampa ou parabólica). Considere a figura 12 que mostra uma entrada em degrau e duas saídas possíveis. A saída 1 representa o erro estacionário nulo, e a saída 2, um erro estacionário, $e(\infty)$, finito.

Figura 12 – erro de estado estacionário com entrada em degrau



Nota: Elaboração própria.

Figura 13 – erro de estado estacionário com entrada em rampa



Nota: Elaboração própria.

Outro exemplo pode ser visto na figura 13, onde a entrada em rampa é comparada com a saída 1, que apresenta erro estacionário nulo (diminui ao longo do tempo), e a saída 2, que apresenta um erro, $e_2(\infty)$, finito, medido verticalmente entre a entrada e a saída 2, depois que os transitórios desaparecem. Também há outra possibilidade para entrada em rampa: se a inclinação da saída for diferente da inclinação de entrada (como na saída 3), o erro estacionário é infinito ao medi-lo verticalmente entre a entrada e a correspondente saída número 3, depois que o transitório desaparece e t tende a infinito.

Em geral, o erro de estado estacionário (ou regime permanente) é o erro que aparece depois que a resposta transitória tenha desaparecido, deixando somente a resposta contínua. (DORF E BISHOP, 2013) Os erros do sistema podem ser originados pelas não linearidades presentes nele, como folgas de engrenagens ou um motor que responde após a tensão aplicada ultrapassar um certo limiar para seu acionamento, ou pela própria configuração do sistema e do tipo de entrada aplicada. (NISE, 2000)

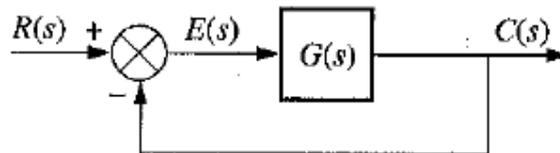
2.1.5.1 Erro estacionário em termos de $G(s)$

É comum encontrar sistemas configurados como sistema com retroação unitária com uma função de transferência, $G(s)$, no percurso direto, como na figura 14. Embora possamos obter a função de transferência a malha fechada, $T(s)$, e proceder então como na subseção

4.5.1, obtemos uma maior compreensão na análise e no projeto exprimindo o erro de estado estacionário em termos de $G(s)$ em vez de $T(s)$.

Considere agora, a figura 14 que representa o erro estacionário em termos de $G(s)$.

Figura 14 – representação do erro estacionário em diagrama de blocos em malha fechada



Fonte: NISE, Norman S. Engenharia de sistemas de controle. 3 ed. Rio de Janeiro: LTC, 2000.

Como a retroação, $H(s)$, é igual a 1, o sistema possui retroação unitária. A implicação disto é que o erro $E(s)$ é realmente o erro entre a entrada, $R(s)$, ea saída, $C(s)$. Portanto, ao calcular-se $E(s)$, chega-se a uma expressão do erro.

Portanto, ao escrever-se $E(s)$ a partir do diagrama de blocos da figura 14, obtém-se

$$E(s) = R(s) - C(s) \quad (9)$$

Porém,

$$C(s) = E(s)G(s) \quad (10)$$

Logo, substituindo a equação 10 em 9, resulta em

$$E(s) = \frac{R(s)}{1+G(s)} \quad (11)$$

Aplica-se agora o teorema do valor final (equação 12). Deve-se lembrar que neste ponto, em um cálculo numérico, devemos verificar se o sistema a malha fechada é estável usando, por exemplo, o critério de Routh-Hurwitz. Admite-se neste ponto, para concluir esta dedução, que o sistema a malha fechada seja estável. Faz-se a substituição da equação 48 na 44, obtém-se

$$e(\infty) = \lim_{s \rightarrow 0} \frac{sR(s)}{1+G(s)} \quad (12)$$

A equação 12 acima permite calcular o erro de estado estacionário, $e(\infty)$, dadas a entrada, $R(s)$, e o sistema, $G(s)$.

2.1.6 Controlador PID e sintonia

O controle Proporcional-Integral-Derivativo (PID) é o algoritmo mais usado e tem sido utilizado em todo o mundo para sistemas de controle industrial. Seu desempenho é robusto para uma ampla gama de condições de funcionamento e apresenta uma simplicidade funcional permitindo sua operação de uma forma simples e direta. A ideia básica por trás de um controlador PID é ler um sensor, calcular a resposta de saída do atuador através do cálculo proporcional, integral e derivativo e então somar os três componentes para calcular a saída. Como o nome sugere, o algoritmo PID é composto por três coeficientes: proporcional, integral e derivativo, que são variados para obter a resposta ideal.

O controlador PID possui ganhos que necessitam ser sintonizados para uma boa resposta do sistema. Para o ajuste dos ganhos proporcional, derivativo e integrativo utiliza-se o método de autossintonia por modelo linearizado (*Model-based PID controller tuning*). Esta técnica permite que se ajuste interativamente os ganhos do controlador enquanto se observa o comportamento da resposta do sistema a ser controlado. Assim, pode-se avaliar de forma visual o desempenho do sistema controlado.

No simulador, o tipo de controlador PID utilizado é paralelo de tempo contínuo. Há quatro parâmetros a serem ajustados:

- Ganho proporcional (P);
- Ganho integrativo (I);
- Ganho derivativo (D);
- Coeficiente de filtro (N)¹;

A FT que representa esta classe de controlador é dada por:

$$k_P + k_I \frac{1}{s} + k_D \frac{N}{1+N\frac{1}{s}} \quad (13)$$

Neste tipo de controlador, o modo integral é usado para eliminar o erro estacionário causado por grandes variações de carga. O modo derivativo, com o seu efeito estabilizador,

¹ “N” é uma constante de filtragem derivativa. A equação 13 é uma representação algébrica alternativa da equação tradicional do controlador PID. Um motivo que leva a adoção deste coeficiente é que o simulador considera o termo matemático $Kd \cdot s$ inadequado. Por isso utiliza-se no lugar de $Kd \cdot s$ a expressão $Kd \cdot \left(\frac{N \cdot s}{s+N}\right) = Kd \cdot \left(\frac{N}{1+N\frac{1}{s}}\right)$.

Se N for suficientemente grande, ele irá tender a $Kd \cdot s$.

Fonte: Disponível em: <<https://www.mathworks.com/matlabcentral/answers/111085-pid-block-and-manual-pid>>. Acesso em 09/2018.

Disponível em: <<https://www.mathworks.com/matlabcentral/answers/282298-what-is-the-n-coefficient-filter-in-simulink>>. Acesso em 09/2018.

permite um aumento do ganho e reduz a tendência para as oscilações, o que conduz a uma velocidade de resposta superior quando comparado com controladores do tipo P e PI.

Os efeitos na resposta do sistema em malha fechada, de adicionar os modos proporcional, integral e derivativo são listados na tabela 3. Note que estas correlações não são exatas, uma vez que, se alterarmos um dos parâmetros do controlador podemos estar a alterar o efeito das outras ações. Por esta razão, a tabela só deve ser usada como referência quando se está a determinar os parâmetros do controlador

Tabela 3 - resposta do sistema controlado em função dos parâmetros do PID.

| Resposta | Tempo de Subida | Sobressinal | Tempo de Acomodação | Erro Estacionário |
|--------------|-----------------|-------------|---------------------|-------------------|
| Proporcional | Diminuição | Aumento | Sem alteração | Diminuição |
| Integral | Diminuição | Aumento | Aumento | Elimina |
| Derivativo | Sem alteração | Diminuição | Diminuição | Sem alteração |

Pode-se projetar um controlador PID que atenda as especificações de resposta desejada (sobressinal, tempo de atraso etc.). As redes neurais por si só não podem ser projetadas em função das especificações, no entanto podem ser treinadas com um conjunto de dados que as capacite a atender as especificações de resposta desejada. Por outro lado, o controlador PID clássico, não possui uma capacidade adaptativa como das redes neurais, embora existam métodos de autossintonia que podem ser implementados a ele, como Ziegler e Nichols em malha aberta e fechada, o método CHR (Chien, Hrone e Reswick), o método de Cohen e Coon (CC), o método da Integral do Erro, o modelo interno (IMC) e outros menos frequentes (ATHERTON, 2014; LI, 2004).

No caso de uma possível alteração nos parâmetros do sistema, é necessário projetar um novo controlador PID. Já as redes neurais podem alterar seus parâmetros durante a etapa de aprendizagem.

2.2 Redes neurais artificiais

O primeiro estudo sobre redes neurais foi realizado em 1943 por Warren McCulloch (neuro-anatomista) e Walter Pitts (matemático). Eles propuseram um modelo de “neurônio biológico como sendo uma unidade lógica de limiar (ou gatilho) contendo N entradas binárias (Valor 0 ou 1 cada), e somente uma saída binária, e pesos associados de valor ± 1 ”. Na Figura 15 tem-se o modelo de McCulloch e Pitts.

Os sinais de entrada, dados por $x_1, x_2 \dots x_n$, representam os dendritos e a saída processada y representa o sinal de saída do axônio. A passagem do sinal de um neurônio para os dendritos de um outro neurônio é chamada de sinapse. Esta pode ser representada matematicamente pela soma ponderada dos sinais de entrada associadas a pesos específicos (função “*net*”) e pela função de ativação que modula o sinal de saída a ser transmitido (“*f(net)*”). A equação 14 representa a soma ponderada²:

$$net = \sum w_m \cdot x_m \quad (14)$$

$$Y = f(net) \quad (15)$$

Em que Y é a saída do neurônio artificial, w_m são os pesos das sinapses e x_m os sinais de entrada. (HAYKIN, 2009)

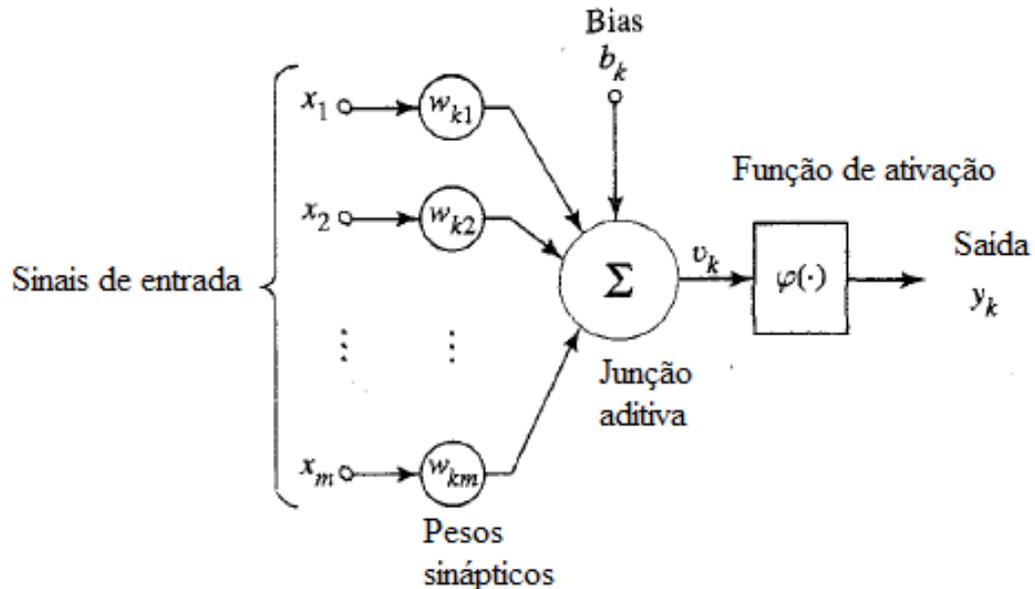
Quando os neurônios se interconectam eles formam estruturas definidas. Uma vez que topologia adequada foi adotada, “deve-se determinar a quantidade de camadas ocultas e a quantidade de neurônios em cada uma delas bem como o algoritmo de aprendizado e a função de ativação que serão utilizados”. (GONÇALVES, SILVA, ALMEIDA, 2015, p. 47).

As redes neurais artificiais (RNAs) são definidas segundo Costa (2009, p. 392) como “um algoritmo computacional de uma rede de neurônios artificiais”. Sua estrutura é baseada na estrutura do cérebro humano, com neurônios interconectados pelas sinapses.

As unidades estruturais de uma rede neural são os neurônios artificiais. Um neurônio é uma unidade de processamento de informação que é fundamental para a operação de uma rede neural. O diagrama em blocos da figura 15 mostra o modelo de um neurônio, que forma a base para o projeto das RNAs.

² A equação 14 foi encontrada originalmente em regime discreto, com termos vetoriais. Mas para simplificação da notação ela foi adaptada.

Figura 15 – Modelo não-linear de neurônio



Fonte: HAYKIN, Simon. *Neural networks and learning machines*. 3 ed. Porto Alegre: Prentice Hall: New Jersey, 2009.

Identificam-se na figura 15 três elementos básicos do modelo neuronal:

- Um conjunto de sinapses ou elos de conexão, cada um caracterizado por um peso ou força própria. Especificamente, um sinal x_j na entrada da sinapse j conectada ao neurônio k e multiplicado pelo peso sináptico w_{kj} . É imponente notar a maneira como são escritos os índices do peso sináptico w_{kj} . O primeiro índice se refere ao neurônio em questão e o segundo se refere ao terminal de entrada da sinapse à qual o peso se refere. Ao contrário de uma sinapse do cérebro, o peso sináptico de um neurônio artificial pode estar em um intervalo que inclui valores negativos bem como positivos.
- Um somador para os sinais de entrada, ponderados pelas respectivas sinapses do neurônio; as operações descritas aqui constituem uma combinação linear.
- Uma função de ativação para restringir a amplitude da saída de um neurônio. Tipicamente, o intervalo normalizado da amplitude da saída de um neurônio é escrito como o intervalo unitário fechado $[1,0]$ ou alternativamente $[-1,1]$.

O modelo neuronal da figura 15 inclui também um *bias* aplicado externamente, representado por b_k . O *bias* b_k tem o efeito de aumentar ou diminuir a entrada líquida da função de ativação por um valor constante, dependendo se ele é positivo ou negativo, respectivamente.

2.2.1 Tipos de funções de ativação

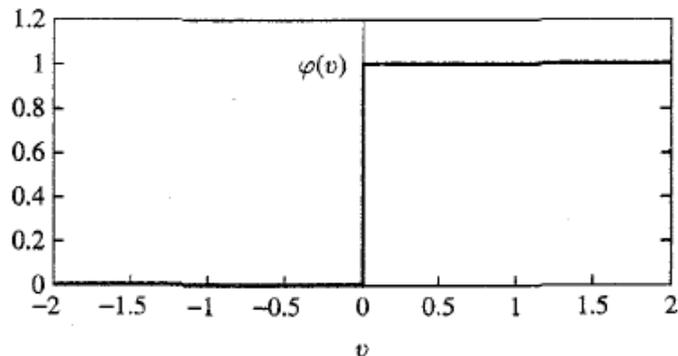
Conforme mencionado na seção 2.2, a operação básica dos neurônios envolve uma soma ponderada dos sinais entrada associada aos pesos sinápticos e uma função de ativação (FAUSETT, 1994). A função de ativação (FA), representada por $\varphi(v)$, define a saída de um neurônio em termos do campo local induzido v . Em outras palavras, ela modula o sinal de saída a ser transmitido (LEAL et al., 2015). Aqui são identificados três tipos básicos de funções de ativação.

Função limiar

Função limiar: para este tipo de FA, descrita na figura 16, tem-se:

$$\varphi(v) = \begin{cases} 1, & \text{se } v \geq 0 \\ 0, & \text{se } v < 0 \end{cases} \quad (16)$$

Figura 16 – Função de ativação limiar



Fonte: HAYKIN, Simon. *Neural networks and learning machines*. 3 ed. Porto Alegre: Prentice Hall: New Jersey, 2009.

Na literatura de engenharia, esta forma de função de limiar é normalmente referida como função de Heaviside. Correspondentemente, a saída do neurônio k que emprega esta função de limiar é expressa como:

$$y_k = \begin{cases} 1, & \text{se } v_k \geq 0 \\ 0, & \text{se } v_k < 0 \end{cases} \quad (17)$$

onde v_k é o campo local induzido do neurônio; isto é

$$v_k = \sum_{j=1}^m w_{kj} x_j + b_k \quad (18)$$

Tal neurônio é referido na literatura como o modelo de McCulloch-Pitts. Neste modelo, a saída de um neurônio assume o valor 1, se o campo local induzido daquele neurônio é não negativo, e 0 caso contrário.

Função linear por partes

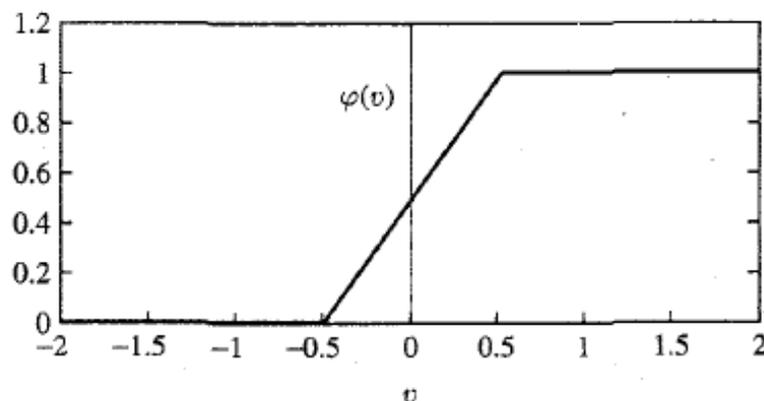
Função linear por partes: Para a função linear por partes descrita na figura 17, tem-se:

$$\varphi(v) = \begin{cases} 1, & v \geq +\frac{1}{2} \\ v, & +\frac{1}{2} > v > -\frac{1}{2} \\ 0, & v \leq -\frac{1}{2} \end{cases} \quad (19)$$

onde assume-se que o fator de amplificação dentro da região linear de operação é a unidade. Esta forma de função de ativação pode ser vista como uma aproximação de um amplificador não-linear. As duas situações seguintes podem ser vistas como formas especiais da função linear por partes:

- Se a região linear de operação é mantida sem entrar em saturação, surge um combinador linear.
- A função linear por partes se reduz função de limiar: se o fator de amplificação da região linear é feito infinitamente grande.

Figura 17 – Função de ativação linear por partes



Fonte: HAYKIN, Simon. *Redes neurais: princípios e prática*. 2 ed. Porto Alegre: Bookman, 2001

Função sigmoide

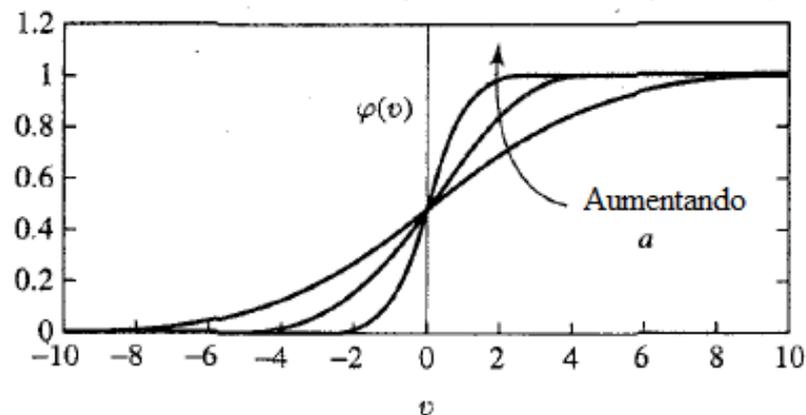
A função sigmoide, cujo gráfico tem a forma de “s” é de longe a forma mais comum de função de ativação utilizada na construção de redes neurais artificiais. Ela é definida como

uma função estritamente crescente que exibe um balanceamento adequado entre comportamento linear e não-linear. Um exemplo de função sigmoide é a função logística, definida por:

$$\varphi(v) = \frac{1}{1+\exp(-av)} \quad (20)$$

onde a é o parâmetro de inclinação da função sigmoide. Variando-se o parâmetro a , obtemos funções sigmóides com diferentes inclinações, como ilustrado na figura 18. Na verdade, a inclinação na origem é igual a $a/4$. No limite, quando o parâmetro de inclinação se aproxima do infinito, a função sigmoide se torna simplesmente uma função de limiar. Enquanto a função de limiar assume o valor 0 ou 1, uma função sigmoide assume um intervalo contínuo de valores entre 0 e 1. Note também que a função sigmoide é diferenciável enquanto a função de limiar não o é.

Figura 18 – Função de ativação sigmoide



Fonte: HAYKIN, Simon. Neural networks and learning machines. 3 ed. Porto Alegre: Prentice Hall: New Jersey, 2009.

As FAs limiar, linear por partes e sigmoide se estendem de 0 a +1. Algumas vezes é desejável que a FA se estenda de -1 a +1, assumindo neste caso uma forma antissimétrica em relação à origem; isto é, a FA é uma função ímpar do campo local induzido. Especificamente, a função de limiar da equação 16 é definida agora como

$$\varphi(v) = \begin{cases} 1, & v > 0 \\ 0, & v = 0 \\ -1, & v < 0 \end{cases} \quad (21)$$

a qual é normalmente denominada função sinal. Para a forma correspondente de uma função sigmoide, podemos utilizar a função tangente hiperbólica, definida por

$$\varphi(v) = \tanh(v) \quad (22)$$

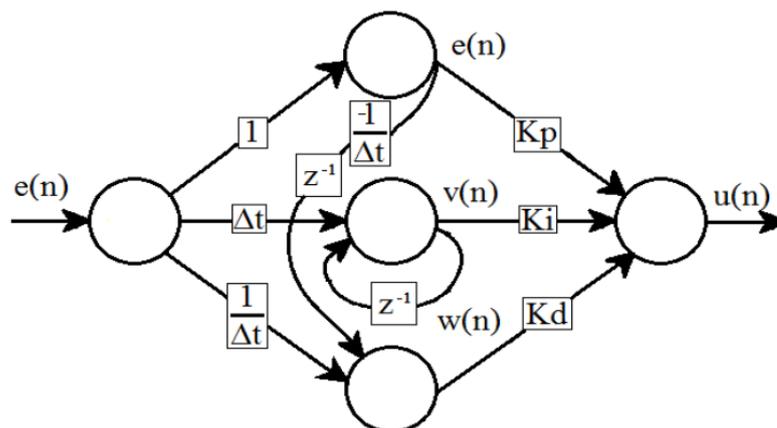
2.2.2 Topologias (ou arquiteturas) de RNAs

A maneira pela qual os neurônios de uma rede neural estão estruturados ligam-se intimamente ao algoritmo de aprendizado usado para treinar a rede. Podemos, portanto, falar de algoritmos (regras) de aprendizado utilizados no projeto de redes neurais como sendo estruturado. Nesta seção, enfatiza-se a atenção nas arquiteturas (estruturas) de rede. Dentre a grande quantidade de arquiteturas de redes neurais idealizadas para resolver os mais variados problemas, serão apresentadas aquelas utilizadas neste trabalho.

Uma classe de arquitetura de RNAs que tem sido explorada para controle de sistemas são as *redes recorrentes*. Uma rede neural recorrente se distingue de uma rede neural alimentada adiante – *feedforward* – por ter pelo menos um laço de realimentação. (HOPFIELD, 1985). A presença de laços de realimentação tem um impacto profundo na capacidade de aprendizado da rede e no seu desempenho. Além disso, os laços de realimentação envolvem o uso de ramos particulares compostos de elementos de atraso unitário (representados por z^{-1}), o que resulta em um comportamento dinâmico não-linear, admitindo-se que a rede neural contenha unidades não-lineares. (HAYKIN, 2001).

Assim, um esquema geral que proponha uma RNA recorrente desempenhando o papel de controle análogo ao consagrado controlador PID discreto pode ser examinado na fig. 19.

Figura 19 - Estrutura primária de um controlador PID neural.



Fonte: FERRARI, A. C. Controlador pid sintonizado por redes neurais artificiais. Trabalho de Conclusão de Curso de Graduação em Engenharia Elétrica. UFPR – Universidade Federal do Paraná. Curitiba. 2010.

2.2.2.1 RNA para aproximação de funções

A escolha de uma RNA com esta característica visa mostrar como este tipo de rede pode reproduzir a função $u(t)$ da saída de um controlador. Algumas estratégias para

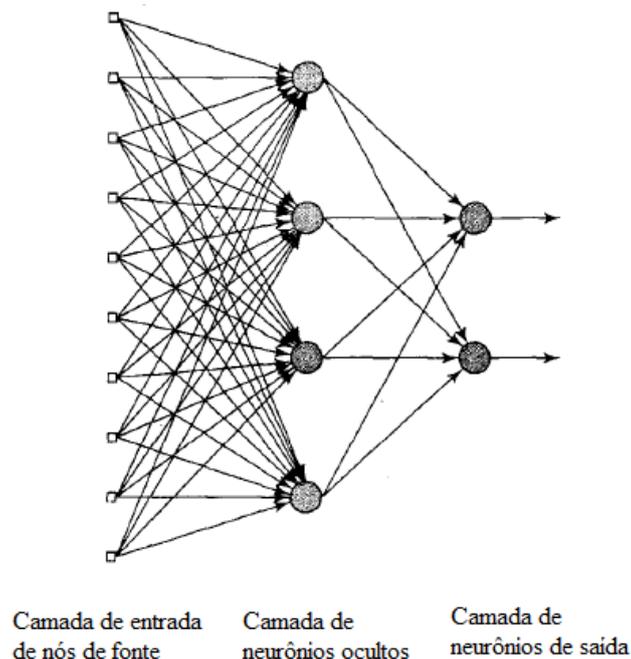
implementação desta arquitetura foram exploradas na literatura (ANTOLINES, QUIROGA, PACHÓN, 2012). No trabalho aqui desenvolvido, para esta tarefa, optou-se uma por RNA do tipo *Multilayer Perceptron* – MLP (fig. 20), com uma camada de neurônios na entrada, uma camada intermediária (também denominada oculta) e uma camada de saída.

Em geral, os MLPs podem trabalhar com qualquer número de neurônios na camada oculta, mas a um custo cada vez mais alto computacionalmente para o treinamento a cada camada adicionada. Como ganho, a partição do espaço N-dimensional das variáveis dos dados se torna mais complexa. O uso de mais de uma camada intermediária se torna necessária quando a função a ser aproximada apresenta descontinuidades. Cybenko (1989) mostra que uma camada é suficiente para aproximar qualquer função contínua.

As funções de ativação normalmente utilizadas são sigmóides nas camadas intermediárias e lineares na camada de saída.

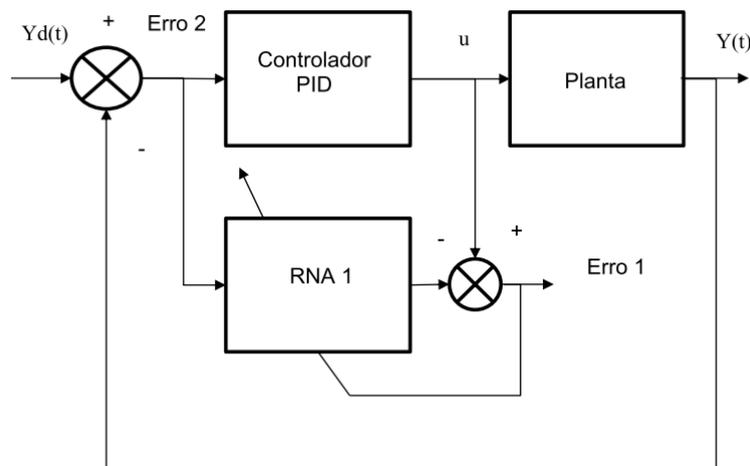
Na fig. 21 observa-se a configuração em que a RNA substitui um controlador por aproximação de sua função de controle. Neste esquema, o controlador PID é sintonizado com o sinal *Erro2* e a RNA é treinada com os conjuntos de entrada *Erro2* e *u*. A saída *Erro1* é utilizada para a correção dos pesos das conexões da RNA, ou seja, o aprendizado da função de controle propriamente.

Figura 20 – Rede tipo *multilayer perceptron* com uma camada oculta e uma camada de saída



Fonte: HAYKIN, Simon. Redes neurais: princípios e prática. 2ª ed. Porto Alegre: Bookman, 2001.

Figura 21 - Configuração de controle modelo de aproximação



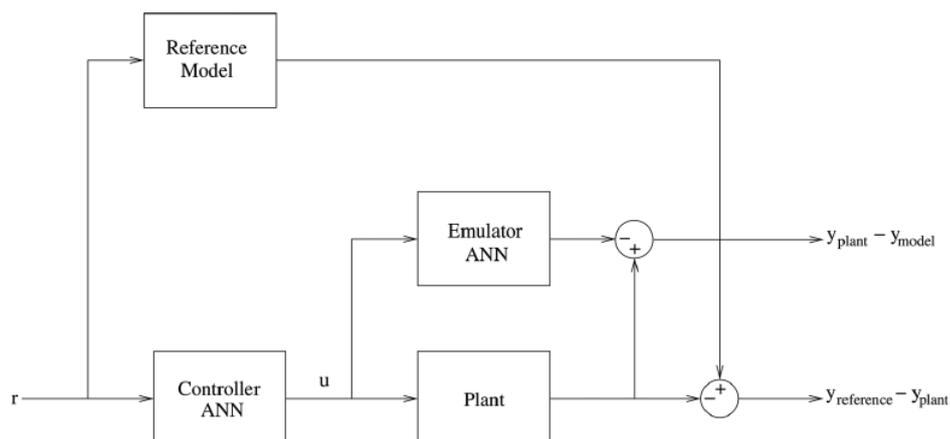
Nota: Elaboração própria.

2.2.2.2 RNA para controle que utiliza um modelo de referência

A configuração *Model Reference Adaptive Control* (MRAC) (Narendra, Parthasarathy, 1990), usa duas redes neurais: uma rede de controle e uma rede para o modelo (Figura 22). A rede do modelo pode ser treinada off-line usando medições históricas da planta. O controlador é treinado de forma adaptável para forçar a saída da planta a rastrear uma saída do modelo de referência. A rede do modelo é usada para prever o efeito das mudanças do controlador na saída da planta, o que permite a atualização dos parâmetros do controlador. No MRAC, definimos um modelo de referência que responde ao sinal de entrada (r) para a resposta desejada da planta. O controlador gera um sinal de controle para a planta, fazendo com que a saída da planta (x_p) siga a saída de referência (x_m). Dependendo do desvio (ou erro), o mecanismo de ajuste atualizará a lei de controle (modificando parâmetros) do controlador. Este processo é repetido iterativamente para que a planta siga o modelo de referência.

A escolha do modelo de referência é norteada pelas características da planta. Se a função de transferência da planta é complicada para ser levantada (uma caixa-preta), um estudo do comportamento não controlado pode fornecer elementos sobre a ordem e a estabilidade do sistema, e parâmetros do sinal de saída. Então, um modelo de função de transferência que contenha estas características (modelo de referência) pode ser elaborado para treinar a respectiva rede a seguir estas características, e a rede controladora para a devida função de controle.

Figura 22 – Configuração *Model Reference Adaptive Control (MRAC)*.



Fonte: BAKSHI, N. A. *Model Reference Adaptive Control of Quadrotor UAVs: A Neural Network Perspective*. Intech, chap. 8, *Adaptive Robust Control Systems*. 2018. Disponível em: <<http://dx.doi.org/10.5772/intechopen.71487>>.

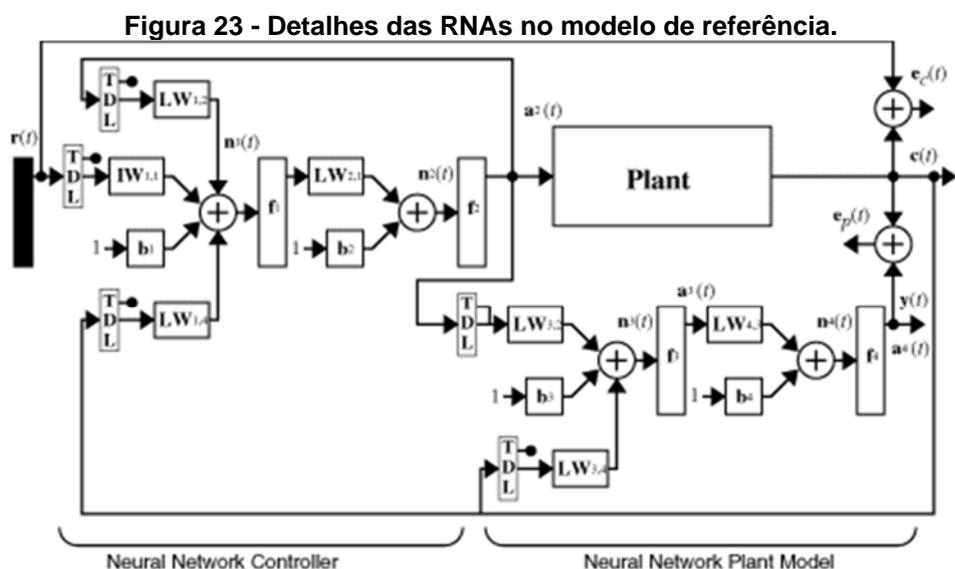
A figura a seguir 23 mostra os detalhes do modelo da planta e do controlador de rede neural. Cada rede tem duas camadas e pode-se selecionar o número de neurônios a serem usados nas camadas ocultas. Existem três conjuntos de dados do controlador:

- Entradas de referência com atraso
- Saídas do controlador com atraso
- Saídas da planta com atraso

Normalmente, o número de atrasos aumenta com a ordem da planta. Existem dois conjuntos de entradas para o modelo de planta de rede neural:

- Saídas do controlador com atraso
- Saídas da planta com atraso

Tal como acontece com a rede neural do controlador, pode-se definir o número de atrasos.



Fonte: Mathworks. Design Model-Reference Neural Controller in Simulink. Disponível em: <<https://www.mathworks.com/help/deeplearning/ug/design-model-reference-neural-controller-in-simulink.html#bss37c6-2>>. Acesso em dez 2018

2.2.2.3 RNA NARMA-L2

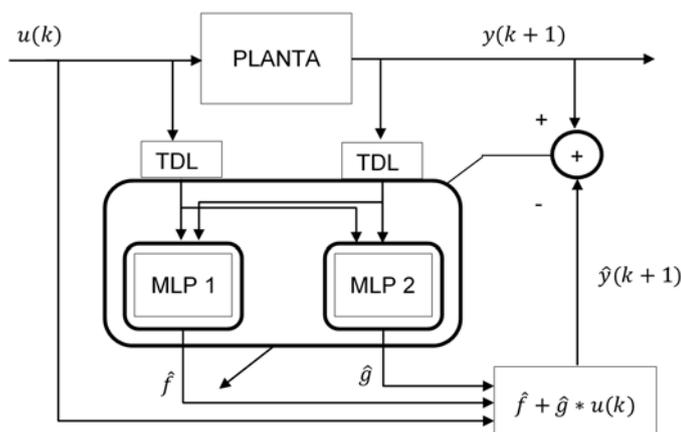
Este modelo foi definido após uma série de outros – resposta ao impulso finita não-linear, média móvel (MA), auto regressivo com entradas externas não-linear (NARX), auto regressivo com média móvel não-linear e com entradas externas (NARMAX), não-linear de erro de saída (NOE). O modelo NARMA-L2 apresenta as características de ser auto regressivo, com linearização da média móvel de uma sequência dos últimos valores do sinal, seja de controle ou de identificação da planta. O sufixo L2 vem do fato de que é uma arquitetura em dois níveis de RNAs (fig. 24).

A ideia central desse tipo de controle é transformar a dinâmica do sistema não linear em dinâmica linear, cancelando as não-linearidades. Este esquema de controle visa aplicar o método de linearização de saída, que se torna uma função linear de uma nova entrada de controle (HAGAN, DEMUTH, 1999).

Há duas etapas envolvidas: identificação do sistema e projeto de controle. No projeto do estágio de identificação do sistema, uma rede neural da planta que precisa ser controlada é desenvolvida usando duas sub-redes para a aproximação do modelo. A rede é então treinada *offline* em forma de lote usando dados coletados da operação da planta. Em seguida, o controlador é construído pelo rearranjo de duas sub-redes do modelo de planta.

As duas sub-redes são usadas para a aproximação do modelo; NN1 e NN2 que são usados para aproximar funções não lineares f e g respectivamente. A estrutura de identificação do sistema NARMA-L2 do manipulador de link único é mostrada na Figura 24.

Figura 24 - Rede NARMA-L2 configurada para a etapa de identificação do sistema.

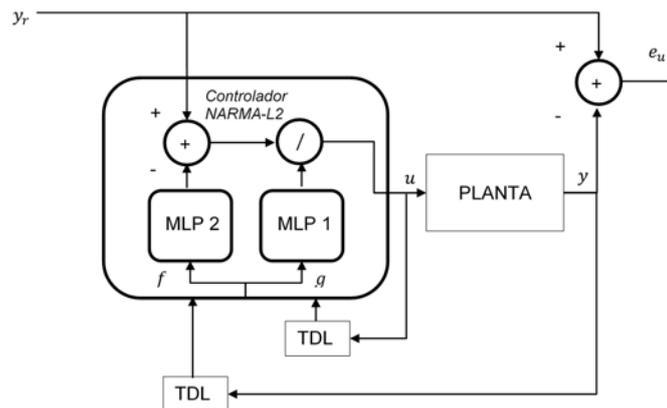


Fonte: KASSEM, A. M. Nonlinear Autoregressive Moving Average Controller for Isolated Wind Generation System Control. Journal of Engineering Sciences, Assiut University, Vol. 40, Nº 1, pp.209-222, janeiro 2012.

A identificação do modelo da planta no controle NARMA-L2 começa com um conjunto de dados de pares de dados de saída de entrada coletados usando o modelo matemático da planta para treinar e validar o modelo da rede neural. Para cada sub-rede, são determinados o número de saídas enviadas n e a entrada anterior m ; que compõem o vetor de entrada e o número de neurônios (p e i) da camada oculta. Subsequentemente, a estrutura da rede neural selecionada é treinada usando o padrão de entrada e a saída desejada do conjunto de dados.

Para a etapa de configuração do controlador, o sistema é rearranjado (fig. 25), de modo a

Figura 25 - Rede NARMA-L2 configurada para a etapa de controle.



Fonte: KASSEM, A. M. Nonlinear Autoregressive Moving Average Controller for Isolated Wind Generation System Control. Journal of Engineering Sciences, Assiut University, Vol. 40, Nº 1, pp.209-222, janeiro 2012.

O processo de treinamento do controlador neural NARMA-L2 se dá em duas etapas:

- Identificação da planta a ser controlada;
- Treinamento do controlador neural;

Na primeira etapa, duas redes neurais recorrentes são treinadas com os dados da saída do controlador e da planta com atrasos. Em seguida, os sinais de saída de ambas as redes são utilizados numa operação discreta, onde juntos com o sinal de saída do modelo de referência fazem gerar o sinal de controle (HOUFAR, SALAHSHOOR e PAIVAR, 2006).

2.2.3 Aprendizado das redes

A propriedade que é de importância primordial para uma rede neural é a sua habilidade de aprender a partir de seu ambiente e de melhorar o seu desempenho através do aprendizado. A melhoria do desempenho ocorre com o tempo de acordo com alguma medida preestabelecida. Uma rede neural aprende acerca do seu ambiente através de um processo interativo de ajustes aplicados a seus pesos sinápticos e níveis de bias. Idealmente a rede se torna mais instruída sobre o seu ambiente após cada iteração do processo de aprendizado. No contexto de redes neurais, define-se o aprendizado como:

Aprendizado é um processo pelo qual os parâmetros livres de uma rede neural são adaptados através de um processo de estimulação pelo ambiente no qual a rede está inserida. O tipo de aprendizado é determinado pela maneira pela qual a modificação dos parâmetros ocorre. (HAYKIN, 2001)

Esta definição do processo de aprendizado implica a seguinte sequência de eventos:

- I. A rede neural é estimulada por um ambiente
- II. A rede neural sofre modificações nos seus parâmetros livres como resultado desta estimulação.
- III. A rede neural responde de uma maneira nova ao ambiente, devido às modificações ocorridas na sua estrutura interna.

Um conjunto preestabelecido de regras bem-definidas para a solução de um problema de aprendizado é denominado um *algoritmo de aprendizado*. Como se pode esperar, não há um algoritmo de aprendizado único para o projeto de redes neurais. Em vez disso, temos um "conjunto de ferramentas" representado por uma variedade de algoritmos de aprendizado, cada qual oferecendo vantagens específicas. Basicamente, os algoritmos de aprendizado diferem entre si pela forma como é formulado o ajuste de um peso sináptico de um neurônio. Outro fator a ser considerado é a maneira pela qual uma rede neural (máquina de aprendizado), constituída de um conjunto de neurônios interligados, se relaciona com o seu ambiente. A escolha de um algoritmo de aprendizado particular é influenciada pela tarefa de aprendizado que uma rede neural deve executar.

Aprendizado supervisionado

Este modo de aprendizado na rede neural implica em um processo supervisorio que organiza dados de entrada como estímulos de diversos padrões e observa as saídas correspondentes, comparando-as com as saídas desejadas que se queira que a rede

“aprenda”. A diferença produz um valor de erro, que é minimizado com algoritmos de correção, pelo reajuste dos valores dos pesos das conexões neuronais. Os algoritmos mais utilizados são a Regra Delta e o *backpropagation* (BRAGA, CARVALHO, LUDERMIR, 2007).

O aprendizado supervisionado é aplicado nos casos em que a rede deve mapear padrões ou valores de entrada para classes ou valores de saída.

Aprendizado não-supervisionado

Não há um processo supervisório. Somente os padrões de entrada estão disponíveis para a rede. A presença de padrões na estrutura dos dados permite com que a rede generalize novos dados, classificando-os nos padrões encontrados por similaridade entre os atributos dos dados existentes e os novos. Os modelos de rede mais conhecidos são os mapas auto-organizáveis de Kohonen e os modelos A.R.T. O algoritmo de aprendizado de Hebb também se encaixa nessa definição, embora necessite dos valores de saída a serem aprendidos (HAYKIN, 2001; BRAGA, CARVALHO, LUDERMIR, 2007)

2.2.4 Treinamento, teste, validação – generalização

O objetivo principal quando se treina uma RNA é obter modelos com capacidade boa de generalização, ou seja, reproduzir a saída desejada dentro de certa variabilidade nos dados de entrada. À medida que a rede vai sendo treinada com os dados de entrada, o erro entre a saída desejada e a obtida vai sendo minimizado. No limite, a rede estará com um acerto de 100% (erro zero), mas apenas para o conjunto de dados de entrada apresentados no treinamento.

Então, é necessário um método para avaliar a capacidade de generalização da rede para outros dados similares (mas não iguais) aos utilizados no treinamento. A técnica para isto é dividir os dados em *treinamento* (cerca de 70%) e *teste* (cerca de 30%). Após a rede ser treinada com os dados de treinamento, a capacidade de acerto da mesma é medida com os dados de teste.

No entanto, pode ocorrer a condição de *superajuste* (*overfitting*), na qual a RNA apresenta uma grande proporção de acertos para o conjunto de treinamento, mas uma proporção bem inferior para o conjunto de teste. Por outro lado, uma RNA que generalize demais para qualquer conjunto de dados pode não ser eficiente no aprendizado e classificação de padrões existentes. Este é um problema conhecido como *dilema bias-variância* (HAYKIN, 2001).

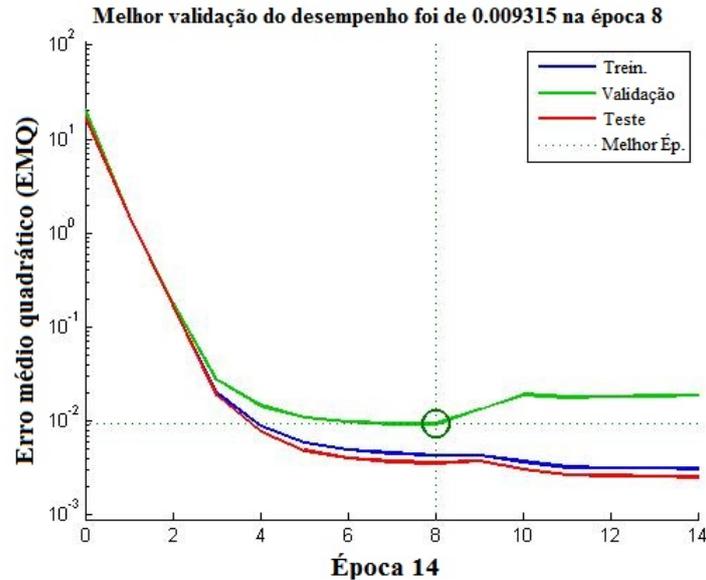
Uma estratégia para minimizar este problema é dividir o conjunto de dados em três partes: *treinamento* (50 a 70%), *teste* (20 a 30%) e *validação* (20 a 30%). À medida que o treinamento ocorre, o algoritmo de correção do erro (pelo reajuste dos pesos das conexões) é aplicado para os dados de treinamento. A cada *época* ou *ciclo* (vezes em que o conjunto de dados de treinamento passa pela rede), é calculada a proporção de acerto para o conjunto de validação e de teste. É importante lembrar que o subconjunto de teste não é usado para correção de erros, apenas para avaliar o desempenho da rede. Os dados de validação podem ser aplicados a cada *N* épocas para que os pesos sejam reajustados, de modo a evitar o *overfitting* sobre os dados de treinamento.

Nos primeiros ciclos ou épocas, os erros para os três subconjuntos de dados vão diminuindo, mas quando a rede neural tende a um *overfitting* sobre os dados de treinamento, o erro relacionado a eles continua tendendo a zero, mas o erro relacionado aos dados de validação começa a aumentar. Ou seja, a rede vai perdendo a capacidade de generalização para outros dados que não os de treinamento. A figura 26 ilustra este fenômeno. O Quadro 1 mostra o pseudoalgoritmo de treinamento da rede por épocas.

Quadro 1 – exemplo de treinamento iterativo

```
Inicialize os pesos da rede com valores pequenos e aleatórios
Repita:
    Embaralhe o conjunto de dados
    Repita:
        Apresente um dado à rede neural
        Se erro na saída for maior que um limiar predeterminado,
            Então executar o algoritmo de reajuste de pesos
    Até que termine todos os dados do conjunto de treinamento
    Calcular os erros para os conjuntos de teste e validação
Até que não haja mais reajuste de pesos
```

Figura 26 - Decaimento do erro quadrático médio durante o treinamento da rede



Nota: elaboração própria.

Conclusão do capítulo

Conforme estabelecido nos objetivos deste trabalho, a simulação de controle de sistemas eletromecânicos por RNAs empregou as seguintes estratégias:

1. Substituição do controlador por uma RNA;
2. Controle da planta pelo uso do controlador neural NARMA-L2;
3. Controle da planta pelo uso do controlador neural Modelo de Referência;

Sendo assim, das tarefas de aprendizagem das RNAs apresentadas neste capítulo, a primeira estratégia utilizou a aproximação da função do controlador pela RNA, treinada por um PID sintonizado e a segunda estratégia utilizou a aprendizagem para a identificação do sistema e aprendizagem por correção de erro. Na terceira estratégia é utilizada a configuração Modelo de Referência.

É importante enfatizar que a escolha das entradas de teste a serem utilizadas para analisar as características do sistema é relativa, pois depende da forma de solicitação a que o sistema será sujeito, ou sob as condições normais de operação.

Assim, para este trabalho é utilizada como entrada de teste a *função degrau* visto que, conforme mencionado no tópico 3.2, são bastante úteis para observar a resposta transitória do sistema. Dessa forma pode-se avaliar a resposta do sistema controlado por meio dos cinco parâmetros de desempenho transitório. Além disso, também foram utilizados sequências de

degraus de intensidade (limitada ao escopo normalizado 0 a 1) e duração aleatória, sendo aqui denominados *degraus aleatórios*.

3 MODELAGEM DOS SISTEMAS DINÂMICOS ESCOLHIDOS

Neste trabalho são estudados três casos de sistemas dinâmicos. O primeiro é um sistema de controle de posicionamento de antena em azimute³ que “converte um comando de posição de entrada em uma resposta de posição de saída” (NISE, 2000).

O segundo é um sistema de controle de um motor elétrico de tração de um veículo de ferrovia (DORF e BISHOP, 2013). Neste projeto, o sistema irá converter uma entrada em tensão numa velocidade de saída para o motor.

O terceiro modela o controle de suspensão ativa de uma das quatro rodas de um veículo (modelo de 1/4 de veículo), representada por dois sistemas massa-mola em série, o primeiro sistema modelando as molas e o amortecedor e o segundo modelando o pneu (sua constante elástica e respectivo amortecimento) (MESSNER et al, 2018)⁴; (VALENTIM, 2017)

Foram simuladas situações de respostas dos sistemas em cinco configurações:

1. sem controle,
2. com controlador PID,
3. com um controlador neural pelo método de aproximação de função
4. com um controlador neural do tipo NARMA-L2 e
5. com um controlador neural do tipo Modelo de Referência

O desempenho dos controladores neurais é comparado com o desempenho de um controlador PID, para avaliar a eficiência das redes no controle de sistemas dinâmicos, por meio dos índices já relatados.

3.1 Análise de estabilidade dos sistemas

Um sistema pode ser caracterizado como estável se sua resposta tende para um valor final ou como instável se sua resposta for oscilatória ou oscilatória com valor final tendendo para o infinito.

Para a verificação de estabilidade de um sistema, utiliza-se frequentemente o critério de Routh Hurwitz. Esse critério informa se há ou não raízes instáveis, isto é, se existem polos

³ Antenas de radiotelescópio, por exemplo, que são usadas em laboratórios de observação de corpos celestes. Seu posicionamento é dado pelo sistema azimutal de medição, uma medida de abertura angular cujo valor em graus perfaz horizontalmente um círculo que vai do norte geográfico até a projeção de um alvo com o horizonte.

Fontes:

NISE, Norman S. Engenharia de sistemas de controle. 3 ed. Rio de Janeiro: LTC, 2000.

BOCZKO, R. 1984. Conceitos de Astronomia. São Paulo: Edgard Blücher, pg. 41. Disponível em:

<https://edisciplinas.usp.br/pluginfile.php/111341/mod_resource/content/1/Conceitos%20de%20astronomia.pdf>.

Acesso em: 21/12/2018.

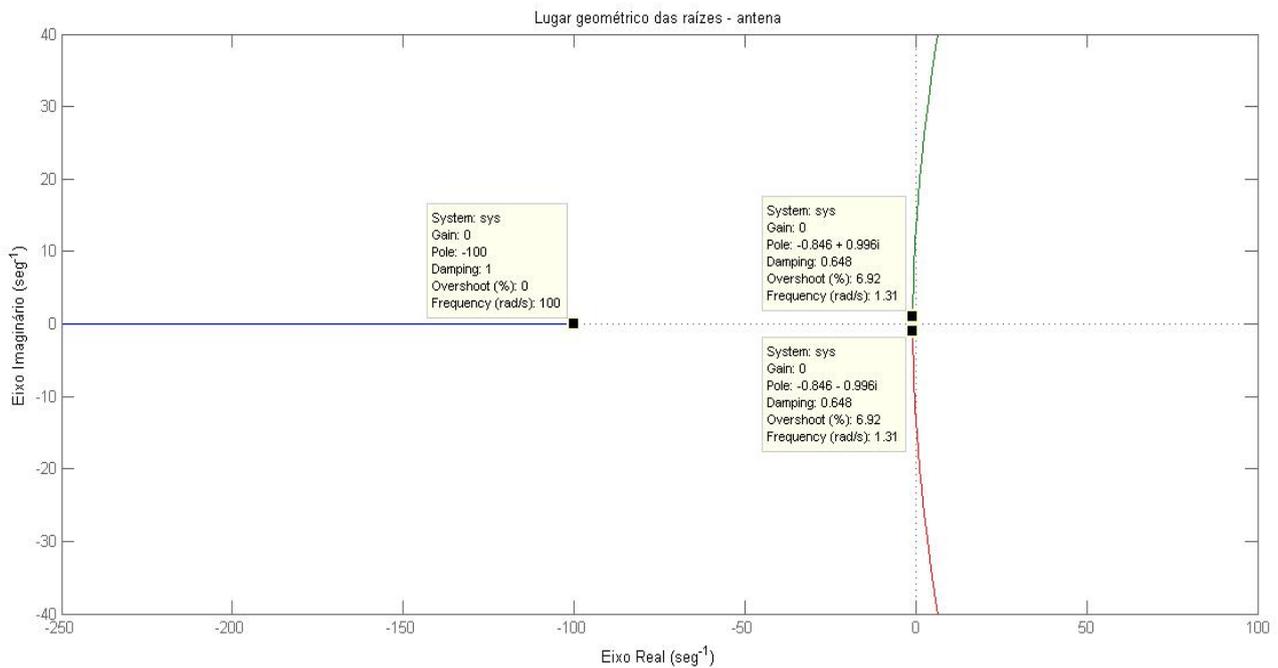
⁴ MESSNER et al, CONTROL TUTORIALS FOR MATLAB & SIMULINK. Suspension. Modeling Simulink. Disponível em:

<<http://ctms.engin.umich.edu/CTMS/index.php?example=Suspension§ion=SimulinkModeling>>. Acesso em: 08/2018.

no semiplano da direita do gráfico das raízes e polos da FT, os quais indicam a instabilidade do sistema.

Obtém-se o seguinte diagrama das raízes e polos da FT do sistema antena de azimute. Realiza-se o mesmo procedimento para os outros sistemas: motor elétrico de tração e suspensão ativa. A análise de estabilidade das figuras 27, 28 e 29 mostram o comportamento do sistema natural, ou seja, sem a ação de controladores.

Figura 27 – Verificação de estabilidade da antena de azimute



Nota: elaboração própria.

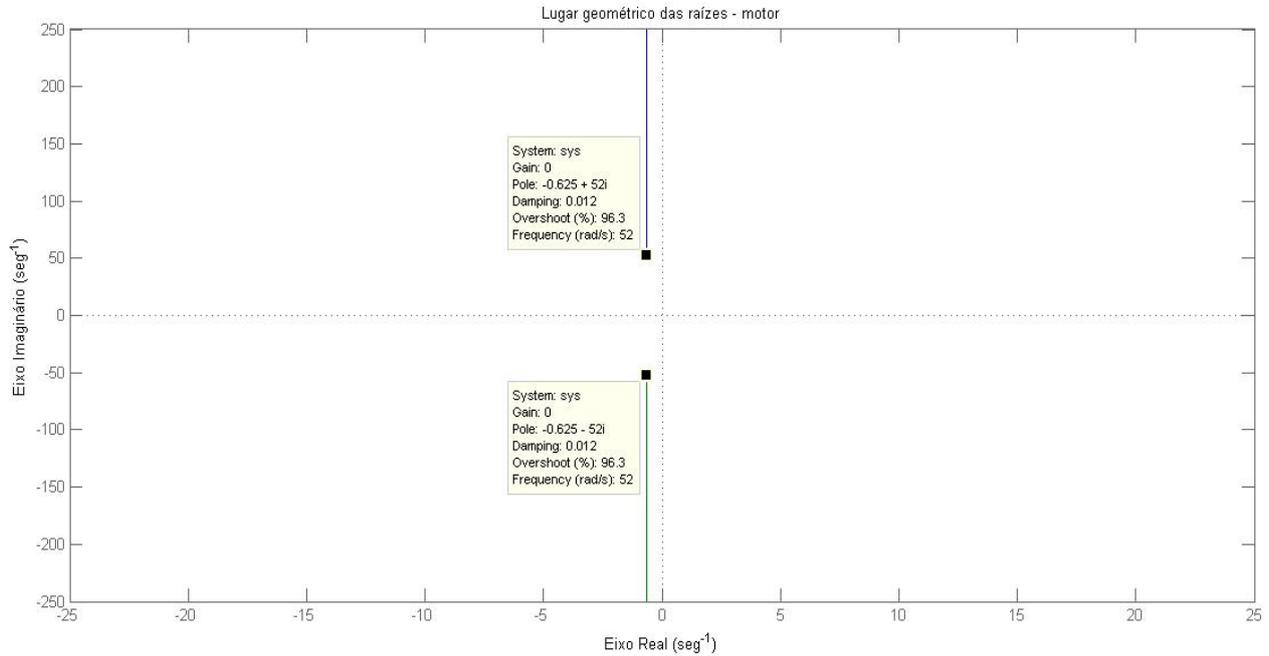
Observa-se na figura 27 que o sistema antena de azimute é um sistema estável com três polos no semiplano esquerdo, o que garante a estabilidade do sistema (NISE, 2000). Isto também demonstra que o sistema é de terceira ordem, visto que o número de polos indica a ordem do sistema.

A figura 27 também mostra que se houver um ganho tendendo ao infinito, os polos irão se deslocar pelas linhas vermelha e azul ao longo do gráfico até chegarem nos zeros (uma vez que o lugar geométrico das raízes inicia em um polo e termina em um zero a medida que um ganho proporcional k se eleva ao infinito), porém em uma região de instabilidade, pois os polos passam para o semiplano direito.

Na figura 28 observa-se que o sistema apresenta dois polos complexos conjugados no semiplano esquerdo. Logo o sistema é estável, pois não há a presença de polos no semiplano direito. A presença destes dois polos indica que o sistema é de segunda ordem. Outra observação importante está no aumento gradativo de um ganho proporcional que faz com que

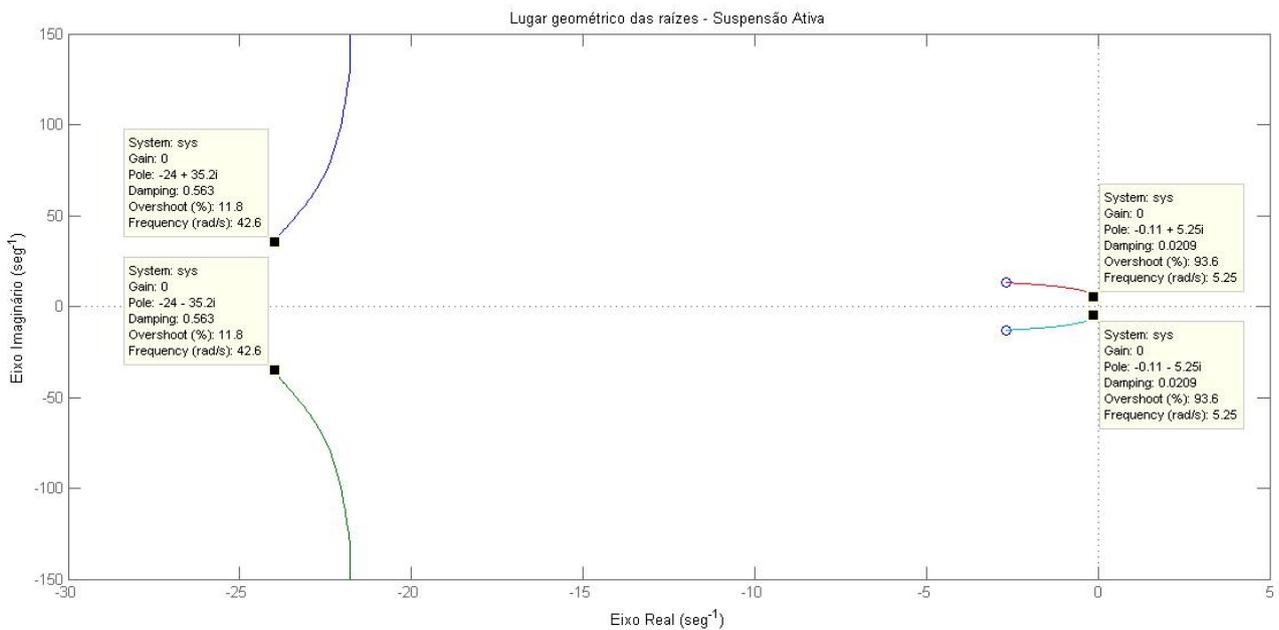
estes polos se deslocem até zeros no infinito do gráfico da figura 28. Estes polos conjugados irão se afastar ao infinito, mas não irão sair da região de estabilidade (semiplano esquerdo).

Figura 28 – Verificação de estabilidade do motor de tração



Nota: elaboração própria.

Figura 29 – Verificação de estabilidade da suspensão ativa



Nota: elaboração própria.

O sistema da figura 29 é estável, pois apresenta todos os quatro polos no semiplano esquerdo. Também apresenta dois zeros no semiplano esquerdo. Embora dois polos estejam

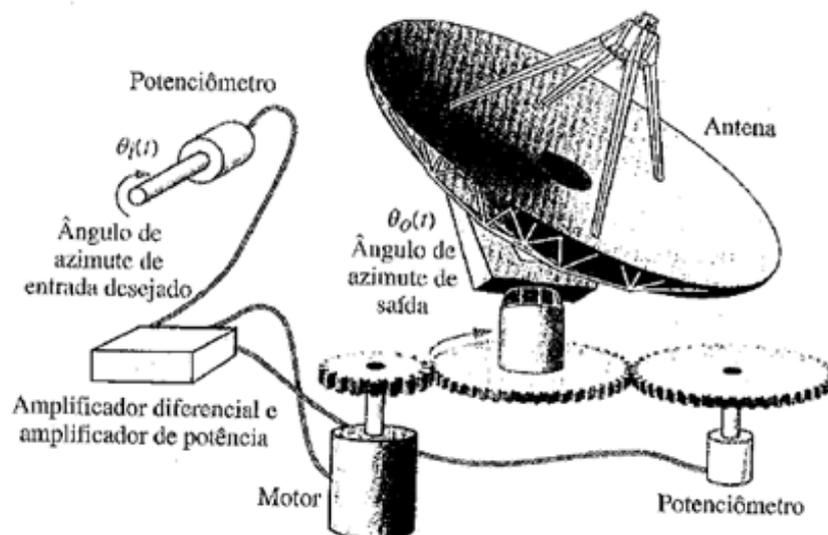
muito próximos do eixo real, estes não passarão ao semiplano direito, uma vez que, com o aumento gradativo de um ganho proporcional estes polos se deslocariam até os zeros que se encontram mais à esquerda. Este sistema é de quarta ordem, visto que possui quatro polos.

Com isto, verifica-se que os sistemas estudados apresentam condições de estabilidade para o controle, sendo viável o estudo da implementação das estratégias descritas nos objetivos do trabalho.

3.2 Antena de azimute

Uma antena de radiotelescópio é controlada através do movimento gerado por um motor acoplado a um conjunto de engrenagens responsáveis por transmitir esse movimento ao eixo da antena. A posição angular do motor, por sua vez, é manipulada por um potenciômetro ligado a um amplificador de potência. O esquema da figura 30 exhibe os componentes da antena.

Figura 30 - componentes da antena de azimute



Fonte: NISE, Norman S. Engenharia de sistemas de controle. 3 ed. Rio de Janeiro: LTC, 2000.

O objetivo desse sistema é descrito por Nise (2000):

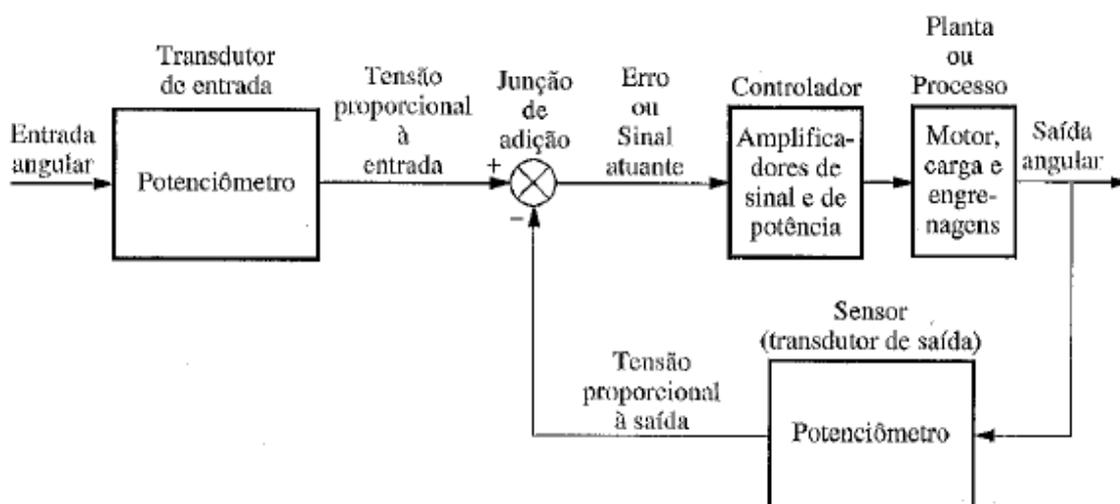
A finalidade desse sistema é fazer com que o ângulo de azimute de saída da antena ($\theta_o(t)$), siga o ângulo de entrada do potenciômetro ($\theta_i(t)$). [...] O comando de entrada é um deslocamento angular. O potenciômetro converte o deslocamento angular em uma tensão. De modo semelhante, o deslocamento angular de saída é convertido em uma tensão por um potenciômetro no canal de retroação. Os amplificadores de sinal e de potência amplificam a diferença entre as tensões de entrada e de saída. Este sinal amplificado atua sobre o processo.

Estabelecido o objetivo de controle, de acordo com Dorf e Bishop (2013) é necessário a definição das variáveis a serem controladas. Para isso, observe primeiro a figura 31 que descreve o sistema antena de azimute em blocos funcionais.

As variáveis a serem controladas, bem como as demais que compõem o sistema antena de azimute, são descritos na tabela 4.

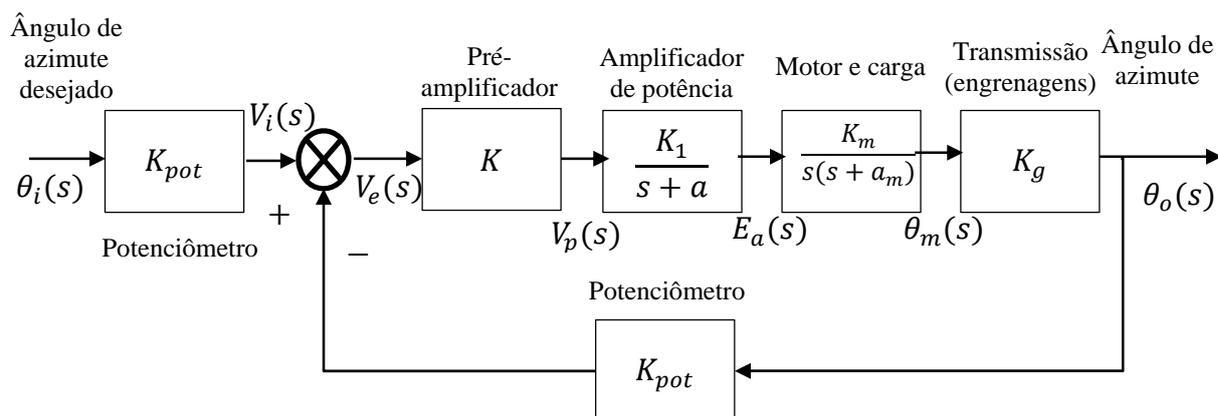
A dinâmica desse sistema é descrita no diagrama de blocos da figura 32, onde os blocos funcionais são substituídos pelas suas respectivas FT's (NISE, 2000).

Figura 31 - sistema antena de azimute em blocos funcionais



Fonte: NISE, Norman S. Engenharia de sistemas de controle. 3 ed. Rio de Janeiro: LTC, 2000.

Figura 32 – Diagrama de blocos da antena de azimute com as funções de transferência



Fonte: NISE, Norman S. Engenharia de sistemas de controle. 3 ed. Rio de Janeiro: LTC, 2000.

Tabela 4 - variáveis do sistema antena de azimute

| Subsistema | Entrada | Saída |
|------------|---------|-------|
|------------|---------|-------|

| | | |
|---------------------------------|--|---|
| Potenciômetro de entrada | Rotação angular vista pelo usuário, $\theta_i(t)$ | Tensão para pré-amplificador $v_i(t)$ |
| Pré-amplificador | Tensão a partir dos potenciômetros $v_e(t) = v_i(t) - v_o(t)$ | Tensão para o amplificador de potência $v_p(t)$ |
| Amplificador de potência | Tensão do pré-amplificador $v_p(t)$ | Tensão para o motor $e_a(t)$ |
| Motor | Tensão do amplificador de potência $e_a(t)$ | Rotação angular para a carga, $\theta_o(t)$ |
| Potenciômetro de saída | Rotação angular vista da carga, $\theta_i(t)$ | Tensão para o pré-amplificador $v_o(t)$ |

Fonte: Adaptado de: NISE, Norman S. Engenharia de sistemas de controle. 3 ed. Rio de Janeiro: LTC, 2000.

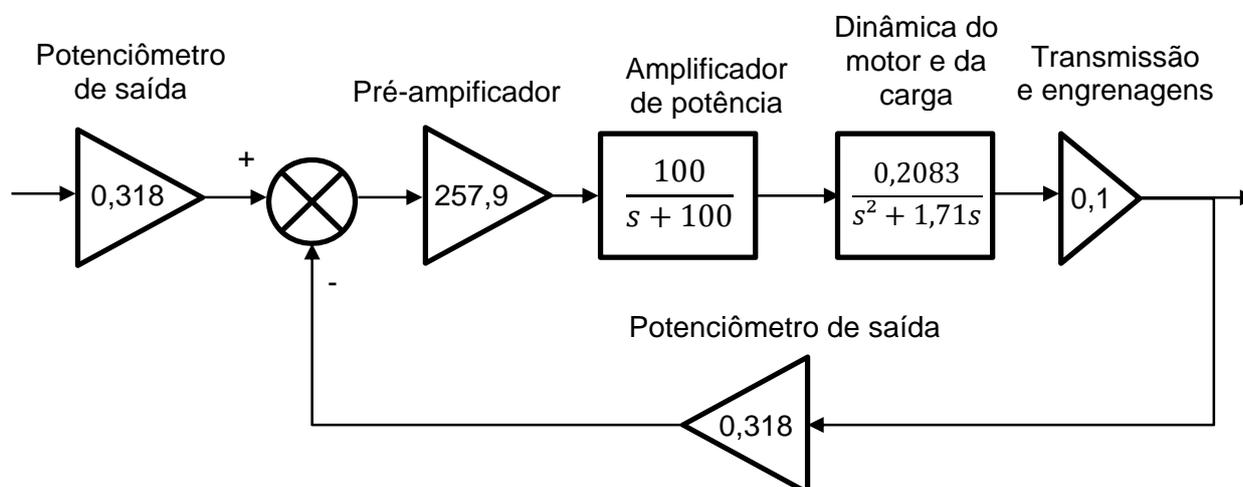
Com o sistema modelado, é necessário inserir os valores das constantes referentes ao diagrama de blocos da figura 32. Substituindo os valores dos parâmetros da tabela 5 no diagrama da figura 32, o sistema antena de azimute corresponde ao seguinte diagrama de blocos da figura 33:

Tabela 5 - contém os parâmetros das constantes do sistema antena de azimute

| | |
|-----------|--------|
| K_{pot} | 0.318 |
| K | 257.9 |
| K_1 | 100 |
| a | 100 |
| K_m | 0.2083 |
| a_m | 1.71 |
| K_g | 0.1 |

Nota: Elaboração própria.

Figura 33 – Diagrama de blocos do sistema antena de azimute com as constantes inseridas



Nota: Elaboração própria.

Pode-se reduzir o diagrama de blocos da figura 33 até chegar a uma FT de malha aberta. Para isto basta utilizar as regras de redução de diagrama dados por Dorf e Bishop (2013).

$$\frac{170,8}{s^3 + 101,7s^2 + 171s + 170,8} \quad (23)$$

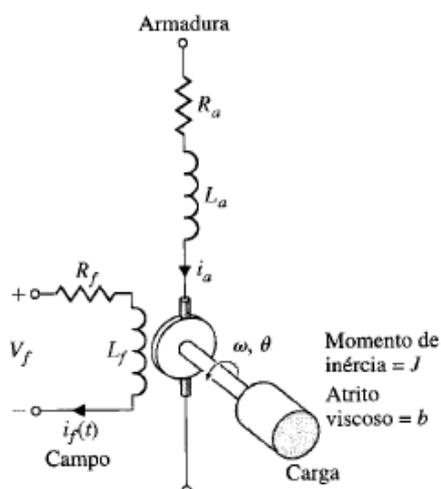
Observa-se pela FT na equação 23 que o sistema antena de azimute é um sistema de terceira ordem. Observa-se na equação característica o grau mais elevado do termo s , e percebe-se que este sistema possui três polos. Não há a presença de zeros neste sistema, uma vez que o numerador da equação 23 não possui nenhum termo em s .

3.3 Motor elétrico CC

O motor CC é um dispositivo atuador de potência que entrega energia a uma carga. Um esboço de um motor está mostrado na figura 34.

Internamente, o motor CC converte energia elétrica de corrente contínua (CC) em energia mecânica rotativa. Uma fração importante do torque gerado no rotor (armadura) do motor está disponível para acionar uma carga externa.

Figura 34 – Esboço do motor CC

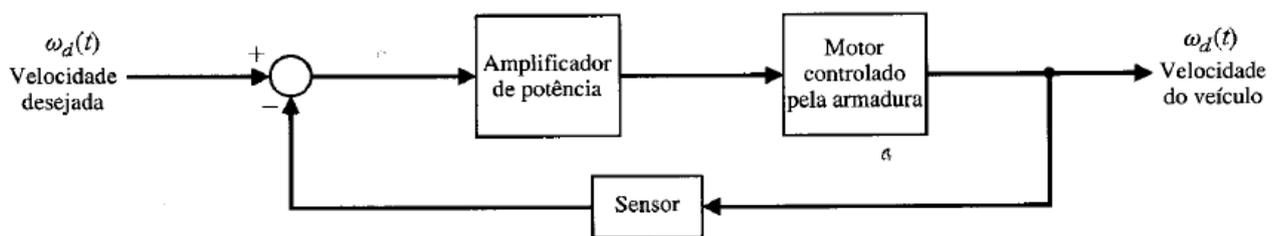


Fonte: DORF, Richard C., BISHOP, Robert H. Sistemas de controle modernos. 12 ed. Rio de Janeiro: LTC, 2013.

A maioria dos trens modernos e dos veículos de trânsito local utiliza motores elétricos de tração. Para este tipo de veículo, é comum ter-se uma alta carga aplicada aos motores, uma vez que eles estão impulsionando uma grande massa composta pela carcaça da locomotiva e os passageiros ou material transportado. Seu deslocamento faz-se de forma gradativa até atingir uma velocidade considerável.

O acionamento de um motor elétrico para veículo de ferrovia está mostrado sob a forma de diagrama de blocos na figura 35, incorporando o controle necessário da velocidade do veículo.

Figura 35 – Diagrama de blocos do acionamento de um motor elétrico para veículo de ferrovia



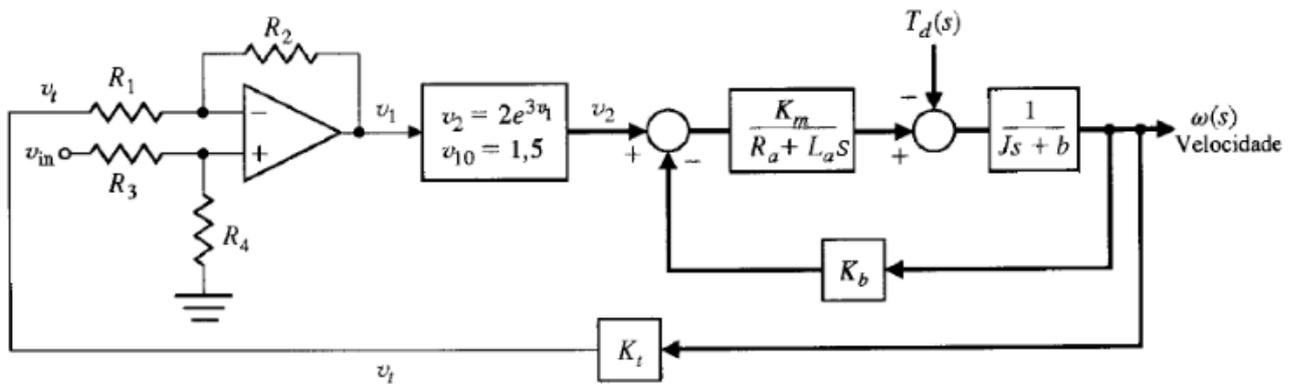
Fonte: DORF, Richard C., BISHOP, Robert H. **Sistemas de controle modernos. 12 ed.** Rio de Janeiro: LTC, 2013.

O objetivo do projeto segundo Dorf e Bishop (2013), é obter um modelo de sistema e a FT a malha fechada do sistema, $\omega(s)/\omega_d(s)$, selecionar resistores apropriados R_1 , R_2 , R_3 e R_4 , e então prever a resposta do sistema.

Os blocos funcionais podem ser substituídos por FT's (figura 36). Sintetizando, tem-se um tacômetro para gerar uma tensão proporcional a velocidade de saída $\omega(s)$ e conectar esta tensão, v_t , a uma das entradas de um amplificador diferencial, como está mostrado na figura 36. O amplificador de potência é não linear e pode ser representado aproximadamente por $v_2 = 2e^{3v_1} = 2 \exp(3v_1) = g(v_1)$, uma função exponencial com um ponto de operação nominal de, $v_{10} = 1,5V$. A relação dos resistores é apresentada por Dorf e Bishop (2013).

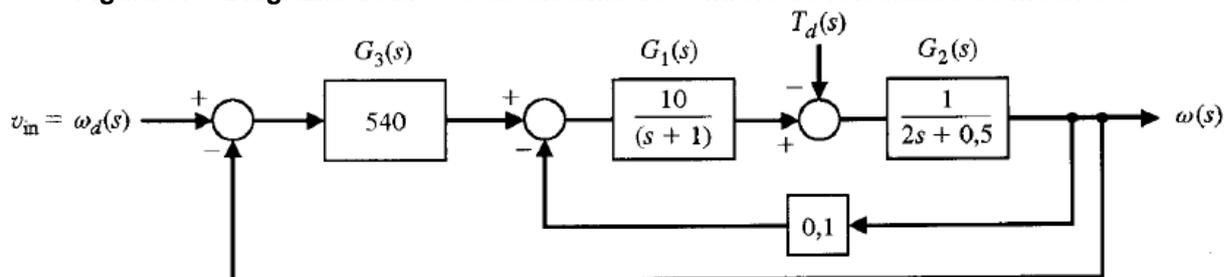
O diagrama de blocos do sistema completo está mostrado na figura 37.

Figura 36 – Controle de velocidade de um motor elétrico de tração



Fonte: DORF, Richard C., BISHOP, Robert H. Sistemas de controle modernos. 12 ed. Rio de Janeiro: LTC, 2013.

Figura 37 – Diagrama de blocos do sistema de controle de velocidade do motor CC



Fonte: DORF, Richard C., BISHOP, Robert H. Sistemas de controle modernos. 12 ed. Rio de Janeiro: LTC, 2013.

Reduzindo o diagrama de blocos da figura 37 (considerando-se $T_d = 0$), obtém-se

$$\frac{\omega(s)}{\omega_d(s)} = \frac{540G_1G_2}{1+0,1G_1G_2+540G_1G_2} = \frac{540G_1G_2}{1+540,1G_1G_2} \quad (24)$$

$$= \frac{5400}{(s+1)(2s+0,5)+5401} = \frac{5400}{2s^2+2,5s+5401,5} \quad (25)$$

$$\frac{2700}{s^2+1,25s+2700,75} \quad (26)$$

Como a equação característica é de segunda ordem, observa-se que $\omega_n = 52$ e $\zeta = 0,012$, e se espera que a resposta do sistema seja altamente oscilatória (subamortecida).

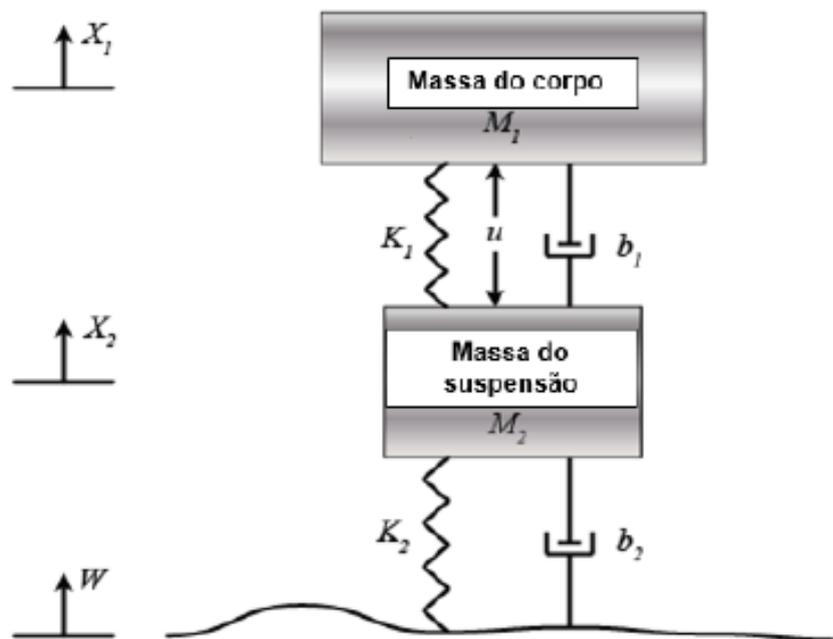
3.4 Suspensão ativa ¼ de roda

Um sistema de suspensão automotiva é um problema comumente abordado em estudos da área de controle. Um bom sistema de suspensão automotiva deve ter uma capacidade de manutenção de estrada satisfatória, ao mesmo tempo em que proporciona

conforto ao andar sobre buracos na estrada. Quando o veículo está experimentando qualquer distúrbio na estrada (ou seja, buracos, rachaduras e pavimento irregular), o corpo do veículo não deve ter grandes oscilações, e as oscilações devem se dissipar rapidamente.

Quando o sistema de suspensão é projetado, um modelo de 1/4 (uma das quatro rodas) é usado para simplificar o problema para um sistema múltiplo em uma dimensão. O sistema é composto basicamente por duas massas (carroceria e pneu), dois corpos elásticos (molas e elasticidade do pneu) e dois amortecimentos (o amortecedor da suspensão e do pneu). Um esquema deste sistema é mostrado na figura 38.

Figura 38 - Modelo de suspensão ativa (1/4).



Fonte: MESSNER, B.; TILBURY, D. CONTROL TUTORIALS FOR MATLAB & SIMULINK. Suspension. Modeling Simulink. Disponível em: <<http://ctms.engin.umich.edu/CTMS/index.php?example=Suspension§ion=SimulinkModeling>>. Acesso em: 08/2018.

A tabela 6 mostra os valores dos parâmetros do sistema de suspensão ativa.

Tabela 6 - valores dos parâmetros do sistema de suspensão ativa

| Simbologia | Descrição | Valores |
|------------|---|------------|
| M_1 | $\frac{1}{4}$ da massa do corpo do ônibus | 2500 kg |
| M_2 | Massa da suspensão | 320 kg |
| K_1 | Mola da suspensão | 80000 N/m |
| K_2 | Mola da roda e pneu | 500000 N/m |

| | | |
|-------|--------------------------------------|-------------|
| b_1 | Amortecedor da suspensão | 350 N.s/m |
| b_2 | Amortecedor da roda e pneu | 15020 N.s/m |
| U | Força de controle do corpo do ônibus | |

Fonte: MESSNER, B.; TILBURY, D. CONTROL TUTORIALS FOR MATLAB & SIMULINK. Suspension. Modeling Simulink. Disponível em: <<http://ctms.engin.umich.edu/CTMS/index.php?example=Suspension§ion=SimulinkModeling>>. Acesso em: 08/2018.

A perturbação da estrada (W) neste problema será simulada por uma entrada de degrau. Este degrau pode representar o veículo saindo de um buraco. Neste projeto, é considerado que a distância X_1-W é muito difícil de medir, e a deformação do pneu (X_2-W) é insignificante. Por isso, usa-se a distância X_1-X_2 em vez de X_1-W como saída do sistema.

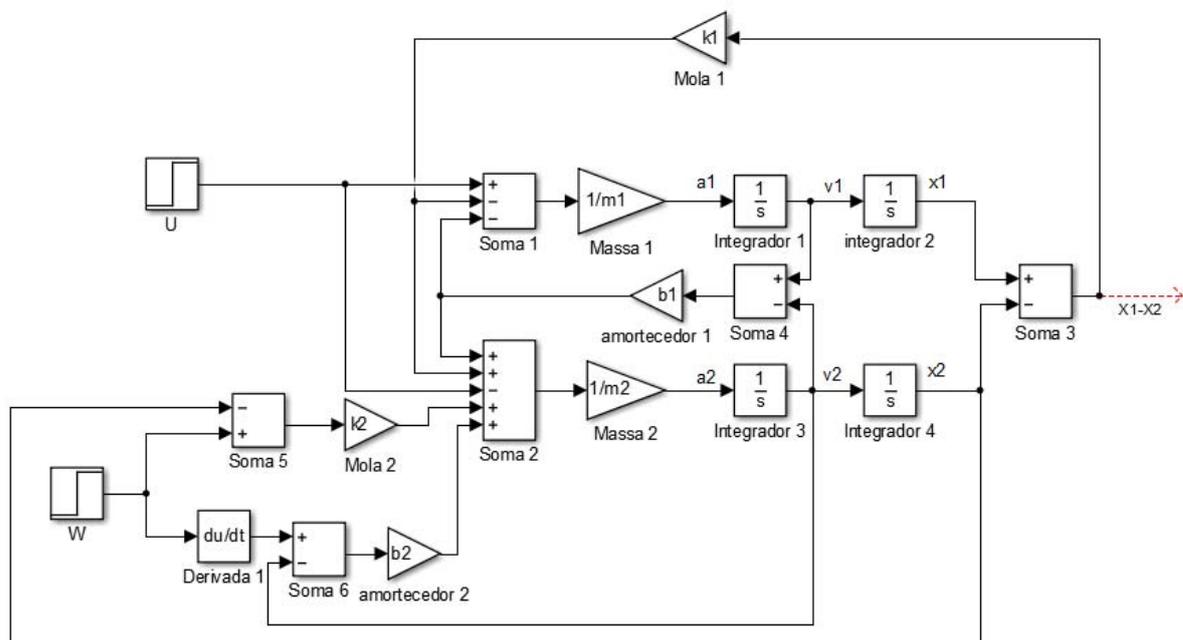
Este sistema pode ser modelado pela soma das forças atuando em ambas as massas (corpo e suspensão) e integrando as acelerações de cada massa duas vezes para dar velocidades e posições. Para isso, a lei de Newton é aplicada a cada massa (equações 27 e 28)

$$\ddot{X}_1 = -\frac{b_1}{M_1}(\dot{X}_1 - \dot{X}_2) - \frac{K_1}{M_1}(X_1 - X_2) + \frac{U}{M_1} \quad (27)$$

$$\ddot{X}_2 = \frac{b_1}{M_2}(\dot{X}_1 - \dot{X}_2) + \frac{K_1}{M_2}(X_1 - X_2) + \frac{b_2}{M_2}(\dot{W} - \dot{X}_2) + \frac{K_2}{M_2}(W - X_2) - \frac{U}{M_2} \quad (28)$$

Essas equações podem ser representadas com blocos de ganho (para $1/M_1$ e $1/M_2$). O diagrama completo pode ser visto na figura 39.

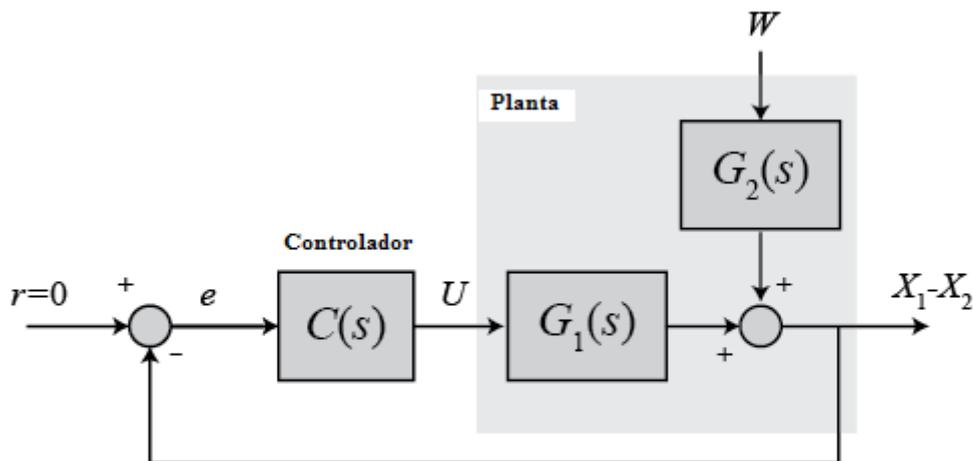
Figura 39 – Diagrama de blocos completo da suspensão ativa.



Nota: elaboração própria.

Este sistema possui uma novidade em relação aos anteriores: possui duas equações dinâmicas que integram o mesmo sistema. Isso se deve ao fato de ao aplicar-se a lei de Newton em relação a cada massa do sistema (ao do corpo e da suspensão), obtém-se como saída as acelerações destas massas. Sendo assim, as FTs são dadas em função da perturbação de entrada entre as massas U ou a perturbação W , originada de alguma saliência do terreno em que se desloca o veículo. Deve-se lembrar que a variável de saída é a diferença entre os deslocamentos das massas $X_1 - X_2$, conforme mencionado no início desta sessão. A figura 40 mostra um diagrama de blocos simplificado do sistema suspensão ativa ¼.

Figura 40 – Diagrama de blocos do sistema suspensão ativa ¼



Fonte: MESSNER, B.; TILBURY, D. CONTROL TUTORIALS FOR MATLAB & SIMULINK. Suspension. Modeling Simulink. Disponível em: <<http://ctms.engin.umich.edu/CTMS/index.php?example=Suspension§ion=SimulinkModeling>>. Acesso em: 08/2018.

A entrada de referência é zero, pois admite-se que o sistema está em equilíbrio. Assim, as FTs são dadas a seguir como:

$$G_1 = \frac{X_1(s) - X_2(s)}{U(s)} \quad (29)$$

$$G_2 = \frac{X_1(s) - X_2(s)}{W(s)} \quad (30)$$

Por fim, a FT deste sistema em malha aberta é:

$$T = \frac{0,003525s^2 + 0,01878s + 0,625}{s^4 + 48,17s^3 + 1851s^2 + 1721s + 50000} \quad (31)$$

Pela equação 31, observa-se que o sistema possui quatro pólos e dois zeros, uma vez que o denominador desta equação possui o termo s^4 , enquanto no numerador ela apresenta o termo s^2 . Por consequência, este é um sistema de quarta ordem.

4 SIMULAÇÕES REALIZADAS E RESULTADOS

Para a simulação foram utilizadas algumas bibliotecas do Simulink do software Matlab.

Neste tópico são analisados tres estudos casos descritos nas sessões 3.2, 3.3 e 3.4: o primeiro é uma antena de azimute, o segundo um motor de tração de veículos de ferrovia e o terceiro é um modelo de suspensão ativa de $\frac{1}{4}$.

Utiliza-se como referência um sinal do tipo degrau e o segundo do tipo sequência de degraus aleatórios cujo propósito é verificar o controle de efeitos não lineares na resposta final dos modelos propostos.

Todos os resultados da simulação são variáveis impressas em função do tempo.

Para cada estrutura de controle neural foram consideradas um número de amostras que varia entre 1000 até 10000 para o treinamento. Para o critério de parada do treinamento são considerados o número de épocas alcançado ou erro mínimo atingido.

O conjunto de amostras foi dividido em dados de treinamento, validação e teste. A quantidade máxima de iterações ou épocas varia entre 10 a 600 épocas dependendo do tipo de neurocontrolador.

4.1 Algoritmos de otimização utilizados para a simulação

São utilizados nas simulações dois tipos de algoritmo de otimização: retro-propagação e Levenberg-Marquardt. Para as simulações dos sistemas controlados pela rede de aproximação, é utilizado o algoritmo de retro-propagação, pois o aplicativo “Neural Network Fitting” foi desenvolvido com este método de otimização. Já para os demais controladores neurais é utilizado o método de Levenberg-Marquardt, pois este algoritmo é estável e converge mais rapidamente em relação aos outros métodos. (ZAMBRANO, 2013)

4.2 Modelos de planta utilizados para o treinamento das redes NARMA-L2 e Modelo de Referência.

O neurocontrolador do tipo NARMA-L2 requer em seu treinamento um modelo de planta que represente a dinâmica do sistema a ser controlado. Para a rede NARMA-L2 são utilizados os modelos de planta dos sistemas sem controle, dados pelas figuras 36 (antena de azimute), 37 (motor de tração) e 39 (suspensão ativa).

O neurocontrolador Modelo de Referência, por sua vez, utiliza dois modelos de planta: um para treinamento da rede de identificação e outro para o treinamento da rede controladora. Para a rede de identificação são utilizados os mesmos modelos de planta utilizados pela rede

NARMA-L2. Já para o treinamento da rede controladora são utilizados os modelos de planta das figuras 33, 37 e 39 com a atuação do controlador PID em série com a planta desses sistemas.

4.3 Especificações de projeto

Conforme Ogata (2010), “as características de desempenho desejadas de sistemas de controle são especificadas em termos de grandezas no domínio do tempo”. Os índices de desempenho transitório mencionados na subseção 2.1.3 são utilizados como parâmetros de desempenho dos sistemas de controle. No entanto, para este trabalho são utilizados apenas dois destes parâmetros como especificações de projeto: a máxima ultrapassagem percentual ou sobressinal (M_p) e o tempo de acomodação (T_s)⁵. Dessa forma, pode-se avaliar se o respectivo controlador atendeu aos requisitos de projeto. Também pode-se realizar uma comparação entre os controladores neurais com o PID pelos demais índices de desempenho transitório.

Além dos índices de desempenho transitório também é considerado como especificação de projeto o erro em regime permanente. Considera-se para todos os sistemas uma tolerância no erro estacionário (e_s) de $\pm 5\%$, que é a faixa de tolerância do tempo de acomodação também.

Assim, as especificações para cada sistema são resumidas a seguir:

- Antena de azimute – $T_s < 3$ segundos, $M_p \leq 10\%$ e $e_{ss} = \pm 5\%$.
- Motor elétrico de tração – $T_s < 3$ segundos, $M_p \leq 15\%$ e $e_{ss} = \pm 5\%$;
- Suspensão ativa $\frac{1}{4}$ – $T_s < 5$ segundos, $M_p \leq 20\%$ e $e_{ss} = \pm 5\%$;

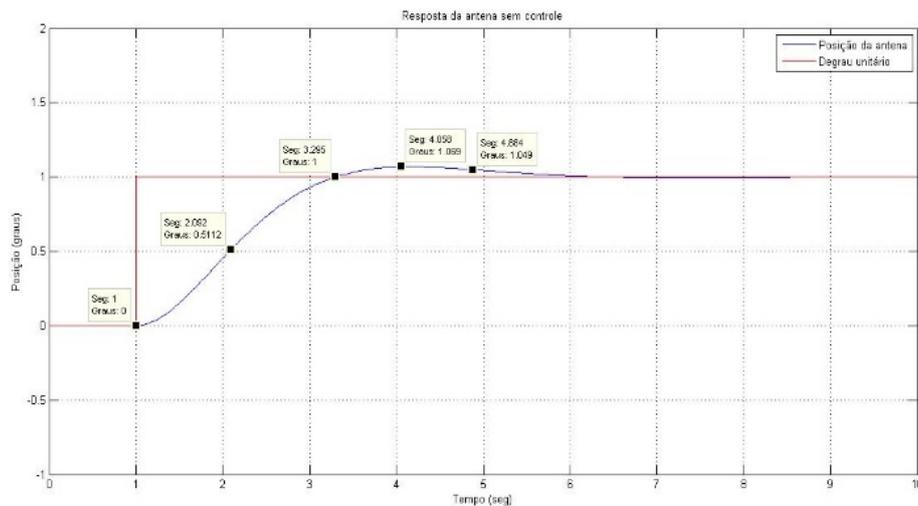
4.4 Antena de azimute, sistema sem controle

Utilizando-se do diagrama de blocos da figura 33 da sessão 3.2 faz-se uma simulação do comportamento dinâmico da resposta do sistema não controlado antena de azimute pelo ambiente gráfico do Simulink do Matlab. Utiliza-se como sinal de entrada o degrau unitário para que se possa obter os índices de desempenho transitórios apresentados na sessão 2.1.3. A curva de resposta é apresentada na figura 41 e os parâmetros de desempenho transitório na tabela 7. Pode-se observar pela figura 41 que o sistema sem um controlador é auto

⁵ As especificações de máximo sobressinal e tempo de acomodação são utilizadas com maior frequência em projetos de controle do que os outros índices.

controlado, ou seja, o sistema consegue acompanhar o sinal de referência aplicado na entrada. A resposta possui pouco sobressinal e com um tempo de acomodação longo de aproximadamente 3,9 segundos. Pode-se melhorar o desempenho desse sistema aplicando-se um controlador que apenas reduza o tempo de acomodação, visto que o sobressinal se encontra dentro do limite das especificações de projeto.

Figura 41 - curva de resposta transitória para uma entrada em degrau unitário do sistema antena de azimuth



Nota: Elaboração própria.

Tabela 7 – Parâmetros de desempenho transitório

| <i>Parâmetros</i> | <i>Valor</i> |
|--|--------------|
| <i>Tempo de atraso (s)</i> | 1.092 |
| <i>Tempo de subida (s)</i> | 2.295 |
| <i>Tempo de pico (s)</i> | 3.058 |
| <i>Tempo de acomodação (s)</i> | 3.884 |
| <i>Máxima ultrapassagem percentual (%)</i> | 6,9 |

Nota: elaboração própria.

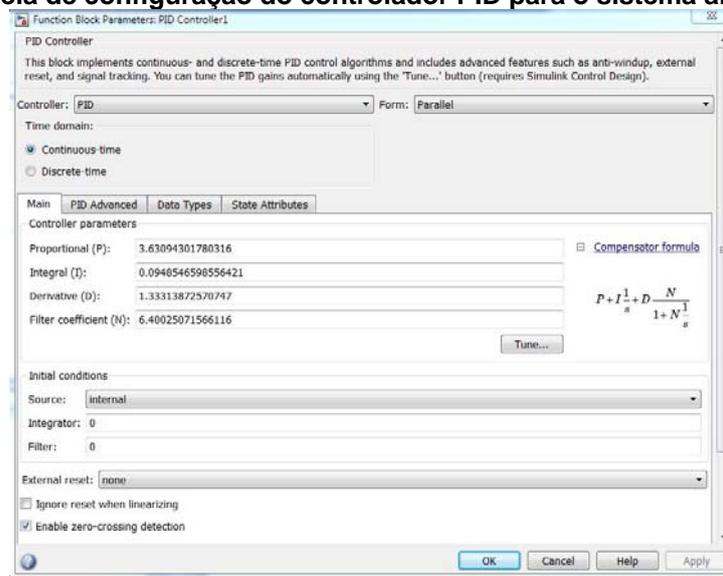
4.4.1 Controle por PID sintonizado

Agora, utilizando-se do mesmo diagrama de blocos da figura 33 (sessão 3.2) faz-se uma nova simulação acrescentando-se um controlador PID em série com o sistema e uma realimentação na entrada para se verificar o desempenho de um controlador PID.

Faz-se então a sintonia do controlador PID para se obter os ganhos do controlador por meio da técnica de sintonia mencionada no apêndice D. Configura-se o controlador com um

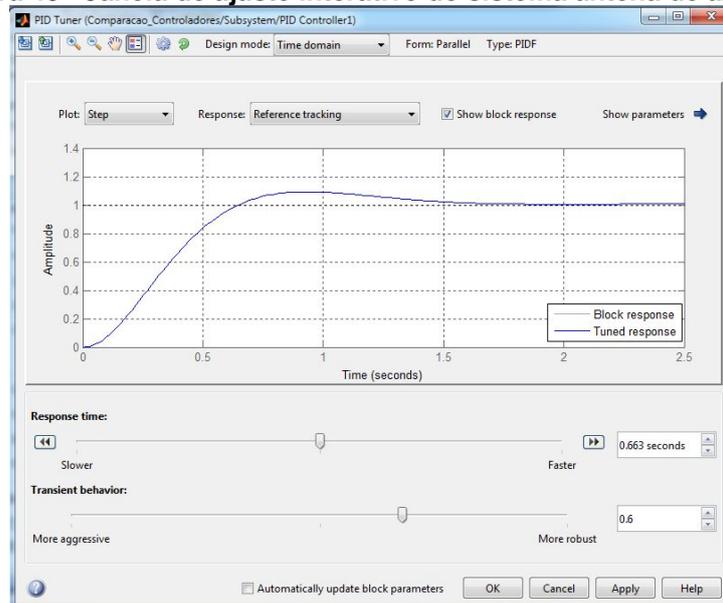
tempo de resposta de 0,663 segundos e a robustez em 0,6. Estes parâmetros devolvem os ganhos $P=3,63094301780316$, $I=0.0948546598556421$, $D=1.33313872570747$ e $N=6,40025071566116$. As figuras 42 e 43 apresentam as janelas de configuração do controlador. A resposta transitória da antena sob ação do controlador PID pode ser visto na figura 44. Na tabela 8 estão os valores registrados pelos pontos do gráfico da figura 44. Pela tabela 8 observa-se que houve uma redução no tempo de acomodação do sistema para 1,29 segundos. Uma diferença de aproximadamente 2,6 segundos em relação ao sistema sem controlador. No entanto, houve um aumento na máxima ultrapassagem percentual de 2,4% em relação ao sistema natural.

Figura 42 - Janela de configuração do controlador PID para o sistema antena de azimute



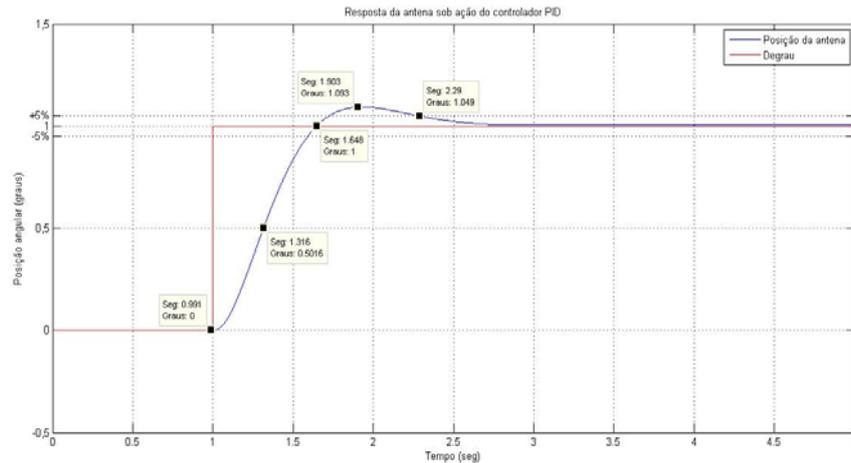
Nota: Elaboração própria.

Figura 43 - Janela de ajuste interativo do sistema antena de azimute



Nota: Elaboração própria.

Figura 44 - resposta transitória da antena sob ação do controlador PID



Nota: elaboração própria.

Tabela 8 - Parâmetros de desempenho transitório da antena

| | |
|--|-------|
| <i>Tempo de atraso (s)</i> | 0.316 |
| <i>Tempo de subida (s)</i> | 0.648 |
| <i>Tempo de pico (s)</i> | 0.903 |
| <i>Tempo de acomodação (s)</i> | 1.29 |
| <i>Máxima ultrapassagem percentual (%)</i> | 9.3 |

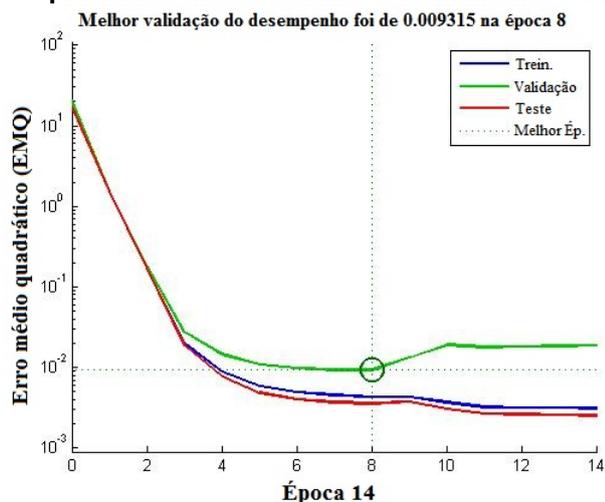
Nota: elaboração própria.

4.4.2 Controle por RNA substituindo PID

Nesta sessão faz-se o controle do sistema antena de azimute pela abordagem de identificação da planta mencionado na sessão 1.3.1. Neste caso, a rede neural irá identificar o modelo de planta de um controlador PID sintonizado para controlar a planta da antena. Para isso, é necessário treinar a rede neural MLP *feed-forward* com um conjunto de dados de entrada e saída provenientes do controlador PID. Em seguida substitui-se o controlador PID pelo rede treinada.

Faz-se o treinamento de uma rede MLP *feed-forward* por meio do aplicativo do matlab “Neural Network Fitting”, descrito no apêndice A. Após o treinamento avalia-se a precisão da rede pelo decaimento do erro médio quadrático, mostrado na figura 45. Mediante a figura 45 é possível afirmar que o treinamento da rede prosseguiu até a época 14. Entretanto o melhor desempenho da rede foi na época oito, onde a validação da rede atingiu um erro entre 10^{-2} e 10^{-3} . Isso significa que, na época oito a rede apresentou a melhor generalização. Da época oito em diante o erro quadrático médio começa a aumentar.

Figura 45 - Decaimento do erro quadrático médio durante o treinamento da rede MLP *feed-forward*.

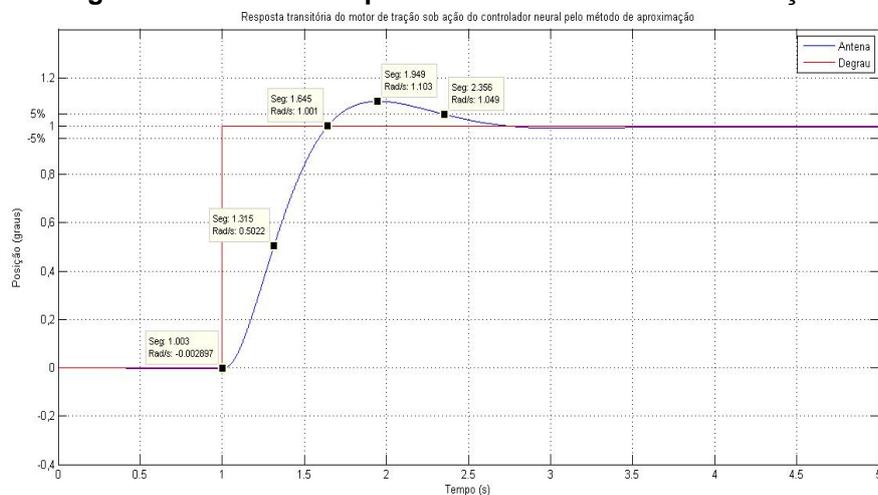


Nota: Elaboração própria.

Finalizado o treinamento, o controlador pode ser substituído pela rede neural e uma nova simulação é gerada. É importante observar que na entrada da rede deve haver a mesma quantidade de variáveis com que a rede foi treinada, conforme é explicado no apêndice A.

Feita a simulação, obtém-se uma curva de resposta transitória da antena sob ação do controlador neural pelo método de identificação mostrado na figura 46. A tabela 9 apresenta os valores dos pontos registrados no gráfico da figura 46. Observa-se pelos dados da tabela 9 que o sistema está controlado com um sobressinal de 10.3% e com um tempo de pico de 949 ms. O tempo de atraso e de subida são curtos (315 ms e 645 ms), o que demonstra que o sistema responde rapidamente. O tempo de acomodação é de 1,356 segundos, uma acomodação rápida também.

Figura 46 - curva de resposta transitória da antena sob ação



Nota: elaboração própria.

Tabela 9 - Parâmetros de desempenho transitório da antena sob ação do controlador neural pelo método de aproximação

| | |
|--|-------|
| <i>Tempo de atraso (s)</i> | 0.315 |
| <i>Tempo de subida (s)</i> | 0.645 |
| <i>Tempo de pico (s)</i> | 0.949 |
| <i>Tempo de acomodação (s)</i> | 1.356 |
| <i>Máxima ultrapassagem percentual (%)</i> | 10.3 |

Nota: elaboração própria.

4.4.3 Rede Neural de controle NARMA-L2

Faz-se nesta sessão uma simulação do controle do sistema antena de azimute pela abordagem de controle NARMA-L2. Nesse controlador neural, duas redes neurais dispostas paralelamente são treinadas simultaneamente com o mesmo conjunto de dados para representar o modelo de planta do sistema antena de azimute.

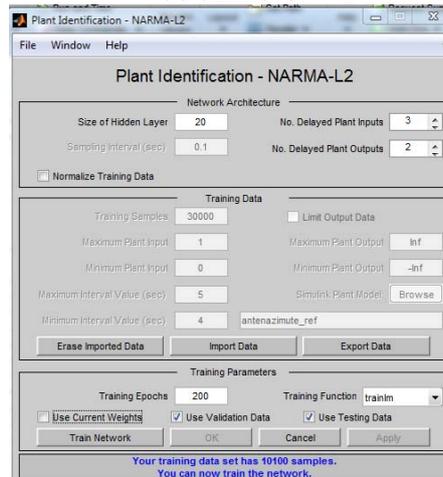
Primeiro é construído o diagrama de blocos do sistema de controle contendo a planta da antena de azimute em série com o controlador neural, conforme a figura 33 da sessão 3.2. Em seguida configura-se a rede neural NARMA-L2 com os parâmetros listados a seguir e executa-se o treinamento conforme o procedimento do apêndice B:

- Arquitetura da rede
 - 20 neurônios na camada oculta;
 - Intervalo de amostragem de 0.1 segundos;
 - Tres passos de atraso no sinal de entrada da planta (z^{-3});
 - Dois passos de atraso no sinal de saída da planta(z^{-2});
 - Desativar a caixa de seleção “Normalizar dados de treinamento”
- Dados de treinamento
 - Número de amostras de treinamento igual a 30000;
 - Valor máximo do sinal de entrada da planta igual a um;
 - Valor mínimo do sinal de entrada da planta igual a 0;
 - Intervalo máximo de constância do sinal gerado de cinco segundos;
 - Intervalo mínimo de constância do sinal gerado de quatro segundos;
 - Desativar a caixa de seleção “Limit output data”
 - Valor máximo do sinal de saída da planta igual a infinito;
 - Valor mínimo do sinal de saída da planta igual a -infinito;
- Parâmetros de treinamento
 - Épocas de treinamento (iterações) do controlador igual a 200;

- Algoritmo de treinamento (função de treinamento) de Levenberg-Marquardt (“*trainlm*”)⁶;
- Desativar caixas de seleção “*Usar os pesos atuais*”;
- Ativar caixas de seleção “*Usar dados de validação*” e “*Usar dados de teste*”;

A figura 47 apresenta a tela de configuração da rede NARMA-L2 com os parâmetros mencionados anteriormente para o sistema antena de azimute.

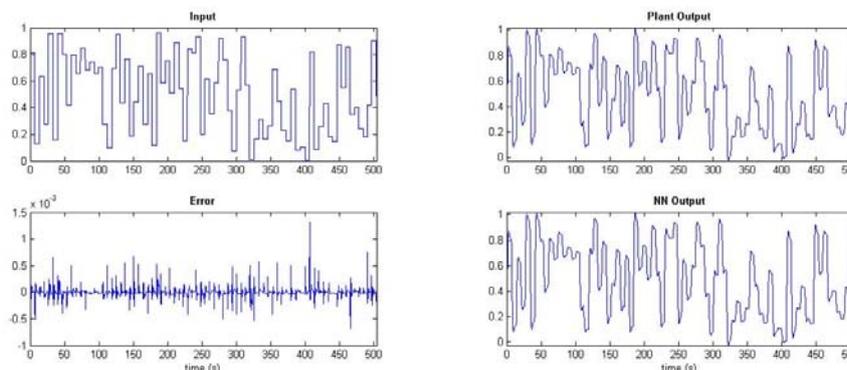
Figura 47 - tela de configuração da rede NARMA-L2 com os parâmetros de simulação do sistema antena de azimute.



Nota: Elaboração própria.

Ao término do treinamento verifica-se o desempenho da rede pelo gráfico do treinamento da rede (figura 48) e pelo decaimento de erro médio quadrático (figura 49). Observa-se pela figura 48 que o sinal de saída da rede acompanha o sinal de saída da planta, o que indica que a rede a rede generalizou bem a função que representa a antena e portanto identificou bem o sistema sem controle. O gráfico de erro da figura 49, dado pela diferença entre a saída da planta e da rede, comprova essa precisão mostrando que a rede atingiu uma precisão de 10^{-3} .

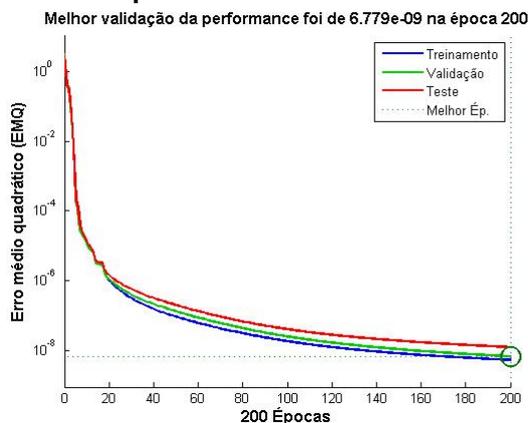
Figura 48 - Gráfico do treinamento da rede NARMA-L2.



Nota: Elaboração própria.

⁶ *trainlm* é um algoritmo de treinamento em rede neurais que atualiza os valores de peso e bias de acordo com a otimização de Levenberg-Marquardt.

Figura 49 - Decaimento do erro quadrático médio durante o treinamento da rede

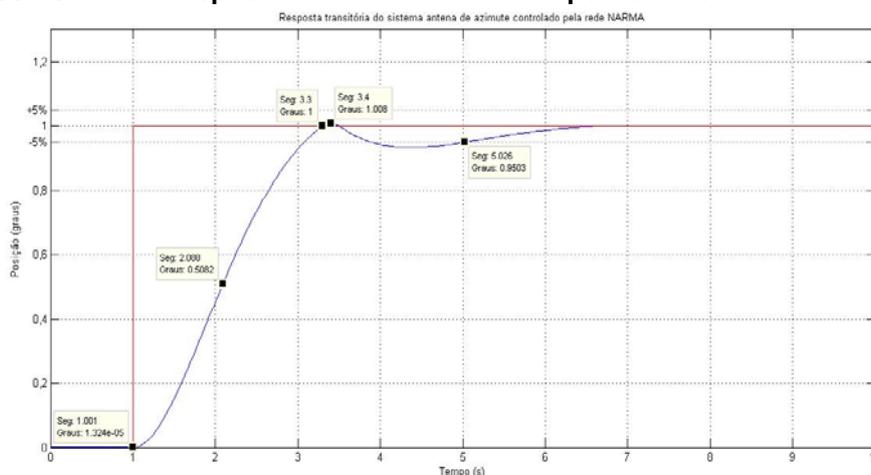


Nota: Elaboração própria.

Pode-se observar pela figura 49 que o melhor desempenho da rede foi na época 200, com um erro de 10^{-8} . É importante notar que a melhor validação coincidiu com a última iteração do processo de treinamento. Isso indica que, caso o sistema fosse submetido a um treinamento com mais iterações, provavelmente o erro iria decair a um valor menor que 10^{-8} .

Por fim, faz-se a simulação do sistema antena de azimute sob ação do controlador neural NARMA-L2 o qual fornece uma curva de resposta (figura 50). A tabela 10 apresenta os índices de desempenho transitório da resposta controlada. Pela tabela 10 observa-se que o sistema está com um sobressinal de 0.8%, e com um tempo de pico igual a 2,4 segundos. O tempo de atraso e de subida são de 1.088 e 2.3 segundos, enquanto que o de acomodação é o de 4,026 segundos, o que demonstra que o sistema apresenta uma resposta mais lenta.

Figura 50 - Curva de resposta da antena controlada pelo controlador neural NARMA-L2



Nota: elaboração própria.

Tabela 10 - Parâmetros de desempenho transitório da antena controlada pelo controlador neural NARMA-L2

| | |
|--|-------|
| <i>Tempo de atraso (s)</i> | 1.088 |
| <i>Tempo de subida (s)</i> | 2.3 |
| <i>Tempo de pico (s)</i> | 2.4 |
| <i>Tempo de acomodação (s)</i> | 4.026 |
| <i>Máxima ultrapassagem percentual (%)</i> | 0.8 |

Nota: elaboração própria.

4.4.4 Controle de modelo de referência

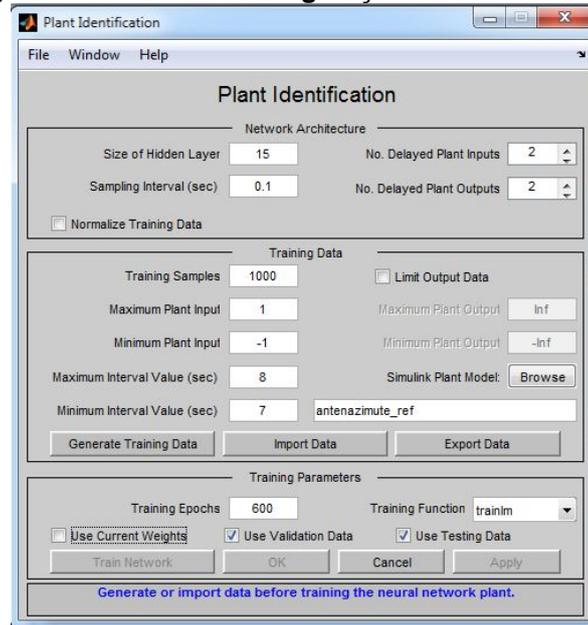
Nesta sessão é apresentada a simulação do controle do sistema antena de azimute pelo estratégia “Modelo de Referência”. Conforme mencionado em 2.2.2.2, nesse controlador neural duas redes neurais são treinadas para controlar uma planta: uma rede controladora e outra de modelo.

Primeiro treina-se a RNA de modelo de maneira independente utilizando um histórico de dados da planta, conforme descrito no apêndice C. Depois treina-se a rede controladora para forçar a saída da planta a seguir a saída do modelo de referência. Para a configuração da rede de modelo são atribuídos os seguintes parâmetros (figura 51):

- Arquitetura da rede identificadora
 - 15 neurônios na camada oculta;
 - Intervalo de amostragem de 0.1 segundos;
 - Dois passos de atraso no sinal de entrada da planta (z^{-2});
 - Dois passos de atraso no sinal de saída da planta (z^{-2});
 - Desabilitar caixa de seleção “*Normalize Training data*”
- Dados de treinamento
 - Número de amostras de treinamento igual a 1000.
 - Valor máximo do sinal de entrada da planta igual a um;
 - Valor mínimo do sinal de entrada da planta igual a -1;
 - Intervalo máximo de constância do sinal gerado de oito segundos;
 - Intervalo mínimo de constância do sinal gerado de sete segundos;
 - Desativar o limite de saída do sinal (caixa de seleção “*Limit output data*”)
- Parâmetros de treinamento
 - Épocas de treinamento (iterações) do controlador igual a 600;
 - Algoritmo de treinamento de Levenberg-Marquardt (“*trainlm*”)
 - Desabilitar caixa de seleção “*Use current weights*”;

- Habilitar caixas de seleção “Use validation data” e “Use testing data”;

Figura 51 - Janela de configuração da rede de modelo



Nota: Elaboração própria.

E para a configuração da rede controladora, são atribuídos os seguintes parâmetros (figura 52):

- Arquitetura da rede controladora⁷
 - 7 neurônios na camada oculta;
 - Dois passos de atraso no sinal de referência (z^{-2});⁸
 - Um passo de atraso no sinal de saída (z^{-1});
 - Dois passos de atraso no sinal de saída da planta (z^{-2});
- Dados de treinamento
 - Valor máximo do sinal de referência randômico gerado igual a um;
 - Valor mínimo do sinal de referência randômico gerado igual a -1;
 - Intervalo máximo de constância do sinal randômico gerado de 4 segundos;
 - Intervalo mínimo de constância do sinal randômico gerado de 3 segundos;
 - Número de amostras de treinamento igual a 6000.
- Parâmetros de treinamento
 - Épocas de treinamento (iterações) do controlador igual a 20.
 - Segmentos de treinamento do controlador igual a 4⁹;

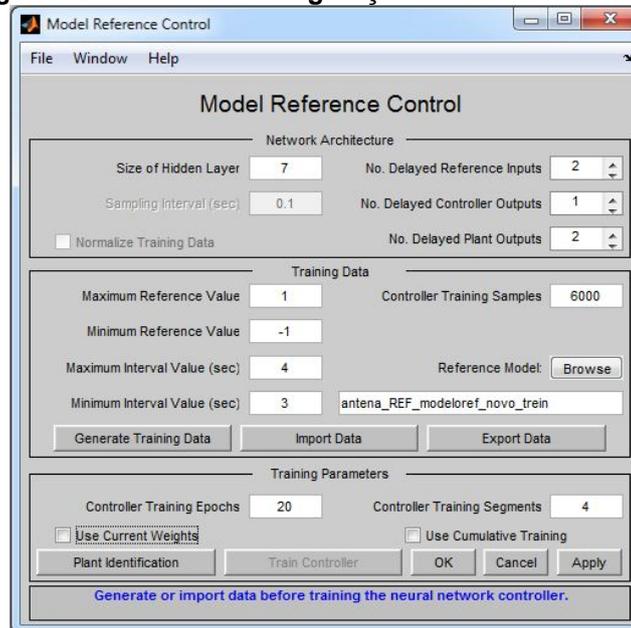
⁷ Na tela de configuração da rede controladora não é possível alterar o intervalo de amostragem porque é o mesmo valor atribuído a rede de modelo.

⁸ O aumento do número de atrasos no sinal pode ser necessário a medida que o sistema se torna mais complexo.

⁹ Defini em quantos segmentos os dados de treinamento serão divididos.

- Desabilitar caixa de seleção “Use current weights”;
- Desabilitar caixa de seleção “Use cumulative training”¹⁰;

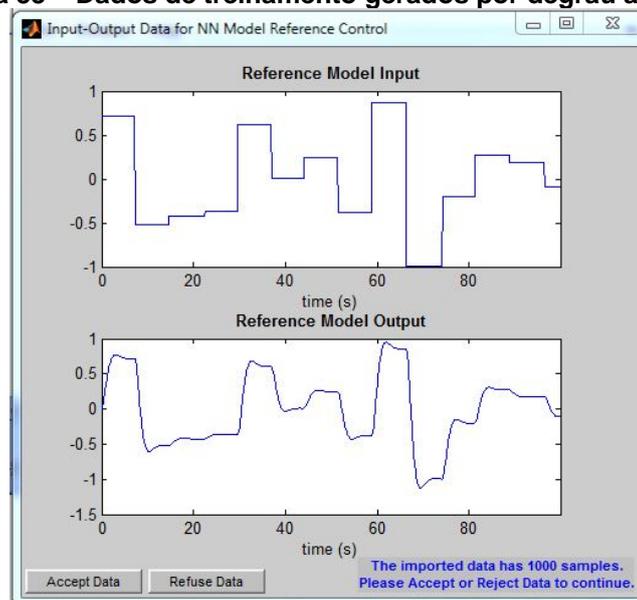
Figura 52 - Janela de configuração da rede controladora



Nota: Elaboração própria.

Faz-se agora o treinamento dessas redes. Primeiro gera-se um conjunto de dados para o treinamento da rede de modelo (figura 53). Realiza-se então o treinamento. Após realizar o treino dessa rede, verifica-se a precisão da rede pelo decaimento do erro médio quadrático (EMQ), que é mostrado na figura 54.

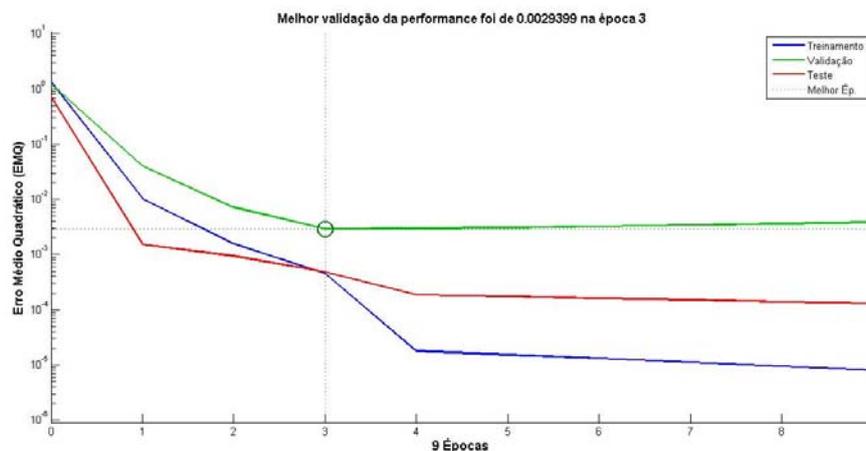
Figura 53 – Dados de treinamento gerados por degrau aleatório



¹⁰ Se esta caixa for selecionada, os segmentos de dados são adicionados ao conjunto de treinamento conforme o treino continua. Do contrário, somente um segmento por vez é utilizado.

Nota: Elaboração própria.

Figura 54 – Desempenho do treinamento da rede de modelo



Nota: Elaboração própria.

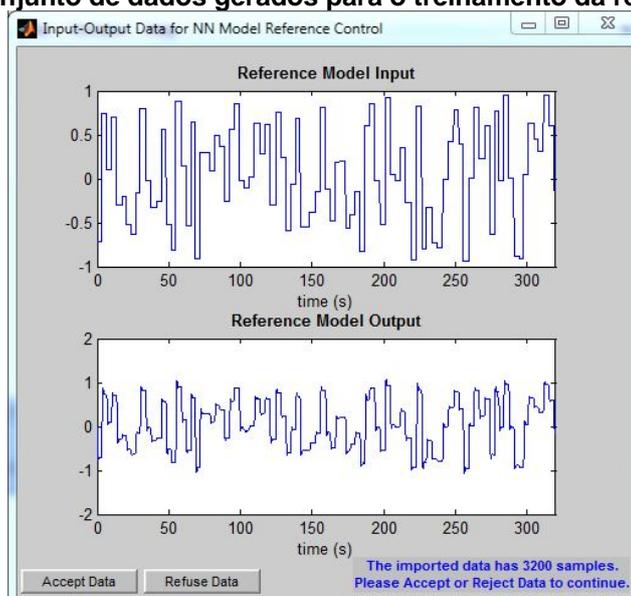
Observa-se pela figura 54 que até a época 9 o EMQ atinge a escala de 10^{-5} . Contudo, a rede identificou o melhor desempenho na época três. O treino é interrompido na época três, visto que a partir desta época o erro do conjunto de validação começa a aumentar. Na época três o EMQ situa-se entre 10^{-2} e 10^{-3} .

Em seguida, faz-se o treinamento da rede controladora pelo mesmo procedimento realizado pela rede de modelo: geração de dados e em seguida o treino da rede.

Um conjunto de 3200 amostras é gerado para o treinamento da rede controladora (figura 55). À medida que o treino é realizado ocorre um decaimento do EMQ desta rede (figura 56), o qual atingi a escala de 10^{-2} após 20 épocas de treinamento realizados sobre quatro segmentos¹¹. Finalizado o treino, avalia-se a atuação da rede neural (linha verde) comparado com o modelo de sistema controlado fornecido pelo usuário (linha azul) conforme visto na figura 57. É possível observar pela figura 57 que para a maioria dos pulsos gerados o sinal de atuação da rede neural consegue acompanhar o sinal do modelo fornecido pelo usuário, no entanto o sinal da rede apresenta um sobressinal para alguns valores de pulso (como nos pulsos de nível alto para baixo).

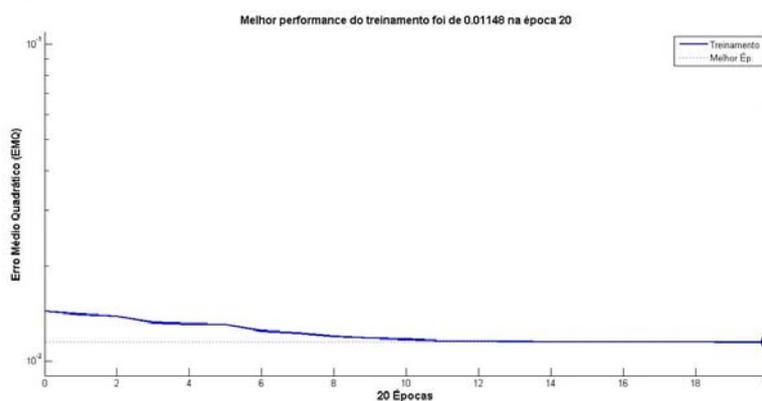
¹¹ A rede controladora foi configurado com quatro segmentos. Isto faz com que o conjunto total de dados seja segmentado em quatro partes, e então é realizado um treinamento de 20 épocas em cada segmento. Nesse caso, o total de épocas é 80. O gráfico da figura 61 refere-se ao decaimento ocorrido no último segmento.

Figura 55 – Conjunto de dados gerados para o treinamento da rede controladora



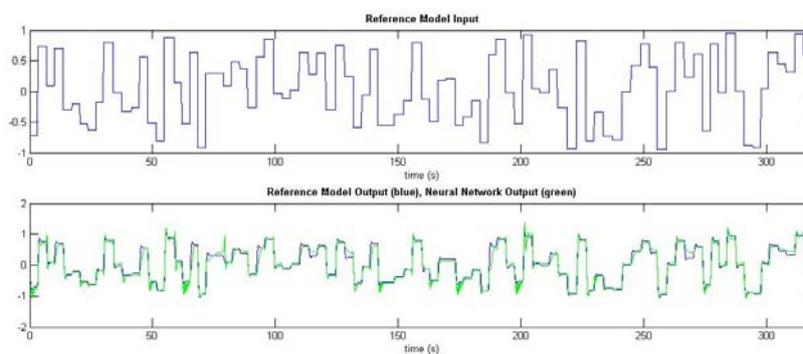
Nota: Elaboração própria.

Figura 56 – Desempenho do treinamento da rede controladora



Nota: Elaboração própria.

Figura 57 – Atuação da rede neural comparada a um modelo de sistema controlado

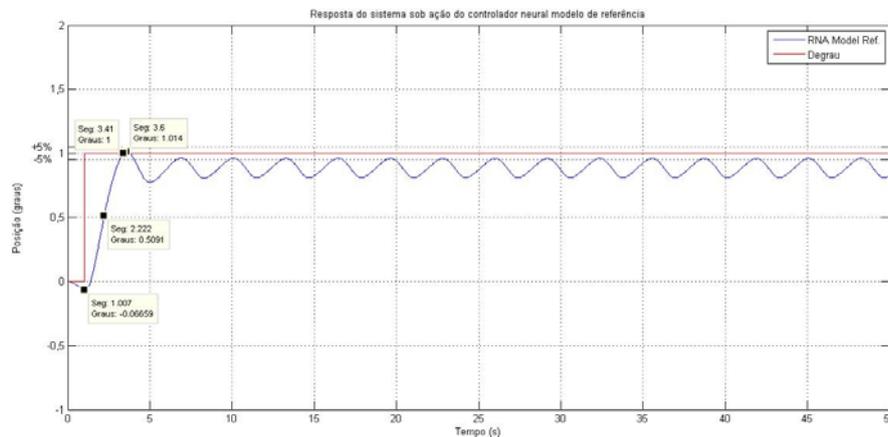


Nota: Elaboração própria.

Na figura 58 é apresentado o comportamento da resposta da antena sob ação de controle da rede neural modelo de referência, e na tabela 11 estão os valores dos índices de

desempenho transitório. Observa-se pela tabela 11 que o sistema responde com um tempo de atraso de 1,2 segundos, um tempo de subida de 2,41 segundos e um tempo de pico de 2,6 segundos. Neste caso não houve acomodação, pois o sistema apresenta uma oscilação constante próxima de seu valor final. Este tipo de resposta é indesejável para sistemas de controle uma vez que estas oscilações podem causar desgastes no sistema, levando-o a fadiga e consequentemente danificá-lo. Nesse caso, o sistema não está controlado, uma vez que a resposta do sistema para a entrada em degrau apresenta esta oscilação constante próximo do valor unitário.

Figura 58 – Resposta da antena sob ação do controlador neural modelo de referência



Nota: elaboração própria.

Tabela 11 - Parâmetros de desempenho transitório

| | |
|--|-------|
| <i>Tempo de atraso (s)</i> | 1.222 |
| <i>Tempo de subida (s)</i> | 2.41 |
| <i>Tempo de pico (s)</i> | 2.6 |
| <i>Tempo de acomodação (s)</i> | - |
| <i>Máxima ultrapassagem percentual (%)</i> | 1.4 |

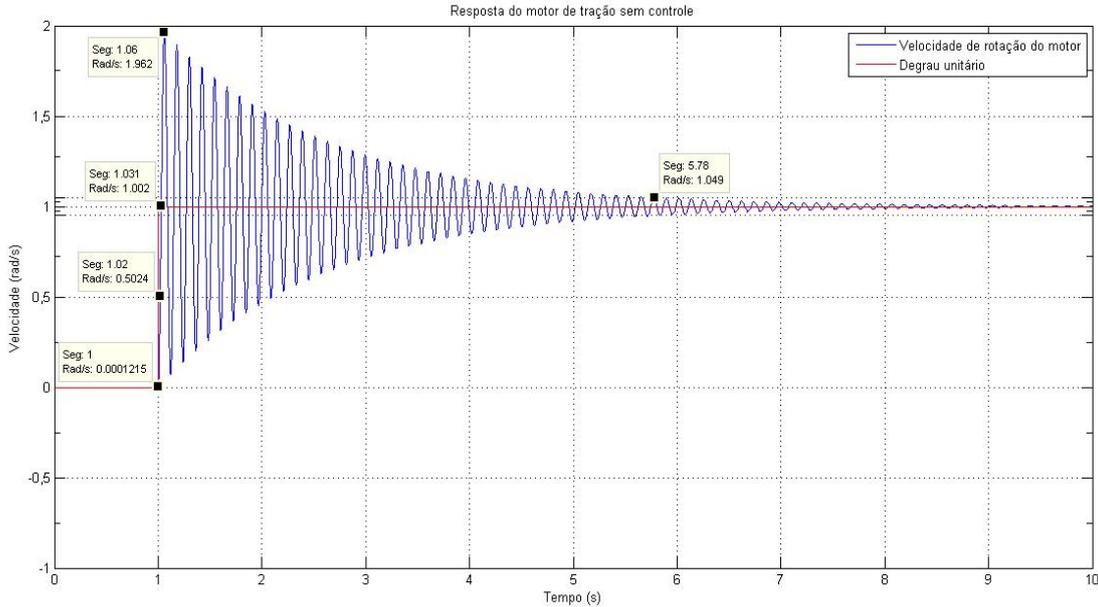
Nota: elaboração própria.

4.5 Motor elétrico de tração de veículo de ferrovia

Utiliza-se do diagrama de blocos da figura 37 da sessão 3.3 para simular o comportamento dinâmico do sistema não controlado “motor de tração do veículo de ferrovia” no ambiente gráfico Simulink do matlab. Também é utilizada a entrada em degrau unitário para obter os índices de desempenho transitório, que são mostrados na tabela 12. Com base nos dados da tabela 12, pode-se observar que o sistema apresenta um sobressinal muito elevado (96,2%), e pela figura 59 observa-se que a resposta dinâmica deste sistema é altamente oscilatória. Conforme Dorf e Bishop (2013) isto já era esperado, uma vez que este sistema apresenta um baixo amortecimento ($\zeta = 0,012$). Pode-se melhorar o desempenho

deste sistema aplicando-se um controlador que reduza o sobressinal e o tempo de acomodação.

Figura 59 – comportamento dinâmico do motor CC de veículo de ferrovia



Nota: elaboração própria.

Tabela 12 - Parâmetros de desempenho transitório

| | |
|--|-------|
| <i>Tempo de atraso (s)</i> | 0.02 |
| <i>Tempo de subida (s)</i> | 0.031 |
| <i>Tempo de pico (s)</i> | 0.06 |
| <i>Tempo de acomodação (s)</i> | 4.78 |
| <i>Máxima ultrapassagem percentual (%)</i> | 96.2 |

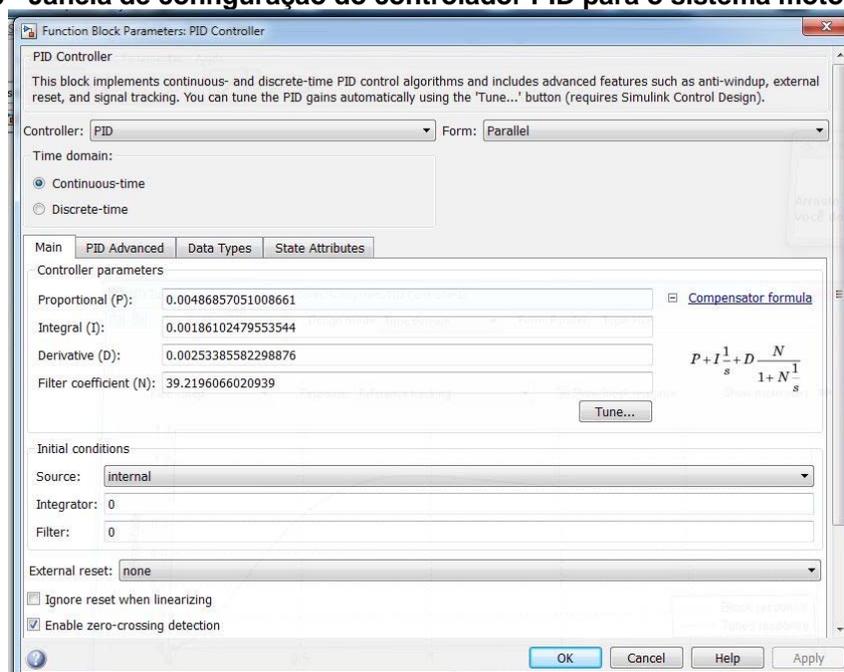
Nota: elaboração própria.

4.5.1 Controle por PID sintonizado

Ao aplicar-se um controlador PID no sistema motor de tração é necessário sintonizar os ganhos. Faz o procedimento descrito no apêndice D. Os ganhos da figura 60 são obtidos configurando o tempo de resposta para 0.278 segundos e o comportamento do transiente em 0.75 (figura 61). Assim, os ganhos obtidos pela sintonização são:

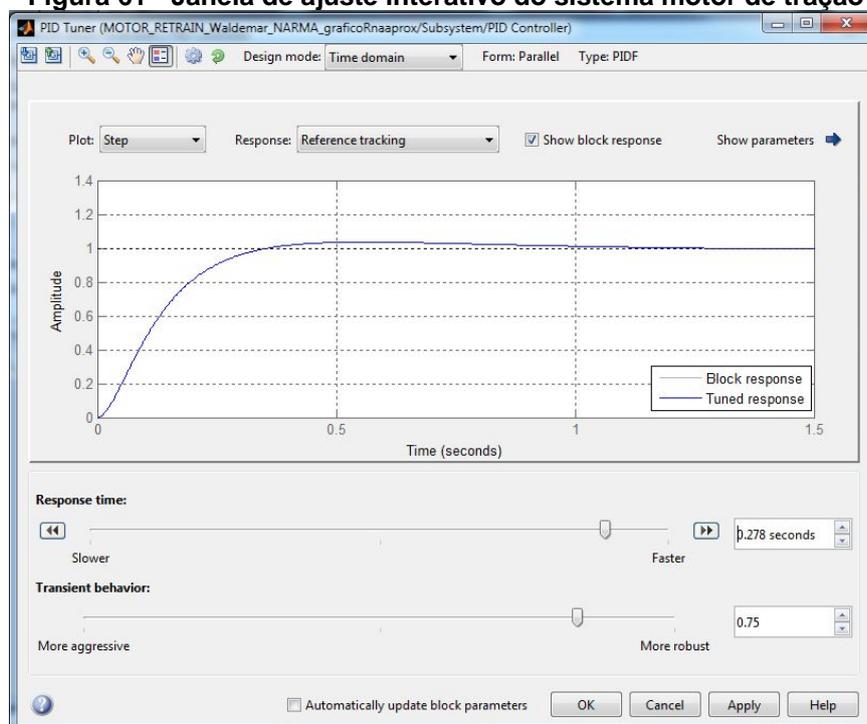
- P = 0.0048686
- I = 0.0018610
- D = 0.0025339
- N = 39.2196

Figura 60 - Janela de configuração do controlador PID para o sistema motor de tração



Nota: Elaboração própria.

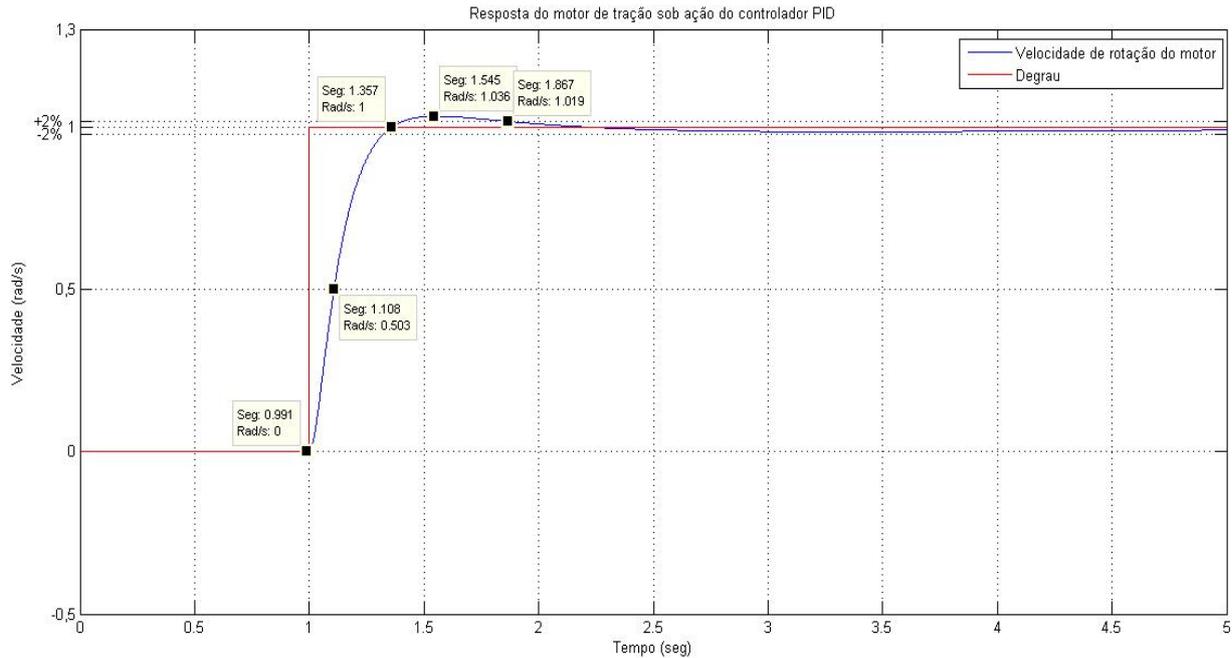
Figura 61 - Janela de ajuste interativo do sistema motor de tração



Nota: Elaboração própria.

Finalizada a sintonia do PID, uma curva de resposta é gerado após a aplicação do controlador PID, conforme mostrado na figura 62. Os parâmetros de desempenho transitório estão mostrados na tabela 13.

Figura 62 – resposta do motor de tração sob ação do controlador PID



Nota: elaboração própria.

Tabela 13 - parâmetros de desempenho transitório do motor de tração sob ação do controlador PID

| | |
|--|-------|
| <i>Tempo de atraso (s)</i> | 0.108 |
| <i>Tempo de subida (s)</i> | 0.357 |
| <i>Tempo de pico (s)</i> | 0.545 |
| <i>Tempo de acomodação (s)</i> | 0.867 |
| <i>Máxima ultrapassagem percentual (%)</i> | 3.6 |

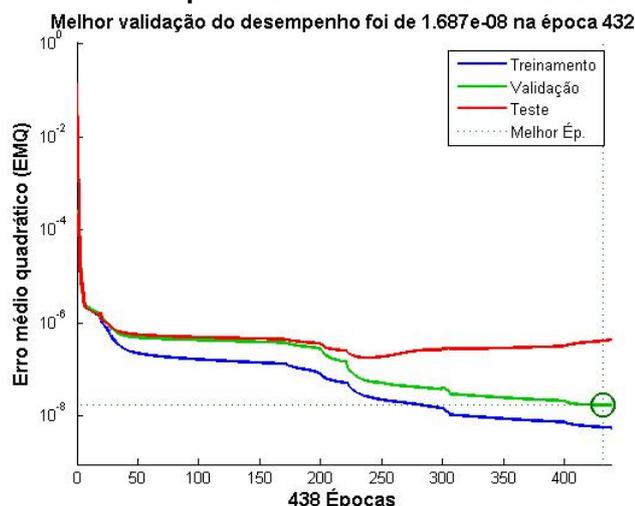
Nota: elaboração própria.

Observa-se pela tabela 13 que o sistema está controlado com um sobressinal muito pequeno de 3.6%, com um tempo de pico de 545ms. O tempo de atraso e de subida são curtos (108ms e 357ms), e o tempo de acomodação é de 867ms, o que demonstra que o sistema responde muito rapidamente.

4.5.2 Controle por RNA substituindo PID

Agora faz-se o controle do sistema motor de tração pela abordagem de identificação da planta mencionado na sessão 1.3.1. Primeiro faz-se o treinamento de uma rede MLP *feed-forward* por meio do aplicativo do matlab “Neural Network Fitting”, descrito no apêndice A. Após o treinamento avalia-se a precisão da rede pelo decaimento do erro médio quadrático, mostrado na figura 63. Observa-se pela figura 63 que o melhor desempenho da rede foi na época 432, alcançando um erro de 10^{-8} . Da época 432 em diante o erro quadrático médio começa a aumentar.

Figura 63 - Decaimento do erro quadrático médio durante o treinamento da rede



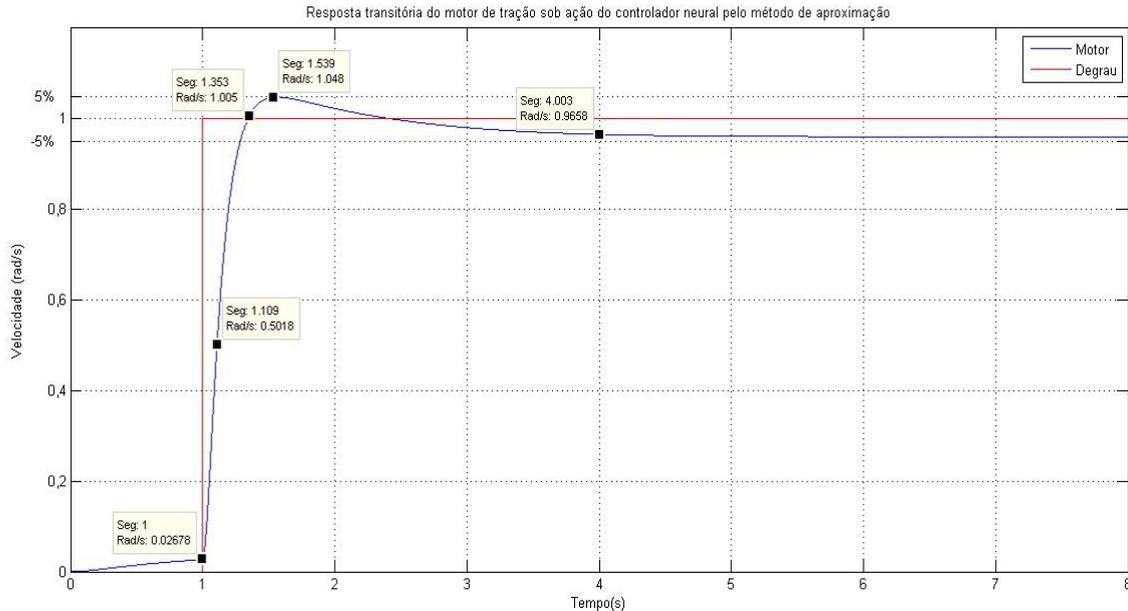
Nota: Elaboração própria.

Finalizado o treinamento, o controlador pode ser substituído pela rede neural e uma nova simulação é gerada. Contudo, deve-se inserir na entrada da rede as mesmas variáveis utilizadas no treinamento (erro e derivada do erro).

Faz-se uma simulação do sistema motor de tração controlado pela método de aproximação. A curva de resposta do sistema controlado pela rede é mostrada na figura 64, e os parâmetros de desempenho transitório da curva de resposta são mostrados na tabela 14. Pela tabela 14, observa-se que o sistema está com um sobressinal muito pequeno de 4.8% e com um tempo de pico de 539ms. O tempo de atraso e de subida são (109ms e 353ms), contudo a acomodação é mais lenta (três segundos), o que demonstra que o sistema responde rápido porém acomoda na faixa de tolerância de forma mais lenta.

Por fim, observa-se que a resposta do sistema apresenta um erro estacionário. Entretanto, este erro está situado na faixa de tolerância dada em 4.3. Logo, o controlador é eficaz.

Figura 64 – resposta do motor de tração sob ação do controlador neural pelo método de aproximação



Nota: elaboração própria.

Tabela 14 – parâmetros de desempenho transitório do motor de tração sob ação do controlador neural pelo método de aproximação

| | |
|--|-------|
| <i>Tempo de atraso (s)</i> | 0.109 |
| <i>Tempo de subida (s)</i> | 0.353 |
| <i>Tempo de pico (s)</i> | 0.539 |
| <i>Tempo de acomodação (s)</i> | 3.003 |
| <i>Máxima ultrapassagem percentual (%)</i> | 4.8 |

Nota: elaboração própria.

4.5.3 Rede Neural de controle NARMA-L2

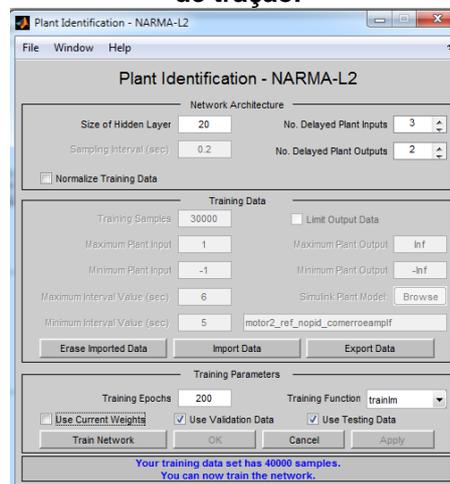
Agora faz-se nesta sessão a simulação do controle do sistema motor de tração pela estratégia de controle NARMA-L2. Primeiro, estrutura-se o sistema conforme o diagrama de blocos da figura 37 da sessão 3.3, com a FT do motor em série com o controlador neural NARMA-L2. Em seguida configura-se a rede neural NARMA-L2 com os parâmetros listados a seguir e executa-se o treinamento conforme o procedimento do apêndice B:

- Arquitetura da rede
 - 20 neurônios na camada oculta;
 - Intervalo de amostragem de 0.2 segundos;
 - Tres passos de atraso no sinal de entrada da planta (z^{-3});
 - Dois passos de atraso no sinal de saída da planta (z^{-2});
 - Desativar a caixa de seleção “Normalizar dados de treinamento”
- Dados de treinamento

- Número de amostras de treinamento igual a 30000;
- Valor máximo do sinal de entrada da planta igual a um;
- Valor mínimo do sinal de entrada da planta igual a -1;
- Intervalo máximo de constância do sinal gerado de seis segundos;
- Intervalo mínimo de constância do sinal gerado de cinco segundos;
- Desativar a caixa de seleção “Limit output data”
- Valor máximo do sinal de saída da planta igual a infinito;
- Valor mínimo do sinal de saída da planta igual a -infinito;
- Parâmetros de treinamento
 - Épocas de treinamento (iterações) do controlador igual a 200;
 - Algoritmo de treinamento (função de treinamento) de Levenberg-Marquardt (“trainlm”);
 - Desativar caixas de seleção “Usar os pesos atuais”;
 - Ativar caixas de seleção “Usar dados de validação” e “Usar dados de teste”;

A figura 65 apresenta a tela de configuração da rede NARMA-L2 com os parâmetros mencionados anteriormente para o sistema motor de tração.

Figura 65 - tela de configuração da rede NARMA-L2 com os parâmetros de simulação do sistema motor de tração.

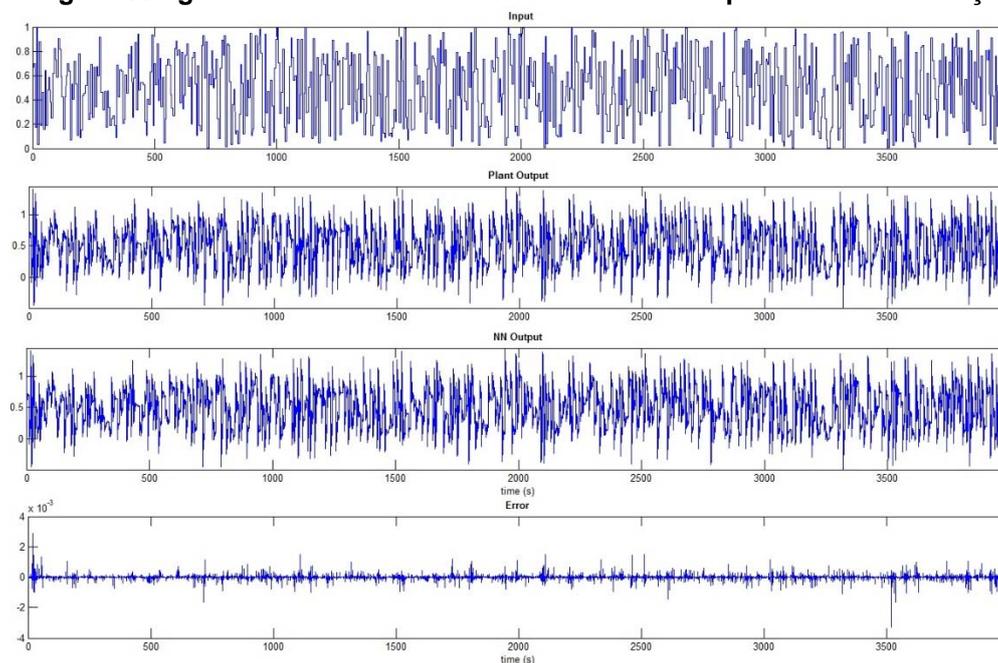


Nota: Elaboração própria.

Finalizado o treinamento verifica-se o desempenho da rede pelo gráfico do treinamento da rede (figura 66) e pelo decaimento de erro médio quadrático (figura 67). Observa-se pela figura 66 que o sinal de saída da rede acompanha o sinal de saída da planta. Isso indica que a rede generalizou bem a função que representa o motor e portanto identificou bem o sistema sem controle. Logo abaixo do gráfico de saída da rede está o erro de precisão, o qual indica o grau de precisão com que a rede aproximou da função do motor,

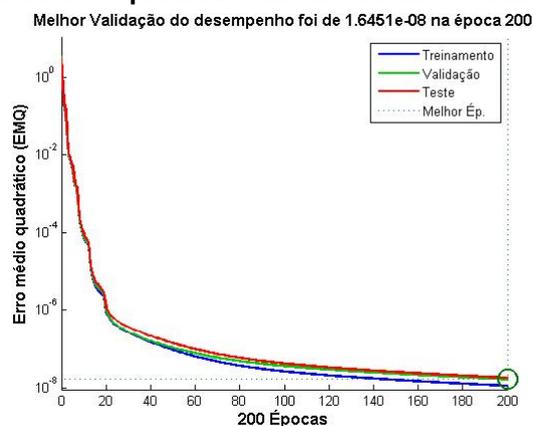
dados pela diferença entre a saída da planta e da rede. A rede alcançou uma escala de 10^{-3} de precisão.

Figura 66 - gráfico do treinamento da rede NARMA-L2 para o motor de tração.



Nota: elaboração própria.

Figura 67 - Decaimento do erro quadrático médio durante o treinamento da rede



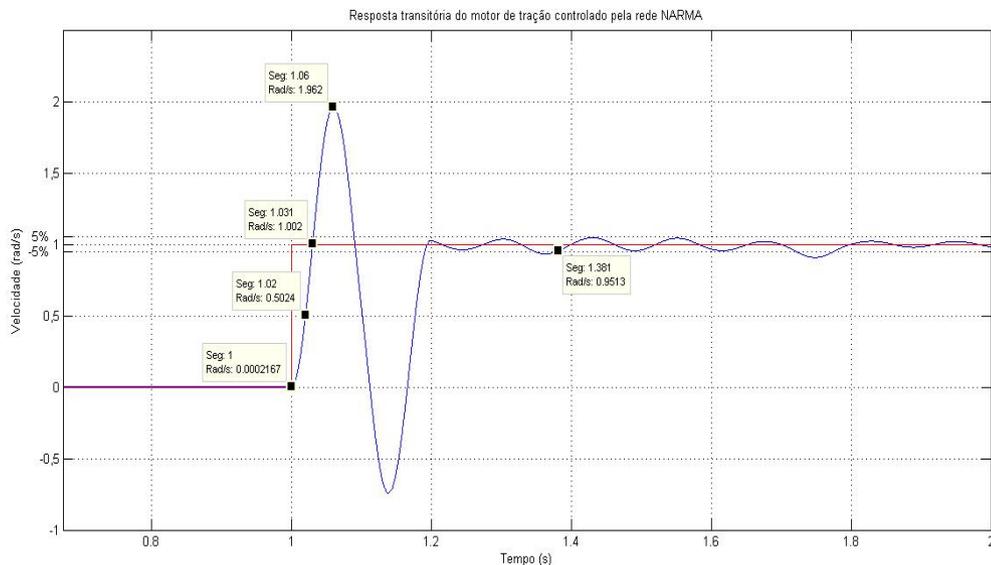
Nota: Elaboração própria.

Na figura 67, a melhor época coincidiu com a última época do treinamento e o erro de treinamento alcançou um mínimo de 10^{-8} .

Agora analisa-se a resposta dinâmica do motor com o controlador neural NARMA-L2, que é apresentada na figura 68. Pela figura 68, pode-se observar que o sistema está com uma oscilação no transitório antes de acomodar. Este tipo de transitório não é aceitável, pois possui um sobressinal muito elevado que não atende à especificação de sobressinal dado em 4.3. Logo, o sistema não está controlado.

Os índices de desempenho transitório estão mostrados na tabela 15. Através da tabela 15 nota-se que o sobressinal é alto, 96.2%, com um tempo de pico de 60ms. O tempo de atraso e de subida são curtos (20ms e 31ms), e o tempo de acomodação é 381ms de duração.

Figura 68 – resposta do motor de tração sob ação do controlador neural NARMA-L2



Nota: elaboração própria.

Tabela 15 – parâmetros de desempenho transitório do motor sob ação do controlador neural

| | |
|--|-------|
| <i>Tempo de atraso (s)</i> | 0.02 |
| <i>Tempo de subida (s)</i> | 0.031 |
| <i>Tempo de pico (s)</i> | 0.06 |
| <i>Tempo de acomodação (s)</i> | 0.381 |
| <i>Máxima ultrapassagem percentual (%)</i> | 96.2 |

Nota: elaboração própria.

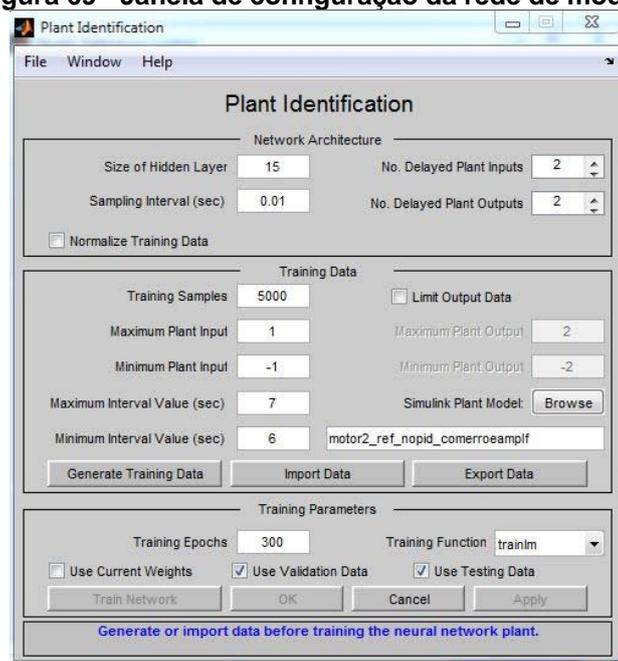
4.5.4 Controle de modelo de referência

Faz-se agora o controle do motor elétrico de tração utilizando o neurocontrolador Modelo de Referência. Primeiro treina-se a RNA de modelo de maneira independente com o histórico de dados da planta do sistema. Para a configuração da rede de modelo são atribuídos os seguintes parâmetros (figura 69):

- Arquitetura da rede identificadora
 - 15 neurônios na camada oculta;
 - Intervalo de amostragem de 0.01 segundos;
 - Dois passos de atraso no sinal de entrada da planta (z^{-2});

- Dois passos de atraso no sinal de saída da planta (z^{-2});
- Dados de treinamento
 - Valor máximo do sinal de entrada da planta igual a um;
 - Valor mínimo do sinal de entrada da planta igual a -1;
 - Intervalo máximo de constância do sinal gerado de sete segundos;
 - Intervalo mínimo de constância do sinal gerado de seis segundos;
 - Número de amostras de treinamento igual a 5000.
 - Desativar o limite de saída do sinal (caixa de seleção "Limit output data")
- Parâmetros de treinamento
 - Épocas de treinamento (iterações) do controlador igual a 300;
 - Algoritmo de treinamento (função de treinamento) de Levenberg-Marquardt ("trainlm")

Figura 69 - Janela de configuração da rede de modelo



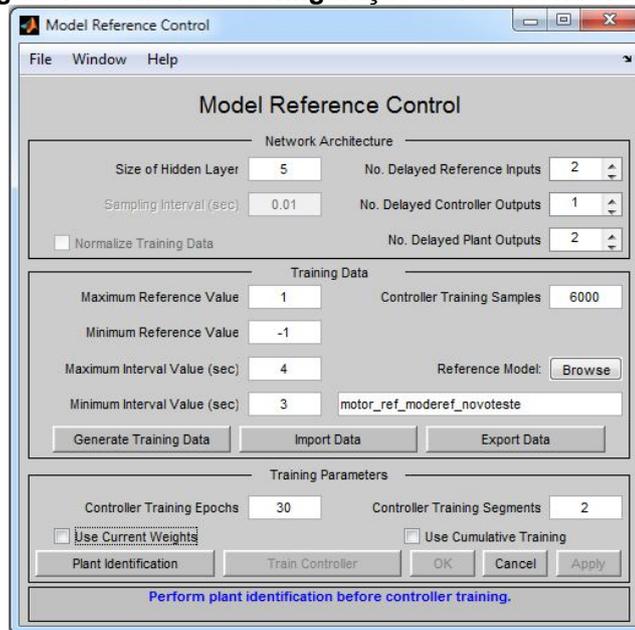
Nota: Elaboração própria.

Em seguida treina-se a rede controladora para forçar a saída da planta a seguir a saída do modelo de referência. Para a configuração da rede controladora, são atribuídos os seguintes parâmetros (figura 70):

- Arquitetura da rede controladora
 - 5 neurônios na camada oculta;
 - Dois passos de atraso no sinal de referência (z^{-2});
 - Um passo de atraso no sinal de saída (z^{-1});

- Dois passos de atraso no sinal de saída da planta (z^{-2});
- Dados de treinamento
 - Valor máximo do sinal de referência randômico gerado igual a um;
 - Valor mínimo do sinal de referência randômico gerado igual a -1;
 - Intervalo máximo de constância do sinal randômico gerado de 4 segundos;
 - Intervalo mínimo de constância do sinal randômico gerado de 3 segundos;
 - Número de amostras de treinamento igual a 6000.
- Parâmetros de treinamento
 - Épocas de treinamento (iterações) do controlador igual a 30.
 - Segmentos de treinamento do controlador igual a 2;

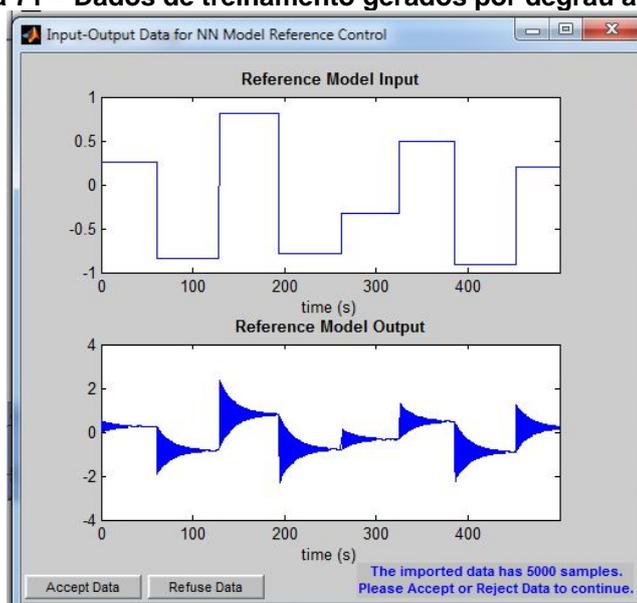
Figura 70 - Janela de configuração da rede controladora



Nota: Elaboração própria.

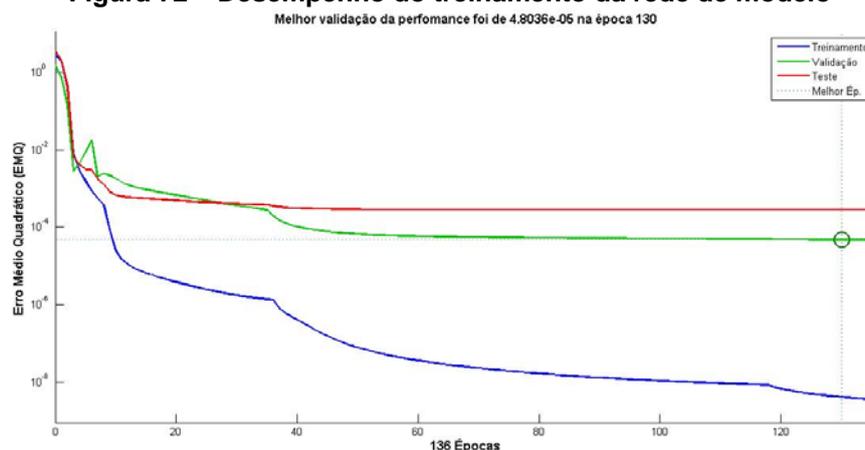
Faz-se o treinamento dessas redes, conforme mencionado no apêndice C. Primeiro gera-se um conjunto de dados para o treinamento da rede de modelo (figura 71). Em seguida, inicia-se o treinamento. Após realizar o treino dessa rede, o índice de desempenho do treinamento, dado pelo decaimento do erro médio quadrático (EMQ), é mostrado na figura 72.

Figura 71 – Dados de treinamento gerados por degrau aleatório



Nota: Elaboração própria.

Figura 72 – Desempenho do treinamento da rede de modelo



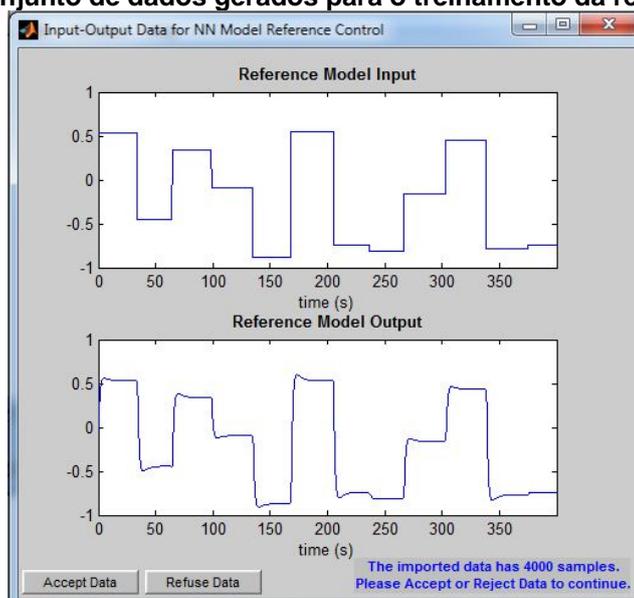
Nota: Elaboração própria.

Pela figura 72, é observado que o treinamento foi interrompido na época 136, onde o EMQ atinge a escala de 10^{-8} .

Em seguida, faz-se o treinamento da rede controladora pelo mesmo procedimento: geração de dados e em seguida o treino da rede.

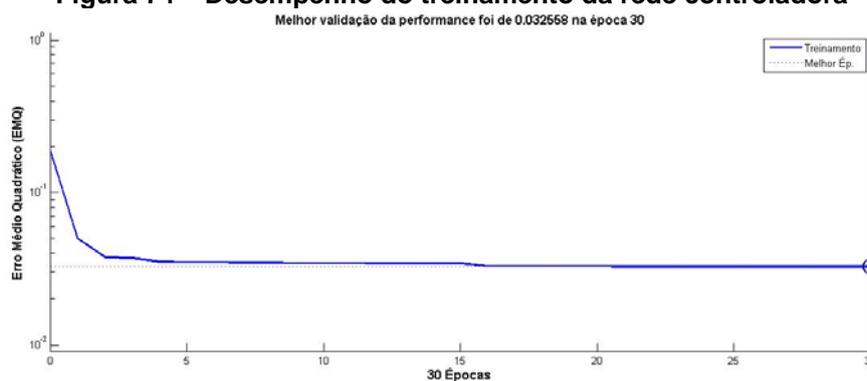
Um conjunto de 4000 amostras é gerado para o treinamento da rede controladora (figura 73). À medida que o treino é realizado ocorre um decaimento do EMQ (figura 74), o qual está situado entre 10^{-1} e 10^{-2} após 30 épocas de treinamento realizados sobre dois segmentos. Finalizado o treino, avalia-se a atuação da rede neural (linha verde) comparado com o modelo de sistema controlado fornecido pelo usuário (linha azul) conforme visto na figura 75.

Figura 73 – Conjunto de dados gerados para o treinamento da rede controladora



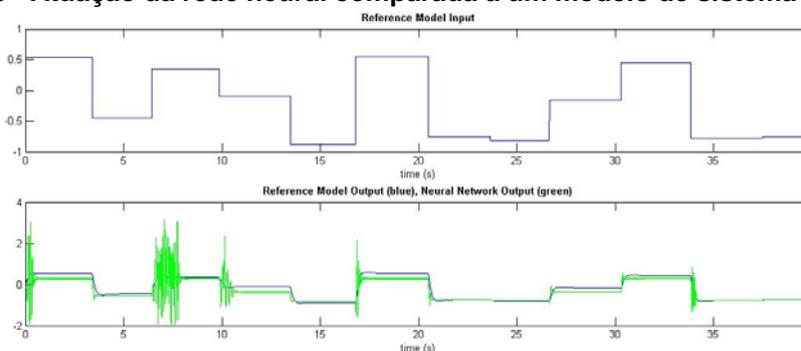
Nota: Elaboração própria.

Figura 74 – Desempenho do treinamento da rede controladora



Nota: Elaboração própria.

Figura 75 - Atuação da rede neural comparada a um modelo de sistema controlado



Nota: Elaboração própria.

Observa-se na figura 75 que a rede neural não consegue acompanhar o sinal do modelo fornecido pelo usuário, apresentando oscilações com elevado sobressinal para alguns valores de pulso. Também é observado a presença de um erro estacionário na saída da rede em

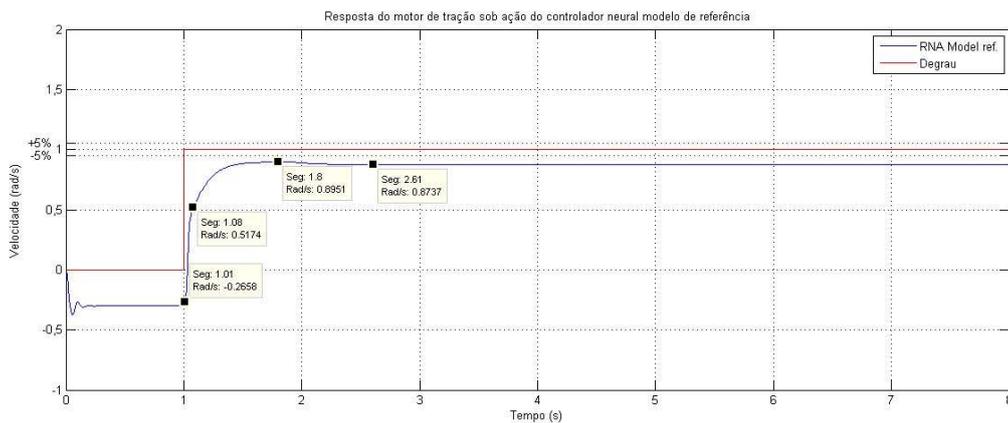
relação aos pulsos. Isso implica em afirmar que o controlador não conseguiu forçar a saída da planta a seguir o modelo de referência fornecido.

Observa-se agora na figura 76 o comportamento da resposta do motor sob ação de controle da rede neural modelo de referência. Neste caso houve uma acomodação da reposta, porém fora do intervalo de tolerância ($\pm 5\%$). O sistema também apresenta um erro estacionário de $-12,6\%$, que se encontra fora da faixa de tolerância dado pelas especificações de projeto, porém não chega a apresentar sobressinal.

Na tabela 16 estão os índices de desempenho transitório da curva de resposta do motor de tração. Nesse caso, a resposta não apresenta sobressinal nem tempo de subida, pois a resposta não alcança o valor final de um.

Em geral, o sistema não está controlado, pois o sistema não atinge seu valor final e estabiliza em um valor fora da faixa de tolerância do erro estacionário ($\pm 5\%$).

Figura 76 – Resposta do motor sob ação do controlador neural modelo de referência



Nota: elaboração própria.

Tabela 16 - Parâmetros de desempenho transitório

| | |
|--|------|
| <i>Tempo de atraso (s)</i> | 0.08 |
| <i>Tempo de subida (s)</i> | - |
| <i>Tempo de pico (s)</i> | 0.8 |
| <i>Tempo de acomodação (s)</i> | 1.61 |
| <i>Máxima ultrapassagem percentual (%)</i> | - |

Nota: elaboração própria.

4.6 Suspensão ativa

Faz-se agora uma simulação do sistema suspensão ativa 1/4 de veículo no Simulink, utilizando-se do diagrama de blocos da figura 39 (sessão 3.4). O sinal de entrada aplicada ao sistema é do tipo degrau unitário. Através da simulação gera-se uma curva de resposta, que está mostrada na figura 77. Os índices de desempenho são mostrados na tabela 17. Com base na tabela 17, observa-se que o sistema possui um elevado sobressinal (91%). Observa-se também pela figura 77 que este sistema é altamente oscilatório assim como o motor de tração. No entanto este sistema apresenta um tempo de acomodação longo de aproximadamente 26 segundos. Isto demonstra que o efeito da suspensão se tornaria desconfortável para os passageiros do veículo. É necessário então aplicar um controlador que reduza o sobressinal e o tempo de acomodação.

Em se tratando dos índices de desempenho transitório, é importante observar que neste caso não é necessário considerar o tempo de atraso e de subida, pois ao projetar um controlador para este sistema, os autores Messner e Tilbury utilizam apenas dois parâmetros, que é a máxima ultrapassagem percentual e o tempo de acomodação. Adota-se também para este trabalho o tempo de pico visto que este parâmetro está relacionado com a máxima ultrapassagem percentual.

Figura 77 – resposta dinâmica do sistema suspensão ativa 1/4.



Nota: elaboração própria.

Tabela 17 - Parâmetros de desempenho transitório do sistema suspensão ativa 1/4 de veículo

| | |
|--------------------------------|-------|
| <i>Tempo de atraso (s)</i> | - |
| <i>Tempo de subida (s)</i> | - |
| <i>Tempo de pico (s)</i> | 0.08 |
| <i>Tempo de acomodação (s)</i> | 25.79 |

| | |
|-------------------------------------|-------|
| Máxima ultrapassagem percentual (%) | 91.48 |
|-------------------------------------|-------|

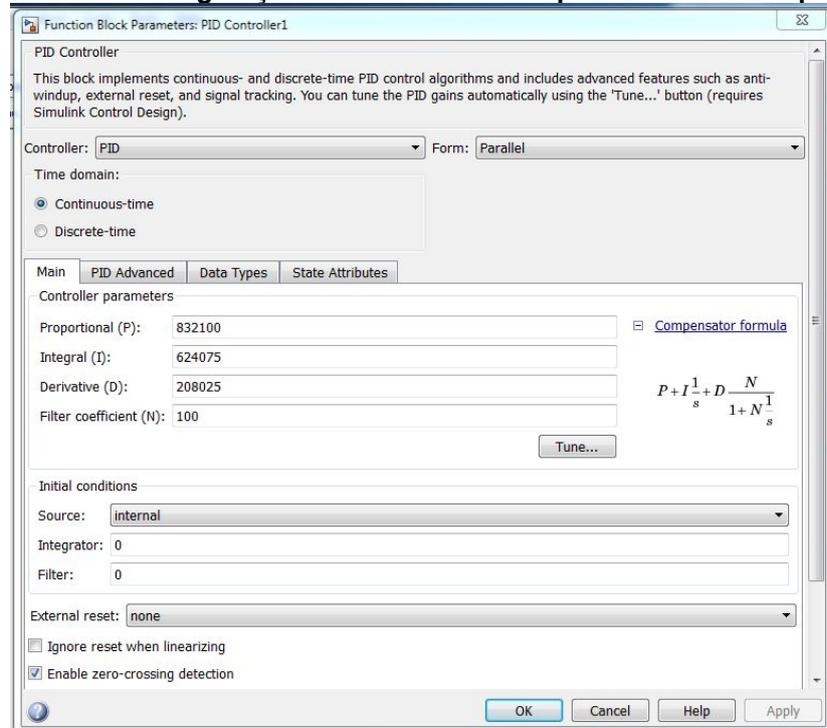
Nota: elaboração própria.

4.6.1 Controle por PID sintonizado

O controle PID aplicado ao sistema suspensão ativa faz-se pela configuração da figura 39 (sessão 3.4) acrescentando-se o controlador PID em série com a entrada “U” e uma linha de realimentação na entrada do PID. Ao aplicar-se um controlador PID ao sistema suspensão ativa $\frac{1}{4}$, deve-se realizar o procedimento de sintonia PID descrito no apêndice D. No entanto, para este sistema pode-se suprimir a etapa de sintonia PID inserindo os ganhos fornecidos por Messner e Tilbury (2018). Na figura 78 estão os valores dos ganhos P, D, I e N¹² fornecidos por esses autores:

- P = 832100
- I = 624075
- D = 208025
- N = 100

Figura 78 - Janela de configuração do controlador PID para o sistema suspensão ativa $\frac{1}{4}$



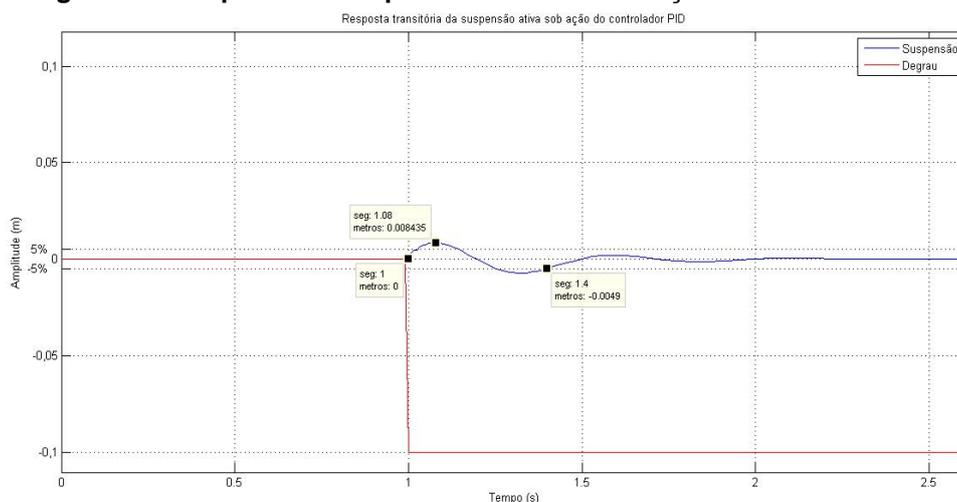
Nota: Elaboração própria.

¹² Para este sistema é importante dizer que os autores que modelaram este sistema não fazem uso deste parâmetro. No entanto, conforme visto no apêndice D, para que a expressão matemática $Kd \cdot \left(\frac{N}{1+N\frac{1}{s}} \right) = Kd \cdot s$ é necessário que o parâmetro “N” seja suficientemente grande. Utilizou-se para este trabalho N = 100.

Após a aplicação do controlador PID faz-se uma nova simulação. A curva de resposta deste sistema é mostrada na figura 79.

Na tabela 18 estão os parâmetros de desempenho transitório da suspensão ativa. Observa-se pela tabela 18 que o sistema está controlado, com um sobressinal de aproximadamente 8,4% e com um tempo de pico de 80ms. O tempo de acomodação é de 0,4 segundos, uma acomodação rápida.

Figura 79 – resposta da suspensão ativa 1/4 sob ação do controlador PID



Nota: elaboração própria.

Tabela 18 – parâmetros de desempenho transitório da suspensão ativa ¼ sob ação do controlador PID

| | |
|--|-------|
| <i>Tempo de atraso (s)</i> | - |
| <i>Tempo de subida (s)</i> | - |
| <i>Tempo de pico (s)</i> | 0,08 |
| <i>Tempo de acomodação (s)</i> | 0,4 |
| <i>Máxima ultrapassagem percentual (%)</i> | 8,435 |

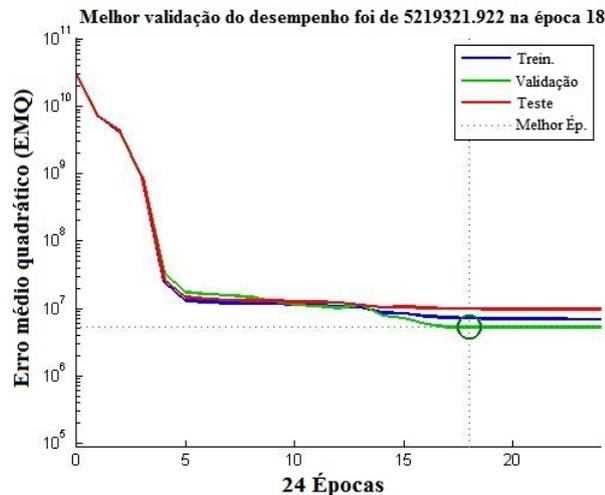
Nota: elaboração própria.

4.6.2 Controle por RNA substituindo PID

Nesta sessão faz-se a simulação do sistema suspensão ativa ¼ com a ação do controlador neural pelo método de aproximação. O procedimento é o mesmo realizado em 4.4.2 e 4.5.2. Primeiro faz-se o treinamento de uma rede MLP *feed-forward* por meio do aplicativo do matlab “Neural Network Fitting”, descrito no apêndice A. Após o treinamento avalia-se a precisão da rede pelo decaimento do erro médio quadrático, mostrado na figura

80. Observa-se pela figura 80 que o melhor desempenho da rede foi na época 18, sendo que o sistema parou de convergir com 24 épocas. Com o treinamento, a rede alcançou um erro entre 10^6 e 10^7 ¹³. A partir da época 24 o erro quadrático médio começa a aumentar.

Figura 80 - Decaimento do erro quadrático médio durante o treinamento da rede



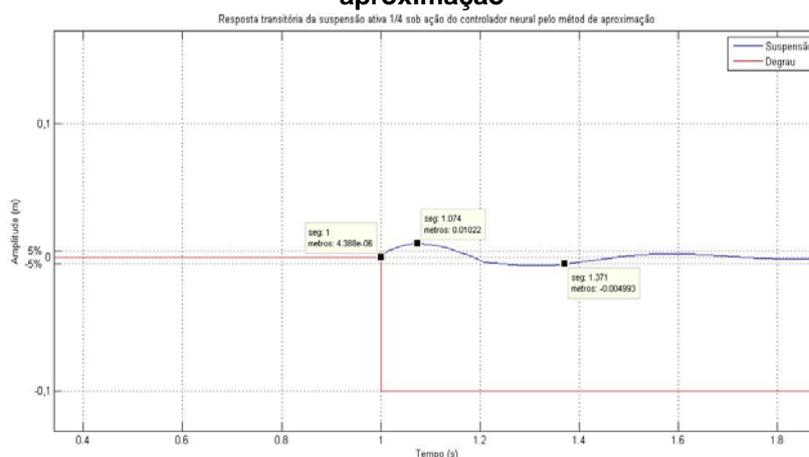
Nota: Elaboração própria.

Finalizado o treinamento, o controlador pode ser substituído pela rede neural e uma nova simulação é gerada.

A ação do controlador neural pelo método de aproximação está mostrada na figura 81, enquanto que os índices de desempenho transitório da curva de resposta da suspensão é mostrado na tabela 19. De acordo com a tabela 19, observa-se que o sistema está com um sobressinal de 10,22% e com um tempo de acomodação de 371ms. Em geral, o sistema atende as especificações dadas em 4.6, portanto é um controlador eficaz.

¹³ Para que o controlador possa amortecer a perturbação externa, é necessária um sinal de atuação elevado, na escala de 10^n , onde n é a potência que determina a intensidade do sinal de controle. Isto faz com que o erro de treinamento da rede seja elevado, visto que o erro é dado pela diferença entre a saída do sistema a ser aproximado (*Target*) e a saída da rede.

Figura 81 – resposta da suspensão ativa 1/4 sob ação do controlador neural pelo método de aproximação



Nota: elaboração própria.

Tabela 19 – parâmetros de desempenho transitório da suspensão ativa 1/4 sob ação do controlador neural pelo método de aproximação

| | |
|--|-------|
| <i>Tempo de atraso (s)</i> | - |
| <i>Tempo de subida (s)</i> | - |
| <i>Tempo de pico (s)</i> | 0.074 |
| <i>Tempo de acomodação (s)</i> | 0.371 |
| <i>Máxima ultrapassagem percentual (%)</i> | 10.22 |

Nota: elaboração própria.

4.6.3 Rede Neural de controle NARMA-L2

Assim como nos sistemas antena de azimute e motor de tração, primeiro constrói-se o diagrama de blocos do sistema de controle contendo a planta da suspensão ativa (figura 39 da sessão 3.4) em série com o controlador neural. É importante enfatizar que a resposta de controle da rede é eficaz se estiver situado após o sinal de perturbação W , pois a configuração deste controlador necessita de um sinal de referência atuante¹⁴. Caso contrário, a rede pode exibir resultados insatisfatórios.

Em seguida configura-se a rede neural NARMA-L2 com os parâmetros listados a seguir e executa-se o treinamento conforme o procedimento do apêndice B:

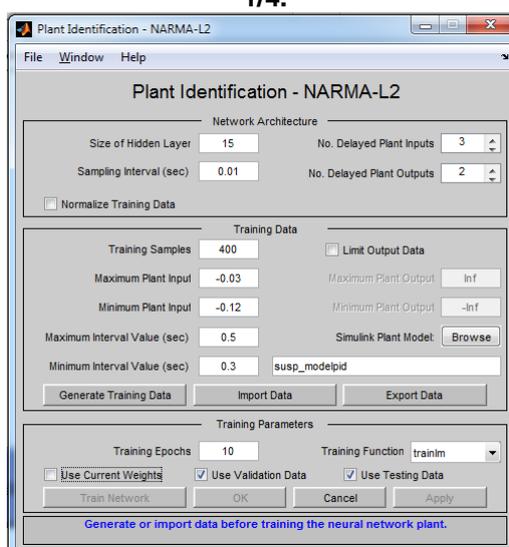
- Arquitetura da rede identificadora
 - 15 neurônios na camada oculta;
 - Intervalo de amostragem de 0.01 segundos;

¹⁴ Ou diferente de zero. Na sessão 3.4, a entrada de referência r do sistema é zero, pois admite-se que o sistema esteja em equilíbrio (figura 40).

- Tres passos de atraso no sinal de entrada da planta (z^{-3});
- Dois passos de atraso no sinal de saída da planta (z^{-1});
- Desativar a caixa de seleção “*Normalizar dados de treinamento*”
- Dados de treinamento
 - Número de amostras de treinamento igual a 400.
 - Valor máximo do sinal de entrada da planta igual a -0.03;
 - Valor mínimo do sinal de entrada da planta igual a -0.12;
 - Intervalo máximo de constância do sinal gerado de 0.5 segundos;
 - Intervalo mínimo de constância do sinal gerado de 0.3 segundos;
 - Desativar a caixa de seleção “*Limit output data*”
 - Valor máximo do sinal de saída da planta igual a infinito;
 - Valor mínimo do sinal de saída da planta igual a -infinito;
- Parâmetros de treinamento
 - Épocas de treinamento (iterações) do controlador igual a 10;
 - Algoritmo de treinamento (função de treinamento) de Levenberg-Marquardt (“*trainlm*”);
 - Desativar caixas de seleção “*Usar os pesos atuais*”;
 - Ativar caixas de seleção “*Usar dados de validação*” e “*Usar dados de teste*”;

Para este caso, é utilizado um número menor de amostras e de épocas de treinamento comparado aos outros sistemas, pois após a realização de um série de simulações constatou-se que a rede apresentava melhor generalização do sistema com um número menor de amostras e de épocas. A figura 82 apresenta a tela de configuração da rede NARMA-L2 com os parâmetros mencionados anteriormente para o sistema suspensão ativa 1/4.

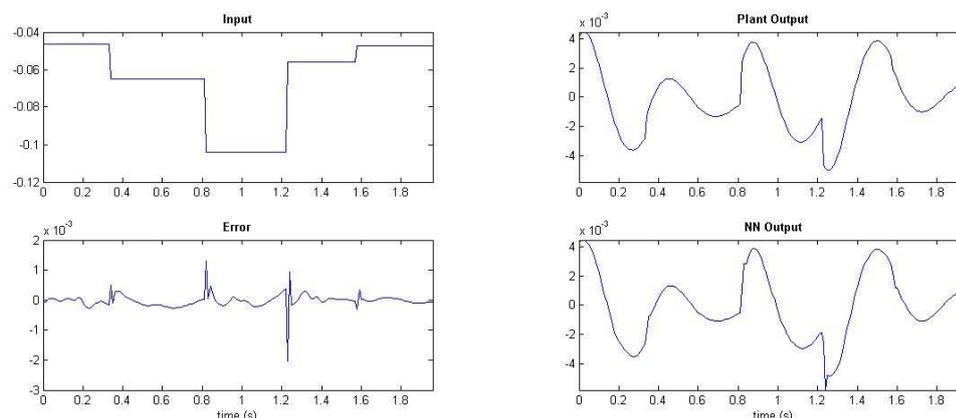
Figura 82 - tela de configuração da rede NARMA-L2 com os parâmetros para o sistema suspensão ativa 1/4.



Nota: Elaboração própria.

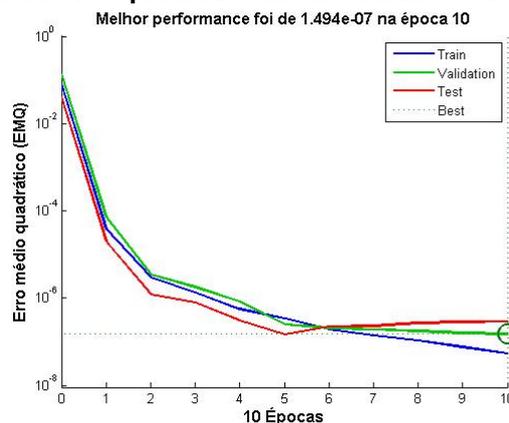
Assim que o treinamento é finalizado, faz-se a verificação do desempenho do treinamento pelo gráfico de treinamento (figura 83) e pelo decaimento do erro médio quadrático (figura 84). Pelo gráfico da figura 84, observa-se que o sinal de saída da rede NARMA-L2 acompanha o sinal de saída da planta o que indica que a rede generalizou bem a função da suspensão ativa. De acordo com a figura 84 a rede alcançou uma erro entre 10^{-6} e 10^{-8} após dez épocas de treinamento. A rede interrompeu o treinamento na época dez porque este foi o número de épocas com que ela fora configurada.

Figura 83 – gráfico de treinamento da rede NARMA-L2 para a suspensão ativa.



Nota: Elaboração própria.

Figura 84 - Decaimento do erro quadrático médio durante o treinamento da rede

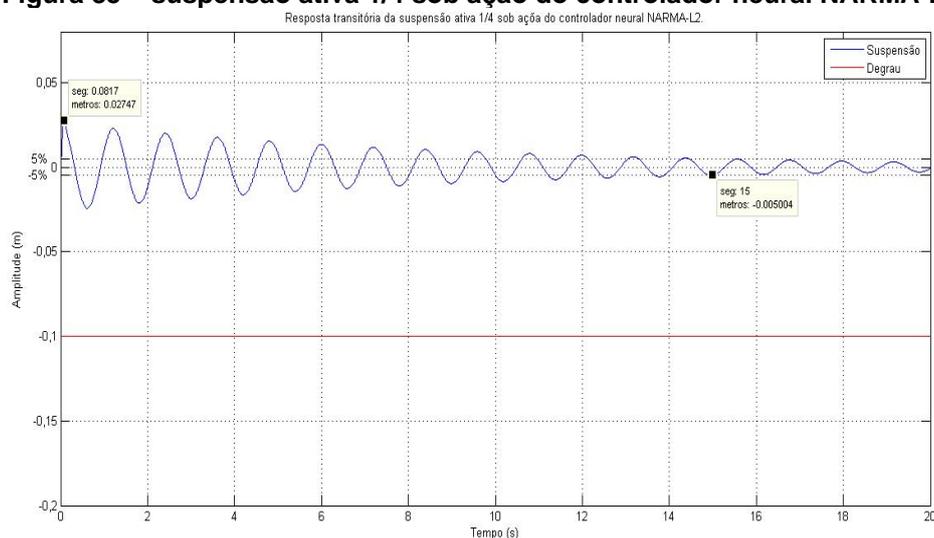


Nota: Elaboração própria.

Em seguida, faz-se a análise da resposta transitória do sistema suspensão ativa sob ação do controlador NARMA-L2. A resposta do sistema de suspensão é apresentada na figura 85. Pelo gráfico da figura 85 é possível observar que o sistema não está controlado, pois está oscilando com um longo período de acomodação. Isso indica que a rede não foi capaz de exercer um controle efetivo que atenuasse todas as oscilações geradas por uma perturbação na entrada. No entanto, comparando-o com o sistema natural (91% de sobressinal e 26 segundos de acomodação), pode-se dizer que houve uma redução significativa no sobressinal e no tempo de acomodação. Em geral, o controlador não é eficaz pois não atende as especificações de sobressinal e acomodação dadas em 4.3.

Os índices de desempenho transitórios deste sistema são mostrados na tabela 20. Observa-se na tabela 20 que o sistema possui um sobressinal de 27,47% e com um tempo de pico de 81,7ms. O tempo de acomodação é de 15 segundos.

Figura 85 – suspensão ativa 1/4 sob ação do controlador neural NARMA-L2



Nota: elaboração própria.

Tabela 20 – parâmetros de desempenho transitório da suspensão ativa 1/4 sob ação do controlador neural modelo de referência

| | |
|--|--------|
| <i>Tempo de atraso (s)</i> | - |
| <i>Tempo de subida (s)</i> | - |
| <i>Tempo de pico (s)</i> | 0.0817 |
| <i>Tempo de acomodação (s)</i> | 15 |
| <i>Máxima ultrapassagem percentual (%)</i> | 27.47 |

Nota: elaboração própria.

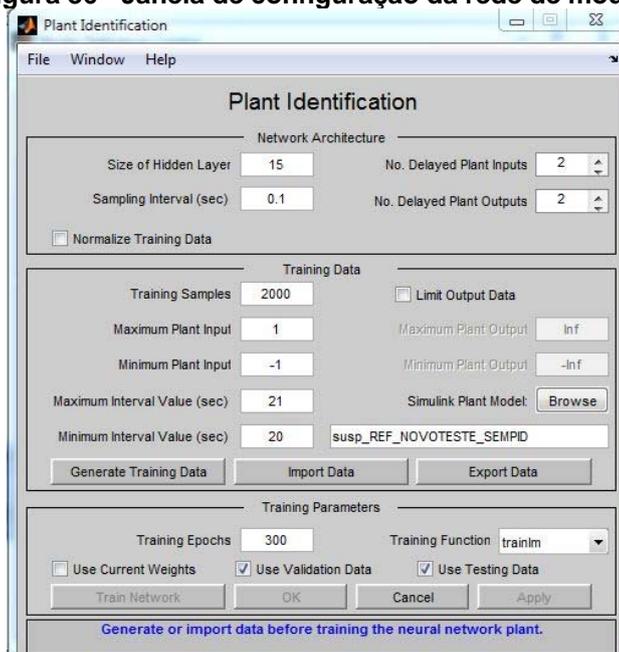
4.6.4 Controle de modelo de referência

Agora faz-se uma simulação do sistema suspensão ativa sob a ação do neurocontrolador Modelo de Referência. Realiza-se o mesmo procedimento aplicado aos sistemas antena de azimute e motor elétrico de tração. Primeiro treina-se a rede de modelo. Depois treina-se a rede controladora para forçar a saída da planta a seguir a saída do modelo de referência.

Para a configuração da rede de modelo são atribuídos os seguintes parâmetros (figura 86):

- Arquitetura da rede identificadora
 - 15 neurônios na camada oculta;
 - Intervalo de amostragem de 0.1 segundos;
 - Dois passos de atraso no sinal de entrada da planta (z^{-2});
 - Dois passos de atraso no sinal de saída da planta (z^{-2});
- Dados de treinamento
 - Valor máximo do sinal de entrada da planta igual a um;
 - Valor mínimo do sinal de entrada da planta igual a -1;
 - Intervalo máximo de constância do sinal gerado de 21 segundos;
 - Intervalo mínimo de constância do sinal gerado de 20 segundos;
 - Número de amostras de treinamento igual a 2000.
 - Desativar o limite de saída do sinal (caixa de seleção “*Limit output data*”)
- Parâmetros de treinamento
 - Épocas de treinamento (iterações) do controlador igual a 300;
 - Algoritmo de treinamento (função de treinamento) de Levenberg-Marquardt (“*trainlm*”)

Figura 86 - Janela de configuração da rede de modelo

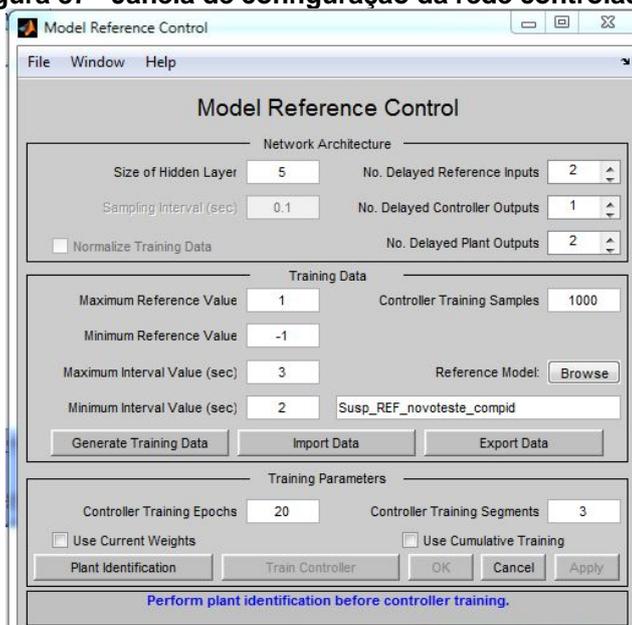


Nota: Elaboração própria.

E para a configuração da rede controladora, são atribuídos os seguintes parâmetros (figura 87):

- Arquitetura da rede controladora
 - 5 neurônios na camada oculta;
 - Dois passos de atraso no sinal de referência (z^{-2});
 - Um passo de atraso no sinal de saída (z^{-1});
 - Dois passos de atraso no sinal de saída da planta (z^{-2});
- Dados de treinamento
 - Valor máximo do sinal de referência randômico gerado igual a um;
 - Valor mínimo do sinal de referência randômico gerado igual a -1;
 - Intervalo máximo de constância do sinal randômico gerado de três segundos;
 - Intervalo mínimo de constância do sinal randômico gerado de dois segundos;
 - Número de amostras de treinamento igual a 1000.
- Parâmetros de treinamento
 - Épocas de treinamento (iterações) do controlador igual a 20.
 - Segmentos de treinamento do controlador igual a 3;

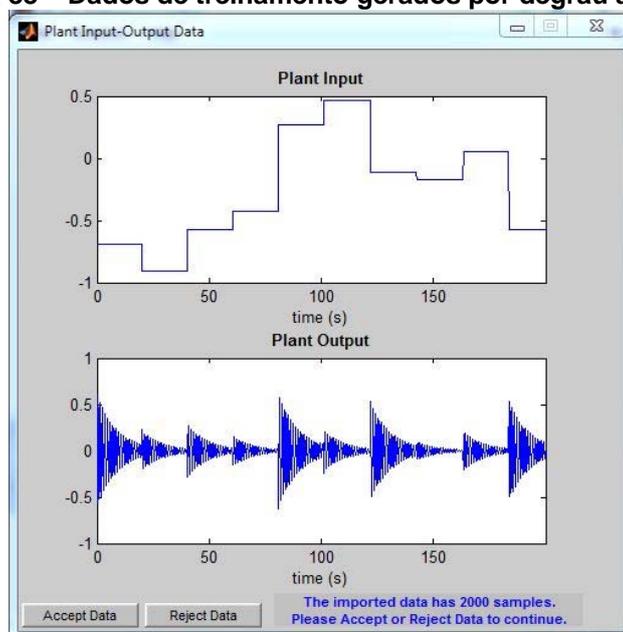
Figura 87 - Janela de configuração da rede controladora



Nota: Elaboração própria.

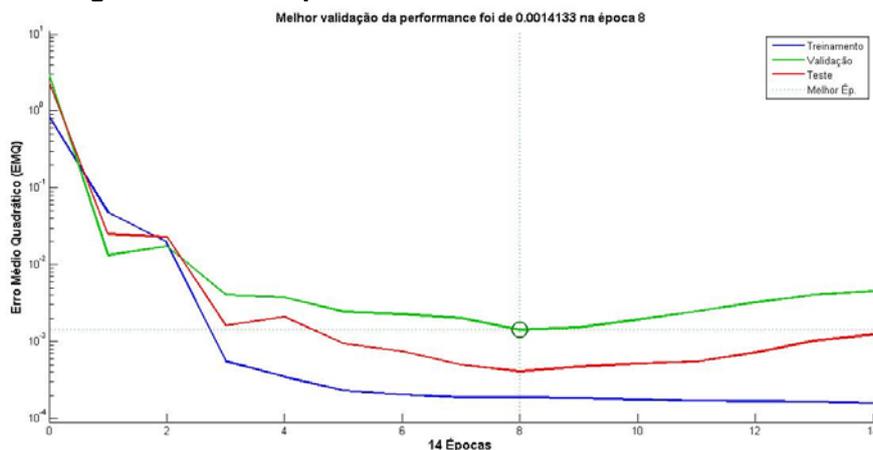
O processo de treinamento consiste em duas etapas: geração de dados e treinamento da rede. Primeiro gera-se um conjunto de dados para o treinamento da rede de modelo (figura 88). Após realizar o treino dessa rede, o índice de desempenho do treinamento, dado pelo decaimento do erro médio quadrático (EMQ), é mostrado na figura 89.

Figura 88 – Dados de treinamento gerados por degrau aleatório



Nota: Elaboração própria.

Figura 89 – Desempenho do treinamento da rede de modelo

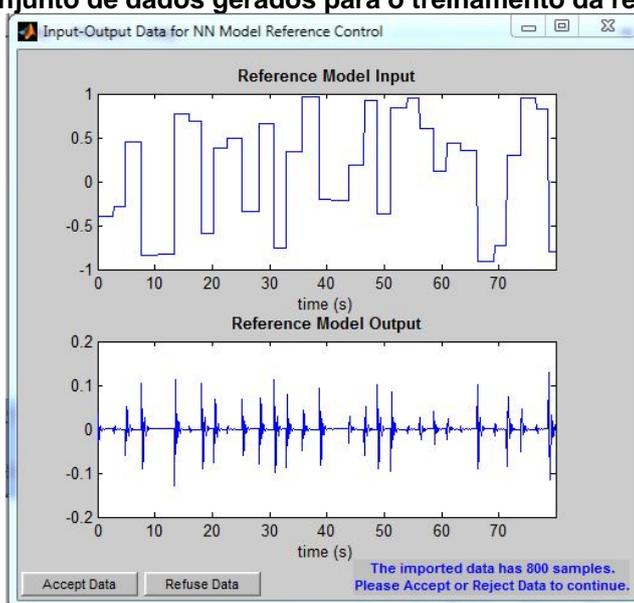


Nota: Elaboração própria.

Pela figura 89, é observado que o EMQ atingiu a escala entre 10^{-3} e 10^{-4} , e o treinamento foi interrompido na época 14, sendo que a melhor época é a oito de acordo com o critério de validação.

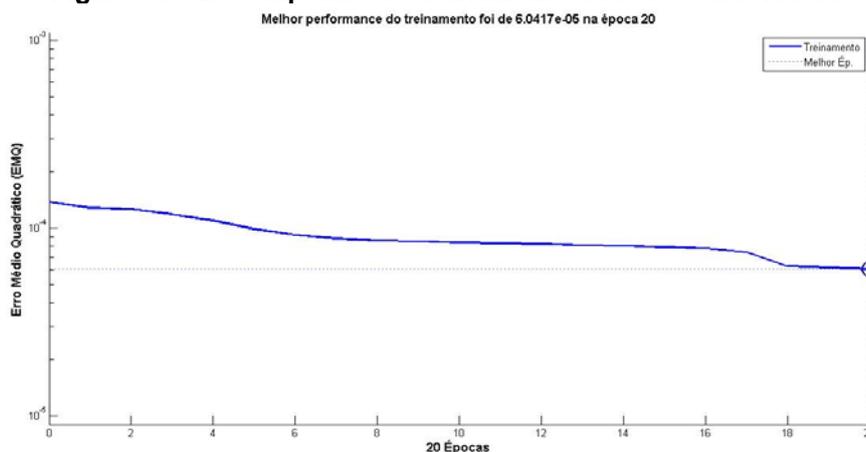
Em seguida, faz-se o treinamento da rede controladora pelo mesmo procedimento. Um conjunto de 800 amostras é gerado para o treinamento da rede controladora (figura 90). À medida que o treino é realizado ocorre um decaimento do EMQ (figura 91), o qual está situado entre 10^{-4} e 10^{-5} após 20 épocas de treinamento realizados sobre três segmentos. Finalizado o treino, avalia-se a atuação da rede neural (linha verde) comparado com o modelo de sistema controlado fornecido pelo usuário (linha azul) conforme visto na figura 92.

Figura 90 – Conjunto de dados gerados para o treinamento da rede controladora



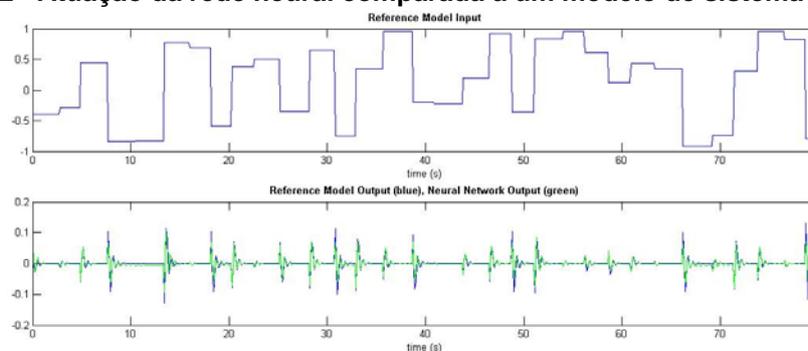
Nota: Elaboração própria.

Figura 91 – Desempenho do treinamento da rede controladora



Nota: Elaboração própria.

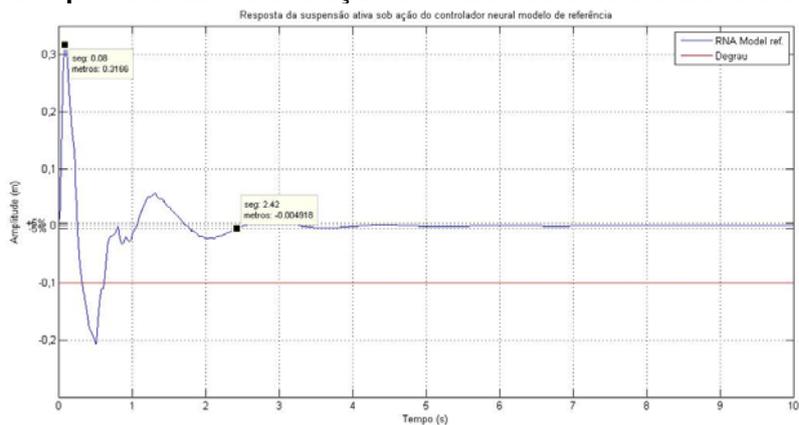
Figura 92 - Atuação da rede neural comparada a um modelo de sistema controlado



Nota: Elaboração própria.

Na figura 93 é apresentado o comportamento da resposta da suspensão ativa sob ação de controle da rede neural modelo de referência, e na tabela 21 estão os valores dos parâmetros do transitório. Pela figura 93 observa-se que houve uma acomodação da resposta do sistema após 1,4 segundos com uma tempo de pico de 0,07 segundos e sua máxima ultrapassagem alcança um valor excedente de 81,1. Pode-se dizer que para este sistema a rede neural Modelo de referência não é eficaz, uma vez que seu desempenho não atende a especificação de sobressinal.

Figura 93 – Resposta do motor sob ação do controlador neural modelo de referência



Nota: Elaboração Própria.

Tabela 21 - Parâmetros de desempenho transitório

| | |
|--|-------|
| <i>Tempo de atraso (s)</i> | - |
| <i>Tempo de subida (s)</i> | - |
| <i>Tempo de pico (s)</i> | 0.08 |
| <i>Tempo de acomodação (s)</i> | 2.42 |
| <i>Máxima ultrapassagem percentual (%)</i> | 216.6 |

Nota: elaboração própria.

5 DISCUSSÃO DOS RESULTADOS

Para a discussão dos resultados são utilizados as informações contidas nos seguintes itens:

- Tabelas 22, 23 e 24: utilizadas para verificar se os controladores atendem aos requisitos de projeto dados em 4.3 e para a comparação do desempenho dos neurocontroladores e relação ao PID;
- Figuras 94, 96 e 98 – utilizadas para verificar o erro estacionário;
- Figuras 95, 97, 99 - para fazer algumas colocações sobre as limitações das redes em relação à complexidade dos sistemas;

Inicialmente verifica-se se os controladores atenderam aos requisitos de projeto para checar sua viabilidade e em seguida, faz-se uma comparação do desempenho dos controladores neurais em relação ao PID.

O primeiro caso (Antena de azimute): observa-se pela tabela 22 que os controladores PID e pelo método de aproximação atendem as especificações de sobressinal (M_p), tempo de acomodação (T_s). A figura 94 mostra que esses controladores atendem a especificação de erro estacionário (e_{ss}).

Os controladores NARMA-L2 e Modelo de Referência atendem a especificação de sobressinal, no entanto não atendem a especificação de acomodação (tabela 22). Em relação ao erro estacionário (figura 94), o controlador NARMA-L2 consegue atender, porém o controlador Modelo de Referência não estabiliza na faixa de tolerância de $\pm 5\%$, conforme fora visto na sessão 4.4.4.

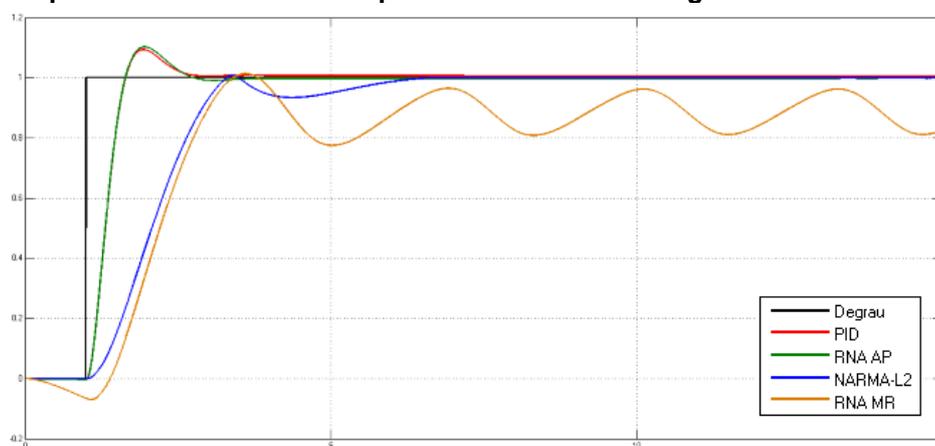
Quanto ao tempo de acomodação do controlador NARMA-L2 deve-se observar que, conforme mencionado em 2.2.2.3, a rede NARMA-L2 depende do comportamento da resposta da planta. Logo, é notório que o controlador apresente uma resposta com o mesmo tempo de acomodação ou próximo ao da planta não controlada. Também deve-se considerar que a ideia central deste tipo de controlador é cancelar as não-linearidades do sistema, não garantindo uma resposta mais rápida.

Tabela 22 - Desempenho dos controladores do sistema antena de azimute

| Antena de Azimute | | | | |
|-------------------------------------|-------|-------------|----------|----------------------|
| Controlador | PID | APROXIMAÇÃO | NARMA-L2 | MODELO DE REFERÊNCIA |
| Tempo de atraso (s) | 0.316 | 0.315 | 1.088 | 1.222 |
| Tempo de subida (s) | 0.648 | 0.645 | 2.3 | 2.41 |
| Tempo de pico (s) | 0.903 | 0.949 | 2.4 | 2.6 |
| Tempo de acomodação (s) | 1.29 | 1.356 | 4.026 | - |
| Máxima ultrapassagem percentual (%) | 9.3 | 10.3 | 0.8 | 1.4 |

Nota: elaboração própria.

Figura 94 – Desempenho dos controladores para uma entrada em degrau do sistema antena de azimute.



Nota: Elaboração Própria.

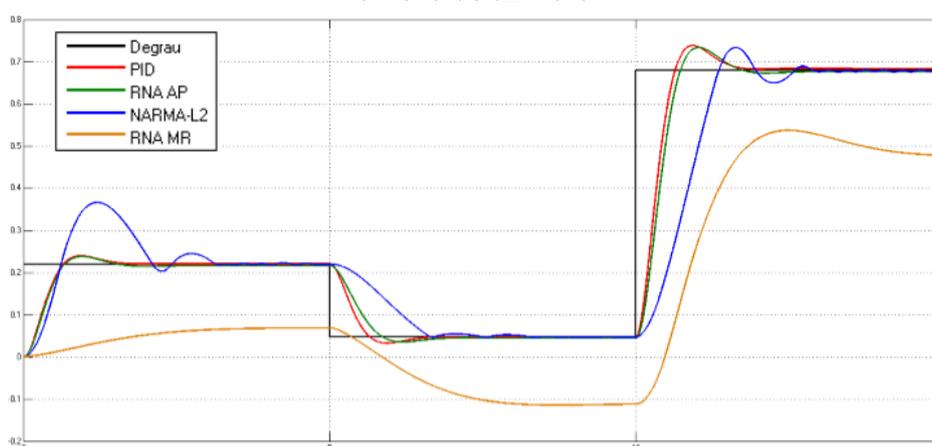
O controlador pelo método de aproximação apresentou índices de desempenho próximos ao PID. Isso demonstra que a rede de aproximação generalizou com eficiência a função que representa o PID. Já os controladores NARMA-L2 e Modelo de Referência apresentaram uma resposta mais lenta em comparação ao PID com relação aos índices Tempo de atraso, de pico e de subida. Contudo, o sobressinal destes controladores é menor em relação ao PID.

A figura 95 apresenta o desempenho de todos os controladores mediante a aplicação de degraus aleatórios com amplitudes entre zero e 1 para o sistema antena de azimute. Observa-se pelas figuras 95 que os controladores PID e pelo método de aproximação não apresentam distinções em suas respostas para ambas as entradas. Já para os controladores NARMA-L2 e Modelo de Referência há algumas considerações. No controlador NARMA-L2 não há sobressinal significativo para entradas próximas de zero ou um. Contudo para valores intermediários (como 0,2 e 0,7) a resposta apresenta sobressinais consideráveis. Observou-se após algumas simulações que as redes neurais costumam limitar a saída quando é aplicado

um sinal de entrada fora do range o qual elas foram treinadas. Logo, para as entradas zero, um ou próximo desses valores a rede limita a saída.

Observa-se a presença de um erro estacionário no controlador modelo de referência. Isso demonstra que a rede de modelo do controlador não generalizou bem o sistema sem controle.

Figura 95 - Desempenho de todos os controladores para uma entrada de degraus aleatórios do sistema antena de azimute.



Nota: Elaboração Própria.

Agora o segundo caso (Motor de tração): pela tabela 23, observa-se que os controladores PID e de Aproximação atenderam as especificações de projeto. A figura 96 mostra que esses controladores também atendem a especificação de erro estacionário.

Quanto aos controladores NARMA-L2 e Modelo de Referência, estes apresentam um tempo de acomodação aceitável pelas especificações, contudo, em relação ao sobressinal o controlador NARMA-L2 se apresenta indesejável enquanto que o Modelo de referência não atinge o valor desejado. Pela figura 96, observa-se que o controlador NARMA-L2 atende à especificação de erro estacionário, porém o Modelo de Referência não.

Percebe-se ao observa-se as tabelas 12 (sessão 4.5) e 23 que o sistema sem controle e o controlado pela rede NARMA-L2 apresentam índices de desempenho iguais, diferindo somente no tempo de acomodação. Isto demonstra que que a rede reproduziu o comportamento inicial da resposta transitória do motor com uma oscilação (figura 96). As demais oscilações são canceladas pela ação do controlador, o que reduz significativamente o tempo de acomodação.

Ao comparar-se o controlador pelo método de aproximação, nota-se que o controlador neural apresenta índices próximos do PID, diferindo consideravelmente no tempo de acomodação. A diferença de sobressinal entre esses controladores é de 1,2%. Isto demonstra que a função do PID não foi completamente generalizada, podendo conter alguns desvios em sua representação.

O controlador NARMA-L2 apresentou uma resposta mais rápida comparado ao PID para os índices de tempo de atraso, subida, pico e acomodação. Porém, o sobressinal é muito elevado comparado ao PID.

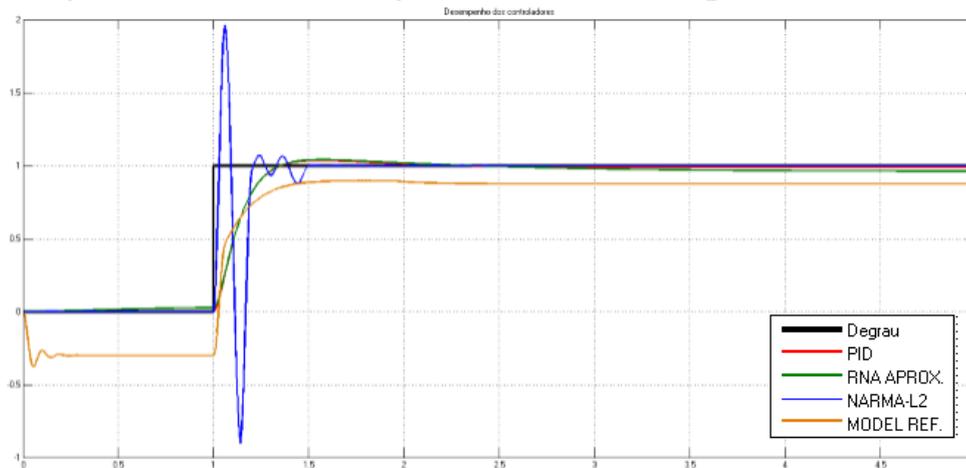
Já o controlador Modelo de Referência apresenta uma resposta mais lenta em relação ao PID, com exceção do tempo de atraso.

Tabela 23 - Desempenho dos controladores do sistema motor de tração

| Motor de Tração | | | | |
|-------------------------------------|-------|-------------|----------|----------------------|
| Controlador | PID | APROXIMAÇÃO | NARMA-L2 | MODELO DE REFERÊNCIA |
| Tempo de atraso (s) | 0.108 | 0.109 | 0.02 | 0.08 |
| Tempo de subida (s) | 0.357 | 0.353 | 0.031 | - |
| Tempo de pico (s) | 0.545 | 0.539 | 0.06 | 0.8 |
| Tempo de acomodação (s) | 0.867 | 3.003 | 0.381 | 1.61 |
| Máxima ultrapassagem percentual (%) | 3.6 | 4.8 | 96.2 | - |

Nota: elaboração própria.

Figura 96 - Desempenho dos controladores para uma entrada em degrau do sistema motor de tração.



Nota: Elaboração Própria.

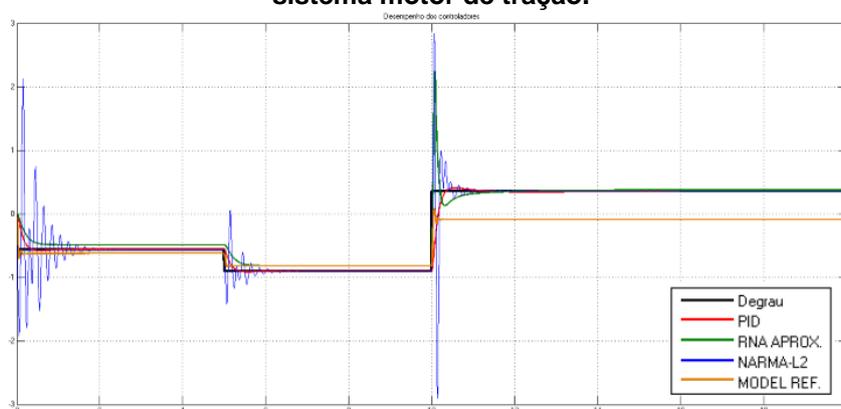
Agora observa-se o desempenho dos controladores para entrada do tipo degrau aleatório com amplitudes entre um e -1 (figura 97). O controlador neural pelo método de aproximação exerce um controle eficaz somente para os dois primeiros pulsos. No terceiro exibe um pico de sobressinal que excede as especificações de projeto dadas em 4.6. Isto demonstra que para alguns valores de entrada o sinal de controle não consegue atuar efetivamente. Portanto, é necessário realizar um novo treinamento com um número de amostras maior ou aumentar o número de épocas de treinamento.

O controlador neural NARMA-L2 apresenta oscilações em seu transitório que torna sua resposta indesejável. Observa-se que para cada amplitude de pulso, a resposta exibe

sobressinais diferentes com mais ou menos oscilações em seu transitório. No primeiro pulso (-0,5), o controlador exibe oscilações semelhantes ao sistema natural. Já no terceiro pulso as oscilações são reduzidas porém o sobressinal é alto. Assim como fora mencionado anteriormente, este controlador depende do comportamento do sistema sem controle. Logo, é esperado que o controlador venha apresentar oscilações na resposta.

O Modelo de Referência possui um erro estacionário em sua resposta para todos os pulsos, sendo o terceiro pulso o mais elevado. É visto na sessão 4.5.4 (figura 75) que a rede neural de controle não generalizou completamente a função do modelo de referência, por isso a presença de erro estacionário em sua resposta.

Figura 97 - Desempenho de todos os controladores para uma entrada de degraus aleatórios do sistema motor de tração.



Nota: Elaboração Própria.

Por fim, o terceiro sistema (Suspensão ativa): pela tabela 24 observa-se que novamente os controladores PID e de aproximação atenderam as especificações de projeto dadas em 4.6. A figura 98¹⁵ mostra que estes controladores atendem também a especificação de erro estacionário.

O controlador neural NARMA-L2 apresenta um tempo de acomodação longo, porém menor que o do sistema sem controle, e um sobressinal que excede em 7% o limite de sobressinal dado em 4.6. A figura 98 mostra que este controlador atende a especificação de erro estacionário. Em geral, este controlador reduz o sobressinal e o tempo de acomodação mas não atende todas as especificações dadas em 4.6.

O controlador Modelo de referência atende a especificação de acomodação, porém apresenta um sobressinal alto que não é desejável. Observa-se pela figura 98 que este controlador atende a especificação de erro estacionário.

¹⁵ Nesta figura, "W" representa a perturbação em degrau, conforme mencionado na sessão 3.4

Ao comparar-se o controlador neural pelo método de aproximação, percebe-se que os índices de desempenho tempo de pico e de acomodação são próximos. Contudo, há uma diferença de 1,78% no sobressinal.

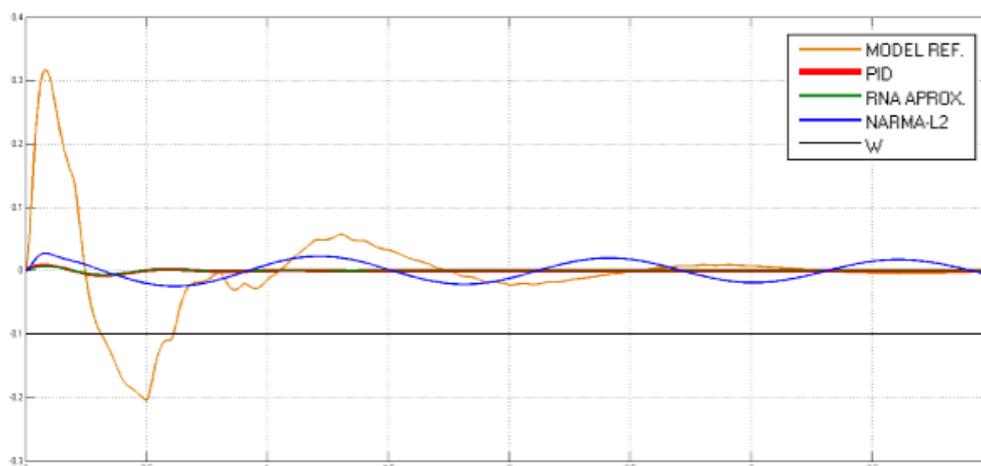
Os controladores neurais NARMA-L2 e Modelo de Referência apresentam um tempo de pico semelhante ao do PID, porém o sobressinal e o tempo de acomodação são mais elevados.

Tabela 24 - Desempenho dos controladores do sistema suspensão ativa

| Suspensão ativa 1/4 | | | | |
|-------------------------------------|-------|-------------|----------|----------------------|
| Controlador | PID | APROXIMAÇÃO | NARMA-L2 | MODELO DE REFERÊNCIA |
| Tempo de atraso (s) | - | - | - | - |
| Tempo de subida (s) | - | - | - | - |
| Tempo de pico (s) | 0,08 | 0.074 | 0.0817 | 0.08 |
| Tempo de acomodação (s) | 0,4 | 0.371 | 15 | 2.42 |
| Máxima ultrapassagem percentual (%) | 8,435 | 10.22 | 27.47 | 216.6 |

Nota: elaboração própria.

Figura 98 - Desempenho dos controladores para uma entrada em degrau do sistema suspensão ativa 1/4.

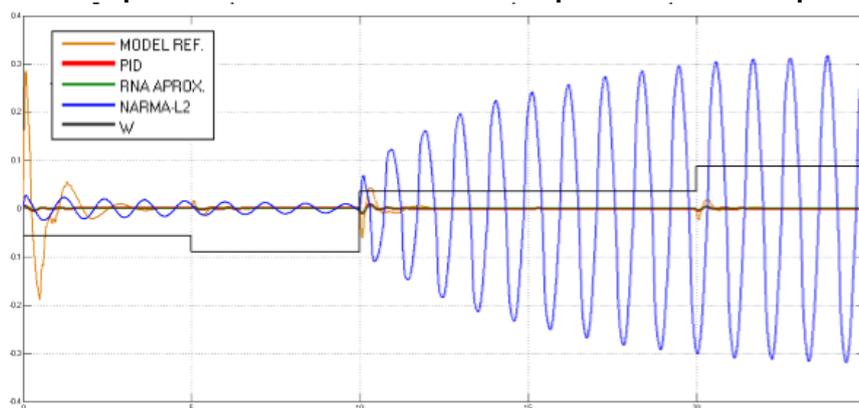


Nota: Elaboração Própria.

A figura 99 apresenta o desempenho dos controladores com uma entrada do tipo degrau aleatório entre um e -1. Observa-se primeiro que a rede de aproximação apresenta uma resposta semelhante ao do PID, o que demonstra que a rede generalizou com sucesso a função do PID. O controlador NARMA-L2, por sua vez se tornou instável. É comum ocorrer em alguns casos da rede convergir para um modelo instável durante o treinamento. Uma medida a ser tomada seria realizar um novo treinamento com mais épocas de treinamento.

O Modelo de referência apresenta uma resposta indesejada somente no primeiro pulso, com um sobressinal elevado. Nos pulsos seguintes o controlador atua efetivamente. Conforme mencionado na sessão 2.2.2.2, a rede de modelo é usada para prever o efeito das mudanças do controlador na saída da planta. Nesse caso, a rede de modelo não convergiu para um modelo fiel à representação da planta sem controle levando a rede de controle a uma atuação insatisfatória.

Figura 99 – Desempenho de todos os controladores para o sistema suspensão ativa 1/4.



Nota: Elaboração própria.

6 CONCLUSÕES

O controle de sistemas dinâmicos por redes neurais artificiais (RNA) pode ser realizado desde que seja selecionada uma configuração de rede adequada ao tipo de sistema. Deve-se observar os seguintes itens ao selecionar uma configuração de rede: tipo de rede (recorrente ou *feedforward*), estrutura das camadas, quantidade de neurônios nas camadas de entrada, ocultas e de saída, tipo de função de ativação, quantidade de variáveis de saída e de entrada, tipos de dados de treinamento e tipo de algoritmo de aprendizagem. Esses itens são considerados segundo a complexidade do sistema. Por exemplo, se um sistema opera dentro de um conjunto de dados fixos, então a rede *feedforward* pode ser adotada, visto que essa rede é treinada com um histórico de dados estático. Do contrário, se o sistema apresenta um conjunto de dados que não se restringe a um intervalo fixo, é necessário adotar uma rede do tipo recorrente, visto que ela possui elos de realimentação que retornam as saídas atuais da rede com atrasos para as entradas. O número de camadas é considerado em função da generalização. Para uma boa generalização utiliza-se comumente uma camada de entrada, uma oculta e uma de saída. No entanto, se o sistema a ser generalizado apresentar elevada complexidade, aumenta-se o número de camadas ocultas. A função de ativação determina a característica do sinal de saída dos neurônios. Se é desejável que o sinal de saída tenha dois estados apenas (zero ou um), então a função limiar é utilizada. Por outro lado, se o sinal de saída está contido dentro de um intervalo de valores relacionadas linearmente por um fator de amplificação, então a função linear por partes é utilizada. Em uma outra circunstância, se o sinal ponderado apresentar valores contínuos com um aspecto intermediário entre linear e não linear, a função sigmóide é utilizada, visto que esta função apresenta um balanceamento adequado entre o comportamento linear e não linear. Em geral, costuma-se utilizar a função sigmóide na camada de entrada e a limiar por partes na oculta. A quantidade de variáveis de entrada e saída é selecionada conforme as características do sistema a ser generalizado. Se o sistema possui uma variável de entrada que gera apenas uma saída, então é necessário uma par de dados de entrada e saída para o treinamento da rede. No entanto, para um melhor desempenho da capacidade de generalização da rede pode-se ampliar o número de variáveis de entrada, como no caso da generalização da função do PID que precisa de duas variáveis de entrada (erro e derivada do erro) para realizar uma generalização eficiente da função de controle. Em relação aos tipos de dados, sabe-se que o conjunto de dados utilizado para o treinamento da rede é dividido em três grupos: treinamento, teste e validação. Essa divisão foi consolidada com intuito de melhorar o desempenho do treinamento da rede. A proporção do conjunto total de dados para cada grupo depende do número de amostras. Se o número de

amostras for baixo é aconselhável atribuir a maior porcentagem dos dados para treinamento. Contudo, se houver um bom número de amostras pode-se separar uma proporção maior dos dados para o conjunto de teste e validação. Isto torna a verificação do desempenho da rede mais eficiente. Na escolha do algoritmo de aprendizagem, deve-se considerar o parâmetro taxa de aprendizado η , pois a escolha do valor deste parâmetro determina o desempenho da aprendizagem da rede, no caso de um algoritmo por correção de erro.

Em se tratando de projetos, pode-se projetar um controlador PID que atenda as especificações de resposta desejada (sobressinal, tempo de atraso etc.). As redes por si só não podem ser projetadas em função das especificações, no entanto podem ser treinadas com um conjunto de dados que as capacite a atender as especificações de resposta desejada. Porém, em alguns casos a rede pode tornar a resposta de saída mais indesejável, como no caso da suspensão controlada pela rede Modelo de Referência que amplia o sobressinal da resposta em vez de reduzi-lo ou no caso da suspensão controlada pela rede NARMA-L2, que acaba instabilizando o sinal de saída para alguns valores de entrada. O controlador PID clássico, por si só, não possui uma capacidade adaptativa como das redes neurais, embora existam métodos de autossintonia que podem ser implementados a ele. No caso de uma possível alteração nos parâmetros do sistema, é necessário projetar um novo controlador PID. Já as redes neurais podem alterar seus parâmetros durante a etapa de aprendizagem.

Observou-se que o treinamento da rede foi bem sucedido nas seguintes situações:

- MLP para “aprender” a função de controle (por aproximação);
- NARMA-L2 e Modelo de Referência na identificação do modelo de planta;

O fato de as redes neurais pelo método de aproximação apresentarem um desempenho aproximado ao do PID é esperado, visto que elas foram treinadas com os dados do próprio PID. Nesse caso, a rede deve apresentar um desempenho igual ou aproximado ao do PID, como é o caso do sistema antena de azimute, onde a rede neural apresentou índices de desempenho muito próximos aos do PID. No entanto os dois últimos neurocontroladores atingiram uma configuração de controle pobre, tanto a NARMA-L2 com duas redes treinadas paralelamente, quanto a do Modelo de Referência, com duas redes treinadas em etapas distintas (identificação e controle). Logo, essas redes podem não ser uma alternativa viável no controle de sistemas dinâmicos visto que elas possuem algumas limitações. Elas podem atingir o valor desejado, entretanto elas não garantem estabilidade e podem não atender todas as especificações de projeto. No caso de sistemas com transitórios muito oscilantes (como no caso do motor de tração e da suspensão ativa), as oscilações podem fazer com que a rede neural fique presa a mínimos globais durante a fase de treinamento. Isso leva a rede a apresentar desvios na resposta de saída, como é o caso da rede modelo de referência. Nesse

caso, há a necessidade de aprimorar ainda mais as técnicas de controle com redes recorrentes.

Por outro lado, o controle neural pelo método de aproximação pode ser uma alternativa viável no controle de sistemas dinâmicos. Em se tratando das especificações de projeto essa abordagem de controle apresenta um melhor desempenho no controle dos três sistemas descritos neste trabalho em relação aos demais controladores.

Por fim, uma das propostas para experimentos consiste em variar outros parâmetros de treinamento, como o número de passos de atraso nas redes NARMA-L2 e no Modelo de Referência. A princípio, um aumento do número de valores de atrasados contribui para uma melhor sintonia na parte de controle integrativa, captando melhor o histórico do sinal de treinamento, no entanto pode aumentar a complexidade do aprendizado, demandando mais épocas para estabilização dos valores dos pesos das conexões. Seria interessante verificar se há um valor que represente um compromisso ótimo entre precisão do histórico e complexidade da convergência do erro.

REFERÊNCIAS

- ANTOLINES, J. D. R.; QUIROGA, D. F. C.; PACHÓN, M. Á. P. Implementação de neurocontroladores em linha: Três configurações, três plantas. **Ing. Univ. Bogotá**. Junho, 2012. Disponível em: <<http://www.scielo.org.co/pdf/inun/v16n1/v16n1a10.pdf>>. Acesso em: 17/05/2018.
- ATHERTON, D. P. Almost Six Decades in Control Engineering. **IEEE Control Systems Magazine**.v. 34-6, pg. 103-110, 2014. doi:10.1109/MCS.2014.2359588
- BAKSHI, N. A. Model Reference Adaptive Control of Quadrotor UAVs: A Neural Network Perspective. **Intech**, chap. 8, Adaptive Robust Control Systems. Acesso: <<http://dx.doi.org/10.5772/intechopen.71487>>.
- BALAKRISHNAN, S. N.; WEIL, R. D. **Mathl. Comput. Modelling**. Neural control: A Literature Survey. 1996. Vol.23. Elsevier: Great Britain.
- BARR, A.; FEIGENBAUM, E. A. **The Handbook of AI**. Vol. 1. Los Altos, CA: W. Kaufmann. 1981.
- BOLTON, W. **Mecatrônica: uma abordagem multidisciplinar**. 4 ed. Porto Alegre: Bookman, 2010.
- BRAGA, A. P., LUDEMIR, T.B., CARVALHO, A.C.P. **Introdução às redes neurais: teoria e aplicações**. Rio de Janeiro: LTC, 2000.
- BREEMEN, V.A.J.N., VEELNTURF, L.P.J. Neural Adaptive Feedback Linearization Control. **Journal A**. vol. 37, pp. 65-71, 1996.
- CAMPOS, M. M. de; SAITO, Kaku. **Sistemas inteligentes em controle e automação de processos**. Rio de Janeiro: Editora Ciência Moderna Ltda., 2004.
- CARVALHO, A. P. **Redes Neurais Artificiais**. Desenvolvimento de aplicações. Coleta de dados e separação em conjuntos. 2009. Disponível em: <<http://conteudo.icmc.usp.br/pessoas/andre/research/neural/desenv.htm>>. Acesso em: 11/2018.
- CERQUEIRA, E. O.; ANDRADE, J. C.; POPPI, R. J. Redes Neurais e suas aplicações em calibração multivariada. **Quim. Nova**. Campinas. Vol. 24. No. 6. Mar. 2001. Disponível em: <<http://www.scielo.br/pdf/qn/v24n6/6797.pdf>>. Acessado em: 06 junho 2016.
- CITISYSTEMS. Motor CC: Saiba como Funciona e de que forma Especificar. Disponível em: <<https://www.citisystems.com.br/motor-cc/>>. Acessado em: 11/07/2018.
- COMSOL. Release. 5.2a. Multibody dynamics. Disponível em: <<https://br.comsol.com/release/5.2a/multibody-dynamics-module>>. Acesso em: 07/2018.
- COSTA, E. J. X. da. Inteligência artificial aplicada à Zootecnia. **Revista brasileira de zootecnia**. Minas Gerais. v.38. Pg.390-396. 2009. Disponível em: <http://www.scielo.br/scielo.php?script=sci_arttext&pid=S1516-35982009001300038>. Acessado em: 25 maio 2016.

- CYBENKO, G. Approximations by superpositions of sigmoidal functions, **Mathematics of Control, Signals, and Systems**, 2 (4), 303-314 (1989).
- DORF, R. C., BISHOP, R. H. **Sistemas de controle modernos**. 12 ed. Rio de Janeiro: LTC, 2013.
- FAUSETT, L. **Fundamentos de redes neurais: arquiteturas, algoritmos e aplicações**. USA. Prentice-Hall: New Jersey, 1994.
- FERRARI, A. C. **Controlador pid sintonizado por redes neurais artificiais**. Trabalho de Conclusão de Curso de Graduação em Engenharia Elétrica. UFPR – Universidade Federal do Paraná. Curitiba. 2010.
- FRANÇA, A. I.; GOMES, J. N. A técnica de ERP: investigando a assimetria sujeito-objeto na interface sintaxe-semântica com EEG. **Revista eletrônica PUCRS**. Porto Alegre. Jul/set 2015. Vol. 50. No. 3. Pg. 360-370.
- GONÇALVES, D. N. S., SILVA, M. A., ALMEIDA, M. A. Procedimento para uso de Redes Neurais Artificiais no planejamento estratégico de fluxo de carga no Brasil. **The Journal of Transport Literature**. 2015. Disponível em: <<http://www.scielo.br/pdf/jtl/v9n1/2238-1031-jtl-09-01-0045.pdf>>. Acesso em: 07/2018.
- GUO, C., SONG, Q., CAI, W. Um sistema de controle em cascata assistido por rede neural para unidade de tratamento de ar. **IEEE Transactions on Industrial Electronics**. Vol.54. No.1. Fev. 2007.
- GRANATYR, J. laexpert. Home. Histórico da IA. IA simbólica x IA conexionista. Disponível em: <<http://iaexpert.com.br/index.php/2017/03/23/ia-simbolica-x-ia-conexionista/>>. Acesso em: 02/06/2018.
- HAGAN, M. T., DEMUTH, H. B. Redes neurais para controle. In: AMERICAN CONTROL CONFERENCE, 1999, USA. **Proceedings of the 1999**. Disponível em: <<http://bme2.aut.ac.ir/~towhidkhah/MPC/seminars-ppt/seminar%20MPC/hamidreza%20iranmanesh/REF-7.pdf>>. Acessado em: 19/05/2018.
- HAYKIN, S. **Redes neurais: princípios e prática**. 2 ed. Porto Alegre: Bookman, 2001.
- HAYKIN, S. **Neural networks and learning machines**. 3 ed. Porto Alegre: Prentice Hall: New Jersey, 2009.
- HYPERPHYSICS. DC motor operation. Disponível em: <<http://hyperphysics.phy-astr.gsu.edu/hbase/magnetic/motdc.html#c1>>. Acessado em: 10/07/2018.
- HOPFIELD, J. J.; TANK, D. W. Neural computation of decisions in optimization problems. **Biolog. Cybern.**, vol. 52, pp. 141-152, 1985.
- HOUFAR F.; SALAHSHOOR K.; PAIVAR A. Controle de linearização adaptativa neural para CSTR, usando o modelo NARMA-L2. Proceedings of the 5th WSEAS Int. Conf. on **Circuits, Systems, Electronics, Control & Signal Processing**, USA: Dallas. November 1-3, 2006.

KASSEM, A. M. Nonlinear Autoregressive Moving Average Controller for Isolated Wind Generation System Control. **Journal of Engineering Sciences**, Assiut University, Vol. 40, Nº 1, pp.209-222, janeiro 2012.

LI, Y. et al. CAutoCSD - Evolutionary search and optimisation enabled computer automated control system design, **Int J Automation and Computing**, vol. 1, No. 1, pp. 76-88. 2004. ISSN 1751-8520. Disponível em: <http://userweb.eng.gla.ac.uk/yun.li/ga_demo/>. Acesso: dez/2018.

LIMA, C. A. M. **Emprego de Teoria de Agentes no Desenvolvimento de Dispositivos Neurocomputacionais Híbridos e Aplicação ao Controle e Identificação de Sistemas Dinâmicos**. Dissertação de Mestrado, Faculdade de Engenharia Elétrica e de Computação, Unicamp, 2000.

LEAL, A. J. F. et al. Redes neurais artificiais na predição da produtividade de milho e definição de sítios de manejo diferenciado por meio de atributos do solo. **Bragantia**. Vol. 74. No. 4. Pg. 436-444. Campinas. 19 jun. 2015. Disponível em: <http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0006-87052015000400436&lng=pt&nrm=iso&tlng=en>. Acessado em : 25 maio 2016.

MESSNER, B.; TILBURY, D. CONTROL TUTORIALS FOR MATLAB & SIMULINK. Suspension. Modeling Simulink. Disponível em: <<http://ctms.engin.umich.edu/CTMS/index.php?example=Suspension§ion=SimulinkModeling>>. Acesso em: 08/2018.

NARENDRA, K. S., ANNASWAMY, A. M. **Stable Adaptive Systems**. Englewood Cliffs. NJ: Prentice-Hall, 1989.

NARENDRA, K. S.; PARTHASARATHY, K. Identificação e controle de sistemas dinâmicos usando redes neurais. **IEEE Transactions On Neural Networks**. Vol. 1. No.1. Março, 1990. Acesso em: 17/05/2018.

NISE, N. S. **Engenharia de Sistemas de Controle**. 3 ed. Rio de Janeiro: LTC, 2000.

NGUYEN, D., WIDROW, B. The truck backer-upper: Na example of self-learning in neural networks. **International Joint Conference on neural networks**. Vol.11. Washington, DC. 1989.

OGATA, K. **Engenharia de controle moderno**. 5 ed. São Paulo: Pearson Prentice Hall, 2010.

OMATU, S., YOSHIOKA, M., KOSAKA, T., YANAGIMOTO, H., DARGHAM, J.A. Neuro-Pid controle de velocidade e torque de um veículo elétrico. **International Journal on Advances in Systems and Measurements**. Vol.3. No 1 e 2, 2010. Acessado em: 22/05/2018.

PINEDA, F. J. Generalization of back propagation to recurrent networks. **Phys. Rev. Lett.**, vol. 59, no. 19, pp. 2229-2232, Nov. 1987.

SAERENS, M., SOQUET, A. Neural controller based on back-propagation algorithm. **IEE Proceedings (London)**. Part F, vol. 138. 1991.

SANTIAGO, R. C. MOYA, N. **Manual de Anatomia Patológica General**. 4 ed. 1905. 560 pg.

SEGATTO, Ê. C., COURY, D. V. Redes neurais artificiais recorrentes aplicadas na correção de sinais distorcidos pela saturação de transformadores de corrente. **Revista Controle & Automação**. Vol. 17. No. 4. Out. 2006. p. 428. Disponível em: <<http://www.scielo.br/pdf/ca/v17n4/a04v17n4>>. Acessado em: 27 maio 2016.

SIMÕES, M. G.; SHAW, I. S. **Controle e modelagem fuzzy**. 2. ed., rev. ampl. São Paulo: Blücher; FAPESP, 2007. xiv, 186 p. ISBN 9788521204169 (broch.)

TEIXEIRA, H. T. **Aprendizado por reforço e programação dinâmica aproximada com máquinas kernel para controle de sistemas não lineares**. Dissertação. Campinas – SP. 2016.

VALENTIM, N. S. da. **Controle de suspensão ativa automotiva por lógica difusa (“fuzzy logic”)**. 2017. 137 f. Dissertação (Mestrado em Processos Tecnológicos e Ambientais) – Universidade de Sorocaba, São Paulo, 2017.

WERBOS, P. J. **Handbook of Intelligent Control: Neural, Fuzzy, and Adaptive Approaches**. Approximate dynamic programming for real-time control and neural modelling. 1992. Disponível em: <<http://www.werbos.com/HICChapter13.pdf>>. Acesso: 06/2013.

WIDROW, B., HOFF, M. E. Adaptive switching circuits. **IRE WESCON Convention Record**. pg. 96-104. 1960.

YADAV, M. K., CHANDRA, D. Controle de Sistemas de Ar Condicionado Usando Rede Neural. **International Journal of Electronics and Electrical Engineering**. Vol. 3, No. 5, Out. 2015.

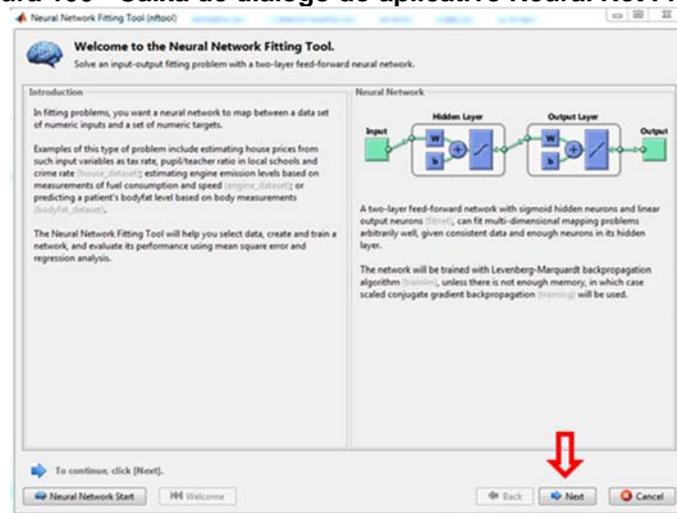
ZAMBRANO, W. C. A. **Controle Ativo de Vibrações usando Redes Neurais Artificiais**. Dissertação. Campinas – SP. 2013.

APÊNDICE A – TREINAMENTO DAS REDES MULTICAMADAS *FEED-FOWARD* VIA SIMULINK

Agora é mostrado o processo de treinamento da rede MLP utilizando um dos aplicativos interativos (APPS) do matlab, chamado “Neural Net Fitting”.

Ao iniciar o aplicativo uma caixa de diálogo é aberta conforme mostrado na figura 100. Clica-se no botão “Next” para à etapa de seleção dos dados de treinamento da rede.

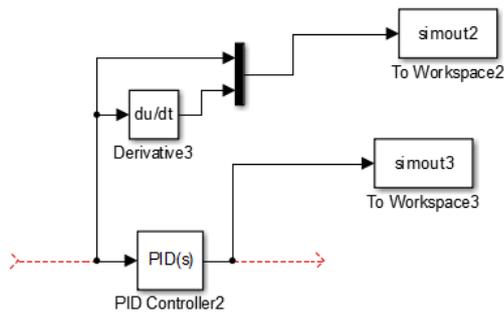
Figura 100 - Caixa de diálogo do aplicativo Neural Net Fitting.



Nota: Elaboração própria.

Na caixa de diálogo seguinte, seleciona-se um conjunto de dados de entrada e saída para o treinamento da rede MLP, que neste trabalho será o conjunto de dados de entrada e saída de um controlador PID (figura 102). É necessário antes realizar uma simulação dos sistemas controlados por um PID para exportar os dados de entrada e saída do controlador PID na forma de um vetor e assim utilizá-los como dados de treinamento. Para os dados de entrada são utilizados dois elementos: o erro proveniente da diferença entre o sinal de realimentação e a entrada do sistema; e a derivada do erro. E para a saída é utilizado apenas o sinal de atuação do PID sobre a planta do sistema, que corresponde a um elemento. Tem-se assim um conjunto de dados de entrada com dois elementos (erro e derivada do erro), e um conjunto de dados de saída com um elemento (sinal de atuação do PID), conforme mostrado na figura 101.

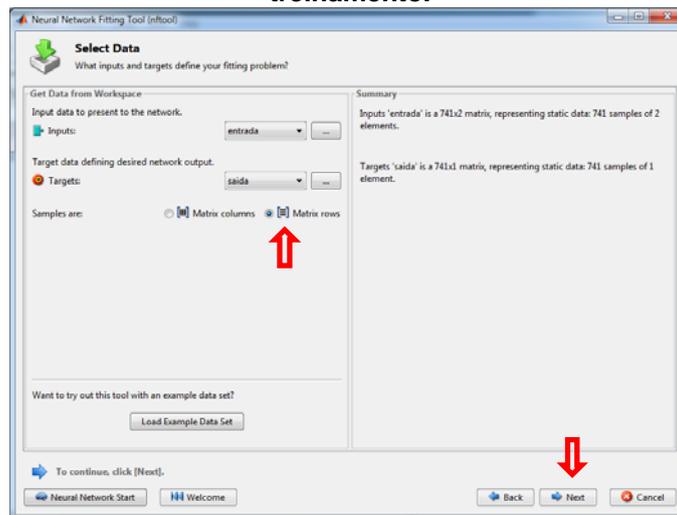
Figura 101 - Coleta de dados do PID para treinamento das redes MLP pelo método de aproximação.



Nota: Elaboração Própria.

Ao selecionar o conjunto de dados apropriado, clica-se em na opção Matrix linha (*Matrix rows*), pois o aplicativo interpreta cada coluna como um elemento e cada linha como uma amostra. Em seguida clica-se em “Next” para seguir a etapa de divisão dos dados em grupos de treinamento, validação e teste.

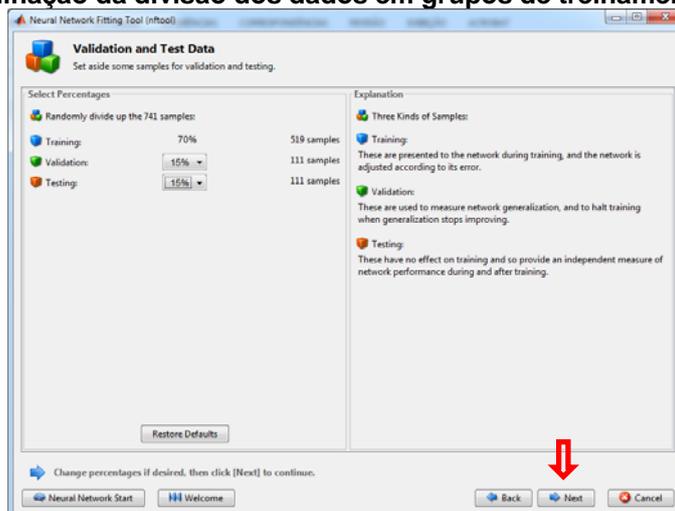
Figura 102 – caixa de diálogo do conjunto de dados de entrada e saída a serem utilizados no treinamento.



Nota: Elaboração própria.

Na etapa de divisão dos dados, pode-se determinar uma porção específica para cada grupo. Cada grupo de dados possui uma função para a rede: o grupo de treinamento é para o treinamento da rede; o grupo de validação indica quando o treinamento deve parar; e o grupo de teste fornece um teste independente que indica a capacidade de generalização da rede. Conforme mencionado anteriormente nesta sessão, os conjuntos são agrupados em 70% dos dados para treinamento, 15% para validação e 15% para teste (figura 103). Em seguida, clica-se em “Next” para seguir a etapa de configuração da arquitetura da rede.

Figura 103 – Determinação da divisão dos dados em grupos de treinamento, validação e teste.



Nota: Elaboração própria.

Para a configuração da arquitetura da rede costuma-se determinar o número de camadas ocultas da rede e o número de neurônios na camada oculta. No entanto, o aplicativo “Neural Net Fitting” permite somente a seleção do número de neurônios da camada oculta (figura 104), pois este aplicativo é limitado a uma rede do tipo feed-foward de duas camadas treinadas com o algoritmo de treinamento Levenberg-Marquardt.

Configura-se então a rede com 10 neurônios na camada oculta e em seguida clica-se em “Next” para proceder a etapa de treinamento.

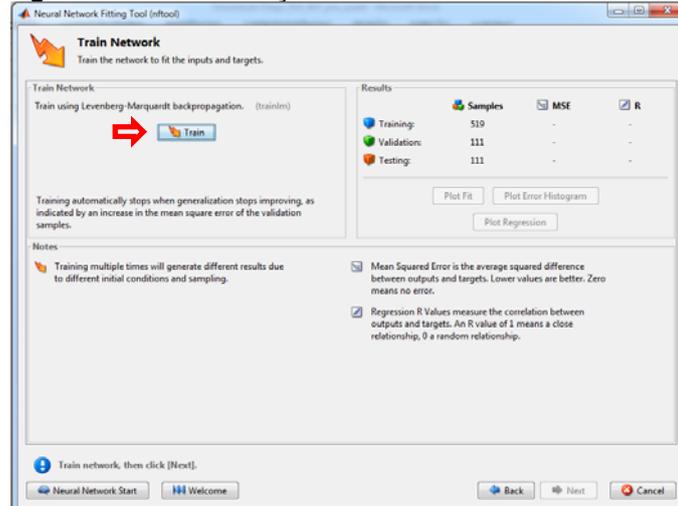
Figura 104 – Configuração do número de neurônios da camada oculta da rede.



Nota: Elaboração própria.

Após a etapa de configuração da rede o aplicativo gera uma rede do tipo *feed-foward* com as configurações selecionadas. É necessário agora treinar esta rede para que esteja apta a exercer o controle do sistema. Para isso, clica-se no botão “Train” na caixa de diálogo da figura 105.

Figura 105 - Inicialização do treinamento da rede MLP.



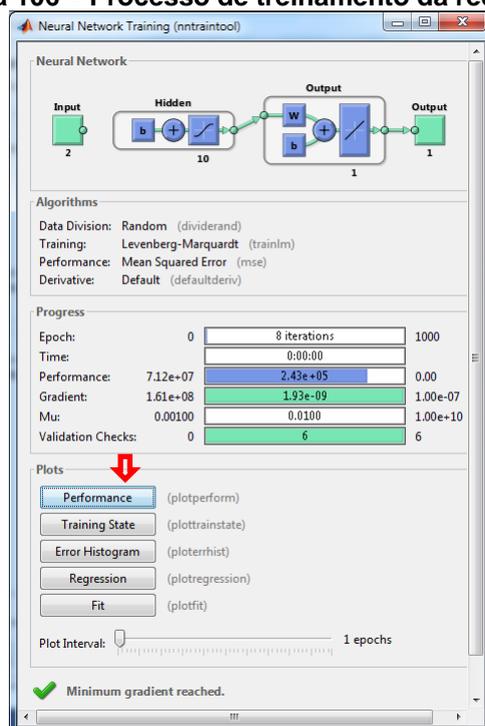
Nota: Elaboração própria.

Após inicializar o treinamento, uma nova tela é exibida a qual apresenta o processo de treinamento sendo realizado (figura 106). O treinamento é interrompido quando: o indicador *Epochs* (Épocas) atinge o valor 1000; quando o indicador a *Gradient* (gradiente) atinge um valor mínimo; ou quando o indicador *Validation checks* (verificação de validação) atinge o valor seis.

Ao término do treinamento, pode-se clicar no botão “Performance” para verificar o decaimento do erro durante o treinamento. Uma nova tela é exibida com o gráfico de decaimento do erro médio quadrático (figura 107). Caso seja necessário, pode-se realizar um novo treinamento da rede seguindo para a tela de avaliação da rede (Figura 108). Nessa tela pode-se treinar a rede novamente ou testá-la com dados adicionais.

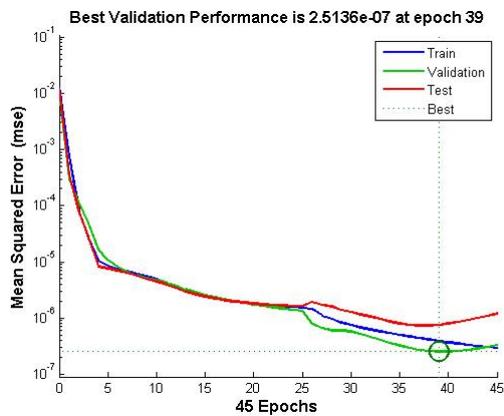
Por fim, segue-se para a tela “Implantação de solução”, onde pode ser obtido o a rede neural treinada na forma de um bloco. Clica-se em “Simulink Diagram” e um modelo em bloco da rede neural é gerado (figura 109). Esse modelo de rede neural pode ser implantado no diagrama de blocos do sistema substituindo o lugar do PID. É importante ressaltar que ao substituir o controlador deve-se aplicar na entrada da rede o mesmo número de variáveis com que ela fora treinada.

Figura 106 – Processo de treinamento da rede MLP



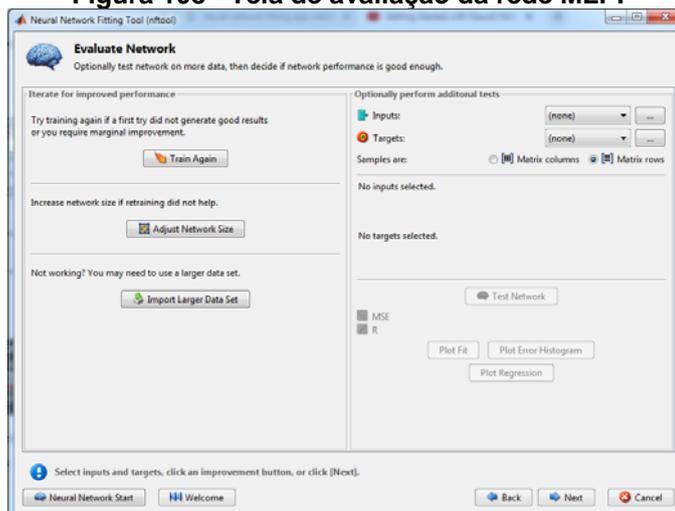
Nota: Elaboração própria.

Figura 107 - Gráfico de decaimento do Erro Médio Quadrático.



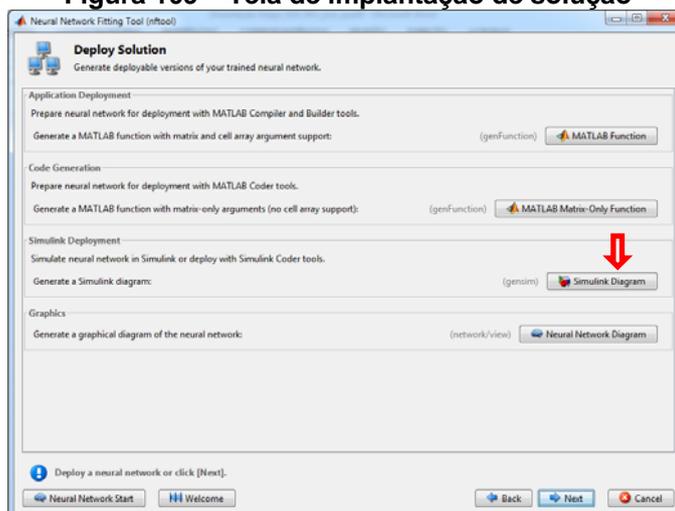
Nota: Elaboração própria.

Figura 108 - Tela de avaliação da rede MLP.



Nota: Elaboração própria.

Figura 109 – Tela de implantação de solução



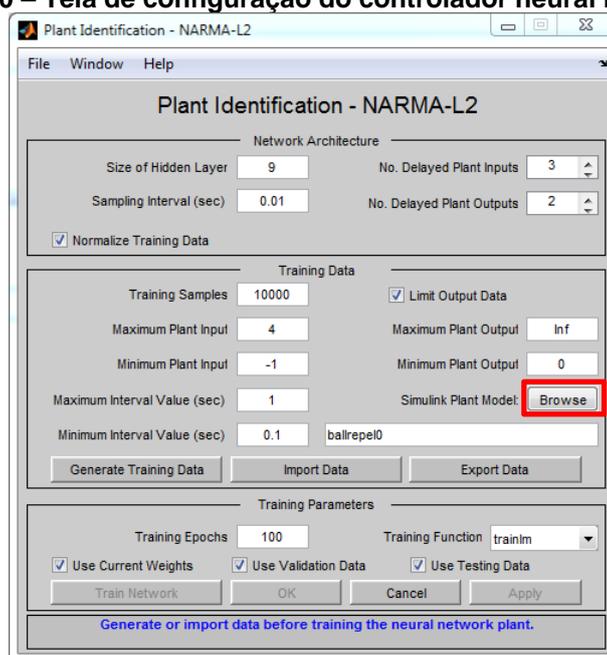
Nota: Elaboração própria.

APÊNDICE B – TREINAMENTO DO CONTROLADOR NARMA-L2 VIA SIMULINK

Aqui é apresentado o processo de treinamento de um controlador neural do tipo NARMA-L2. Para isso é utilizado o controlador neural NARMA-L2 da biblioteca “Neural Network Toolbox” na categoria “Control Systems” do matlab.

Primeiro deve-se clicar no bloco do controlador neural para abrir a caixa de diálogo apresentada na figura 110.

Figura 110 – Tela de configuração do controlador neural NARMA-L2.



Nota: Elaboração própria.

Esta é a tela inicial de configuração do controlador neural. Nessa tela podem ser configurados os seguintes parâmetros:

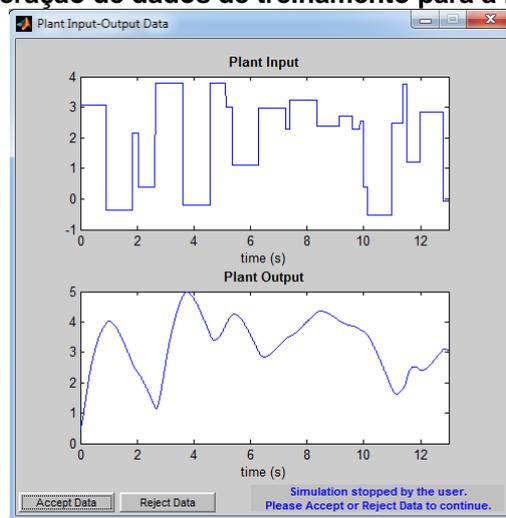
- Arquitetura da rede
 - Neurônios na camada oculta (*Size of Hidden Layer*);
 - Intervalo de amostragem em segundos (*Sampling Interval*);
 - Número de atrasos no sinal de entrada da planta (*No. Delayed Plant Inputs*);
 - Número de atraso no sinal de saída da planta (*No. Delayed Plant Outputs*);
 - Caixa de seleção para Normalizar dados de treinamento (*Normalize Training Data*)
- Dados de treinamento
 - Número de amostras de treinamento (*Training Samples*);
 - Valor máximo do sinal de entrada da planta (*Maximum Plant Input*);
 - Valor mínimo do sinal de entrada da planta (*Minimum Plant Input*);
 - Intervalo máximo de constância do sinal gerado (*Maximum Interval Value*);
 - Intervalo mínimo de constância do sinal gerado (*Minimum Interval Value*);

- Caixa de seleção para Limitar Dados de saída (*Limit Output Data*);
- Valor máximo do sinal de saída da planta (*Maximum Plant Output*);
- Valor mínimo do sinal de saída da planta (*Minimum Plant Output*);
- Parâmetros de treinamento
 - Épocas de treinamento (*Training Epochs*);
 - Algoritmo de treinamento (*Training Function*);
 - Caixa de seleção para usar os pesos atuais (*Use current weights*);
 - Caixa de seleção para usar dados de validação (*Use validation data*);
 - Caixa de seleção para usar dados de teste (*Use Testing data*);

Antes de inicializar o treinamento deve-se primeiro inserir um modelo de planta a ser controlado. Para isso clica-se no botão “Browse” conforme mostrado na figura 110. O modelo de planta a ser utilizado neste trabalho são os diagramas de blocos dos sistemas não controlados: antena de azimute; motor de tração de veículo de ferrovia; e suspensão ativa $\frac{1}{4}$.

Após selecionar um modelo de planta, deve-se gerar um conjunto de dados de treinamento para a rede. Ao clicar-se no botão “Generate Training data”, um conjunto de dados é gerado, conforme mostrado na figura 111.

Figura 111 – Geração de dados de treinamento para a rede NARMA-L2.



Nota: Elaboração própria.

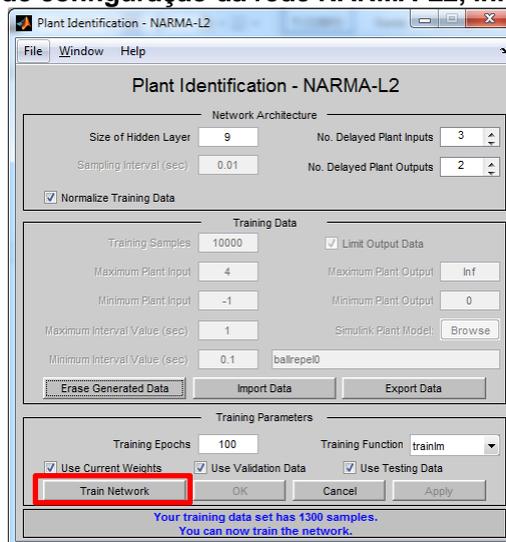
Pode-se gerar um novo conjunto de dados caso seja necessário. Para isso, rejeita-se o conjunto de dados atual pelo botão “Reject Data” e repete-se o processo de geração de dados.

Em seguida inicia-se o treinamento da rede com o conjunto de dados gerados. Para isso, clica-se no botão “Train Network” (figura 112). Durante o treinamento, uma tela exibe o desempenho da rede (figura 113). O treinamento é interrompido quando a rede atingir o número máximo de épocas (*Epochs*), quando atingir o número máximo de Verificação de

validações (Validation checks), ou atingir o gradiente mínimo. Ao término do treinamento três telas exibem a capacidade de generalização da rede em relação a cada grupo de dados apresentado (treinamento, teste e validação). Por exemplo, a figura 114 mostra o desempenho da rede em relação aos dados de treinamento.

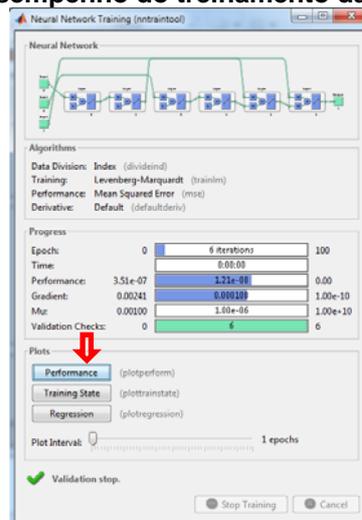
Para analisar o decaimento do erro médio quadrático clica-se no botão performance na tela da figura 113. A figura 115 apresenta o decaimento do erro médio quadrático após o treinamento da rede.

Figura 112 – Tela de configuração da rede NARMA-L2, iniciando treinamento



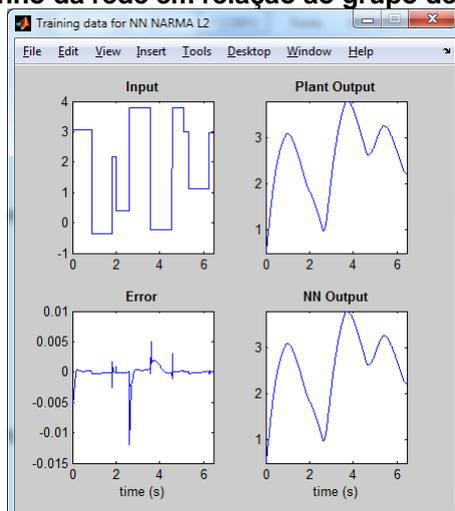
Nota: Elaboração própria.

Figura 113 – Desempenho do treinamento da rede NARMA-L2.



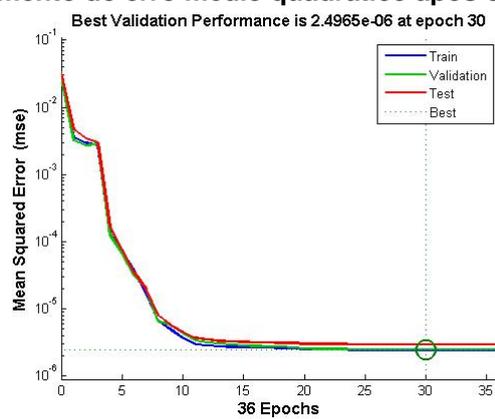
Nota: Elaboração própria.

Figura 114 - Desempenho da rede em relação ao grupo de dados de treinamento.



Nota: Elaboração própria.

Figura 115 - Decaimento do erro médio quadrático após o treinamento da rede.



Nota: Elaboração própria.

APÊNDICE C – TREINAMENTO DO CONTROLADOR NEURAL MODELO DE REFERÊNCIA

Nesta sessão é apresentado o processo de treinamento de um controlador neural do tipo Modelo de Referência. Para isso é utilizado o controlador neural *Model reference* da biblioteca “Neural Network Toolbox” na categoria “Control Systems” do matlab. Esse tipo de neurocontrolador utiliza duas redes para exercer o controle de um sistema. O treinamento dessas redes funciona de maneira independente. Primeiro deve-se treinar a rede de modelo, a qual irá identificar um modelo de sistema adotado pelo usuário. Em seguida treina-se a rede de controle, a qual é responsável por gerar o sinal de controle.

Agora é descrito uma sequência de passos para realizar o treinamento dessas redes tomando como referência as figuras 116, 117 e 118, que apresenta as janelas de configuração de cada rede:

- 1 – Abre-se a janela de configuração da rede controladora (figura 116);
- 2 – Abre-se a janela de configuração da rede identificadora ao clicar no botão *Plant identification* (Identificação da planta);
- 3 – Inserir um modelo que represente a dinâmica do sistema a ser controlado, pelo botão *Browse*¹⁶ (figura 117);
- 4 – Inserir os parâmetros de configuração da rede identificadora¹⁷;
- 5 – Desabilitar ou habilitar as caixas de seleção *Use current weights*¹⁸, *Use Validation Data* e *Use Testing Data*¹⁹ conforme as especificações do usuário.
- 6 – Ao clicar no botão *Generate Training Data*, gera-se um conjunto de dados para o treinamento da rede de modelo. Finalizado o processo, pode-se aceitar ou rejeitar o conjunto de dados. Supõem-se aceitável o conjunto de dados para que se possa prosseguir com as etapas de treinamento (figura 118);
- 7 – Iniciar o treino da rede identificadora clicando no botão *Train network* (figura 117);
- 8 - Caso o desempenho da rede de modelo não for preciso, pode-se selecionar *Train network* novamente, o que continua o treinamento da rede com o mesmo conjunto de dados. Se for desejável usar um novo conjunto de dados para continuar o treinamento, deve-se selecionar novamente *Generate Training Data*²⁰ antes de selecionar *Train network*. Deve-se

¹⁶ Para este trabalho utiliza-se os modelos dos sistemas não controlados: antena de azimute, motor de tração e suspensão ativa ¼.

¹⁷ Para cada sistema estudado neste trabalho é realizado uma configuração específica dessas redes.

¹⁸ *Use current weights* implica em usar pesos já gerados previamente, caso seja necessário realizar outro treinamento com o mesmo conjunto de dados.

¹⁹ Essas opções é utilizada quando deseja-se separar parte do conjunto total de dados de treinamento para realizar a validação e teste dos dados.

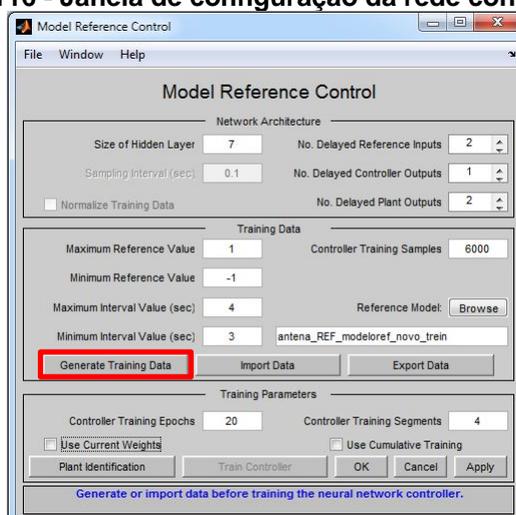
²⁰ Pode-se também importar um conjunto de dados pelo botão *Import Data*.

certificar de que a caixa de seleção *Use current weights* esteja selecionado se quiser continuar treinando com os mesmos pesos.

9 – Ao finalizar o treinamento da rede de modelo, deve-se prosseguir com o treinamento da rede controladora. Para isso realiza-se as etapas três a oito para a janela da rede controladora, lembrando que o modelo a ser inserido na etapa três deve ser um modelo de sistema controlado (figura 116);

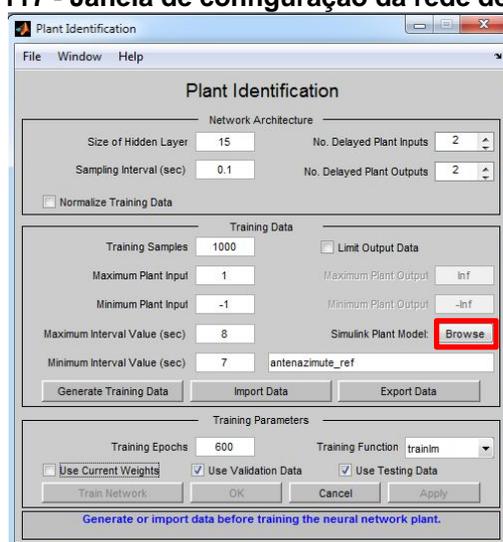
10 - Após realizar o treinamento da rede controladora, uma janela será aberta, mostrando o desempenho do treinamento (figura 119). Deve-se observar se a linha verde (que representa a rede controladora) está seguindo a azul (o modelo de sistema controlado fornecido pelo usuário). Isto demonstra se a rede foi capaz seguir o modelo de referência fornecido pelo usuário.

Figura 116 - Janela de configuração da rede controladora



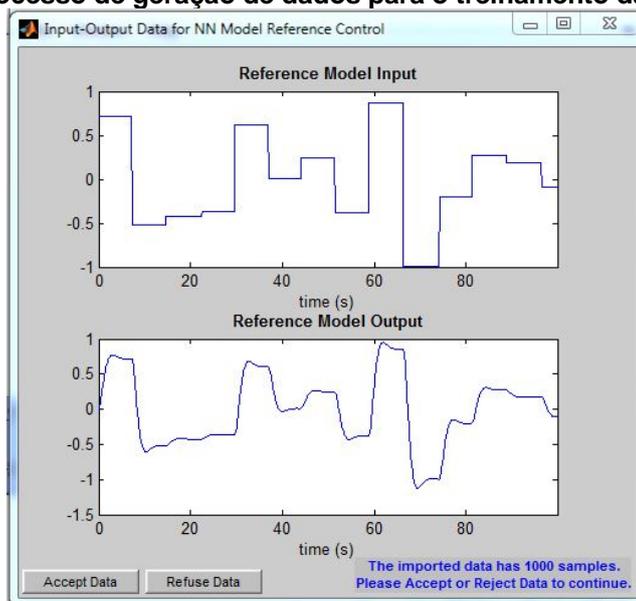
Nota: Elaboração própria.

Figura 117 - Janela de configuração da rede de modelo



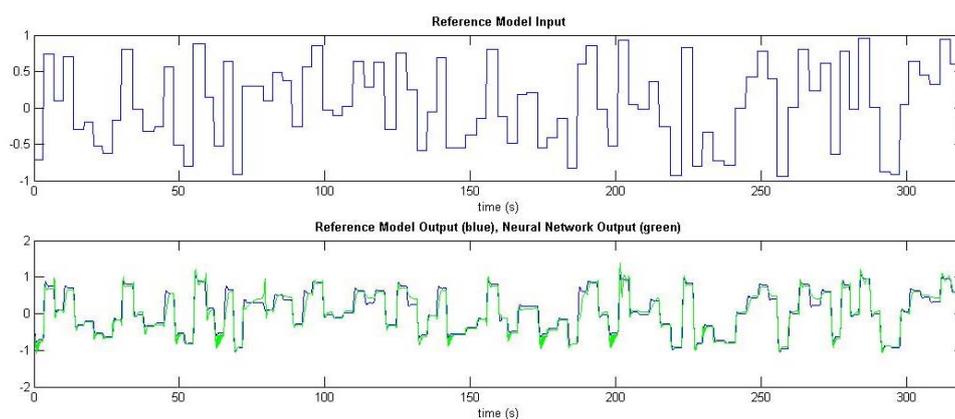
Nota: Elaboração própria.

Figura 118 – Processo de geração de dados para o treinamento da rede de modelo



Nota: Elaboração própria.

Figura 119 – Desempenho do treinamento da rede controladora



Nota: Elaboração própria.

APÊNCICE D – SINTONIA DO CONTROLADOR PID NO SIMULINK

Existem muitos métodos disponíveis para se determinar valores aceitáveis dos ganhos PID. Dentre eles, pode-se citar o método de sintonia manual, nos quais os ganhos do controlador são obtidos por tentativa e erro com mínima análise analítica usando respostas ao degrau obtidas via simulação, e o método de Ziegler-Nichols, um método mais analítico baseado em respostas em malha aberta ou fechada a uma entrada em degrau. (DORF e BISHOP, 2013)

Neste trabalho será utilizado o método de sintonia por modelo linearizado, presente nos blocos PID do Simulink do software de simulação gráfica Matlab.

Os blocos de Controlador PID do Simulink possuem um sintonizador que fornece um método rápido de sintonia de PID de malha única. Com este método, você pode ajustar os parâmetros do controlador PID para obter um controle robusto com o tempo de resposta desejado.

A seguir está uma sequência típica de etapas para realizar a sintonização do controlador PID pelo Simulink.

(1) Iniciar o sintonizador PID. Ao iniciar, o software calcula automaticamente um modelo de planta linear do modelo Simulink e projeta um controlador inicial.

(2) Ajuste o controlador no Sintonizador PID pelo ajuste manual de dois parâmetros. O sintonizador então calcula os parâmetros PID que estabilizam o sistema de maneira robusta.

(3) Exporte os parâmetros do controlador projetado de volta para o bloco do Controlador PID e verifique o desempenho do controlador no Simulink.

As etapas desse processo de sintonização são explicadas em mais detalhes a seguir.

Para iniciar o Sintonizador PID, clica-se duas vezes no bloco Controlador PID para abrir sua caixa de diálogo. Aqui, não é necessário alterar nenhuma configuração, no entanto é importante observar que para as simulações deste trabalho, o tipo de controlador PID utilizado é paralelo (*Parallel*) de tempo contínuo com quatro parâmetros:

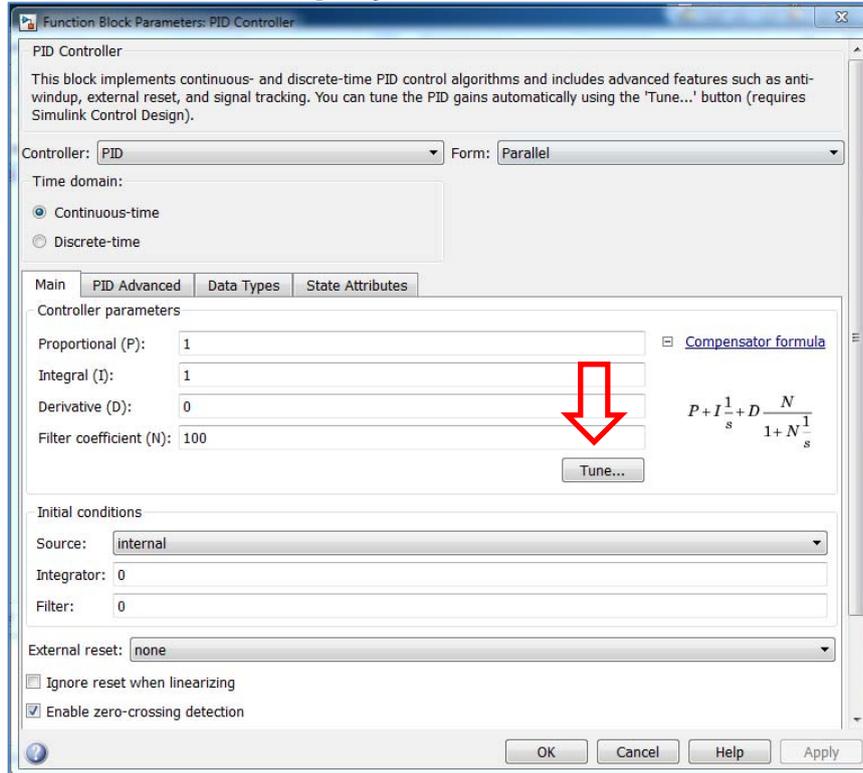
- Ganho proporcional (P);
- Ganho integrativo (I);
- Ganho derivativo (D);
- Coeficiente de filtro (N);

A FT que representa esta classe de controlador é dada por:

$$P + I \frac{1}{s} + D \frac{N}{1 + N \frac{1}{s}} \quad (32)$$

Na guia principal, clica-se em “Tune...” (*Sintonizar*) conforme mostrado na figura 120.

Figura 120 - Janela de configuração do controlador PID do software matlab.



Nota: Elaboração própria.

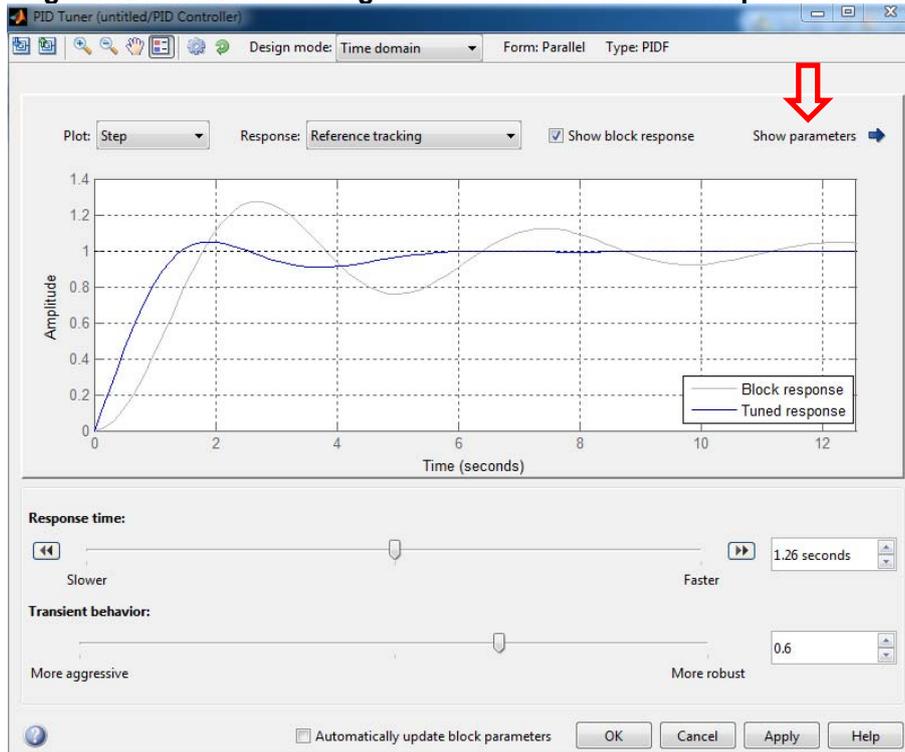
Quando o Sintonizador PID é iniciado, o software calcula um modelo de planta linearizado visto pelo controlador. O software identifica automaticamente a entrada e a saída da planta e usa o ponto operacional atual para a linearização²¹. A planta pode ter qualquer ordem e pode ter atrasos de tempo.

O Sintonizador PID calcula um controlador PID inicial para obter uma proporção razoável entre desempenho e robustez. Por padrão, o desempenho inicial a uma entrada em degrau é exibido na plotagem.

A figura 121 mostra a caixa de diálogo do bloco PID com o desempenho inicial:

²¹ Técnica de aproximação em torno de um ponto de operação.

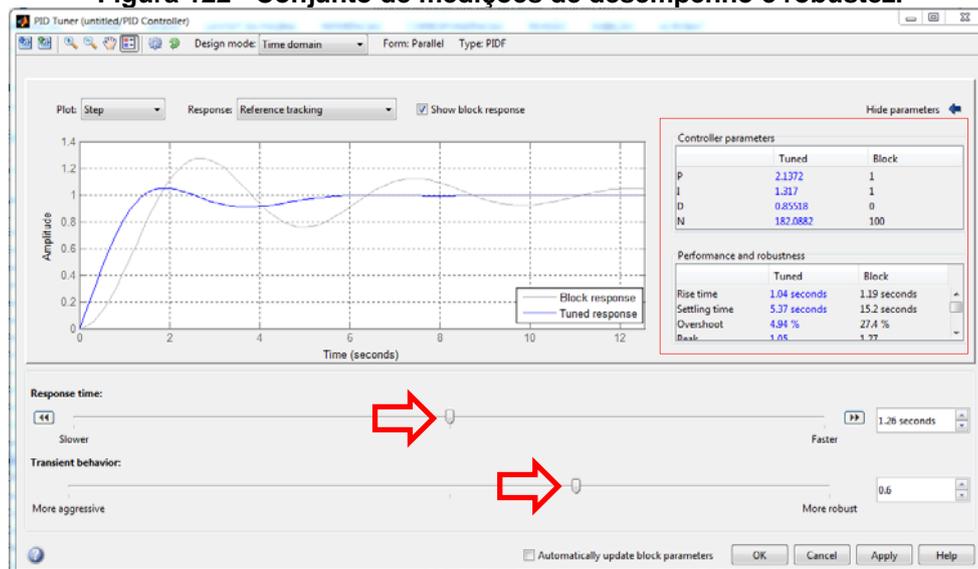
Figura 121 - Caixa de diálogo do bloco PID com o desempenho inicial



Nota: Elaboração própria.

Clica-se em “Show parameters” (*Mostrar parâmetros*) para visualizar os parâmetros P, I, D e N do controlador e um conjunto de medições de desempenho e robustez mostrados na figura 122. Neste exemplo, o projeto inicial do controlador PID fornece um tempo de acomodação de 5,37 segundos e um sobressinal de aproximadamente 5%.

Figura 122 - Conjunto de medições de desempenho e robustez.



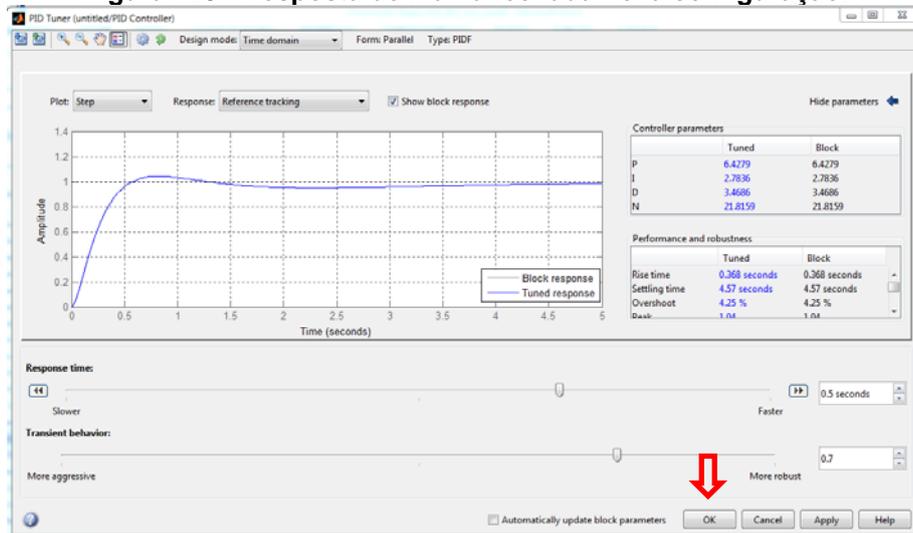
Nota: Elaboração própria.

Para reduzir o sobressinal e o tempo de estabilização, é necessário utilizar os dois controles deslizantes mostrados na figura 122. Com esses controles deslizantes pode-se tornar a resposta de controle mais rápida para reduzir o tempo de estabilização e aumentar a

robustez para reduzir o sobressinal. Por exemplo, pode-se reduzir o tempo de resposta (*Response Time*) de 1,26 para 0,5 segundos e aumentar a robustez (*Transient Behavior*) de 0,6 para 0,7.

A figura 123 mostra a resposta de malha fechada com essas configurações:

Figura 123 - Resposta de malha fechada nova configuração.

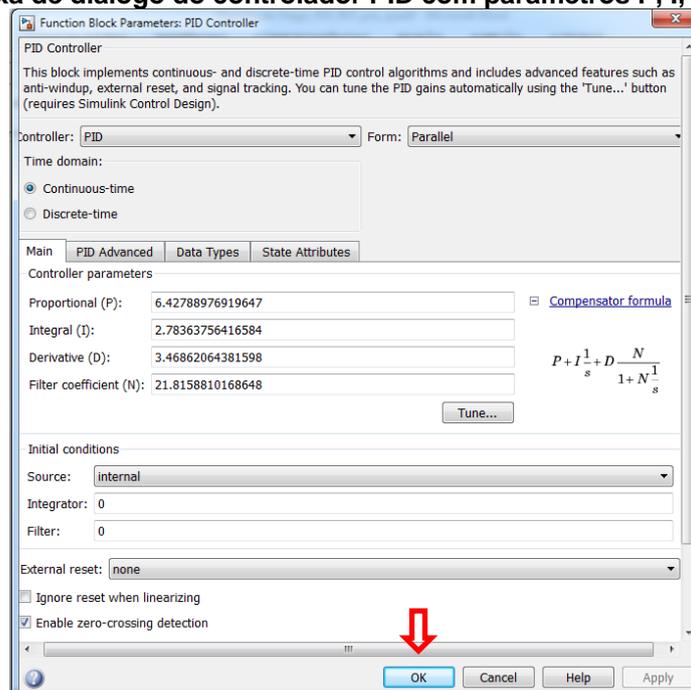


Nota: Elaboração própria.

Encontrado um desempenho satisfatório do controlador no modelo de planta linear, pode-se então testá-lo no modelo de planta do sistema a ser controlado. Para fazer isso, clica-se em “OK” na caixa de diálogo da figura 123. Esta ação grava os parâmetros de volta ao bloco do Controlador PID no modelo Simulink.

A figura 124 mostra a caixa de diálogo atualizada do bloco do Controlador PID:

Figura 124 - Caixa de diálogo do controlador PID com parâmetros P, I, D e N atualizados.



Nota: Elaboração própria.

Para finalizar o processo de sintonização, clica-se em “OK” na caixa e diálogo mostrada na figura 124 e o controlador PID já estará sintonizado.