

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
CÂMPUS PONTA GROSSA
DEPARTAMENTO DE ELETRÔNICA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA
MESTRADO EM ENGENHARIA ELÉTRICA**

EDUARDO MOLETTA

**REDES NEURAIS COM ESTADOS DE ECO APLICADAS EM CONTROLE
DEPENDENTE DOS ESTADOS**

DISSERTAÇÃO

**PONTA GROSSA
2015**

EDUARDO MOLETTA

**REDES NEURAIIS COM ESTADOS DE ECO APLICADAS EM CONTROLE
DEPENDENTE DOS ESTADOS**

Dissertação de mestrado apresentada ao Programa de Pós-Graduação em Engenharia Elétrica do Departamento de Eletrônica do Câmpus Ponta Grossa da UTFPR como requisito parcial para a obtenção do título de mestre em Engenharia Elétrica.

Orientador: Prof. Dr. José Carlos Alberto de Pontes

Coorientador: Prof. Dr. Marcos Cesar Vergès

**PONTA GROSSA
2015**

Ficha catalográfica elaborada pelo Departamento de Biblioteca
da Universidade Tecnológica Federal do Paraná, Câmpus Ponta Grossa
n.50/15

M719 Moletta, Eduardo

Redes neurais com estados de eco aplicadas em controle dependente dos estados/ Eduardo Moletta – Ponta Grossa, 2015.

83 f.: il. ; 30 cm

Orientador: Prof. Dr. José Carlos Alberto de Pontes

Coorientador: Prof. Dr. Marcos Cesar Vergès

Dissertação (Mestrado em Engenharia Elétrica). Programa de Pós-Graduação em Engenharia Elétrica. Universidade Tecnológica Federal do Paraná, Ponta Grossa, 2015.

1. Redes neurais (Computação). 2. Teoria de controle. 3. Engenharia elétrica. I. Pontes, José Carlos de. II. Vergès, Marcos Cesar. III. Universidade Tecnológica Federal do Paraná. IV. Título.

CDD 621.3



Universidade Tecnológica Federal do Paraná
Câmpus Ponta Grossa
Diretoria de Pesquisa e Pós-Graduação
PROGRAMA DE PÓS GRADUAÇÃO EM
ENGENHARIA ELÉTRICA

FOLHA DE APROVAÇÃO

Título da Dissertação Nº 8/2015

REDES NEURAIS COM ESTADOS DE ECO APLICADAS EM CONTROLE DEPENDENTE DOS ESTADOS

por

Eduardo Moletta

Esta dissertação foi apresentada às **9 horas** do dia **6 de fevereiro de 2015** como requisito parcial para a obtenção do título de MESTRE EM ENGENHARIA ELÉTRICA, com área de concentração em Controle e Processamento de Energia, Programa de Pós-Graduação em Engenharia Elétrica. O candidato foi arguido pela Banca Examinadora composta pelos professores abaixo assinados. Após deliberação, a Banca Examinadora considerou o trabalho aprovado.

Prof. Dr. Levy Boccato (UNICAMP)

Prof. Dr. Angelo Marcelo Tuset (UTFPR)

Prof. Dr. Hugo Valarares Siqueira (UTFPR)

**Prof. Dr. José Carlos Alberto de Pontes
(UTFPR)**
Orientador

Visto do Coordenador:

Prof. Dr. Claudinor Bitencourt Nascimento (UTFPR)
Coordenador do PPGEE

AGRADECIMENTOS

Agradeço especialmente à minha família.

Agradeço ao meu orientador José Carlos Pontes pela orientação e dedicação em todos os trabalhos.

Agradeço aos meus sócios Ciro e Nelson por me concederem a oportunidade de participar do programa.

Agradeço à Dra. Lucia Maria pelas correções.

Agradeço aos membros da banca pelas considerações de grande importância para este trabalho.

Também agradeço à Universidade Tecnológica Federal do Paraná, por disponibilizar os seus laboratórios e ambientes de ensino.

RESUMO

MOLETTA, Eduardo. **REDES NEURAIS COM ESTADOS DE ECO APLICADAS EM CONTROLE DEPENDENTE DOS ESTADOS**. 2015. 115 f. Dissertação – Programa de Pós-Graduação em Engenharia Elétrica, Universidade Tecnológica Federal do Paraná. Ponta Grossa, 2015.

Por volta de 1764 aparece um novo ramo da ciência - A teoria de controle - quando James Watt consertou uma máquina Newcomen e percebeu que essa era ineficiente, e criou um sistema de controle de velocidades. A evolução destes sistemas controladores pode ser observada no controle utilizando a equação de Riccati dependente de estados (SDRE). Apesar de ser uma técnica muito avançada em relação à capacidade de realizar o controle, alguns problemas precisam ser encarados quanto à sua utilização, como a necessidade de se ter recursos computacionais de alto nível e custo. Essas questões podem impedir o uso da técnica SDRE em alguns sistemas. Uma solução para este problema é apontada através do uso de uma rede neural (RNA) chamada de Rede Neural com Estados de Eco (ESN). As RNAs possuem arquiteturas baseadas em redes neurais biológicas para que tenhamos resultados desejados na saída. Para que essa saída seja satisfatória a rede neural passa por um processo de treinamento. Sendo assim, usa-se os dados de comportamento do SDRE para a realização do treinamento da ESN. Depois disso, realizam-se testes quanto à eficiência da rede neural no controle do sistema a ser controlado e ao custo computacional. Os resultados são comparados aos obtidos com o controle ESN. Este teste foi realizado para um sistema micro eletromecânico e o controle da suspensão ativa de um *half-car*. Os resultados obtidos foram positivos, pois a ESN conseguiu realizar o controle utilizando menos tempo de processamento em relação ao SDRE, além de possuir uma estrutura base fixa, possibilitando ajustes para realização de diferentes tipos de controle.

Palavras-chave: Redes neurais recorrentes. Echo state network. SDRE.

ABSTRACT

MOLETTA, Eduardo. **REDES NEURAIS COM ESTADOS DE ECO APLICADAS EM CONTROLE DEPENDENTE DOS ESTADOS**. 2015. 115 s. Masters Thesis – Post-Graduation Program in Electrical Engineering, Federal University of Technology, Paraná. Ponta Grossa, 2015. In around 1764, it emerged a new branch of science – the theory of control - when James Watt was given a model Newcomen engine to repair. He realised that it was hopelessly inefficient and began to work to improve the design. He did a velocity controller to solve the problem. The evolution of these systems is shown in State-dependent Riccati equation (SDRE) techniques. Although it is a very advanced technique in relation to the capacity of performing control, some problems have to be faced for its use, as the necessity of computational resources of high level and cost, which may impede the use of SDRE in some systems. The solution for these problems is pointed out in this study by the use of Echo State Neural Networks (ESNs). These neural networks have inputs and outputs and the inputs are processed through the use of algorithms in order to reach the desired results, and for that the neural network has to be under a task of training. After that we use the behavioral SDRE data for the training followed by the neural network efficiency test for the system control and for the computational cost. The results are compared to the ones obtained with the ESN control. This test was realized for a micro electromechanical system and the control of the active suspension of a half-car. The results were positive as the ESN could perform the control in a short time in relation to the SDRE. There is also a fixed structure which makes possible some adjustments for different kinds of control.

Keywords: Recurrent neural network. Echo State Network. Control with neural network.

LISTA DE FIGURAS

Figura 1 – Malha fechada (Autoria própria)	18
Figura 2 – Malha aberta (Autoria própria)	18
Figura 3 – Diagrama de blocos controlador PID (Autoria própria)	20
Figura 4 – Diagrama de Blocos em Malha Aberta (Autoria própria)	21
Figura 5 – Diagrama de Blocos do Controle por Realimentação de estados (Autoria própria)	22
Figura 6 – Fluxograma SDRE (Autoria própria)	32
Figura 7 – Representação do sistema nervoso (Autoria própria)	35
Figura 8 – Neurônio Biológico (MACHADO, 2006)	36
Figura 9 – Neurônio Artificial (SIQUEIRA, 2013)	36
Figura 10 – Função tangente hiperbólica (Autoria própria)	38
Figura 11 – Rede neural MLP (<i>multilayer perceptron</i>) (SIQUEIRA, 2013)	39
Figura 12 – Echo State Network (SIQUEIRA, 2013)	43
Figura 13 – Fluxograma para realização da simulação (Autoria própria)	49
Figura 14 – Sistema microeletromecânico (BASSINELLO, 2011)	51
Figura 15 – Deslocamento de u_1 para o sistema microeletromecânico com controle (SDRE) (Autoria própria)	54
Figura 16 – Erro de posição SDRE (Autoria própria)	55
Figura 17 – Erro de velocidade SDRE (Autoria própria)	55
Figura 18 – Atuação SDRE Sistema MEMs (Autoria própria)	55
Figura 19 – Teste de atuação da rede neural comparando com o SDRE, usando 60 neurônios (Autoria própria)	57
Figura 20 – Teste de atuação da rede neural comparando com o SDRE, usando 60 neurônios aproximado (Autoria própria)	57
Figura 21 – Comparativo controle SDRE e ESN (Autoria própria)	58
Figura 22 – Comparativo controle SDRE e ESN aproximado (Autoria própria)	59
Figura 23 – Histograma de erros do controlador SDRE e ESN no sistema MEMs demonstrando a igualdade do comportamento da ESN em relação ao SDRE (Autoria própria)	59
Figura 24 – Tempos para cálculo de atuação para SDRE e ESN com 60 neurônios no reservatório dinâmico sistema MEMs (Autoria própria)	60
Figura 25 – Histograma Tempos SDRE x ESN sistema MEMs 60 neurônios (Autoria própria)	61
Figura 26 – Histograma MSEs (Autoria própria)	62
Figura 27 – Erro com distúrbio de 5% na entrada do sistema MEMs (Autoria própria)	63
Figura 28 – Erro com distúrbio de 5% na entrada do sistema MEMs zoom (Autoria própria)	64
Figura 29 – Modelo de meio carro (TUSSET et al., 2012b)	66
Figura 30 – Estados SDRE do sistema 2 (Autoria própria)	70
Figura 31 – Entradas da rede neural para treinamento (Autoria própria)	71
Figura 32 – Atuação SDRE sistema 2 (Autoria própria)	72
Figura 33 – Teste resposta da rede neural em relação ao SDRE sistema <i>half-car</i> amortecedor dianteiro (Autoria própria)	73
Figura 34 – Teste resposta da rede neural em relação ao SDRE sistema <i>half-car</i> amortecedor traseiro (Autoria própria)	73
Figura 35 – Estados do sistema <i>half-car</i> com atuação ESN (Autoria própria)	74

Figura 36 – Tempos para cálculo de atuação para SDRE e ESN com 120 neurônios no reservatório dinâmico sistema <i>half-car</i> (Autoria própria)	74
Figura 37 – Histograma ESN x SDRE para controle do sistema <i>half-car</i> (Autoria própria) .	75
Figura 38 – Histogramas de erros da ESN em relação ao SDRE roda dianteira e roda traseira respectivamente (Autoria própria)	76
Figura 39 – Simulação com erro de leitura dos sensores de 5%, 2% e sem erro (Autoria própria)	76

LISTA DE TABELAS

Tabela 1 – Erros para SDRE e ESN com 60 neurônios sistema MEMs	61
Tabela 2 – Erros para SDRE e ESN com 60 neurônios sistema <i>Half-car</i>	75
Tabela 3 – Erros para SDRE e ESN com 60 neurônios com ruído sistema <i>Half-car</i>	77

LISTA DE SIGLAS

ESN	Rede com estados de eco (<i>Echo State Network</i>)
LSM	<i>Liquid State Machine</i>
RC	Computação de reservatório (<i>Reservoir Computing</i>)
MEMS	Sistema microeletromecânico
SDRE	Equação de Riccati dependente de estado (<i>State Dependent Riccati Equation</i>)
RMS	Raiz do valor quadrático médio (<i>Root Mean Square</i>)
DC	Tensão contínua
RNN	Rede Neural Recorrente (<i>recurrent neural network</i>)

SUMÁRIO

1	INTRODUÇÃO	13
2	SISTEMAS DE CONTROLE	16
2.1	INTRODUÇÃO E HISTÓRICO	16
2.2	DEFINIÇÕES	17
2.3	CLASSIFICAÇÃO DE SISTEMAS DE CONTROLE	17
2.3.1	Controle em Relação Ao Tipo de Malha (Sistemas em Malha Aberta e Malha Fechada)	18
2.4	CLASSIFICAÇÃO E TIPOS DE CONTROLE QUANTO À QUANTIDADE DE ENTRADAS E SAÍDAS	18
2.4.1	Controle Proporcional, Integral e Derivativo (PID)	19
2.4.2	Controle Por Estação de Estado	20
2.4.2.1	Alocação de polos	21
2.4.2.2	Regulador linear quadrático (LQR)	23
2.4.2.3	SDRE	25
3	REDES NEURAIS ARTIFICIAIS	33
3.1	HISTÓRICO	33
3.2	O FUNCIONAMENTO DO CÉREBRO BIOLÓGICO	34
3.3	NEURÔNIOS	35
3.3.1	Neurônio Biológico	35
3.3.2	Neurônio Artificial	36
3.4	ARQUITETURA DAS REDES NEURAIS ARTIFICIAIS	37
3.4.1	Redes Neurais <i>Feedforward</i> , a <i>MLP (multilayer perceptron)</i>	38
3.4.2	Redes Neurais Recorrentes	40
3.4.3	Treinamento de Redes Neurais	40
3.4.4	Rede Neural Com Estado de Eco	42
3.4.4.1	Propriedade de estados de eco	43
3.4.4.2	Reservatório dinâmico	44
3.4.4.3	Estado do reservatório de dinâmicas	44
3.4.4.4	Saída da rede	44
3.4.4.5	Determinação dos pesos \mathbf{W} e \mathbf{W}^{in}	45
3.4.4.6	Treinamento da ESN	45
3.4.4.7	Aplicações e desenvolvimento das ESNs	47
4	ESNS PARA O CONTROLE DO SISTEMA MICROELETROME CÂNICO (MEMS)	48
4.1	INTRODUÇÃO	48
4.2	SISTEMAS MICROELETROME CÂNICOS	50
4.2.1	Aquisição de Dados Para o Treinamento da Rede	54
4.2.2	Treinamento, Ajuste e Teste da Rede Neural	56
4.2.3	Atuação da Rede de Estado de Eco Diretamente na Planta do Sistema MEMs	58
4.2.4	Comparativo Entre a ESN e SDRE Quanto ao Tempo de Processamento para o sistema Microeletromecânico	59
4.2.5	Influência dos Números Aleatórios de \mathbf{W} e \mathbf{W}^{in}	61
4.2.6	Sensibilidades a Erros Paramétricos	61

5	USANDO CONTROLE SDRE PARA TREINAMENTO DA ESN E SEU CONTROLE DE UM SISTEMA DE MEIO CARRO (<i>HALF-CAR</i>)	65
5.1	SISTEMA <i>HALF-CAR</i> (Meio Carro).....	65
5.1.1	Aquisição de Dados Para o Treinamento da Rede	69
5.1.2	Treinamento, Ajuste e Teste da Rede Neural	70
5.1.3	Atuação da Rede de Estado de Eco Diretamente na Planta do Sistema <i>Half-car</i> ...	71
5.1.4	Comparativo Entre a ESN e SDRE Quanto ao Tempo de Processamento Para o Sistema <i>Half-car</i>	72
5.1.5	Sensibilidades a Erros Paramétricos	76
6	CONSIDERAÇÕES FINAIS	78
	REFERÊNCIAS	80

1 INTRODUÇÃO

Sistemas de controle surgiram no século XVIII principalmente a partir da necessidade do homem automatizar sistemas a fim de melhor desempenho, precisão e diminuição de mão de obra. Sua constante evolução fez com que surgissem várias técnicas para a produção desses controladores, os quais são sistemas matemáticos que, muitas vezes, necessitam de uma grande quantidade de cálculos para o seu funcionamento. Um exemplo desses controladores é o controlador SDRE, que utiliza a equação de Riccati dependentes de estados.. Essa técnica possui uma grande dificuldade de ser implementada e processada, e neste presente trabalho uma alternativa será demonstrada para que um controlador análogo ao SDRE possa ser utilizado mantendo a mesma eficiência.

A técnica de controle SDRE é muito eficiente e vem sendo usada em várias aplicações. Utiliza-se de uma grande quantidade de processamento computacional, pois é necessário refazer vários cálculos para obtenção da atualização do controle do sistema. Para a simulação em softwares matemáticos a técnica demonstra uma ótima capacidade de controlar sistemas não-lineares, porém para implementar os resultados numéricos em sistemas reais há limitações como: (1) aquisição dos estados através de sensores, (2) limitação de processamento, de forma que o recurso programável do hardware (processadores e microcontroladores, por exemplo) normalmente não é dimensionado para este tipo de aplicação. Isso torna a implementação em hardware mais complicada e com possíveis atrasos significativos nos cálculos dos sinais para controle. O exemplo pode ser visto em (MENON; LAM; CHENG, 2002), onde são apontadas as dificuldades da utilização do SDRE em tempo real, que requer mais recursos que outros algoritmos de controle - nesse exemplo são usados computadores com altas frequências de processador para controle em tempo real. Em (MAURO et al., 2013) também é implementada a técnica SDRE usando processadores de computadores. A proposta deste trabalho visa diminuir estes tempos de processamento através do uso de uma rede neural se comportando similarmente a um controlador SDRE.

As redes neurais artificiais (RNAs) foram construídas a partir da observação das redes neurais biológicas, formando estruturas simuladas em algoritmos computacionais para realizar determinada função. As RNAs são amplamente estudadas e formam uma nova alternativa para aplicações em vários âmbitos, inclusive aplicações em controle. As RNAs possuem entradas e saídas, as entradas são processadas através de algoritmos que simulam redes neurais biológicas, a fim de se ter um resultado desejado na saída (SILVA; SPATTI; FLAUZINO, 2010).

Recentemente, um novo tipo de categoria de redes neurais foi criado, chamado de computação de reservatório (FERREIRA; LUDERMIR; AQUINO, 2013). Trata-se de redes neurais que possuem um elemento de memória (redes neurais recorrentes), ou seja, a rede é capaz de armazenar informação de várias entradas anteriores da rede e esse conjunto de informações anteriores da rede, formam o que é chamado aqui de estado, que também será base para a rede formar a próxima saída. Jaeger criou uma arquitetura dessa e a chamou de Rede de Estado de

Eco (em inglês *Echo State Network* (ESN)) (JAEGER, 2001) e, praticamente ao mesmo tempo, (MAASS; NATSCHLÄGER; MARKRAM, 2002) descreveu uma rede chamada por ele de *Liquid State Machine* (LSM) em 2002, e as duas funcionam de forma similar com a presença deste reservatório dinâmico (KEITH, 2013).

A ESN é um sistema não-linear e tem uma capacidade de resolver muito tipos de problemas, os quais têm dependência no tempo (por conta do reservatório). Assim, exemplos de aplicações são: previsão de séries temporais, controle e outras aplicações dependentes do tempo (ERHAN, 2004; LUKOSEVICIUS; JAEGER, 2009).

As redes neurais, apesar de muitas vezes necessitarem de grandes quantidades de memória por conta principalmente da quantidade de neurônios, são sistemas simples computacionalmente e envolvem basicamente cálculos de multiplicação de matrizes e cálculos das funções de ativação dos neurônios. São estruturas muitas vezes fixas, nas quais apenas parâmetros da rede neural são alterados (quantidades de neurônios e os pesos das ligações entre eles por exemplo), possibilitando assim que seja implementado um padrão fixo de estrutura, e ajustado para diferentes tipos de aplicações.

Neste trabalho será realizado o treinamento da rede neural de estado de eco (ESN) a partir de dados de estados e atuações do controle baseado em SDRE, o que será feito através de simulações do controle do SDRE. Com as aquisições dos estados do sistema e os sinais de controle calculados, uma rede ESN será devidamente treinada, tentando diminuir ao máximo a quantidade de tipos de entradas observadas para o treinamento, diminuindo a quantidade de sensores num sistema real. Após conseguir um bom treinamento da rede neural é realizada a simulação do sistema de controle anteriormente controlado pelo SDRE, sendo agora controlado pela ESN treinada. Assim, os dados do resultado são comparados demonstrando que a atuação da rede neural foi tão eficaz quanto a do SDRE. Após esse resultado é comparado o esforço computacional entre as técnicas de controles, através da medição da quantidade de tempo gasto ao fazer os cálculos de sinal de controle. Isso será realizado para dois sistemas: microeletromecânico (MEMS) (TUSSET et al., 2012a) e o controle de suspensão veicular de um meio carro (*half-car*) (TUSSET et al., 2012b).

O que motiva o uso da ESN são algumas características como o fato de serem estruturas recorrentes (possuem realimentações), não-lineares e que principalmente podem ser treinadas de maneira relativamente simples.

Este estudo começa a partir de uma descrição sobre controle no Capítulo 2, onde entre outras técnicas de controle, também é apresentado sobre a técnica de controle SDRE. O Capítulo 3 fala sobre as redes neurais em geral, como surgiram, qual é a relação com o sistema nervoso biológico, funcionamento e aplicações. Então, se passa a aprofundar nos conhecimentos sobre as ESNs, vendo como funciona a rede neural: sua arquitetura, sua implementação, parâmetros da rede e a influências deles, processo de treinamento e testes. No Capítulo 4 é apresentado o método para treinar a rede neural a partir de dados do controle SDRE, e o controle do sistema microeletromecânico é realizado e comparado com o SDRE. No Capítulo 5 a

prática é feita para o controle de suspensão de *half car* (2 amortecedores). No Capítulo 6 estão presentes as considerações finais e perspectivas de trabalhos futuros.

2 SISTEMAS DE CONTROLE

2.1 INTRODUÇÃO E HISTÓRICO

Um sistema pode ser chamado de sistema de controle quando sua saída é controlada para assumir um valor particular ou seguir uma determinada entrada (BOLTON, 1993), sendo aplicado em vários ramos da tecnologia: controles de estabilidade de helicópteros e aviões, controle de demanda de energia, controle de pragas em uma plantação, entre outros (OGATA, 2011).

Em controle, as únicas variáveis de interesse são as variáveis de entrada e de saída, ou seja, não existe a necessidade de explorar as interações entre os componentes do sistema. O controle é efetuado a partir da leitura de variáveis de estado e então o controle interage com o sistema fazendo este alcançar o comportamento desejado.

O primeiro trabalho significativo no âmbito de controle aconteceu no século XVIII por James Watt, que utilizou um regulador centrífugo para o controle automático da velocidade de uma máquina a vapor. Em 1940, métodos de resposta em frequência tornaram possível o controle de sistemas lineares em malha fechada com bom desempenho, já que em 1940 e 1950 os controladores Proporcional Integral e Derivativos (PID) passaram a ser usados para controle de pressão, temperatura, etc. Regras para ajustes de ganhos de PID foram utilizadas no método Ziegler-Nichols (criadores da técnica). Outro método muito difundido e importante é conhecido como “Lugar das raízes”, e foi desenvolvido por Evans por volta de 1950 (OGATA, 2011).

Com o passar dos anos sistemas de controle modernos tornaram-se mais complexos, por envolver uma grande quantidade de variáveis de entrada e um grande número de variáveis de saída, em que o sinal de controle de uma variável pode influenciar na saída de outra variável. A partir daí, para alguns problemas tecnológicos, o controle clássico passou a não ser suficiente para suprir as necessidades, pois o controle clássico trata basicamente sistemas com uma única entrada e uma única saída. Assim, surgem os controles de múltiplas entradas e múltiplas saídas (*multiple input and multiple output (MIMO's)*). A evolução computacional (hardware e software) no início dos anos 60 permitiu a análise de sistemas complexos diretamente no domínio do tempo, dessa forma, a teoria de controle moderno passou a ser criada para operar nas crescentes complexidades dos sistemas e seus requisitos de precisão, como por exemplo, nos equipamentos militares, espaciais e industriais (WIENER, 1948) (MOLTER, 2008). O controle moderno se diferencia do controle clássico principalmente por controlar mais de uma variável de entrada e saída.

A teoria de controle moderno se baseia em equações diferenciais ordinárias, que são os modelos dos sistemas físicos, para a análise dos sistemas no domínio do tempo, e simplificando a realização do projeto de controle, isso tendo como referência o modelo de um sistema de controle real. Porém, a estabilidade é comprometida devido a fatores aleatórios como: erros de medições, erros nas atuações, desgastes de componentes físicos. Esses fatores podem

fazer o controlador não levar o sistema para o estado controlado. Se os erros no sistema real forem previstos no controlador, então o sistema continua sendo estável apesar das diferenças dos sistemas (modelo e real). Isso é chamado de controle robusto e esse controle aborda a análise tanto no domínio do tempo, quanto na frequência para realização do projeto (SCHMID; RAFIKOV, 2005).

2.2 DEFINIÇÕES

Nesta subseção é definida a terminologia básica de sistemas de controle, necessária para a compreensão do demais capítulos dessa dissertação:

1. Variável Controlada: é a grandeza ou a condição que é medida e controlada. Representa qualquer variável no sistema onde se deseja realizar o controle. Por exemplo, a temperatura de um tanque a qual deve ser mantida em 50° C.
2. Sinal de controle: é a grandeza ou a condição que influenciará na variável manipulada, a qual interfere direta ou indiretamente na variável controlada. No exemplo do controle da temperatura do tanque, sinal (sinal de controle) que atua na resistência (variável manipulada) que realiza o aquecimento do fluido do tanque.
3. Variável Manipulada: é onde ocorre diretamente a atuação do controlador na planta.
4. Plantas: é um conjunto de equipamentos que funcionam de forma integrada.
5. Processos: toda operação a ser controlada, onde há um sequenciamento de estados dependentes e sucessivos que geram um determinado comportamento.
6. Sistema: é um conjunto de componentes combinados que constroem uma determinada estrutura para um objetivo.
7. Distúrbios: os distúrbios agem no sistema adversamente alterando de forma não desejada a variável controlada. Estes distúrbios podem ser gerados dentro do próprio sistema, ou, então, surgidos a partir de um evento externo, sendo visto como uma entrada adicional no sistema.

2.3 CLASSIFICAÇÃO DE SISTEMAS DE CONTROLE

Um sistema de controle pode ser classificado de diferentes formas, como por exemplo em relação ao tipo de malha e ao número de variáveis de entradas e saídas do sistema.

2.3.1 Controle em Relação Ao Tipo de Malha (Sistemas em Malha Aberta e Malha Fechada)

Os sistemas de controle podem ser classificados, quanto a sua malha: sistemas de malha fechada e malha aberta. Essa classificação é de grande importância e envolve a forma em que o controlador é projetado.

Um sistema de controle em malha fechada (Figura 1) tem uma realimentação da informação da variável controlada, que é comparada com o valor desejado, obtendo assim o erro que é a nova entrada do controlador. O controlador gera o sinal de controle que atuará na planta e modificará a variável controlada que realimentará o sistema novamente. O objetivo é fazer com que o erro seja zero, ou o menor possível (DORF, 2009).

O controle em malha aberta (Figura 2) não possui realimentações do sistema, e a saída do sistema não é comparada com um comportamento desejado. Neste caso, o controle é realizado tendo como base outros parâmetros como o tempo de execução. Os sistemas em malha aberta são calibrados com valores fixos de operação, e funcionam com base de uma programação pré-determinada (OGATA, 2011).

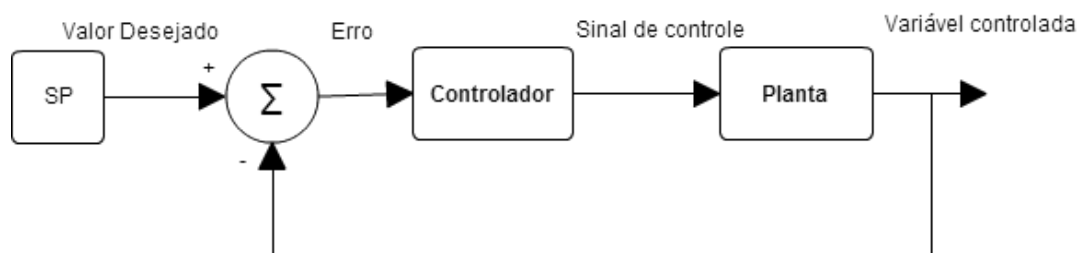


Figura 1 – Malha fechada (Autoria própria)

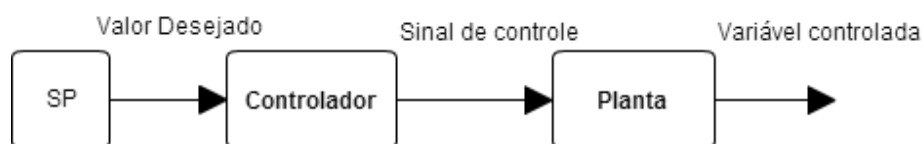


Figura 2 – Malha aberta (Autoria própria)

2.4 CLASSIFICAÇÃO E TIPOS DE CONTROLE QUANTO À QUANTIDADE DE ENTRADAS E SAÍDAS

Em sistemas simples, muitas vezes, é necessária apenas uma entrada e uma saída para o controle. Esses são os sistemas SISO (*single input e single output*), chamados de controle clássico, como exemplo o controlador PID. Em outros projetos de controle são necessárias a utilização de n entradas e m saídas, caracterizando sistemas multivariáveis, ou ainda, sistemas MIMO (*multiple input and multiple output*).

Para a análise e implementação de um controlador é necessária a utilização de um modelo matemático para o sistema a ser controlado. Esse modelo é composto por equações

diferenciais ordinárias, obtidas pelas leis físicas que regem o comportamento dos elementos do sistema.

Após a obtenção das equações pode ser feita a adimensionalização das equações, obtendo o espaço de estados, que é um modelo no qual é estudado o controle do comportamento do processo físico. O estudo do controle, no espaço de estado, faz a utilização do mesmo conjunto de equações adimensionais em outros modelos. Por exemplo, o modelo de estado de um sistema massa-mola com atrito linear e uma perturbação externa senoidal é igual ao modelo de um circuito elétrico RLC, em série, com uma tensão senoidal. Isso faz o controle de um sistema ser utilizado como inspiração inicial para outro sistema.

Os sistemas multivariáveis são expressos no espaço de estados, por um sistema de equações diferenciais ordinárias. Desta forma, todos os estados de entrada e saída devem ser apresentados nas equações de estado:

$$\begin{cases} \dot{\mathbf{x}} &= \mathbf{Ax} + \mathbf{Bu} \\ \dot{\mathbf{y}} &= \mathbf{Cx} + \mathbf{Du} \end{cases}, \quad (2.1)$$

onde: $\mathbf{x} \in \mathbb{R}^n$ é o vetor do estado; $\mathbf{u} \in \mathbb{R}^m$ o vetor de entradas; $\mathbf{A} \in \mathbb{R}^{n \times n}$ a matriz de estado; $\mathbf{B} \in \mathbb{R}^{n \times m}$ é a matriz de controle; \mathbf{C} é a matriz de saída do sistema; $\mathbf{D} \in \mathbb{R}^{m \times n}$ a matriz de transferência direta. Sendo n o número de estados e m o número de entradas (NISE, 2000).

2.4.1 Controle Proporcional, Integral e Derivativo (PID)

O controle PID (proporcional, integral e derivativo) é um sistema em malha fechada, SISO, que calcula o sinal de controle a partir do valor do erro na variável controlável da planta. O cálculo deste erro é obtido através da diferença do valor desejado e o valor da variável controlada. Para o controlador PID este erro é analisado de 3 formas diferentes: (1) o proporcional do erro, o qual é o próprio erro em si; (2) a integral do erro, o somatório dos erros, e (3) o derivativo que é calculado através da taxa de erro, ou seja, a diferença do erro no instante atual e instante anterior dividido pelo intervalo de tempo entre as medidas. Cada um desses erros, proporcional, integral e derivativo, são multiplicados por constantes chamadas normalmente de K_p , K_i e K_d . A soma dos três termos define o sinal de controle, que será utilizado para controle da planta. Isso pode ser verificado na Figura 3.

Equacionando o sinal de controle $u(t)$ que entra na planta tem-se:

$$u(t) = K_p e(t) + K_i \int e(t) + K_d \frac{de(t)}{dt}. \quad (2.2)$$

O PID é muito usado, pois possui uma série de vantagens como: facilidade de implementação, baixo custo, e lógica pré-configurada em vários dispositivos como Controladores Lógico Programáveis (CLPs). Seu ajuste, determinação de K_p , K_i e K_d , pode ser conseguido mesmo sem o conhecimento do funcionamento da planta, sendo que vários métodos existentes

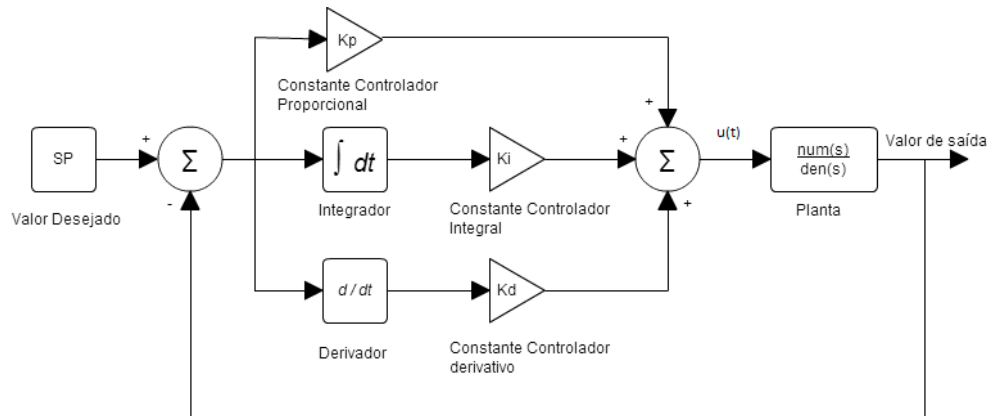


Figura 3 – Diagrama de blocos controlador PID (Autoria própria)

podem ser utilizados. Existem também técnicas *auto tunnings* onde o PID é regulado automaticamente (OGATA, 2011). Pode-se também usar apenas o controlador proporcional e derivativo, ou proporcional e integral, formando assim controladores PD e PI, respectivamente.

Uma das técnicas para determinar os ganhos K_p , K_i e K_d foi proposta por Ziegler-Nichols chamada de “curva de reação de processo”, a qual funciona apenas para sistemas de primeira ordem e consiste em analisar a resposta da planta sem o controlador, e a partir da curva se extraem os valores do ganho do controlador PID.

Outro método também proposto por Ziegler-Nichols é o “método do ciclo máximo”, que funciona não apenas para sistemas de primeira ordem, mas consiste em analisar o comportamento do sistema através do ganho proporcional K_p e em seguida se definem os valores dos ganhos K_p , K_i e K_d . Outro método para regular os ganhos é o de alocação de polos, no qual o objetivo é o cancelamento dos polos da planta (DORF, 2009).

2.4.2 Controle Por Estação de Estado

De acordo com (OGATA, 2011) muitos problemas de controle podem ser representados por equações em espaço de estados conforme a equação 2.3.

$$\begin{aligned}\dot{\mathbf{x}} &= \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \\ \mathbf{y} &= \mathbf{C}\mathbf{x} + \mathbf{D}\mathbf{u}\end{aligned}\quad (2.3)$$

onde \mathbf{A} é uma matriz de coeficiente $n \times n$ (matriz de controle do processo), \mathbf{B} é uma matriz de controle $n \times r$, \mathbf{D} e \mathbf{C} é uma matriz resposta $m \times n$. A variável \mathbf{x} representa o vetor de estados, \mathbf{u} é um vetor de controle r-dimensional e \mathbf{y} é um vetor resposta m-dimensional. Pode-se verificar a representativa do sistema em espaço de estados através de diagramas de blocos conforme Figura 4.

Um sistema (2.3) é dito controlável se existir um u que o leve para a origem em um tempo finito (DORF, 2009). Assim, se as matrizes \mathbf{A} e \mathbf{B} da equação (2.3) escritas na matriz

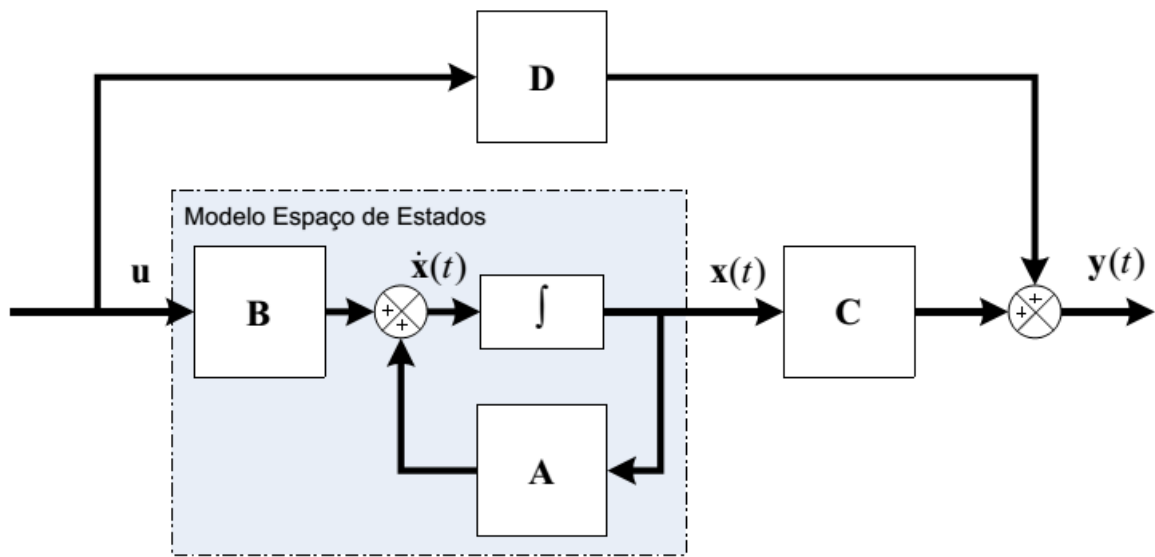


Figura 4 – Diagrama de Blocos em Malha Aberta (Autoria própria)

(2.4), e esta tiver posto igual a n (OGATA, 2011) então o sistema é dito controlável.

$$CO = \begin{bmatrix} B & AB & A^2B & \dots & A^{n-1}B \end{bmatrix} \quad (2.4)$$

Considerando $u = -\mathbf{K}\mathbf{x}$ a lei de controle, onde \mathbf{K} é a matriz de ganho de realimentação de estados e substituindo $u = -\mathbf{K}\mathbf{x}$ em (2.3), obtém-se a correspondente em malha fechada conforme a equação (2.5).

$$\dot{\mathbf{x}}(t) = (\mathbf{A} - \mathbf{BK})\mathbf{x}(t) \quad (2.5)$$

A representação em diagramas de blocos do sistema com controle por realimentação de estados pode ser visto pela figura 5.

A solução da equação (2.4) pode ser encontrada na equação (2.6)

$$\mathbf{x}(t) = e^{(\mathbf{A}-\mathbf{BK})(t)}\mathbf{x}(0), \quad (2.6)$$

sendo $\mathbf{x}(0)$ o estado inicial causado por distúrbios externos (OGATA, 2011).

Percebe-se pela equação (2.6) que as características de resposta do sistema, e critério de estabilidade, podem ser determinadas pelos autovalores de $(\mathbf{A}-\mathbf{BK})$, sendo que para o caso de sistemas estáveis é necessário que a parte real de todos os autovalores sejam negativos. Existem várias técnicas para encontrar a matriz \mathbf{K} , e algumas destas técnicas serão demonstradas a seguir.

2.4.2.1 Alocação de polos

Para encontrar a matriz \mathbf{K} se utilizando da técnica de alocação de polos, primeiramente deve-se definir os polos desejados para o sistema em malha fechada, e determinar o valor da

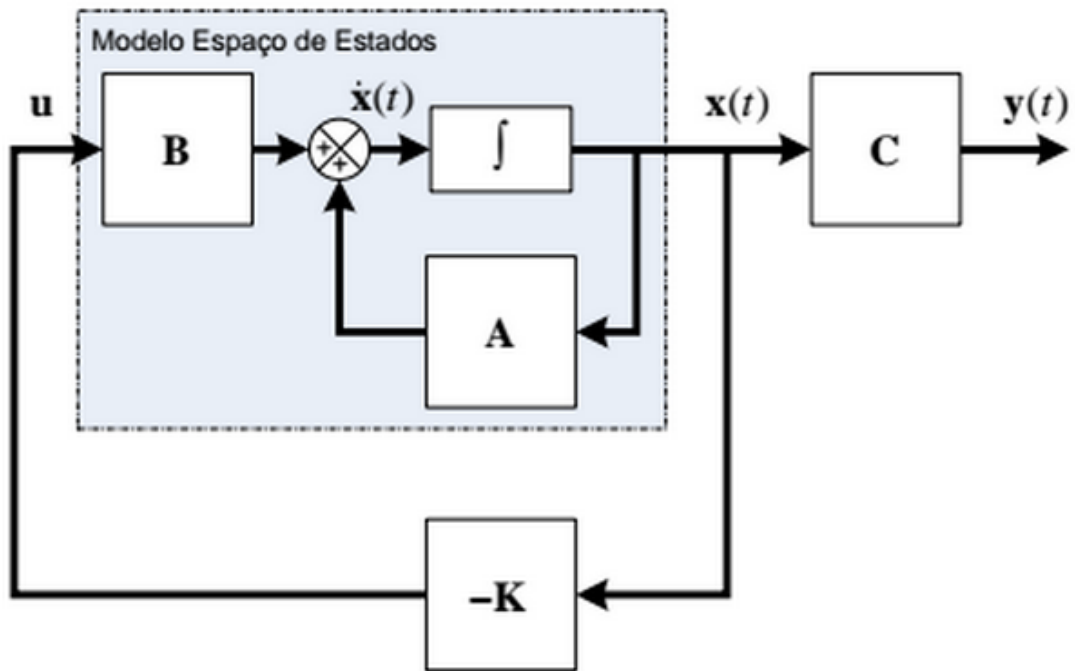


Figura 5 – Diagrama de Blocos do Controle por Realimentação de estados (Autoria própria)

matriz K de forma a impor estes polos. De acordo com Ogata a matriz K pode ser obtida de três formas distintas (OGATA, 2011):

Primeiro Método

Escrever o polinômio característico para o sistema com os polos desejados:

$$(s - \mu_1)(s - \mu_2) \dots (s - \mu_n) = s^n + \alpha_1 s^{n-1} + \dots + \alpha_n s + \alpha_n, \quad (2.7)$$

onde: μ_i são os polos desejados.

A matriz de ganho de retroação de estado é obtida de:

$$K = \begin{bmatrix} \alpha_n - a_n & \alpha_{n-1} - a_{n-1} & \dots & \alpha_2 - a_2 & \alpha_1 - a_1 \end{bmatrix} T^{-1}.$$

Sendo que α_i é obtido de (2.7), a_1 de:

$$|sI - A| = s^n + a_1 s^{n-1} + \dots + a_{n-1} s + a_n, \quad (2.8)$$

e $T = MW$ onde $M = \begin{bmatrix} B & AB & \dots & A^{n-1}B \end{bmatrix}$ e W :

$$W = \begin{bmatrix} a_{n-1} & a_{n-2} & \dots & a_1 & 1 \\ a_{n-2} & a_{n-3} & \dots & 1 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_1 & 1 & \dots & 0 & 0 \\ 1 & 0 & \dots & 0 & 0 \end{bmatrix} \quad (2.9)$$

Segundo Método

Escrever a matriz de ganho de retroação de estados na seguinte forma:

$$\mathbf{K} = \begin{bmatrix} k_1 & k_2 & \dots & k_n \end{bmatrix}. \quad (2.10)$$

A matriz \mathbf{K} é obtida da seguinte igualdade:

$$|s\mathbf{I} - \mathbf{A} + \mathbf{BK}| = (s - \mu_1)(s - \mu_2) \dots (s - \mu_n), \quad (2.11)$$

onde: μ_i são os polos desejados.

Terceiro Método

Considerando a fórmula de Ackermann:

$$\phi(\mathbf{A}) = \mathbf{A}^n + \alpha_1 \mathbf{A}^{n-1} + \alpha_2 \mathbf{A}^{n-2} + \dots + \alpha_{n-1} \mathbf{A} + \alpha_n \mathbf{I}, \quad (2.12)$$

α_i é obtido de (2.7). E a matriz \mathbf{K} é obtida de:

$$\mathbf{K} = \begin{bmatrix} 0 & 0 & \dots & 1 \end{bmatrix} \mathbf{M}^{-1} \phi(\mathbf{A}), \quad (2.13)$$

onde \mathbf{M} é a matriz de controlabilidade $\begin{bmatrix} \mathbf{B} & \mathbf{AB} & \dots & \mathbf{A}^{n-1} \mathbf{B} \end{bmatrix}$.

2.4.2.2 Regulador linear quadrático (LQR)

Como visto em (OGATA, 2011), este método desempenha um papel fundamental em diversos métodos de controle moderno para sistemas MIMO. Ele está presente, como por exemplo, nos controladores LQG (*linear quadratic Gaussian*), horizonte finito ou H_m , e principalmente o método SDRE, o qual será usado como controlador neste trabalho.

Formação do Problema de Controle Ótimo Quadrático

A obtenção do controle ótimo (LQR) é obtido da seguinte forma:

Dado o sistema:

$$\dot{\mathbf{x}} = \mathbf{Ax} + \mathbf{Bu}. \quad (2.14)$$

Determinar a matriz \mathbf{K} do controle ótimo:

$$\mathbf{u} = -\mathbf{Kx}. \quad (2.15)$$

De modo a minimizar o índice de desempenho:

$$J = \int (\mathbf{x}^T \mathbf{Qx} + \mathbf{u}^T \mathbf{Ru}) dt. \quad (2.16)$$

sendo \mathbf{Q} uma matriz hermetiana ou simétrica real definida positiva (ou semidefinida positiva), e \mathbf{R} uma matriz hermetiana ou real simétrica definida positiva. As matrizes \mathbf{Q} e \mathbf{R} determinam a importância relativa do erro e do dispêndio de energia.

Em consequência, se os elementos da matriz \mathbf{K} forem determinados de modo a minimizar o índice de desempenho, então $\mathbf{u} = -\mathbf{K}\mathbf{x}$ é ótimo qualquer que seja o estado inicial $\mathbf{x}(0)$.

Para resolver o problema de otimização, substitui-se (2.15) em (2.14):

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} - \mathbf{B}\mathbf{K}\mathbf{x} = (\mathbf{A} - \mathbf{B}\mathbf{K})\mathbf{x}. \quad (2.17)$$

Admitindo-se que a matriz $\mathbf{A} - \mathbf{B}\mathbf{K}$ seja estável, ou seja, que os autovalores tenham parte real negativa.

Substituindo (2.17) em (2.16):

$$J = \int_0^{\infty} (\mathbf{x}^T \mathbf{Q} \mathbf{x} + \mathbf{x}^T \mathbf{K}^T \mathbf{R} \mathbf{K} \mathbf{x}) dt = \int_0^{\infty} (\mathbf{x}^T (\mathbf{Q} + \mathbf{K}^T \mathbf{R} \mathbf{K}) \mathbf{x}) dt, \quad (2.18)$$

considerando que:

$$\mathbf{x}^T (\mathbf{Q} + \mathbf{K}^T \mathbf{R} \mathbf{K}) \mathbf{x} = -\frac{d}{dt} (\mathbf{x}^T \mathbf{P} \mathbf{x}), \quad (2.19)$$

onde \mathbf{P} é uma matriz hermetiana ou real simétrica definida positiva. Obtém-se, então:

$$\mathbf{x}^T (\mathbf{Q} + \mathbf{K}^T \mathbf{R} \mathbf{K}) \mathbf{x} = -\dot{\mathbf{x}}^T \mathbf{P} \mathbf{x} - \mathbf{x}^T \mathbf{P} \dot{\mathbf{x}} = -\mathbf{x}^T \left[(\mathbf{A} - \mathbf{B}\mathbf{K})^T \mathbf{P} + \mathbf{P} (\mathbf{A} - \mathbf{B}\mathbf{K}) \right] \mathbf{x}. \quad (2.20)$$

Comparando ambos os membros desta última equação, e se observando que ela deve ser verdadeira para qualquer \mathbf{x} , deve-se ter:

$$(\mathbf{A} - \mathbf{B}\mathbf{K})^T \mathbf{P} + \mathbf{P} (\mathbf{A} - \mathbf{B}\mathbf{K}) = -(\mathbf{Q} + \mathbf{K}^T \mathbf{R} \mathbf{K}). \quad (2.21)$$

Pelo segundo método de Liapunov, se $\mathbf{A} - \mathbf{B}\mathbf{K}$ é uma matriz estável, então existe uma matriz \mathbf{P} definida positiva que satisfaz (2.21). Por conseguinte, o procedimento a adotar é o da determinação dos elementos de \mathbf{P} a partir desta equação, e verificar se ela é definida positiva (OGATA, 2011).

O índice de desempenho J pode ser calculado considerando-se apenas as condições iniciais do sistema $\mathbf{x}(0)$ da seguinte forma:

$$J = \int_0^{\infty} \mathbf{x}^T (\mathbf{Q} + \mathbf{K}^T \mathbf{R} \mathbf{K}) \mathbf{x} dt = -\mathbf{x}^T (\infty) \mathbf{P} \mathbf{x} (\infty) + \mathbf{x}^T (0) \mathbf{P} \mathbf{x} (0). \quad (2.22)$$

Como se admite que todos os autovalores de $\mathbf{A} - \mathbf{B}\mathbf{K}$ têm parte real negativa, tem-se $\mathbf{x}(\infty) \rightarrow \mathbf{0}$. Portanto:

$$J = \mathbf{x}^T (0) \mathbf{P} \mathbf{x} (0). \quad (2.23)$$

Assim, o índice de desempenho J pode ser obtido em termos de estado inicial $\mathbf{x}(0)$ e \mathbf{P} .

Para obter a solução do problema de controle ótimo quadrático, procede-se como a seguir: supondo que \mathbf{R} é uma matriz hermetiana ou real simétrica definida positiva, pode-se escrever $\mathbf{R} = \mathbf{T}^T \mathbf{T}$, onde \mathbf{T} é uma matriz não singular, então tem-se:

$$(\mathbf{A}^T - \mathbf{K}^T \mathbf{B}^T) \mathbf{P} + \mathbf{P}(\mathbf{A} - \mathbf{B} \mathbf{K}) + \mathbf{Q} + \mathbf{K}^T \mathbf{T}^T \mathbf{T} \mathbf{K} = 0, \quad (2.24)$$

que pode ser escrita sob a forma:

$$\mathbf{A}^T \mathbf{P} + \mathbf{P} \mathbf{A} + \left[\mathbf{T} \mathbf{K} - (\mathbf{T}^T)^{-1} \mathbf{B}^T \mathbf{P} \right]^T \left[\mathbf{T} \mathbf{K} - (\mathbf{T}^T)^{-1} \mathbf{B}^T \mathbf{P} \right] - \mathbf{P} \mathbf{B} \mathbf{R}^{-1} + \mathbf{Q} = 0. \quad (2.25)$$

A minimização de J com relação a \mathbf{K} requer a minimização de (2.26) com relação a \mathbf{K} :

$$\mathbf{x}^T \left[\mathbf{T} \mathbf{K} - (\mathbf{T}^T)^{-1} \mathbf{B}^T \mathbf{P} \right]^T \left[\mathbf{T} \mathbf{K} - (\mathbf{T}^T)^{-1} \mathbf{B}^T \mathbf{P} \right] \mathbf{x}. \quad (2.26)$$

Como esta expressão é não negativa, o mínimo ocorre quando ela vale zero, ou seja, quando $\mathbf{T} \mathbf{K} = (\mathbf{T}^T)^{-1} \mathbf{B}^T \mathbf{P}$, assim:

$$\mathbf{K} - \mathbf{T}^{-1} (\mathbf{T}^T)^{-1} \mathbf{B}^T \mathbf{P} = \mathbf{R}^{-1} \mathbf{B}^T \mathbf{P}. \quad (2.27)$$

A equação 2.27 fornece a matriz ótima \mathbf{K} . Em consequência a lei de controle ótimo para o problema de controle ótimo quadrático, quando o índice de desempenho é dado pela equação 2.27, é linear e é dada por:

$$\mathbf{u} = -\mathbf{K} \mathbf{x} = -\mathbf{R}^{-1} \mathbf{B}^T \mathbf{P} \mathbf{x}. \quad (2.28)$$

A matriz \mathbf{P} na equação 2.27 deve satisfazer a equação (2.25) ou a seguinte equação reduzida:

$$\mathbf{P} \mathbf{A} + \mathbf{A}^T \mathbf{P} - \mathbf{P} \mathbf{B} \mathbf{R}^{-1} \mathbf{B}^T \mathbf{P} + \mathbf{Q} = 0. \quad (2.29)$$

A equação (2.29) chamada de equação matricial de Riccati é usada para encontrar a matriz \mathbf{K} de ganho de retroação do controlador, de modo que a lei de controle (2.28) minimiza o índice de desempenho (2.16) sujeito à equação de restrição (2.14).

2.4.2.3 SDRE

O controle SDRE é uma extensão do Regulador Linear Quadrático (LQR) para sistemas não-lineares. O método é realizado reescrevendo as equações dinâmicas do sistema não-linear de forma que seus coeficientes dependam dos estados (SILVA, 2014).

Quando existe a necessidade de controlar sistemas com múltiplas saídas, o controle PID se torna uma ferramenta que, em determinadas situações, não se mostra capaz de manter

as variáveis de saída com níveis de erros aceitáveis (TUSSET et al., 2012b). Assim, devem ser utilizados outros métodos de controle para que os objetivos propostos sejam alcançados. A técnica de controle SDRE, além de efetuar o controle de sistemas não-lineares podem atuar em sistemas de múltiplas entradas e múltiplas saídas (SILVA, 2014).

Alguns trabalhos que usaram a técnica de controle SDRE: (GARRARD; MCCLAMROCH; CLARK, 1967; BURGHART, 1969; WERNLI; COOK, 1975) são trabalhos que apresentam soluções para sistemas não-lineares realizando controle sub-ótimo. (CLOUTIER; D'SOUZA; MRACEK, 1996) desenvolveram uma técnica para otimalidade, robustez e estabilidades relacionadas à técnica SDRE. (MRACEK; CLOUTIER, 1997) aplicaram as técnicas e realizaram uma avaliação do estudo aplicando no problema de referência proposto por (BUPP; BERNSTEIN; COPPULA, 1998), o qual trata de sistemas não-lineares.

A partir de então, a técnica passou a ser mais difundida e vários são os exemplos de aplicações: (BANKS B. M. LEWIS, 2007) faz a estimação não-linear de estados para o controle de crescimento de filme em reator de deposição de vapor químico; (BANKS et al., 2011) fizeram um trabalho de controle terapêutico de infecções por HIV; (ABDELRAHMAN; CHANG; PARK, 2011) desenvolveram um controle não-linear para controle de atitude (orientação) de um satélite com barras magnéticas; (SOUZA; GONZÁLES, 2012) também realizaram o controle de atitude de satélite com a técnica SDRE em conjunto com o Filtro de Kalman; (BANK; LEWIS; TRAN, 2007) abordaram o problema de convergência assintótica de forma didática além de discutirem métodos numéricos para a aproximação da solução do método SDRE.

(ÇIMEN, 2008) discutiu a existência de soluções para problemas não-lineares quanto à estabilidade e otimalidade associadas à tecnologia. (CHANG, 2009) realizou uma estratégia nova na análise de contração para investigar a estabilidade não-linear da metodologia. (ÇIMEN, 2010) analisou a seleção das matrizes do modelo colocado na estrutura de aplicação da técnica SDRE, sintetizou a execução do projeto utilizando SDRE e utilizou sistemas em conformidade ou não para a aplicação da técnica.

Hoje existem muitos trabalhos que realizam a análise do comportamento em sistemas em malha fechada e em tempo real. (SOUZA; ARENA, 2013) fez um estudo de controle e estimação de estados num modelo não-linear de um simulador de movimento para manobras de rastreamento de ângulos grandes. Através da simulação do modelo o sistema foi analisado quanto à robustez e desempenho. (JUNG et al.,) aplicaram a técnica para sincronização de atitude de um satélite. O sistema foi fisicamente simulado para validar o controle em tempo real. Todos estes trabalhos demonstram um pouco da diversificação de aplicações onde pode estar trabalhando com a técnica SDRE.

Dois trabalhos publicados por Tusset (TUSSET et al., 2012a; TUSSET et al., 2012b) serão demonstrados nos capítulos de desenvolvimento. O primeiro trata de um sistema microeletromecânico MEMs, no qual é efetuado o controle para levar o sistema para uma órbita desejada. O outro sistema é um controle de suspensão *half-car* (meio carro).

Controle ótimo com realimentação para sistemas não-lineares usando equações de Riccati dependentes do estado

A técnica foi iniciada por (PEARSON, 1962), que apresenta um procedimento para aplicação de índice de desempenho quadrático otimizando um sistema não-linear, utilizando-se de uma aproximação linear dependente do tempo dos estados.

Um problema de controle ótimo na forma:

$$\begin{aligned}\dot{\mathbf{x}}(t) &= \mathbf{A}_e(t)\mathbf{x}(t) + \mathbf{B}_e(t)\mathbf{u}(t), \\ \mathbf{y} &= \mathbf{C}_e\mathbf{x}, \\ \mathbf{x}(0) &= \mathbf{x}_0,\end{aligned}\tag{2.30}$$

e minimizando o funcional na forma quadrática:

$$\mathbf{J}[\mathbf{u}] = \int_0^{t_f} [\mathbf{x}^T(t)\mathbf{Q}_e(t)\mathbf{x}(t) + \mathbf{u}^T(t)\mathbf{R}_e(t)\mathbf{u}] dt.\tag{2.31}$$

para um sistema com os coeficientes das matrizes de estado, dependentes do estado e horizonte infinito, pode ser formulado da seguinte forma (MRACEK; CLOUTIER, 1998):

$$\mathbf{J}[\mathbf{u}] = \frac{1}{2} \int_{t_0}^{\infty} \mathbf{x}^t \mathbf{Q}_e(x) \mathbf{x} + \mathbf{u}^T \mathbf{R}_e(x) \mathbf{u} dt.\tag{2.32}$$

minimizar o funcional de custo em relação ao estado \mathbf{x} e o controle \mathbf{u} , sujeito ao sistema de restrições não-lineares

$$\begin{aligned}\dot{\mathbf{x}} &= \mathbf{f}(x) + \mathbf{B}_e(x)\mathbf{u}, \\ \mathbf{y} &= \mathbf{C}_e\mathbf{x}, \\ \mathbf{x}(0) &= \mathbf{x}_0,\end{aligned}\tag{2.33}$$

onde: $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{u} \in \mathbb{R}^m$ e $\mathbf{y} \in \mathbb{R}^s$ (\mathbb{R}^s é a dimensão do vetor de saída do sistema). $\mathbf{Q}_e(x) \in \mathbb{R}^{n \times n}$ é a matriz de pesos semidefinida positiva e $\mathbf{R}_e(x) \in \mathbb{R}^{m \times m}$ é definida positiva.

A aproximação pelas equações de Riccati dependentes do estado (SDRE), para resolver o problema de controle subótimo (2.32) e (2.33) se dá por:

1. *usar parametrização direta para transformar a dinâmica não-linear do estado em matrizes de coeficientes dependentes do estado (SDC):*

$$\dot{\mathbf{x}} = \mathbf{A}_e(x)\mathbf{x} + \mathbf{B}_e(x)\mathbf{u},\tag{2.34}$$

com $\mathbf{f}(x) = \mathbf{A}_e(x)\mathbf{x}$. Em geral, $\mathbf{A}_e(x)$ é única somente se x for escalar (BANKS; LEWIS; TRAM, 2007). A escolha das parametrizações a serem feitas deve ser apropriada, de acordo com o sistema de controle de interesse. Um fator importante para esta escolha é não violar a controlabilidade do sistema (a capacidade do sistema ser controlado através

da atuação de componentes possíveis de serem manipulados), ou seja, a matriz de controlabilidade dependente do estado $\begin{bmatrix} \mathbf{B}_e(x) & \mathbf{A}_e(x)\mathbf{B}_e(x) & \dots & \mathbf{A}_e^{n-1}(x)\mathbf{B}_e(x) \end{bmatrix}$ ter posto completo.

2. resolver a equação de Riccati dependente do estado (BANKS; LEWIS; TRAM, 2007):

O hamiltoniano para o problema de controle ótimo (2.32) e (2.33) é dado por:

$$\mathbf{H}(x, u, \lambda) = \frac{1}{2}(\mathbf{x}^T \mathbf{Q}_e(x) \mathbf{x} + \mathbf{u}^T \mathbf{R}_e(x) \mathbf{u}) + \lambda^T (\mathbf{A}_e(x) \mathbf{x} + \mathbf{B}_e(x) \mathbf{u}). \quad (2.35)$$

Do hamiltoniano, as condições necessárias para o controle ótimo são dadas por

$$\dot{\lambda} = -\mathbf{Q}_e(x) \mathbf{x} - \frac{1}{2} \mathbf{x}^T \frac{\partial \mathbf{Q}_e(x)}{\partial x} \mathbf{x} - \frac{1}{2} \mathbf{u}^T \frac{\partial \mathbf{R}_e(x)}{\partial x} \mathbf{u} - \left[\frac{\partial (\mathbf{A}_e(x) \mathbf{x})}{\partial x} \right]^T \lambda - \left[\frac{\partial (\mathbf{B}_e(x) \mathbf{u})}{\partial x} \right]^T \lambda, \quad (2.36)$$

$$\dot{\mathbf{x}} = \mathbf{A}_e(x) \mathbf{x} + \mathbf{B}_e(x) \mathbf{u}, \quad (2.37)$$

$$0 = \mathbf{R}_e(x) \mathbf{u} + \mathbf{B}_e(x) \lambda. \quad (2.38)$$

Seja \mathbf{A}_i a i -ésima linha de $\mathbf{A}_e(x)$ e \mathbf{B}_i a i -ésima linha de $\mathbf{B}_e(x)$. Os termos de derivada parcial, na forma matricial, podem ser escritos como:

$$\frac{\partial (\mathbf{A}_e(x) \mathbf{x})}{\partial x} = \mathbf{A}_e(x) + \frac{\partial (\mathbf{A}_e(x))}{\partial x} \mathbf{x} = \mathbf{A}_e(x) + \begin{bmatrix} \frac{\partial \mathbf{A}_1}{\partial x_1} \mathbf{x} & \dots & \frac{\partial \mathbf{A}_1}{\partial x_n} \mathbf{x} \\ \dots & \ddots & \dots \\ \frac{\partial \mathbf{A}_n}{\partial x_1} \mathbf{x} & \dots & \frac{\partial \mathbf{A}_n}{\partial x_n} \mathbf{x} \end{bmatrix}, \quad (2.39)$$

e

$$\frac{\partial (\mathbf{B}_e(x) \mathbf{u})}{\partial x} = \begin{bmatrix} \frac{\partial \mathbf{B}_1}{\partial x_1} \mathbf{u} & \dots & \frac{\partial \mathbf{B}_1}{\partial x_n} \mathbf{u} \\ \dots & \ddots & \dots \\ \frac{\partial \mathbf{B}_n}{\partial x_1} \mathbf{u} & \dots & \frac{\partial \mathbf{B}_n}{\partial x_n} \mathbf{u} \end{bmatrix}. \quad (2.40)$$

O co-estado é assumido na forma $\lambda = \mathbf{P}_e(x) \mathbf{x}$, que tem dependência do estado. Usando esta forma do co-estado, da equação (2.38) obtém-se o controle realimentado

$$\mathbf{u} = -\mathbf{R}_e^{-1}(x) \mathbf{B}_e^T(x) \mathbf{P}_e(x) \mathbf{x}. \quad (2.41)$$

Substituindo este controle na equação (2.37) tem-se

$$\dot{\mathbf{x}} = \mathbf{A}_e(x) \mathbf{x} - \mathbf{B}_e(x) \mathbf{R}_e^{-1}(x) \mathbf{B}_e^T(x) \mathbf{P}_e(x) \mathbf{x}. \quad (2.42)$$

Para encontrar o valor da função $P_e(x)$ diferencia-se $\lambda = P_e(x)\mathbf{x}$ no tempo ao longo da trajetória

$$\begin{aligned}\dot{\lambda} &= \dot{P}_e(x)\mathbf{x} + P_e(x)\dot{\mathbf{x}} \\ &= \dot{P}_e(x)\mathbf{x} + P_e(x)\mathbf{A}_e(x)\mathbf{x} - P_e\mathbf{B}_e(x)\mathbf{R}_e^{-1}(x)\mathbf{B}_e^T(x)P_e(x)\mathbf{x},\end{aligned}\quad (2.43)$$

onde é usada a notação $\dot{P}_e(x) = \sum_{i=1}^n P_{xi}(x)\dot{x}_i(t)$.

Substituindo a equação (2.43) na primeira condição necessária do controle ótimo, mostrado na equação (2.36), referente à $\dot{\lambda}$, obtém-se

$$\begin{aligned}\dot{P}_e(x)\mathbf{x} + P_e(x)\mathbf{A}_e(x)\mathbf{x} - P_e\mathbf{B}_e(x)\mathbf{R}_e^{-1}(x)\mathbf{B}_e^T(x)P_e(x)\mathbf{x} \\ = -\mathbf{Q}_e(x)\mathbf{x} - \frac{1}{2}\mathbf{x}^T \frac{\partial \mathbf{Q}_e(x)}{\partial x} \mathbf{x} - \frac{1}{2}\mathbf{u}^T \frac{\partial \mathbf{R}_e(x)}{\partial x} \mathbf{u} \\ - \left[\mathbf{A}_e(x) + \frac{\partial(\mathbf{A}_e(x)\mathbf{x})}{\partial x} \right]^T P_e(x)\mathbf{x} - \left[\frac{\partial(\mathbf{B}_e(x)\mathbf{u})}{\partial x} \right]^T P_e(x)\mathbf{x}\end{aligned}\quad (2.44)$$

Assumindo que $P_e(x)$ é solução da equação de Riccati dependente de estado (SDRE), dada por

$$P_e(x)\mathbf{A}_e(x) + \mathbf{A}_e^T(x)P_e(x) - P_e\mathbf{B}_e(x)\mathbf{R}_e^{-1}(x)\mathbf{B}_e^T(x)P_e(x) + \mathbf{Q}_e(x) = 0, \quad (2.45)$$

então a seguinte condição necessária de otimalidade precisa ser satisfeita:

$$\begin{aligned}\dot{P}_e(x)\mathbf{x} + \frac{1}{2}\mathbf{x}^T \frac{\partial \mathbf{Q}_e(x)}{\partial x} \mathbf{x} + \frac{1}{2}\mathbf{u}^T \frac{\partial \mathbf{R}_e(x)}{\partial x} \mathbf{u} + \\ \mathbf{x}^T \left[\frac{\partial(\mathbf{A}_e(x)\mathbf{x})}{\partial x} \right]^T P_e(x)\mathbf{x} + \left[\frac{\partial(\mathbf{B}_e(x)\mathbf{u})}{\partial x} \right]^T P_e(x)\mathbf{x} = 0\end{aligned}\quad (2.46)$$

Esta é uma condição de otimalidade que satisfaz a solução de controle sub-ótimo localmente. No tempo infinito, no caso padrão do Regulador linear quadrático (com matrizes de peso do funcionamento com coeficientes constantes) verifica-se que esta equação é localmente satisfeita.

3. construir o controle não-linear realimentado (BANKS; LEWIS; TRAM, 2007)

$$\begin{aligned}\mathbf{u} &= -\mathbf{S}_e(x)\mathbf{x} \\ \mathbf{S}_e(x) &= \mathbf{R}_e^{-1}(x)\mathbf{B}_e^T(x)P_e(x)\end{aligned}\quad (2.47)$$

Para alguns casos especiais, como sistemas com pouca dependência do estado ou com poucas variáveis de estado, a equação (2.45) pode ser resolvida de forma analítica (SHAWKY et al., 2007). Por outro lado, uma solução numérica pode ser obtida com uma taxa de amostragem suficientemente grande. Uma aproximação, com estabilidade local, do sistema de malha fechada é resultado do uso da técnica das equações de Riccati dependentes do estado, não-lineares.

Lema: Para uma parametrização dos coeficientes dependentes do estado $\mathbf{A}_e(x)\mathbf{x}$, $\mathbf{A}_e(0)$ é a linearização de $f(x)$ sobre o zero equilíbrio.

Prova: sejam $\mathbf{A}_1(x)$ e $\mathbf{A}_2(x)$ duas parametrizações distintas de $f(x)$ e seja $\tilde{\mathbf{A}}(x) = \mathbf{A}_1(x) - \mathbf{A}_2(x)$. Então, $\tilde{\mathbf{A}}(x)\mathbf{x} = 0$ para todo x e

$$\frac{\partial \tilde{\mathbf{A}}(x)\mathbf{x}}{\partial x} = \tilde{\mathbf{A}}(x) + \frac{\partial \tilde{\mathbf{A}}(x)}{\partial x}\mathbf{x} = 0. \quad (2.48)$$

Como o segundo termo do lado direito é zero em $x = 0$, segue que $\tilde{\mathbf{A}}(x) = 0$. Isto implica que $\mathbf{A}_1(0) = \mathbf{A}_2(0)$. Portanto, a parametrização avaliada em zero é única. Sem perder a generalidade, considere uma parametrização dada por $\mathbf{A}_1(x)$. A linearização do sistema é dada por

$$\dot{\mathbf{z}} = \nabla f(0)\mathbf{z}, \quad (2.49)$$

mas

$$\nabla f(x) = \mathbf{A}_1(x) + \frac{\partial \tilde{\mathbf{A}}(x)}{\partial x}\mathbf{x}, \quad (2.50)$$

logo $\nabla f(x) = \mathbf{A}_1(x)$, a qual foi mostrada ser a única para todas as parametrizações.

É assumido que existe solução da SDRE para todo x na vizinhança da origem considerada. Desta forma, o par $(\mathbf{A}_e(x), \mathbf{B}_e(x))$ é uma parametrização estabilizável. Uma consequência lógica é que a solução existe em $x = 0$, e que $\mathbf{P}_0 = \mathbf{P}_e(0)$ é solução do sistema de equações algébricas de Riccati:

$$\mathbf{A}_e^T \mathbf{P}_e + \mathbf{P}_e \mathbf{A}_e - \mathbf{P}_e \mathbf{B}_e \mathbf{R}_e^{-1} \mathbf{B}_e^T \mathbf{P}_e + \mathbf{Q}_e = 0. \quad (2.51)$$

Teorema 3: (MRACEK; CLOUTIER, 1998) Assume-se que a parametrização dos coeficientes dependentes do estado seja escolhida de forma que a coluna $\mathbf{A}_e(x) \in C^1$ em torno da vizinhança sobre a origem, e que os pares $(\mathbf{A}_e(x), \mathbf{B}_e(x))$ e $(\mathbf{C}_e(x), \mathbf{A}_e(x))$ sejam, no sentido linear para todo x pertencente à vizinhança sobre a origem, ponto a ponto, estabilizáveis e detectáveis, respectivamente. Então o regulador não-linear SDRE produz uma solução em malha fechada que é localmente e assintoticamente estável (MOLTER, 2008).

Segundo (MRACEK; CLOUTIER, 1998), um fator importante do método SDRE é que ele não cancela os benefícios que podem provir das não-linearidades do sistema dinâmico. A razão para isto pode ser que ele não exige nenhuma inversão dinâmica e nem linearizações na realimentação do sistema não-linear.

O procedimento para aplicar o método SDRE em um sistema dinâmico é (SHAWKY et al., 2007):

1. Transformar o sistema dinâmico em espaço de estados e parametrizar o modelo na forma de coeficientes dependentes do estado;
2. Mensurar o estado do sistema $x(t)$, aplicar as condições iniciais;

3. Analisar a controlabilidade da matriz $\begin{bmatrix} \mathbf{B}_e & \mathbf{A}_e\mathbf{B}_e & \dots & \mathbf{A}_e^{n-1}\mathbf{B}_e \end{bmatrix}$;
4. Se controlável:
 - a) Resolver as equações de Riccati dependentes do estado. Calcular o sinal de entrada através da equação dada por \mathbf{u} ;
5. Se não controlável:
 - a) Usa a solução da iteração anterior no passo 4.a para o cálculo da equação dada por \mathbf{u} ;
6. Assumir o valor da saída do sistema como o novo valor inicial e repetir o procedimento para mensurar o próximo valor do estado, isto é, voltar ao passo 3;
7. Repetir até o critério de parada adotado ser alcançado. Fim.

Na Figura 6 e a partir dos procedimentos matemáticos do método de controle SDRE descritos, pode-se notar a dificuldade e o número de operações matemáticas envolvidas para o cálculo de cada sinal de controle, principalmente a necessidade de analisar a controlabilidade e resolver a equação de Riccati a cada operação. Isso deixa a técnica muito difícil de ser implementada em sistemas reais. Como pode ser visto nos artigos: (MAURO et al., 2013; MENON; LAM; CHENG, 2002) nos quais os autores necessitam implementar o controle SDRE em computadores de alto desempenho, devido à grande dificuldade de linguagem e processamento dos dispositivos para controle embarcado.

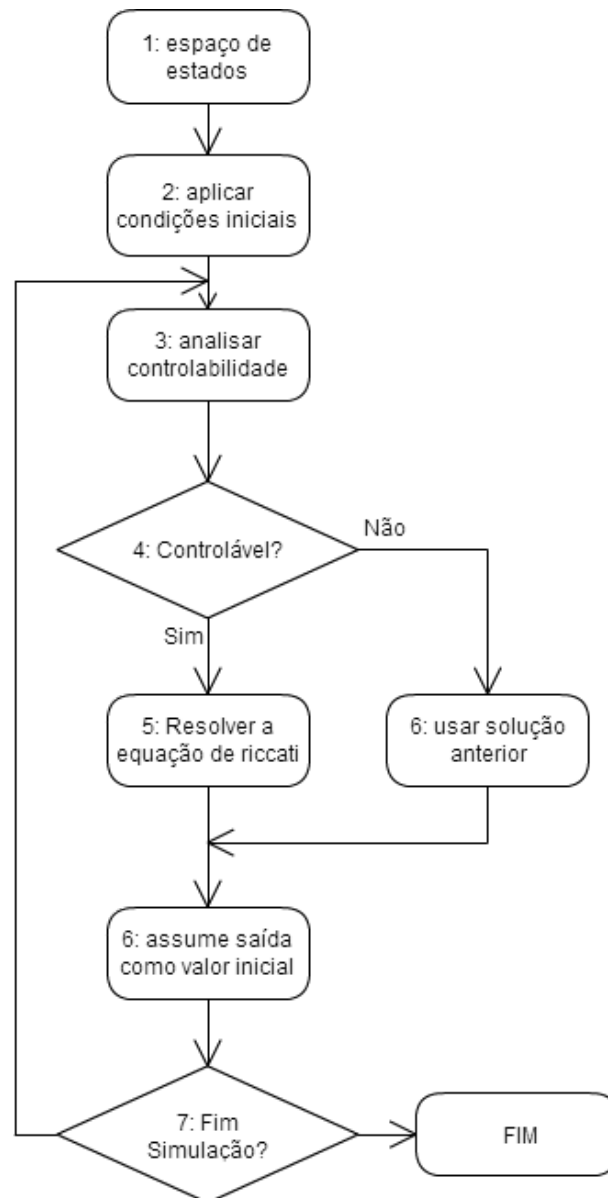


Figura 6 – Fluxograma SDRE (Autoria própria)

3 REDES NEURAIS ARTIFICIAIS

As redes neurais artificiais (RNAs) são modelos matemáticos inspirados no funcionamento do cérebro biológico, o qual possui uma arquitetura que tem basicamente neurônios e interligações entre eles. Cada neurônio é uma unidade de processamento no cérebro que possui milhares de neurônios que executam um processamento individual simultaneamente, caracterizando assim um sistema de processamento paralelo. Isso é muito diferente dos processadores digitais atuais, nos quais a maioria do processamento é realizado em série (HAYKIN, 2001). Do estudo do funcionamento do cérebro biológico foram arquitetadas formas de criação de redes neurais artificiais, implementadas normalmente em linguagens computacionais ou ainda em circuitos eletrônicos.

Neste capítulo esses modelos matemáticos são estudados, analisando suas principais características e tipos. Com a evolução das pesquisas sobre redes neurais artificiais, várias redes foram criadas, as quais diferem em uma série de características, tais como: o funcionamento do neurônio, o padrão de conectividade entre eles e a forma com que a rede neural será treinada.

Uma maneira de classificar é a partir da forma das interligações entre os neurônios. Essa classificação pode ser dividida em dois grupos: as *feedforwards*, são redes que não possuem realimentação, ou seja, as informações anteriores não influenciam no valor corrente da rede; outro tipo são as redes neurais recorrentes (*recurrent neural networks* (RNN)), as quais possuem dependência temporal, desta forma, sinais passados exercem influência sobre o comportamento do sistema (HAYKIN, 2001; SILVA; SPATTI; FLAUZINO, 2010). Para os dois tipos serão mostradas alguns modelos criados, modo de funcionamento, treinamento e outras informações importantes.

3.1 HISTÓRICO

Os primeiros idealizadores que começaram os estudos sobre as redes neurais artificiais foram Warren McCulloch e Walter Pitts (MCCULLOCH; PITTS, 1943). Propuseram o primeiro modelo matemático o qual se comportou de maneira análoga ao neurônio biológico. O objetivo do trabalho eram cálculos lógicos para expressar comportamentos existentes em sistemas nervosos biológicos.

Para (ROSENBLATT, 1958) o cérebro não funcionava apenas como um circuito lógico, e sim como um associador adaptável de padrões. Nesse trabalho também se denominou a arquitetura da rede neural como *perceptron*, para o qual foi elaborado um algoritmo de treinamento, o qual é a principal influência na resposta da rede neural. Além disso, o pesquisador comprovou a capacidade da rede neural artificial em convergir para padrões linearmente separáveis.

(WIDROW; HOFF, 1960) apresentaram um novo método de treinamento da rede neural

mantendo a mesma arquitetura da rede neural *perceptron*, utilizando uma representação matemática diferente da de McCulloch e Pitts. Esses autores apresentaram o algoritmo LMS (*Least Mean Square*) utilizando-o para formar a rede neural chamada de *Adaline*. Porém, (MINSKY; PAPER, 1969) comprovaram matematicamente de que o *perceptron* não era capaz de executar funções lógicas como o ou-exclusivo (HAYKIN, 2001).

Em 1974, Werbos (WERBOS, 1974) propôs uma forma geral de treinamento de rede neural. (RUMELHART; HINTON; WILLIAMS, 1986) utilizaram esse mesmo método para formar uma das redes neurais mais utilizadas na prática, a rede *perceptron de múltiplas camadas* (MLP), na qual o algoritmo de treinamento baseia-se na noção de retropropagação de erro (*error backpropagation*).

Conforme exposto anteriormente, os trabalhos publicados até o início da década de 1980, em sua maioria, tratavam apenas de redes neurais *feedforward*, e então em (HOPFIELD, 1982) foi elaborada uma rede neural recorrente. O autor colocou realimentações dos neurônios no sistema e demonstrou que a rede possuía um número finito de estados de equilíbrio. Isso fez com que um elemento memória passasse a fazer parte da rede neural.

Na mesma época que a rede neural estava sendo criada em 1948, Alan Turing fez uma proposta sobre as redes neurais e gerou o termo *máquinas desorganizadas* (TURING, 1968). Tais redes apresentavam paradigmas conexionistas com uma grande relação de proximidade dos trabalhos anteriormente citados. Turing relacionou o comportamento da rede sendo semelhante ao do cérebro humano. Porém, este trabalho foi apenas publicado após a sua morte, em 1968.

Em (JAEGER, 2001) foi proposta a rede neural com estado de eco (*Echo State Network*), a qual será utilizada para o desenvolvimento deste trabalho, e será apresentada ainda neste capítulo. Em (HUANG; ZHU; SIEW, 2004) outra rede proposta foi chamada de máquina de aprendizado extremo (*extreme learning machine*).

3.2 O FUNCIONAMENTO DO CÉREBRO BIOLÓGICO

O cérebro é o centro do sistema nervoso, que funciona como um processador que toma as decisões para realizar ações a partir de estímulos de entrada. Como exemplificado na Figura 7, que é uma versão simplificada do funcionamento do sistema nervoso, um estímulo é realizado nos receptores, os quais passam os estímulos para a rede neural, que os processa a fim de decidir quais serão as informações enviadas aos atuadores que irão responder de alguma forma. Nota-se que nessa figura as setas são bidirecionais, dando assim ideia de recorrência no processamentos das informações vindas dos receptores, declarando que não existem apenas informações no sentido Estímulo para Resposta. Os receptores são sensores que leem informações externas e fazem parte dos sentidos como: audição, tato, paladar e etc. Os receptores recebem estes sinais externos, e então passam a informação na forma de impulsos elétricos. Através destes impulsos também é realizada toda a comunicação interna

entre os neurônios, e também para a realização da atuação (uma saída), como por exemplo o movimentar de um músculo (HAYKIN, 1994).



Figura 7 – Representação do sistema nervoso (Autoria própria)

3.3 NEURÔNIOS

O componente básico de uma rede neural é o neurônio. Ele é basicamente composto por entradas, saídas e uma unidade de processamento. São unidades de processamento capazes de produzir e transmitir sinais de informação em resposta a um conjunto de estímulos de entrada.

Nas arquiteturas de redes neurais, tanto biológicas quanto artificiais, uma série de neurônios interligam-se com outros neurônios, e neste conjunto cada um deles toma uma decisão expressada em sua saída. As redes neurais dão ideia de processamento em paralelo, pois todos os neurônios executam o processamento de informações ao mesmo tempo (SILVA; SPATTI; FLAUZINO, 2010).

3.3.1 Neurônio Biológico

Apesar do funcionamento do cérebro não ser ainda completamente compreendido, sabe-se que sua estrutura é basicamente formada por neurônios e as conexões entre eles. A complexidade é presente devido à grande quantidade de neurônios e suas interligações. O conjunto de neurônios realizando processamento faz com que a rede neural biológica dê ideia de processamento da forma distribuída, e forme ações apropriadas para a sua funcionalidade, como se movimentar, pensar, falar e etc. (SILVA; SPATTI; FLAUZINO, 2010).

Os neurônios biológicos, nos quais são baseados os neurônios artificiais, são formados minimamente por dendritos, corpo (núcleo celular) e o axônio como pode ser visualizado na Figura 8. Um único neurônio possui vários dendritos, os quais representam as entradas do neurônio, cada dendrito recebe um sinal de entrada, o qual vem de outros neurônios ou de sensores. Esse sinal vindo dos dendritos chegam ao corpo do neurônio, que representa uma unidade processadora que irá decidir se o neurônio transmite ou não um sinal. Esse sinal é transmitido a outros neurônios através do axônio, que leva sinal através da sua extensão até os dendritos de outros neurônios (SILVA; SPATTI; FLAUZINO, 2010). A região na qual se passa a informação de um axônio para um dendrito chama-se sinapse (SILVA, 2011).

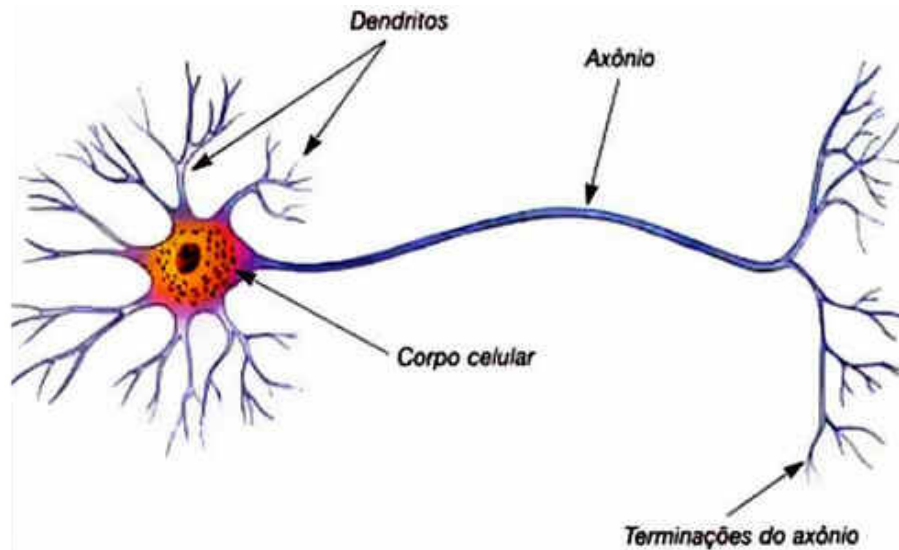


Figura 8 – Neurônio Biológico (MACHADO, 2006)

3.3.2 Neurônio Artificial

Através da observação dos neurônios artificiais se realizam modelos matemáticos para simular o funcionamento. Com este modelo matemático é possível fazer com que a rede neural artificial execute uma tarefa específica através de regulagens da rede neural como treinamento, arquitetura e outros fatores.

O primeiro modelo matemático foi feito por McCulloch e Pitts no ano de 1943 (MCCULLOCH; PITTS, 1943) apud (SILVA; SPATTI; FLAUZINO, 2010) mantendo a mesma estrutura básica do neurônio biológico.

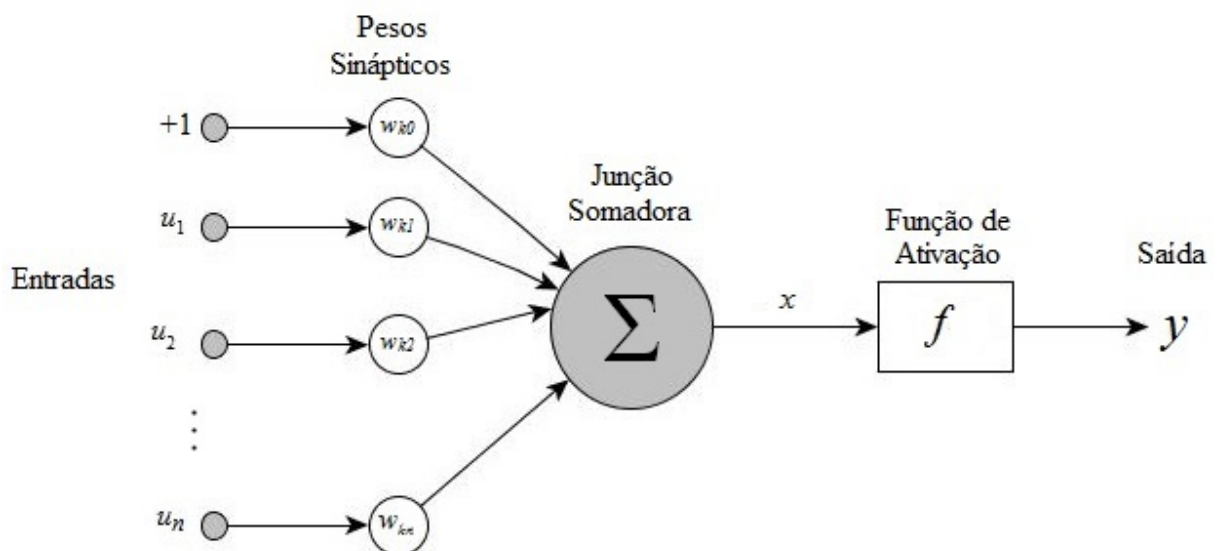


Figura 9 – Neurônio Artificial (SIQUEIRA, 2013)

A Figura 9 representa o modelo matemático de um neurônio artificial, o qual é usado pela maioria das rede neurais, no qual cada entrada u_n do neurônio possui um peso w_{kn} , pelo qual o valor da entrada será multiplicado. Os valores de entrada multiplicados pelos pesos são

somados na Junção Somadora retornando um resultado x , que passa pela Função de Ativação formando o valor de saída da rede neural. A equação que representa o valor da saída de um neurônio é:

$$y = f(\mathbf{w}_k^T \mathbf{u}), \quad (3.1)$$

onde \mathbf{w}_k é o vetor de pesos do neurônio k e \mathbf{u} o vetor de entrada, cada entrada i de \mathbf{u} possui um peso no vetor \mathbf{w}_k . (SILVA; SPATTI; FLAUZINO, 2010)

A função de ativação vai fornecer o valor da saída do neurônio e pode ter vários comportamentos, dependendo da aplicação e o modelo da rede neural a ser utilizado. Ela é usada para limitar o tamanho da saída, seu mínimo, seu máximo e os estados, se existirem, entre esse mínimo e máximo. Geralmente a função de ativação é não-linear para os neurônios das camadas escondidas, enquanto nos neurônios na camada de saída é usada a função identidade (HAYKIN, 1994). A seguir serão apresentadas algumas funções de ativação que podem ser utilizadas:

A forma mais simples, usada nos primeiros modelos por McCulloch e Pitts, é quando o neurônio responde segundo uma função degrau.

$$f(x) = \begin{cases} 1, & \text{se } x \geq 0 \\ 0, & \text{se } x < 0 \end{cases} \quad (3.2)$$

A função tangente hiperbólica é uma função muito utilizada como ativação, e sempre assumirá valores entre -1 e 1, sendo definida por:

$$f(x) = \frac{1 - e^{(-ax)}}{1 + e^{(-ax)}}. \quad (3.3)$$

O valor de a dá a inclinação da função. O comportamento da função hiperbólica está na Figura 10. A Tangente hiperbólica é muito utilizada como função de ativação pois (HAYKIN, 2001):

- é uma função contínua e diferenciável em todos os pontos, permitindo ajustes com algoritmos baseados no gradiente descendente;
- existe uma saturação da saída do neurônio, o que evita que a saída divirja;
- podem ser utilizadas para criar vários tipos de mapeamento, pois próximo da origem apresentam caráter quase linear, e próximos à saturação são fortemente não-lineares.

3.4 ARQUITETURA DAS REDES NEURAIS ARTIFICIAIS

Uma rede neural artificial é um modelo computacional que processa informações e retorna respostas através de um modelo matemático. Este modelo opera através dos seus neurônios bem como a ligação entre esses. Existem várias arquiteturas diferentes de redes

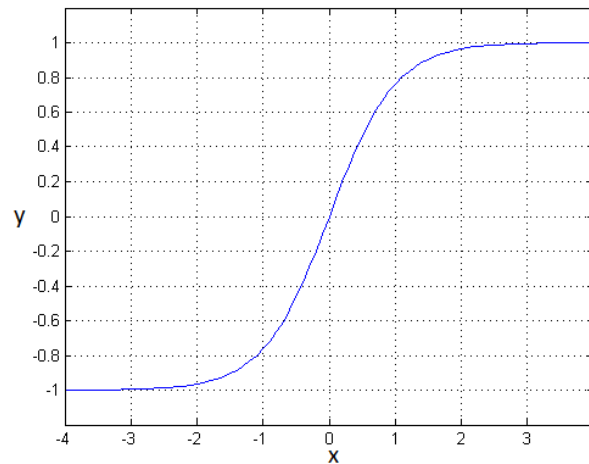


Figura 10 – Função tangente hiperbólica (Autoria própria)

neurais, e as diferenças estão em várias características da rede como: modo de interligação entre os neurônios; diferentes funções de ativação; se as informações são realimentadas na rede (redes neurais recorrentes) ou não (*Feedforward*); a forma de treinamento da rede; e se será ou não colocado um limiar de ativação (valor fixo em uma das entradas dos neurônios) (SILVA, 2011).

3.4.1 Redes Neurais *Feedforward*, a MLP (*multilayer perceptron*)

As redes neurais *feedforward*, possuem um mapeamento estático da entrada para a saída. As informações oriundas da camada de entrada (u) seguem sempre o mesmo fluxo em direção à saída para saídas (y), e nunca para o sentido contrário ou para neurônios de uma mesma camada. Depois de chegar à saída, aquela informação não é mais usada. Portanto, não existe realimentação dentro de uma rede neural do tipo *feedforward* (SILVA; SPATTI; FLAUZINO, 2010).

A Figura 11 representa uma rede neural de multicamadas *feedforward* chamada de MLP (*multilayer perceptron*). A característica de multicamadas pode ser vista na Figura 11 na qual existem a camada de entrada, as camadas intermediárias (chamadas também de camadas ocultas ou escondidas) e mais uma camada de saída. A camada intermediária é a que constrói uma estrutura de multicamadas, podendo existir mais camadas intermediárias. Normalmente todos os neurônios possuem a mesma função de ativação, sendo que o que diferencia um neurônio do outro são os pesos e o padrão de conectividade entre eles na rede (BARRETO, 1997).

A rede neural artificial MLP é uma generalização do *perceptron* de (ROSENBLATT, 1958). Essa rede possui capacidade de aproximar qualquer tipo de função não-linear contínua, limitada, diferenciável e com as entradas com espaços e precisão definidas. Essa definição foi demonstrada por (CYBENKO, 1989), e chamada de teorema de aproximação universal. Contudo, ele não fornece indicativos com respeito ao número de neurônios (e/ou camadas), tam-

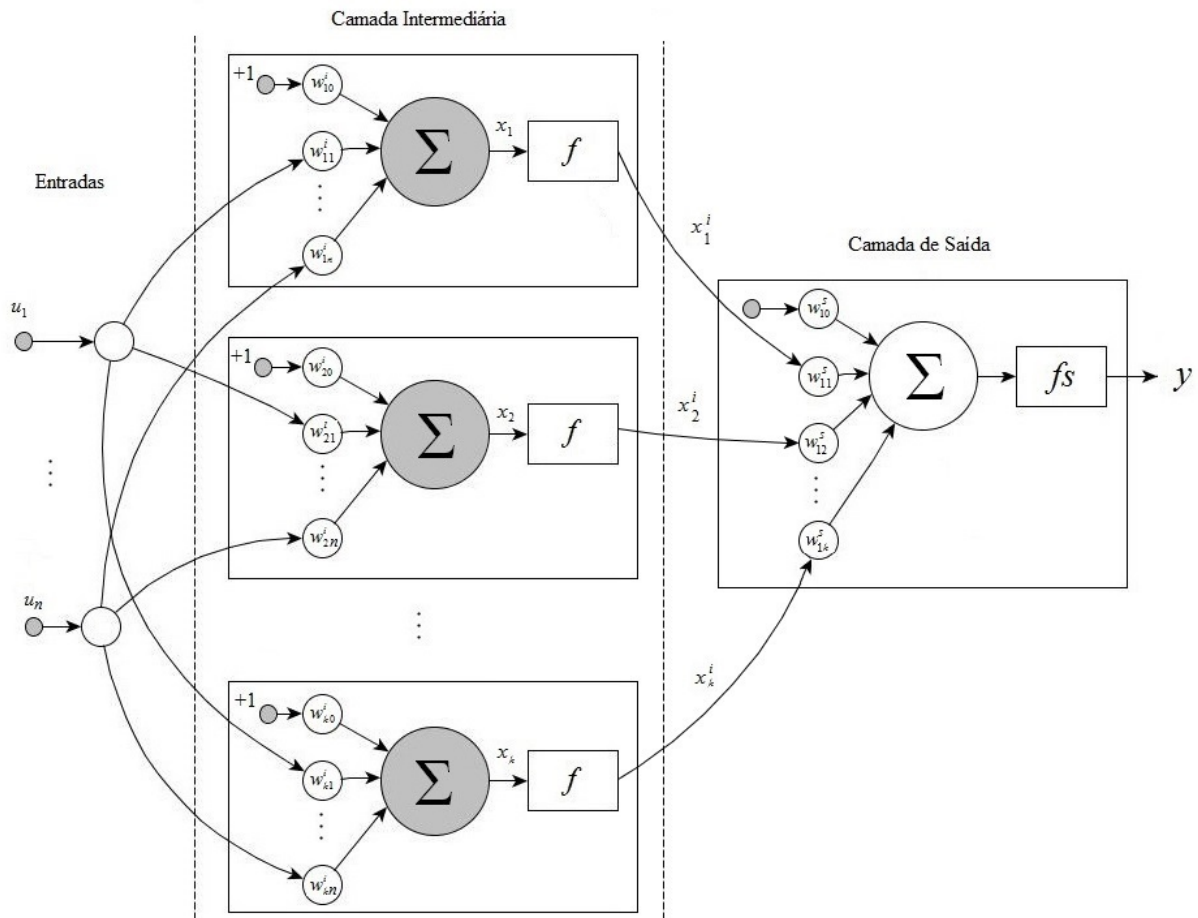


Figura 11 – Rede neural MLP (*multilayer perceptron*) (SIQUEIRA, 2013)

pouco indica um método eficiente para os ajustes dos parâmetros que nos leva seguramente a obter uma rede ótima.

Ainda pela Figura 11 pode-se verificar que existem n entradas u , as quais alimentam todos os K neurônios da primeira camada intermediária representada por i . Essas entradas são multiplicadas por cada peso de cada neurônio, e então são passadas pela função de ativação, resultando na saída do neurônio. Esses valores de saída dos neurônios serão as entradas para a próxima camada, a intermediária i ou a camada de saída s . No caso de chegar à camada de saída, a entrada da camada é multiplicada pelos pesos, soma-se e aplica-se a função de ativação f_s , obtendo-se assim, o resultado da rede neural. A equação 3.4 representa a equação de uma MLP com uma camada intermediária e L saídas.

$$y_l = f_s \left(\sum_{k=0}^K w_{lk}^s \left[f \left(\sum_{n=0}^N w_{kn}^i u_n \right) \right] \right), l = 1, 2, \dots, L \quad (3.4)$$

3.4.2 Redes Neurais Recorrentes

As redes neurais recorrentes possuem realimentações entre os seus neurônios, ou seja, suas conexões não são apenas no sentido entrada para a saída como na *feedforward*. Se existe uma ligação que caracterize como um retorno de uma informação de um neurônio ou entrada anterior, então se considera esta rede neural como recorrente. Os dados da entrada anterior influenciam no valor corrente da rede e isso confere um elemento de memória, pois os valores dos estados anteriores da rede influenciam nos valores internos dos neurônios da rede (estado) e conseqüentemente, na saída da rede. Este tipo de rede neural se caracteriza como uma possível solução para problemas com dependência temporal entre as amostras. Essas redes são usadas principalmente em previsão de séries temporais, otimização de funções e identificação de sistemas, controle de processos. O próprio cérebro biológico funciona desta maneira (HAYKIN, 2001).

As redes neurais recorrentes, da mesma forma que as *feedforward*, também são aproximadores universais, porém com a funcionalidade de possuir memória. No entanto, elas estão associadas a um processo de treinamento custoso e podem ter comportamentos indesejados como convergência prematura ou lenta. Um novo tipo de rede chamado por Verstraeten por Computação de Reservatório (RC) descreve redes neurais recorrentes que se utilizam de uma arquitetura e treinamento que facilitam a utilização de redes neurais recorrentes. Redes neurais do tipo RC foram descritas em 2001, quando dois autores de forma independente idealizaram uma nova forma de implementar uma rede neural recorrente: Wolfgang Mass chamando de Liquid State Machine (LSM) (MAASS; NATSCHLÄGER; MARKRAM, 2002) e Herbert Jaeger chamando de Echo State Network (ESN) (JAEGER, 2001). Ambas as redes possuem um reservatório de dinâmicas, o qual é uma estrutura de neurônios recorrentes formando um sistema dinâmico não linear, e as conexões entre estes neurônios são geradas aleatoriamente, assim como os pesos destas conexões. Nesse modelo, além desse reservatório existe uma camada de saída a qual é treinada para que processe os dados do reservatório de forma desejada para então mostrar uma resposta adequada (VERSTRAETEN, 2009) .

3.4.3 Treinamento de Redes Neurais

Cada diferente tipo de arquitetura de rede neural é capaz de gerar um sinal de saída a partir de entradas, porém, para que a saída da rede seja uma saída desejada a partir de um sinal de entrada, além do ajuste de como os neurônios estarão conectados na rede, um ajuste dos pesos dos neurônios da rede deve ser realizado. Esse ajuste dos pesos da rede é chamado de treinamento, o qual pode ser feito de diversas formas diferentes dependendo das características da rede, na maioria das vezes, através de amostras de dados, devendo existir um conjunto de entradas associados a um conjunto de saídas desejadas. O treinamento é, normalmente, expressado em um algoritmo, e regula os pesos dos neurônios a partir da amostra de dados. (SILVA, 2011).

O treinamento é a forma na qual os pesos serão regulados a partir de amostras de entradas e suas respectivas saídas. Esse treinamento tem como objetivo final fazer com que a rede tenha um comportamento semelhante com erros pequenos em relação ao comportamento dos dados usados para o treinamento dessa rede.

A forma de se regular os pesos dos neurônios são expressados em algoritmos de treinamentos, os quais definem os passos a serem dados para que o ajuste aconteça da forma correta. É importante também que a rede neural não só atenda aos dados de treinamento da rede, mas a rede deve generalizar uma solução, onde novos dados de entradas, que não foram mostrados à rede no treinamento, devem gerar na saída respostas desejadas (BARRETO, 1997; FRANK, 2008).

O conjunto de dados de treinamento e testes possuem dados de entradas e saídas desejadas, e são amostras do comportamento ideal para várias possibilidades de entrada na rede. Desse conjunto de amostras, uma parte é usada para o ajuste dos pesos da rede neural (treinamento), e outra parte é usada para testar a rede. O teste é feito colocando na entrada da rede o conjunto de entradas, com o objetivo que na saída da rede esteja o valor da saída desejada. O objetivo de deixar uns dados reservados para realizar o teste da rede neural é fazer com que haja um teste da rede neural com dados os quais a rede neural nunca viu antes (durante o treinamento), e isso é chamado de generalização (SILVA, 2011).

Ainda em se falando em treinamento de rede neural, podemos citar três principais maneiras para a realização do treinamento da rede neural:

- supervisionado: os valores dos pesos da rede são modificados a partir da análise do erro: diferença entre o valor desejado e o valor de saída da rede. Operações matemáticas são utilizadas para então determinar o novo valor dos pesos dos neurônios.
- não-supervisionado: o treinamento é realizado de forma que os pesos sejam atualizados para que a rede passe a identificar padrões a partir do conjuntos de dados de entrada.
- por reforço: toda saída da rede é comparada com um saída desejada e, a partir disso, modificações no pesos da rede são realizadas para se verificar melhorias no comportamento.

O treinamento da rede neural MLP da Figura 11, é realizado através de métodos que adequem os pesos para uma melhor resposta a cada iteração. Para estas adequações dos pesos é utilizado um algoritmo de retropropagação de erro (*backpropagation*) (RUMELHART; HINTON; WILLIAMS, 1986). O processo de treinamento da MLP normalmente é feito de forma supervisionada com duas principais etapas: a primeira etapa consiste em alimentar a rede com dados de entrada do treinamento, a fim de se ter o valor que estas entradas gerarão na saída da rede. Esta etapa é chamada etapa *forward*. A segunda etapa chamada de *backward* é realizada a partir da comparação entre a saída que a rede trouxe na etapa *forward* e o valor desejado dos

dados de treinamento. Esse erro é propagado para a entrada da rede pelo algoritmo de retropropagação de erro. A partir destes valores de erros retropropagados os valores dos pesos são ajustados com o objetivo de diminuir o erro na saída da rede. Esse método é repetido até que se chegue a erros aceitáveis na saída da rede.

Para o treinamento de redes neurais recorrentes, por conta das realimentações existentes, normalmente são usados métodos massivos e complicados para a obtenção dos pesos. Exemplos desses métodos podem ser vistos em (JAEGER, 2008) no qual o autor apresenta algumas técnicas de treinamento de redes neurais recorrentes como *backpropagation through time (BPTT)*, *real-time recurrent learning (RTRL)*, e *Kalman filtering based techniques (EKF)*. Essas técnicas necessitam, normalmente, de familiarização e conhecimento aprofundado na rede recorrente para realização de ajustes necessários para que não haja divergências na saída da rede. Por outro lado, as Redes Neurais com estados de eco (ESN), estão sendo amplamente usadas por conta de serem redes neurais recorrentes, que necessitam de um treinamento relativamente fácil e com grande eficiência.

3.4.4 Rede Neural Com Estado de Eco

Uma ESN é uma rede neural recorrente, pois possui vários neurônios cujas saídas são realimentadas na rede. Essas realimentações acontecem predominantemente na camada intermediária, conhecida como reservatório de dinâmicas, gerando assim memória dos estados anteriores da rede. O conjunto das saídas de todos estes neurônios do reservatório de dinâmicas é denominado de estado simbolizado aqui pelo vetor $\mathbf{x}(t)$ (MASSAR; MASSAR, 2013). Este estado depende de todas as entradas \mathbf{u} anteriores para que o seu valor seja definido, ou seja, existe uma função E que retorna o valor do estado $\mathbf{x}(t)$ a partir de todas as entradas: $\mathbf{x}(t) = E(\mathbf{u}(t), \mathbf{u}(t-1), \mathbf{u}(t-2)\dots)$, onde \mathbf{x} é um reflexo das entradas correntes e das entradas anteriores da rede. Isso acontece graças à propriedade de estado de eco.

As redes neurais recorrentes (RNNs) são sistemas não-lineares complexos por conta dos laços de realimentações. As RNNs são conhecidas por serem sistemas muito instáveis e com algoritmos de treinamentos complicados e de grande custo computacional (HAYKIN, 2001). Isso fazia com que as redes neurais recorrentes fossem pouco utilizadas em relação às RNAs *feedforward*. Uma das alternativas de contornar a maneira difícil de utilizar as redes neurais recorrentes clássicas é a partir da ESN, cujo o treinamento é realizado de forma relativamente fácil, e esse treinamento se dá basicamente com o ajuste apenas da camada de saída (JAEGER, 2001).

A Figura 12 representa uma rede de estado de eco, possuindo as entradas \mathbf{u} na camada de entrada, cuja matriz \mathbf{W}^{in} representa os pesos dos neurônios de entrada para a próxima camada denominada intermediária. Na ESN tal camada é chamada de reservatório de dinâmicas, cujos pesos da realimentação dos sinais de seus neurônios são representados pela matriz \mathbf{W} . A saída da camada intermediária, que é o estado \mathbf{x} do reservatório dinâmico, são os valores que entram na camada de saída, e seus pesos são representados pela matriz \mathbf{W}^{out} . A

forma que o reservatório de dinâmicas é elaborado é o principal diferencial da rede com estado de eco, sendo que os pesos que realizam as interligações entre os neurônios do reservatório e as conexões que vêm da camada de entrada são gerados aleatoriamente (SALMEN; PLOGER, 2005) apud (JAEGER, 2008) .

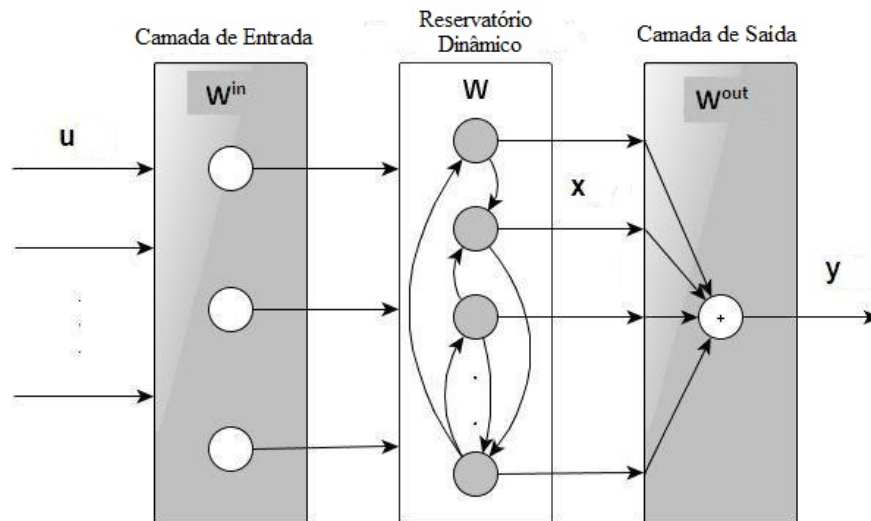


Figura 12 – Echo State Network (SIQUEIRA, 2013)

3.4.4.1 Propriedade de estados de eco

O reservatório dinâmico da ESN é a principal razão pelo processo simples de determinar os pesos da camada de saída, pois a ativação de cada um dos neurônios do reservatório de dinâmicas é, na verdade, uma transformação não-linear do histórico mais recente do sinal de entrada. (JAEGER, 2001) comprova a *propriedade de estados de eco* em seu trabalho que se refere aos estados $x(t)$ da rede que são assintoticamente independentes da condição inicial. Dessa forma, pode-se dizer que apenas as entradas mais recentes influenciam no valor de x e as entradas que estão mais distantes influenciam em valores muito pequenos no valor do estado x .

O trabalho de Jaeger aponta as condições para que a propriedade de estado de eco seja válida: os sinais de entrada da rede neural devem ser extraídos de um espaço compacto \mathbf{B} ; os estados tem de estar contidos em um conjunto compacto $A \subset \mathbb{R}^N$ de estados admissíveis, caracterizando que os valores de x_n sempre estarão dentro de A (JAEGER, 2001).

A parametrização na rede neural, a qual faz com que essas condições para que a propriedade aconteça, estão no processo de geração da matriz de pesos \mathbf{W} . As condições da matriz \mathbf{W} são: que o seu valor singular deve estar dentro do círculo real unitário ($|\sigma_{max}(\mathbf{W})| < 1$), e que o maior autovalor esteja menor que 1, ou seja, ($|r_{max}(\mathbf{W})| \leq 1$) (JAEGER, 2001).

Em 3.4.4.5 formas de realizar esses ajustes são demonstradas.

3.4.4.2 Reservatório dinâmico

Para a construção do reservatório em uma linguagem de programação, armazenam-se os pesos de forma matricial. É preciso uma matriz que armazene os pesos de cada uma das K entradas para cada um dos N neurônios do reservatório dinâmico. Ou seja, cada entrada tem uma ligação entre todos os neurônios do reservatório. Esses valores são armazenados em uma matriz chamada aqui de $\mathbf{W}^{\text{in}} \in \mathbb{R}^{K \times N}$.

Para as ligações que fazem parte da realimentação do reservatório dinâmico, ou seja, ligações do reservatório para o próprio reservatório, é necessária uma estrutura que armazene os pesos de cada N neurônio para todos os próprios N neurônios do reservatório. A matriz $\mathbf{W} \in \mathbb{R}^{N \times N}$ é usada para representar estes pesos (JAEGER, 2008).

3.4.4.3 Estado do reservatório de dinâmicas

O estado do reservatório é o valor das saídas de seus neurônios. Os estados dos neurônios devem ser armazenados em uma estrutura para que seja usada pelo próprio reservatório e pela camada de saída. Para armazenar este estado é necessário um vetor de N posições, ou seja, um valor para cada neurônio. Como já visto anteriormente, esse estado é chamado aqui de \mathbf{x} , e para o seu cálculo é necessária a Equação (3.5) (JAEGER, 2008).

$$\mathbf{x}(t + 1) = f(\mathbf{W}\mathbf{x}(t) + \mathbf{W}^{\text{in}}\mathbf{u}(t + 1)) \quad (3.5)$$

em que: o termo $\mathbf{W}\mathbf{x}(t)$ representa a parte da realimentação do estado anterior do reservatório. O termo $\mathbf{W}^{\text{in}}\mathbf{u}(t + 1)$ representa as novas entradas do sistema para o reservatório. A função de ativação f normalmente usada na ESN é a tangente hiperbólica, a qual é necessária para satisfazer a propriedade de estado de eco. Em \mathbf{x} haverá o estado do reservatório corrente. Esse estado será usado para o cálculo das saídas da rede. (OUBBATI, 2006).

3.4.4.4 Saída da rede

A saída da rede é definida pelo valor dos neurônios da camada de saída, a qual recebe como entrada o estado do reservatório (\mathbf{x}). As entradas da ESN devem estar ligadas ao reservatório e podem estar ligadas diretamente à camada de saída também. Desta forma, a saída é definida pela equação (3.6).

$$\mathbf{y}(t + 1) = f(\mathbf{W}^{\text{out}}[\mathbf{u}(t + 1), \mathbf{x}(t + 1)]) \quad (3.6)$$

Os únicos pesos que são calculados (treinados) na rede de estado de eco são os pesos da camada de saída que são representados por uma matriz \mathbf{W}^{out} que possui dimensão $(K + N) \times L$, mas o uso das ligações vindas da entrada diretamente à saída podem não ser consideradas ficando assim uma matriz de dimensão $N \times L$. O termo $[\mathbf{u}(t + 1), \mathbf{x}(t + 1)]$ significa a concatenação dos vetores de entrada e estado.

3.4.4.5 Determinação dos pesos \mathbf{W} e \mathbf{W}^{in}

Como visto anteriormente, a matriz $\mathbf{W} \in \mathbb{R}^{N \times N}$ carrega o valor dos pesos de todas as conexões entre todos os neurônios do reservatório. Para que se possa obter um reservatório que satisfaça a *propriedade de estado de eco*, primeiramente pode-se gerar aleatoriamente os valores para a matriz (JAEGER, 2008). A arquitetura da rede possibilita que todos os neurônios conectem-se a todos os neurônios da rede neural, porém, geralmente apenas algumas conexões existem, ou seja, com pesos diferentes de 0. Assim, é necessário uma matriz esparsa para representar. O padrão esparsa de conexões é utilizado para haver um desacoplamento de grupos de neurônios desenvolvendo uma relação baixa entre os neurônios do reservatório dinâmico. Para estes valores, Jaeger, fez um experimento empírico cujos pesos da matriz \mathbf{W} eram $-0,4$, 0 e $0,4$, onde a probabilidade do valor de um peso ser nulo era de 95%, 2,5% para o valor $0,4$ e 2,5% para o valor $-0,4$ (JAEGER, 2001). Jaeger também usou números aleatórios de -1 à 1 para o reservatório dinâmico em (JAEGER, 2008), também com um grau de esparsidade de 95%. Em (DUTOIT et al., 2009) é explorada a possibilidade de desativar as conexões entre os neurônios de forma a conseguir um melhor grau de esparsidade, sendo o método chamado de poda (em inglês *pruning*).

Depois de gerada a matriz \mathbf{W} esparsa e aleatória, ainda há a necessidade de regular o raio espectral da matriz \mathbf{W} . Para essa regulagem, primeiramente deve-se achar o raio espectral, ou seja, o máximo autovalor da matriz \mathbf{W} . Dividindo a matriz inteira por esse raio espectral, obtém-se então uma matriz \mathbf{W} com raio espectral unitário. Depois disso, a matriz está pronta para ser multiplicada por um valor que determinará o raio espectral da matriz que deve ser regulado de acordo com a forma que se quer que o reservatório se comporte. O raio espectral será a unidade que regulará o eco do reservatório (JAEGER, 2008). Quanto maior o raio espectral, menor será a volatilidade dos dados anteriores no interior do reservatório. Teoricamente quanto maior a necessidade da memória dos estados anteriores da rede, mais próximo de 1 deve ser o raio espectral.

Os valores de \mathbf{W}^{in} são gerados aleatoriamente e regulados arbitrariamente. Normalmente se usam números aleatórios entre -1 e 1 gerados em uma distribuição uniforme.

3.4.4.6 Treinamento da ESN

O treinamento se dá apenas em calcular os pesos dos neurônios da camada de saída, expressadas pela matriz \mathbf{W}^{out} . Esse cálculo se realiza de acordo com amostras de entradas \mathbf{U} e de saída desejadas \mathbf{D} . \mathbf{U} e \mathbf{D} são matrizes relacionadas, cada linha da matriz \mathbf{U} representa uma entrada que tem uma relação com a mesma linha da matriz \mathbf{D} . As linhas estão sequenciadas no tempo, demonstrando o comportamento temporal desejado (JAEGER, 2008).

Determinação dos valores da matriz \mathbf{W}^{out}

Para a determinação dos pesos dos neurônios da camada de saída, representados aqui pela matriz \mathbf{W}^{out} deve-se, primeiramente, organizar os dados para obter o que estará na camada de saída da ESN. Começando pelos dados de entrada: deve-se definir quais dados estarão na entrada da camada de saída.

Como já visto em 3.4.4.4 podem haver ligações da entrada diretamente para a saída, ou apenas do reservatório para a saída (ERHAN, 2004). Esses dados são representados na Equação (3.6) pelo termo $[\mathbf{u}(t+1), \mathbf{x}(t+1)]$ que é multiplicado por \mathbf{W}^{out} pela direita. Para a amostra \mathbf{U} e \mathbf{D} precisa-se guardar o histórico do comportamento do reservatório a cada nova entrada (cada linha da matriz \mathbf{U}). As entradas são mostradas à rede neural e assim, o reservatório mostra o seu comportamento para esses dados de entradas. Esse comportamento é um histórico de estados calculados pela Equação (3.5). A cada nova entrada, o estado do reservatório vai se alterando e se caracterizando pela entrada atual e pelas entradas anteriores. Esse histórico de estados deve ser calculado e armazenado em uma matriz chamada \mathbf{M} . Para montar essa matriz \mathbf{M} pode-se fazer da seguinte forma (JAEGER, 2008).

$$\mathbf{M}(t+1) = [\mathbf{U}(t+1), \mathbf{x}(t+1)] \quad (3.7)$$

Para o treinamento da rede os termos $\mathbf{U}(t+1)$ podem não ser utilizados dependendo da configuração usada. Para cada $\mathbf{M}(t)$ deve ter um respectivo resultado desejado para ajuste dos pesos de \mathbf{W}^{out} . Esse resultado desejado é o próprio \mathbf{D} , que é o resultado da saída da camada. Portanto, é preciso descobrir o resultado do termo $\mathbf{W}_{\text{out}}[\mathbf{u}(t+1), \mathbf{x}(t+1)]$ da Equação 3.6. Deve-se então calcular o valor da saída desejada \mathbf{D} passando pelo inverso da função de ativação da camada de saída. A matriz destes resultados esperados será chamado de \mathbf{T} e é calculado da seguinte forma:

$$\mathbf{T}(t+1) = f^{-1}(\mathbf{D}(t+1)), \quad (3.8)$$

onde f^{-1} representa o inverso da função de ativação usada nos neurônios da camada de saída. Na prática deste trabalho a função de ativação da camada de saída utilizada é a função identidade, não precisando assim utilizar o termo f^{-1} .

Os vetores \mathbf{M} e \mathbf{T} são os dados de entrada e saída da camada de saída antes de passar pela função de ativação. Para calcular os pesos Jaeger propõe, entre outras técnicas, fazer o cálculo de \mathbf{W}^{out} utilizando-se de regressão linear da seguinte forma:

$$\mathbf{W}^{\text{out}} = \mathbf{M}^+ \mathbf{T}, \quad (3.9)$$

onde a matriz \mathbf{M} não é uma matriz quadrada, portanto não possui matriz inversa. O termo \mathbf{M}^+ representa a matriz pseudo-inversa de \mathbf{M} . Para uma matriz \mathbf{E} e sua pseudo-inversa \mathbf{S} , a pseudo-inversa de Moore-Penrose possuem as seguintes propriedades, $\mathbf{ESE} = \mathbf{E}$ e $\mathbf{SES} = \mathbf{S}$,

onde $(SE)^T = SE$ e $(ES)^T = ES$. No MatLab é usado o método de Moore-Penrose para a obtenção da matriz inversa.

Tendo o valor dos pesos no vetor W^{out} , o próximo passo na rede neural é a realização de testes normalmente fazendo a rede funcionar como entrada as amostras U de treinamento. A saída que a rede neural gera é então comparada com os dados da saída do treinamento D . Compara-se a diferença entre a saída da rede e os valores de D através de métodos estatísticos, como o erro médio quadrático, sendo possível assim, calcular a eficiência da rede neural. Para um melhor teste é importante colocar na entrada da rede neural uma sequência de dados que a rede nunca viu (conjunto de testes), mostrando o quanto a rede generalizou o problema quando recebeu os dados de treinamento a partir de cálculos de erro (JAEGER, 2001).

Esta forma de treinar a ESN é relativamente muito fácil em relação aos processos de treinamento das RNNs, os quais normalmente são acompanhados com grande dificuldade no treinamento, e elevam a possibilidade de instabilidade.

3.4.4.7 Aplicações e desenvolvimento das ESNs

(KEITH, 2013) implementado o algoritmo da ESN em processadores gráficos (GPUs) que se utilizam de muitos núcleos em paralelo para realizar operações com características paralelas. Esse trabalho mostrou a possibilidade de se utilizar grandes redes neurais com alto desempenho computacional.

Em (JAEGER, 2008) Jaeger realiza um tutorial de treinamento de redes neurais recorrentes. Para a ESN o autor lançou um diferencial da utilização de matrizes de realimentação da camada de saída para o reservatório dinâmico, demonstrando como a rede neural pode gerar um sinal em sua saída mesmo sem possuir um sinal de entrada.

(BOCCATO, 2013) utiliza a ESN em conjunto com outras arquiteturas de redes neurais, gerando quatro diferentes arquitetura. Realiza testes com as quatro redes neurais a partir de simulações de equalização de canais de comunicação, separação de fontes e predição de séries temporais.

(SIQUEIRA, 2013) realizou um trabalho fazendo uso de uma arquitetura ESN em conjunto com uma rede neural denominada de máquina de aprendizado extremo para previsões de séries e vazões.

Esses exemplos da utilização da ESN entre muitos existentes demonstram a capacidade de diversificação de solução desse tipo de rede neural. No próximo capítulo mais uma forma de utilização da ESN é realizada a partir da rede operando como controle de sistemas. A rede neural é treinada a partir do comportamento do controle SDRE sobre os mesmos sistemas a controlar.

4 ESNS PARA O CONTROLE DO SISTEMA MICROELETROME CÂNICO (MEMS)

4.1 INTRODUÇÃO

A complexidade computacional para a realização do controle através das equações de Riccati dependentes de estado, como visto no capítulo 2, existe por conta da necessidade da resolução da equação de Riccati a cada iteração, fazendo com que o uso desta técnica necessite de processadores de alto desempenho, conseqüentemente, processadores com alto valor comercial. Isso, muitas vezes, pode inviabilizar o uso do controlador em algumas aplicações. Outro fator contra o uso do SDRE é a implementação das equações de estado no dispositivo de controle, pois como visto anteriormente, para calcular o sinal de controle são necessárias as equações de estado do sistema. Para cada novo controle uma grande parte da programação do controlador deveria ser refeita.

As redes neurais com estados de eco se caracterizam por serem sistemas não-lineares, e seu uso se dá principalmente por isso e por possuírem uma relativa facilidade de treinamento. Por conta dessas características, é apresentada neste e no próximo capítulo, a capacidade da rede com estado de eco se comportar como um controlador baseado em SDRE. Isso será realizado a partir dos dados de erro e sinal de controle no controlador SDRE, que serão coletados e utilizados para o treinamento da rede neural ESN. Então, com a rede treinada, ela é colocada para controlar o sistema.

O objetivo de fazer com que a rede se comporte como controlador SDRE é, primeiramente, reduzir o custo computacional, porque a rede trata basicamente de cálculos de multiplicação de matrizes. Outro objetivo é fazer com que a implementação seja facilitada, pois a ESN é sempre a mesma, mudando apenas algumas propriedades e os dados de treinamento. Diferentemente do SDRE, onde a planta a ser controlada deve ser descrita a partir de equações de estados, fazendo com que grandes modificações na programação do controlador sejam necessárias de planta para planta. Para o treinamento da rede neural é necessário apenas que o comportamento do SDRE seja simulado.

Dessa forma, o objetivo é realizar um controlador usando uma rede de estado de eco que se comporte como um controlador SDRE com menos gasto de recurso computacional, e menos gasto para implementação do dispositivo controlador.

Para essa demonstração serão usados dois sistemas com comportamento caótico: o primeiro representa um modelo de um sistema microeletromecânico (MEMS) controlado de acordo com (BASSINELLO, 2011; TUSSET et al., 2012a) já neste capítulo. No Capítulo 5 será analisado o segundo sistema, que é um controle de suspensão veicular complexo composto, demonstrado em (TUSSET et al., 2012b), sendo ambos controlados com sucesso através do controlador SDRE.

O procedimento usado é descrito no fluxograma da Figura 13. Inicialmente modelamos

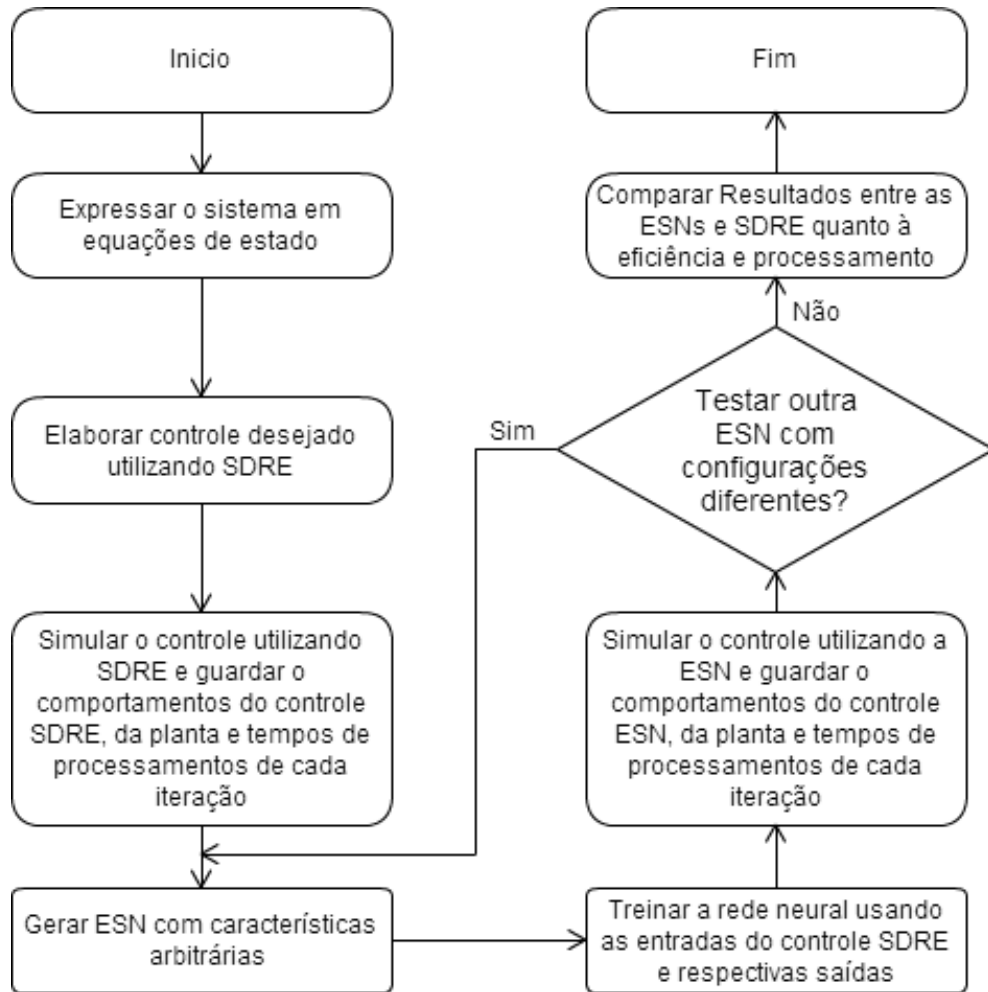


Figura 13 – Fluxograma para realização da simulação (Autoria própria)

o sistema a ser controlado em equações de estados, faz-se a adimensionalização do sistema, e então implementa-se um controle SDRE para levar o sistema a um comportamento desejado. Simulando o controle realizado, obtém-se o conjunto de treinamento a partir do comportamento do SDRE para o aprendizado da rede. No próximo passo é gerada uma ESN com características arbitrárias como: quantidade de neurônios, raios espectrais, tipo de geração de números aleatórios e grau de esparsidade do reservatório de dinâmicas. Tendo a rede neural formada é necessário o treinamento dessa a partir dos dados coletados do comportamento do controle SDRE. Treinada a rede neural, realizam-se testes para capturar o comportamento da rede neural para a solução do controle. Após a realização de diferentes redes neurais, é então analisado os resultados comparando a eficiência da rede neural em relação ao controle e ao custo computacional.

Os passos deste fluxograma serão demonstrados separadamente para os dois sistemas a serem controlados com o auxílio de ESNs.

4.2 SISTEMAS MICROELETROMECHANICOS

Com a evolução dos sistemas microeletromecânicos, esses estão sendo produzidos também a partir de estruturas das indústrias de semicondutores, ou seja, baixo custo e grandes quantidades, fazendo com que os sistemas comecem a entrar em projetos de engenharia como: cabeçotes de impressão a jato de tinta, matrizes de exposição da projeção, acelerômetros de *air bags* e microbombas. Esses são exemplos do uso de sistemas microeletromecânicos. Alguns sistemas, para certos parâmetros, podem ter comportamento caótico, dificultando assim a realização do controle desses (TUSSET et al., 2012a). Para um sistema MEMS existem três elementos importantes: o transdutor de entrada, o ressonador mecânico e o transdutor de saída. O transdutor de entrada transforma o sinal elétrico de entrada em uma força mecânica. O ressonador do MEMS contém um capacitor variante no tempo como elemento central, e é isolado do seu entorno. O transdutor de saída gera um sinal elétrico na saída através do seu movimento mecânico (LUO; WANG, 2002).

A não-linearidade destes sistemas pode surgir pela parte elétrica (capacitores, resistores e indutores) ou mecânica (massa, mola e amortecimento), e certamente pela interação entre eles. Essa não-linearidade faz com que haja necessidade da utilização de controladores robustos para que estes sistemas se comportem de maneira desejada. Como analisado em (TUSSET et al., 2012a), a técnica de controle SDRE se mostra capaz de guiar adequadamente o sistema até o modo de operação desejado.

A ilustração do sistema microeletromecânico está na Figura 14, representado por uma microviga acionada eletrostaticamente. O sistema é composto por duas placas fixas e uma placa móvel entre elas, sobre a qual é aplicada uma tensão $V(t)$ que é composta por uma tensão de polarização DC, V_p , e uma tensão alternada AC, $V_i \text{sen}(\omega t)$. A tensão DC aplica uma força eletrostática na viga, e geralmente muda a posição de equilíbrio (TUSSET et al., 2012a). As placas têm as funções de fornecer eletrodos para formar um capacitor armazenando energia elétrica e fornecer elasticidade ou rigidez mecânica.

O modelo matemático proposto para este sistema é dado por (TUSSET et al., 2012a):

$$m\ddot{x} = -F_k - F_c + F_e, \quad (4.1)$$

onde: F_k é a força conservativa da mola, F_c amortecimento termoelástico e F_e a força elétrica.

A força elétrica entre a carga positiva no movimento da placa central e a carga negativa nas placas laterais é dada por:

$$F_e = -\frac{\partial W^*}{\partial x}, \quad (4.2)$$

onde W^* é a quantidade de energia elétrica armazenada em um capacitor, a qual é dada por:

$$W^* = \frac{1}{2}CV^2. \quad (4.3)$$

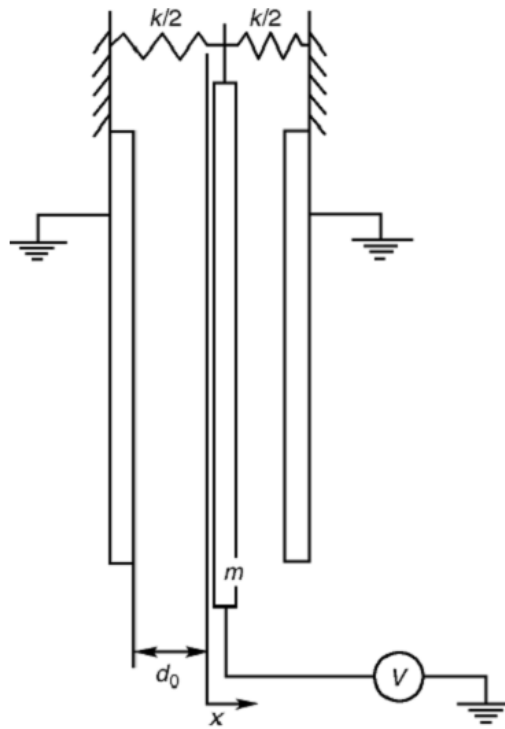


Figura 14 – Sistema microeletromecânico (BASSINELLO, 2011)

Sendo V a tensão elétrica e C a capacitância. A capacitância pode ser obtida de:

$$C = \frac{\varepsilon_0 A}{d}, \quad (4.4)$$

onde, ε_0 é a permissividade no vácuo, d é a distância entre as placas e A a área da placa.

Substituindo (4.4) em (4.3), obtemos:

$$W^* = \frac{1}{2} \frac{\varepsilon_0 A}{d} V^2. \quad (4.5)$$

Pode-se ver na Figura 14 que a distância d entre as placas fixas e a móvel depende da posição de x e de d_0 (distância inicial entre as placas). Considerando que as placas fixas tenham as mesmas características, a quantidade de energia elétrica total armazenada no sistema pode ser obtida de:

$$W^* = \frac{\varepsilon_0}{2} V^2 A \left(\frac{1}{d_0 - x} \right) + \frac{\varepsilon_0}{2} V^2 A \left(\frac{1}{d_0 + x} \right) = \varepsilon_0 V^2 A \left(\frac{d_0}{d_0^2 - x^2} \right), \quad (4.6)$$

de modo que a força elétrica F_e é uma função não-linear do deslocamento em x e uma função quadrática da tensão:

$$F_e = -\frac{\partial W^*}{\partial x} = 2\varepsilon_0 d_0 A V^2 \frac{x}{(d_0^2 - x^2)^2}. \quad (4.7)$$

A rigidez da mola também é um parâmetro importante, que pode ser afetado por fenômenos termoelástico e não-linearidades. Considerando estas variações a força conservativa da mola F_k pode ser representada por:

$$k = k_1x + k_3x^3. \quad (4.8)$$

Em (BASSINELLO, 2011) é descrito que em sistemas MEMS tem-se considerado um amortecimento termoelástico como uma possível fonte dominante de amortecimento intrínseco. Para o modelo estudado o amortecimento termoelástico é resultado do fluxo irreversível de calor gerado pela oscilação da massa m conforme a posição de x .

Esta variação na temperatura provoca um gradiente térmico no interior do material da estrutura, que se ajusta para permitir um equilíbrio térmico. No entanto, a energia utilizada neste ajuste não pode ser restaurada, mesmo com a estrutura retornando ao seu estado original.

Considerando que o amortecimento termoelástico seja uma função de \dot{x} , a força de dissipação F_c pode ser obtida de:

$$F_c = c\dot{x}. \quad (4.9)$$

Com isto, substituindo (4.7), (4.8) e (4.9) em (4.1) obtêm-se a equação do movimento:

$$m\ddot{x} + c\dot{x} + k_1x + k_3x^3 = 2\varepsilon_0d_0AV^2 \frac{x}{(d_0^2 - x^2)^2}, \quad (4.10)$$

onde:

$$V = (V_p + V_i \text{sen}(\omega t)). \quad (4.11)$$

(TUSSET et al., 2012a) considerou as condições iniciais como um repouso $x(0) = x_0, \dot{x}(0) = 0$ e definindo novas variáveis $T = \omega_0 t, u = x/x_0$

onde: $\omega_0^2 = \frac{k_1}{m}, t = \frac{T}{\omega_0}, x = x_0 u, \dot{x} = \omega_0 x_0 \dot{u}, \ddot{x} = \omega_0^2 x_0 \ddot{u}$

A equação do movimento pode ser representada da forma adimensional:

$$m\omega_0^2 x_0 \ddot{u} + k_1 x_0 u + k_3 (x_0 u)^3 + c\omega_0 x_0 \dot{u} = 2\varepsilon_0 d_0 A V^2 \frac{x_0}{(d_0^2 - x_0 u^2)^2}. \quad (4.12)$$

Considerando as seguintes substituições: $b = (c/m\omega_0), \alpha_1 = (k_1/m\omega_0^2), \alpha_3 = (k_3 x_0^2/m\omega_0^2), \beta = (2\varepsilon_0 d_0 A/m\omega_0^2 x_0^4), w = \omega/\omega_0, d = d_0/x_0$.

O sistema adimensional 4.12 fica na seguinte forma:

$$\ddot{u} + b\dot{u} + a_1 u + a_3 u^3 = \beta V^2 \frac{u}{(d^2 - u^2)^2}, \quad (4.13)$$

onde, V^2 , pode ser representada na seguinte forma:

$$V_p^2 + \frac{1}{2}V_i^2 + 2V_p V_i \text{sen}(\omega T) - \frac{1}{2}V_i^2 \cos(2\omega T). \quad (4.14)$$

Reescrevendo a equação 4.14, em variáveis de estado:

$$\begin{aligned} \dot{u}_1 &= u_2 \\ \dot{u}_2 &= -bu_2 - a_1u_1 - a_3u_1^3 + \beta \frac{u}{(d^2 - u^2)^2} [2V_p V_i \text{sen}(\omega T) - \frac{1}{2}V_i^2 \text{cos}(2\omega T)] \end{aligned} \quad (4.15)$$

Para a simulação numérica (BASSINELLO, 2011) usou os seguintes valores: $b = 1$, $\alpha_1 = 1$, $\alpha_3 = 0,445$, $w = 6,28$, $d = 25$, $V_p = 2$, $V_i = 10$, $\beta = 69141,6$.

O controle utilizando a Equação de Riccati dependente de estado (SDRE) é usado para ver qual sinal de controle deve ser utilizado para levar o sistema para órbita desejada. Escrevendo o sistema dinâmico definido por 4.15 na forma:

$$\dot{u} = f(u) + B(u)u_r, \quad (4.16)$$

onde: $f(u) = A(u)u$

O vetor $u = \begin{bmatrix} u_1 & u_2 \end{bmatrix}^T$ representa os estados do sistema dependente do tempo, $\dot{u} \in \mathbb{R}^2$ é o vetor de derivadas primeira do estado, $u_r \in U$ é a função de controle.

Considerando as condições iniciais $u(T_0) = u_0$, e que $f(0) = 0$ implica que a origem seja um ponto de equilíbrio, podendo assim estar escrevendo (4.15) na forma (4.16):

$$\dot{u} = \begin{bmatrix} 0 & 1 \\ -\alpha_1 - \alpha_3 u_1^2 + \frac{\beta}{(d^2 - u_1^2)^2} [V_p^2 + \frac{1}{2}V_i^2 + 2V_p V_i \text{sen}(\omega T) - \frac{1}{2}V_i^2 \text{cos}(2\omega T)] & -b \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u_r \quad (4.17)$$

A matriz A, e a matriz B, são representadas por:

$$A = \begin{bmatrix} 0 & 1 \\ -\alpha_1 - \alpha_3 u_1^2 + \frac{\beta}{(d^2 - u_1^2)^2} [V_p^2 + \frac{1}{2}V_i^2 + 2V_p V_i \text{sen}(\omega T) - \frac{1}{2}V_i^2 \text{cos}(2\omega T)] & -b \end{bmatrix}$$

$$B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

e definindo: $Q = \begin{bmatrix} 2000 & 0 \\ 0 & 2 \end{bmatrix}$, $R = [0,0001]$. Assim o funcional custo a ser minimizado através do Controle SDRE, é dado por:

$$J = \int_0^{\infty} (e^T Q(e)e + u_r^T R(u)u_r) dt, \quad (4.18)$$

onde: $\dot{u} = Au + Bu_r$, $e(0) = e_0$, $e(\infty) = 0$ e $e = [u - u^*]$.

O controle de realimentação de estados não-linear é obtido de:

$$u_r = -R^{-1}(e)B^T(e)P(e) = -K(e)e, \quad (4.19)$$

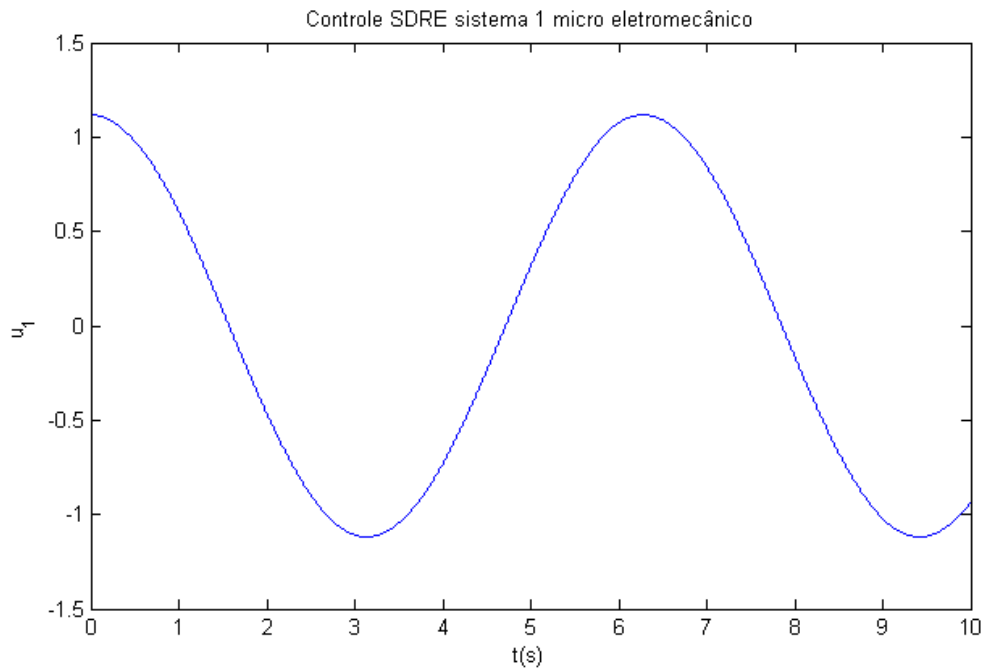


Figura 15 – Deslocamento de u_1 para o sistema microeletromecânico com controle (SDRE) (Autoria própria)

onde $P(e)$ é a solução da Equação de Riccati dependente do Estado:

$$P(e)A(e) + A^T(e)P(e) - P(e)B(e)R^{-1}(e)B^T(e)P(e) + Q(e) = 0. \quad (4.20)$$

Pode-se observar, na Figura 15, o deslocamento $u_1(x)$ quando o controle é aplicado.

4.2.1 Aquisição de Dados Para o Treinamento da Rede

Para o treinamento foram adquiridos os dados através de uma simulação da planta sendo controlada pelo SDRE, e os dados coletados são as diferenças entre o estado da planta e o estado desejado, ou seja, o erro na saída. Esse sistema tem dois estados: a posição, gerando assim o erro de posição (Figura 16), e a velocidade gerando o erro de velocidade (Figura 17). O erro é a entrada do SDRE para cálculo da atuação na planta e para cada erro de entrada o SDRE gera um valor de atuação na planta. Nos dados de treinamento, além dos erros de posição e velocidade, também são armazenados, para cada erro, a respectiva atuação do SDRE para esse erro representado pela Figura 18. O treinamento da rede neural é feito a partir desses dados, com um conjunto de erros e a atuação desejada para cada um desses erros.

Foram coletados cem mil (100.000) dados de erros e atuações para o treinamento da rede neural, formando 100 segundos de simulação com passo de 0,001 entre cada conjunto de dado guardado. Essa coleta de dados é realizada através da simulação do controle SDRE na planta e a quantidade de coleta foi feita para que houvesse dados de sobra para realizar o treinamento da ESN.

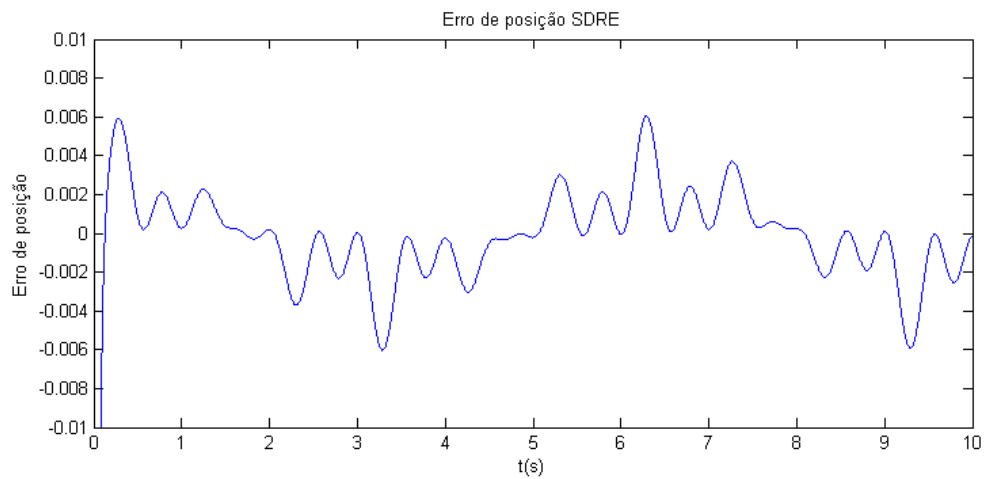


Figura 16 – Erro de posição SDRE (Autoria própria)

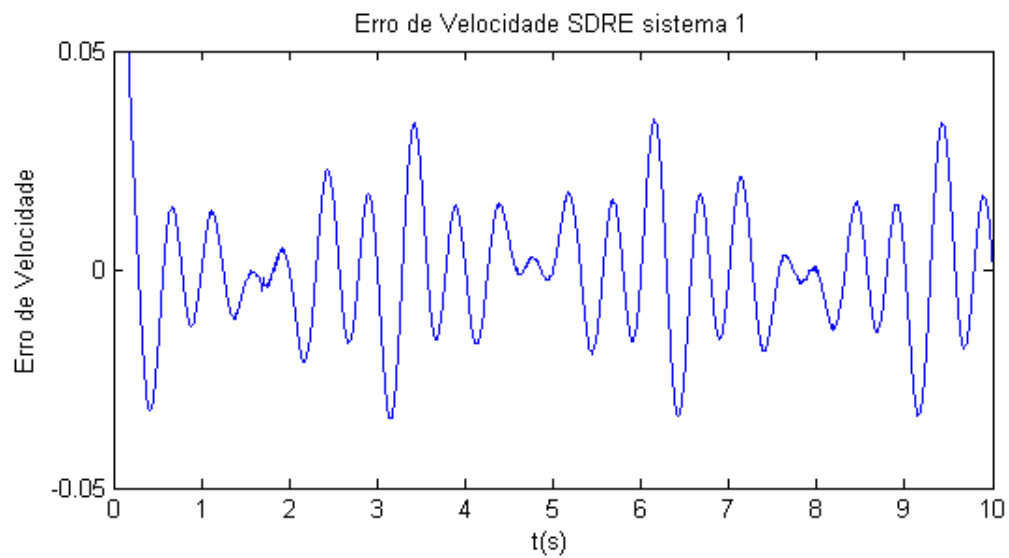


Figura 17 – Erro de velocidade SDRE (Autoria própria)

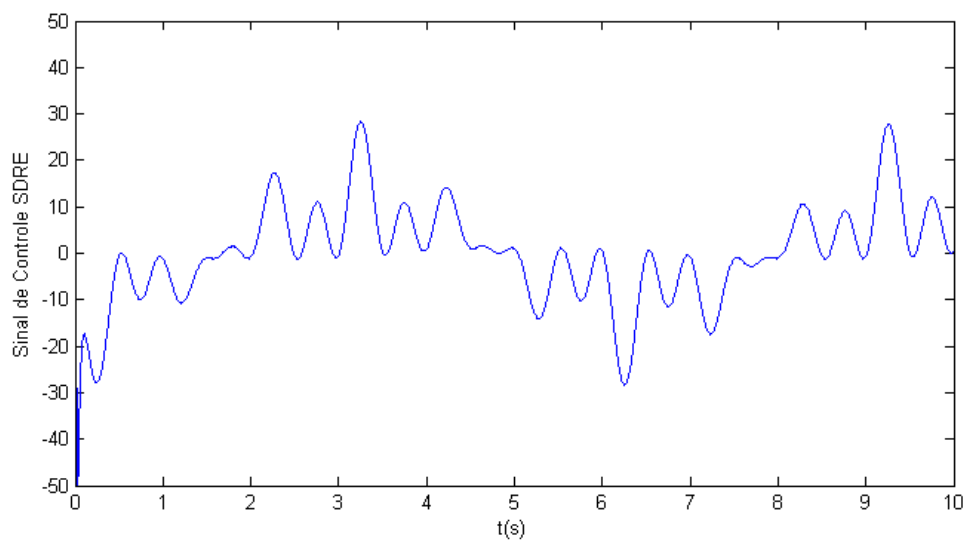


Figura 18 – Atuação SDRE Sistema MEMs (Autoria própria)

4.2.2 Treinamento, Ajuste e Teste da Rede Neural

Tendo os dados de erro e atuação do SDRE, então é necessário treinar a rede para que ela consiga responder aos erros de entrada da mesma forma que o SDRE. Para isso é necessário ajustar as configurações da rede neural e do treinamento da rede neural, e então realizar testes e novos ajustes na rede neural até convergir para valores aceitáveis. As configurações que serão ajustadas aqui serão:

- Quantidade de neurônios no reservatório;
- O raio espectral da matriz \mathbf{W} ;
- Quão esparsa a matriz \mathbf{W} deve ser;
- Se todas as entradas disponíveis na planta serão utilizadas;
- Dos 100.000 dados amostrados do SDRE, quantos destes devem ser usados para treinar a rede?

O primeiro resultado satisfatório foi obtido com as seguintes configurações:

- Quantidade de neurônios no reservatório: 60
- Pesos das matrizes \mathbf{W} e \mathbf{W}^{in} gerados aleatoriamente de -1 à 1 uniformemente.
- Raio Espectral da matriz \mathbf{W} : 0,3
- Grau de esparsidade da matriz \mathbf{W} : 5%
- Quantidade de amostras utilizadas para o treino: 3000

Para esse caso foi usado apenas o erro de posição como entrada, ou seja, a rede não usa o erro de velocidade, apesar de ser realizada a captura desta informação. A velocidade é a variação do deslocamento, portanto, a rede já possui esse dados visto que a rede neural é recorrente e possui esta informação no reservatório de dinâmicas.

Com esses parâmetros esta rede foi treinada e então testada comparando com as 100.000 amostras do comportamento do controlador SDRE. O resultado obtido pode ser verificado na Figura 19. Nela está representada a atuação da rede neural em pontilhado e a atuação do SDRE (do treinamento) em traço contínuo, porém é praticamente impossível ver a linha pontilhada pois a rede neural atuou praticamente da mesma forma que o SDRE. O erro médio quadrático para este resultado é de apenas 0,0045652, o que representa um erro muito pequeno em relação à grandeza das atuações que variam aproximadamente entre -30 e 30. A Figura 20 apresenta o mesmo resultado, mas aproximando entre o tempo 1,6 e 2 segundos para então verificar que realmente existem diferenças na atuação.

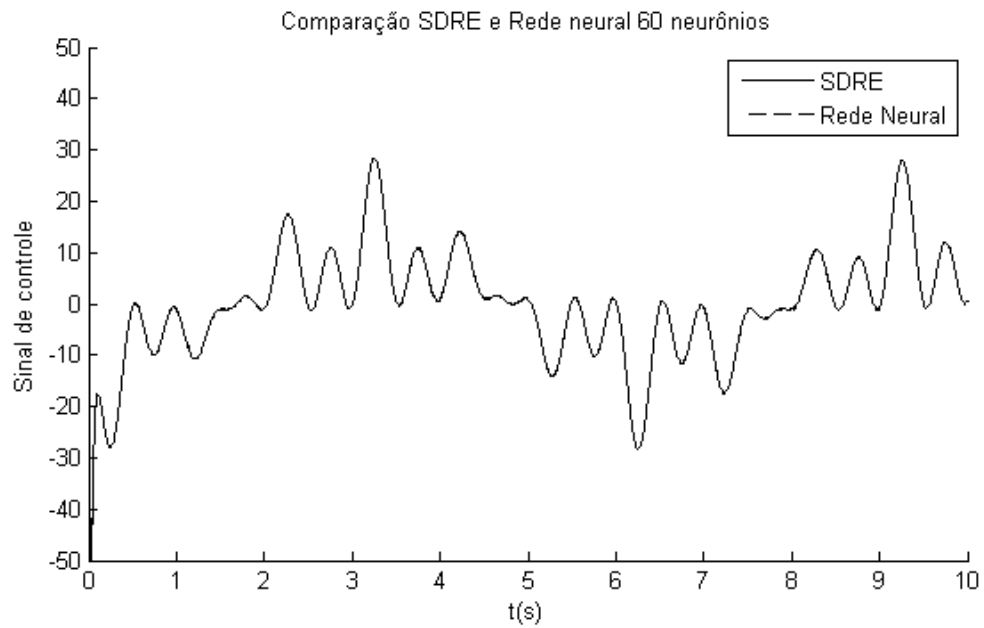


Figura 19 – Teste de atuação da rede neural comparando com o SDRE, usando 60 neurônios (Autoria própria)

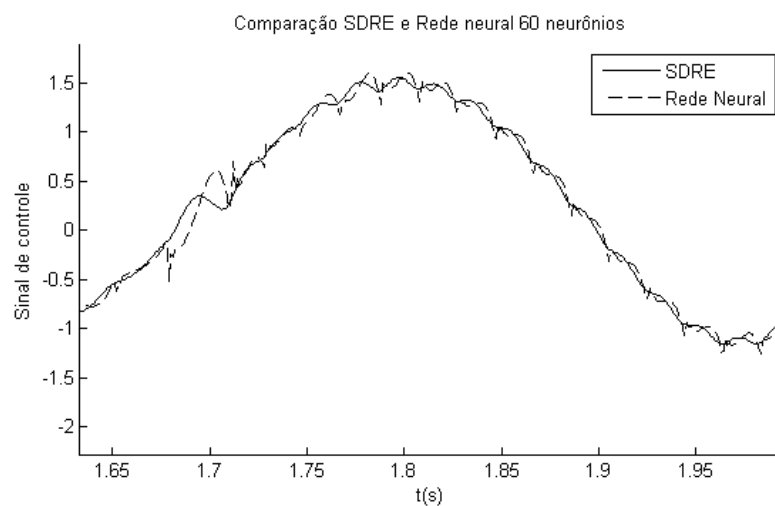


Figura 20 – Teste de atuação da rede neural comparando com o SDRE, usando 60 neurônios aproximado (Autoria própria)

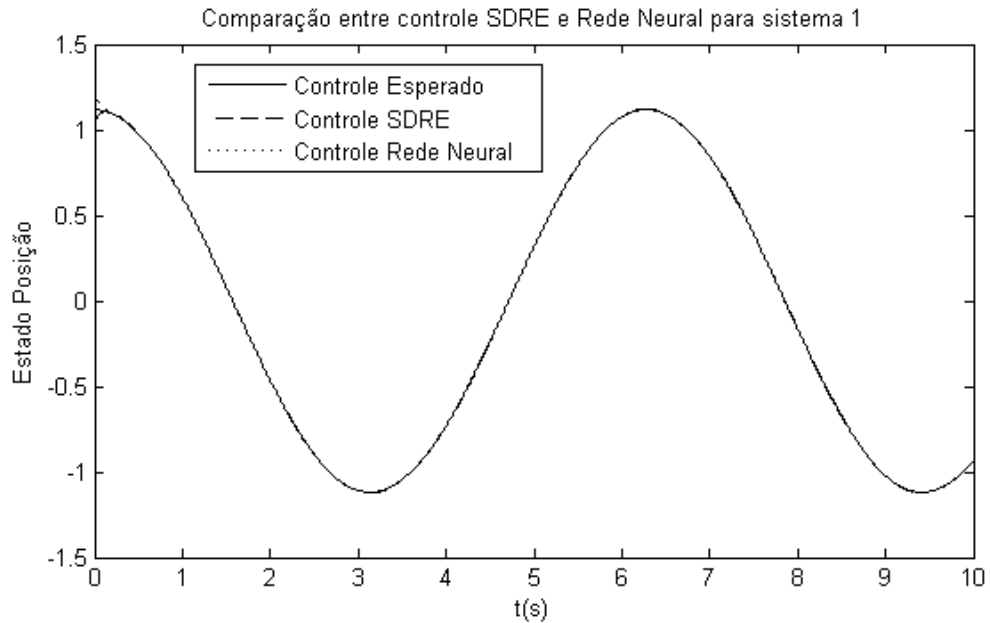


Figura 21 – Comparativo controle SDRE e ESN (Autoria própria)

Esses resultados representam visualmente um bom comportamento da rede neural, porém o resultado mais importante é o da atuação da rede neural diretamente na planta, fazendo assim com que a rede neural interfira diretamente na saída da planta, conseqüentemente gerando os erros para a próxima atuação da rede. Dessa maneira se consegue realmente avaliar se a rede neural se comporta como um controlador SDRE.

4.2.3 Atuação da Rede de Estado de Eco Diretamente na Planta do Sistema MEMs

Para fazer com que a rede neural atuasse na planta foi modificado o código fonte, escrito em *MatLab*, para gerar as simulações do SDRE, o qual gerou os dados para treinamento da rede neural. Neste código as plantas foram duplicadas de forma que uma fosse controlada pelo SDRE, e outra planta fosse controlada pela rede neural para haver comparação entre os resultados. Para o treinamento com 60 neurônios (Figura 21), a linha contínua é o valor desejado, em tracejado o resultado da saída da planta controlada pelo SDRE, e em pontilhado a planta controlada pela rede neural notando-se que o comportamento da planta controlada, tanto pelo SDRE quanto pelo ESN estão praticamente o mesmo.

Na Figura 22 pode ser visualizado o mesmo resultado, porém com uma maior aproximação, apresentando uma amostra apenas entre os segundo 3 e os 3,5 segundos, podendo assim ser observada a diferença entre os resultados. Pode-se notar que a rede e o SDRE, como esperado, estão respondendo praticamente da mesma maneira.

Outro resultado que pode ser observado são os histogramas dos erros do SDRE e da Rede neural, apresentados na Figura 23, onde é verificado um comportamento similar na quantidade de erros.

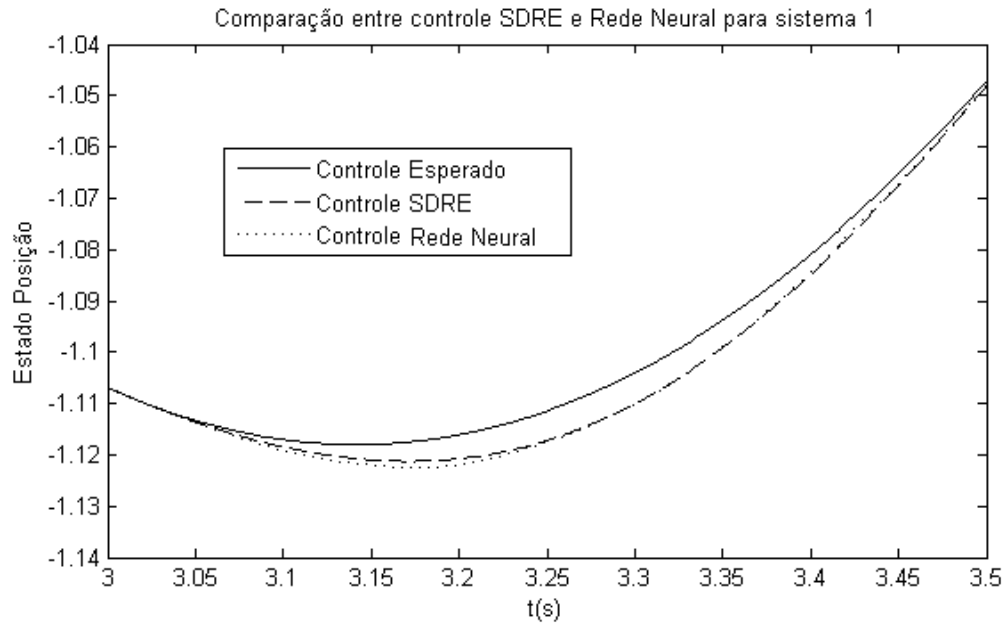


Figura 22 – Comparativo controle SDRE e ESN aproximado (Autoria própria)

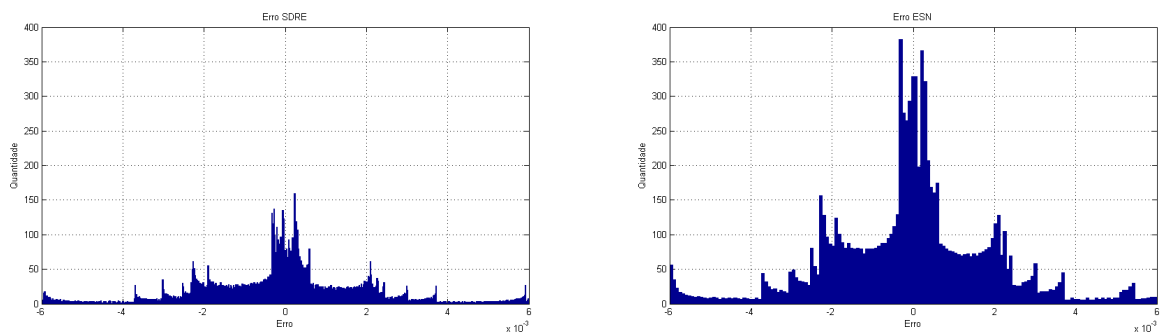


Figura 23 – Histograma de erros do controlador SDRE e ESN no sistema MEMS demonstrando a igualdade do comportamento da ESN em relação ao SDRE (Autoria própria)

4.2.4 Comparativo Entre a ESN e SDRE Quanto ao Tempo de Processamento para o sistema Microeletromecânico

O processamento computacional para a ESN cresce à medida que a quantidade de neurônios utilizada no reservatório dinâmico aumenta. É possível, portanto, existir uma quantidade de neurônios no reservatório que faça com que a rede neural tenha um custo computacional superior a do controle SDRE. Uma pergunta que pode ser levantada aqui é: “A Rede Neural com Estados de Eco é capaz de se comportar eficientemente como um controlador SDRE se utilizando de menos tempo de processamento que este controlador SDRE?” Uma das formas de compararmos o custo computacional das técnicas é através do tempo de processamento, o qual deve ser o mínimo possível.

Foram realizadas medições do tempo necessário para realizar o cálculo para a atuação do SDRE e para redes neurais de estado de eco com diferentes quantidades de neurônios. Para isso foram colocados medidores de tempo (funções *tic()* e *toc()* do *MatLab*), medindo o

tempo utilizado para realizar os cálculos para cada atuação do ESN e SDRE. A Figura 24 representa o gráfico dos tempos para o cálculo de cada atuação, e além do SDRE ter valores mais altos de tempo para cálculo de atuação, nota-se uma grande variação entre os tempos em relação aos tempos do ESN. Para verificar estas diferenças, na Figura 25, dois histogramas são utilizados, um representando os tempos do SDRE e outro da ESN, e nota-se que o histograma da ESN contém um maior número de ocorrência na média, e em relação ao SDRE pouca variação de tempo para cada cálculo de sinal de controle. Numericamente na tabela 1, se pode confirmar isso através da média e desvio padrão. Nesta tabela também estão os resultados das simulações quanto à performance em relação aos erros. Os resultados foram realizados em um mesmo computador em condições computacionais, porém variações de disponibilidade do processador para o *MatLab* podem ocorrer por conta dos serviços ativos no sistema operacional multitarefas.

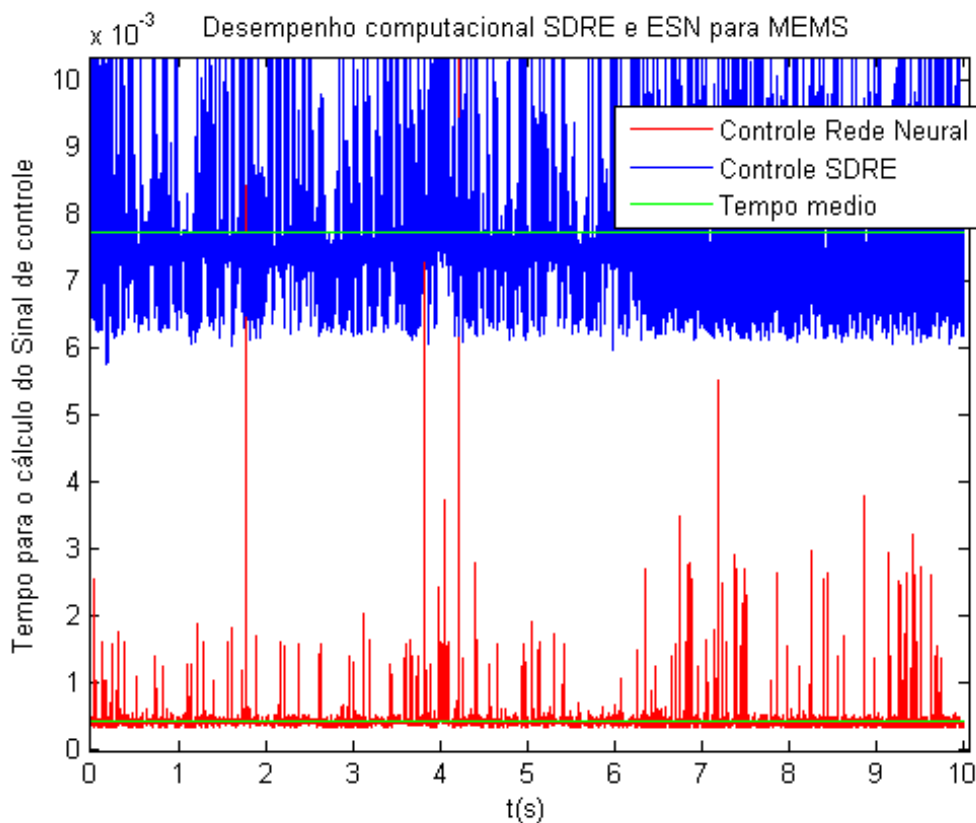


Figura 24 – Tempos para cálculo de atuação para SDRE e ESN com 60 neurônios no reservatório dinâmico sistema MEMS (Autoria própria)

Observando os dados da tabela 1 nota-se o aumento do tempo médio à medida que se aumenta a quantidade de neurônios no reservatório dinâmico, sendo que quando é simulado o sistema com uma rede de 1.000 neurônios, os valores de tempo para os cálculos de atuação praticamente igualam-se ao tempo de cálculo do SDRE. Porém, o desvio padrão continua sendo menor que o desvio padrão do SDRE. Estes testes também foram realizados para demonstrar quando a rede neural em questão começa a ter um mesmo custo de processamento que o

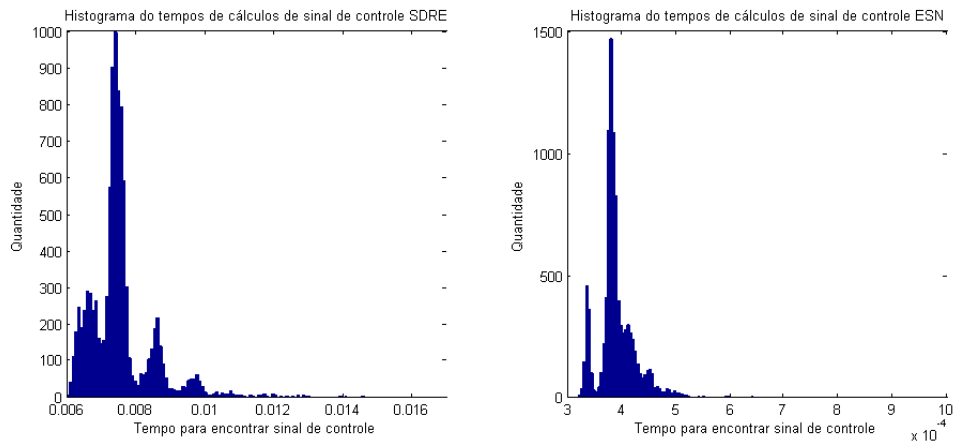


Figura 25 – Histograma Tempos SDRE x ESN sistema MEMs 60 neurônios (Autoria própria)

SDRE, mas com 60 neurônios já foi possível obter um ótimo resultado na atuação da planta.

	SDRE	ESN 30	ESN 60	ESN 200	ESN 500	ESN 1000
Erro Médio Quad.	0	0,0070035	0,0067917	0,0076214	0,0070674	0,0070954
Erro médio	-0,000469	-4,3026e-05	4,7391e-06	-1,044e-05	-7,3907e-06	-1,7841e-05
Erro médio absol.	0,001968	0,0020336	0,0020144	0,0021285	0,0020706	0,0020825
Erro médio quad.	0,0058705	0,0056459	0,0054752	0,006144	0,0056974	0,00572
Média dos tempos	0,0077	2,6672e-04	4,1757e-04	0,0016	0,0039	0,0069
Desvio padrão tem	0.0023	1,7003e-04	3,2645e-04	8,6474e-04	9,7294e-04	5,7421e-04

Tabela 1 – Erros para SDRE e ESN com 60 neurônios sistema MEMs

4.2.5 Influência dos Números Aleatórios de \mathbf{W} e \mathbf{W}^{in}

A cada novo teste da ESN são gerados novos valores aleatórios para as matrizes \mathbf{W} e \mathbf{W}^{in} , para verificar a influência destes valores nos resultados da rede foram geradas 130 ESN's com 100 neurônios no reservatório de dinâmicas. Estas ESNs foram treinadas e o MSE de cada uma foi armazenado em um vetor. O comportamento desses erros é mostrado na Figura 26 na qual foi usado um histograma para tanto. Nota-se que a maioria dos erros convergiram para um erro de $4,6e-04$, mas não todos, demonstrando assim que existe a influência dos valores aleatórios. Porém, estas influências não são significativas, pois todas as redes conseguiram realizar o controle desejado do sistema MEMs.

4.2.6 Sensibilidades a Erros Paramétricos

Em sistemas reais os erros dos sensores podem fazer com que o controle não aconteça, pois o sistema pode divergir do comportamento desejado. Para o sistema MEMs, foi verificada a possibilidade da realização de controle com erros de até 1% e 5% tanto para a rede neural quanto para o SDRE. O aumento desse erro, a partir de valores como 10%, deixaram o sistema instável. Com os testes de inserção de 1% verificou-se que se o comportamento

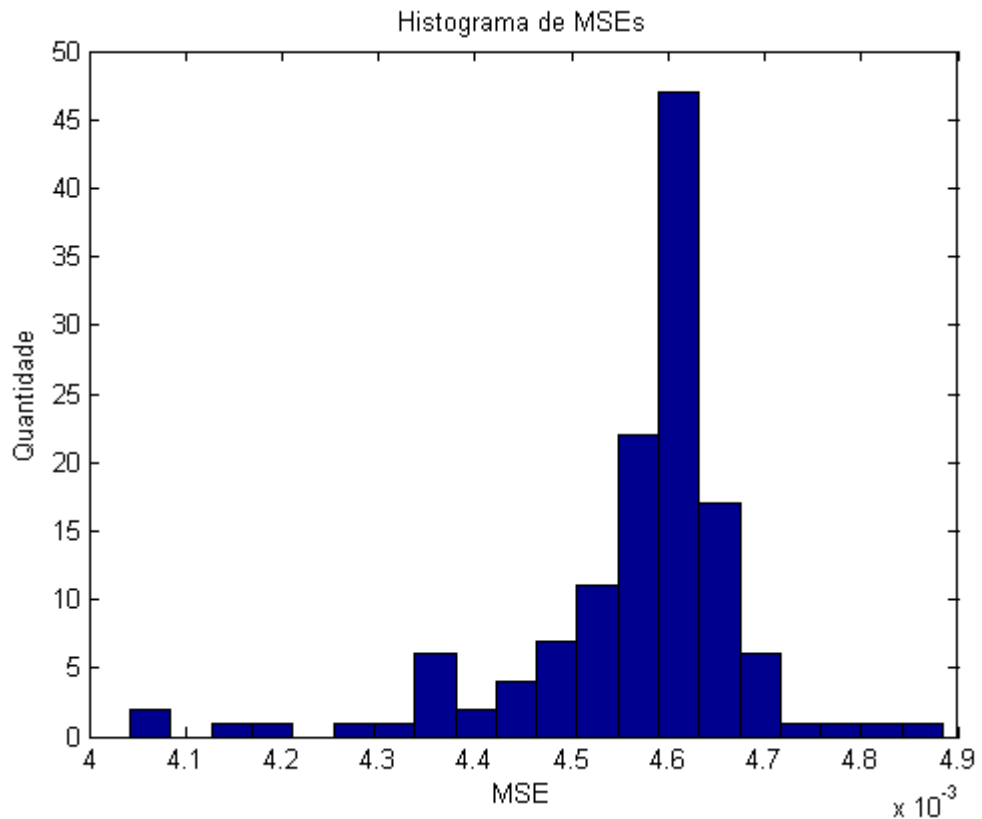


Figura 26 – Histograma MSEs (Autoria própria)

do controle SDRE consegue controlar o sistema mesmo com um erro simulado de leitura das variáveis. Felizmente, a ESN também consegue controlar o sistema.

Um teste mais significativo aconteceu quando foi simulado um distúrbio de $\pm 5\%$ onde visualmente nota-se um comportamento diferente no erro da saída na Figura 27 em relação à figura 19, podendo-se observar a diferença gerada pelo distúrbio. Na Figura 28 na qual o gráfico o qual é aproximado entre os tempos 5,67 e 5,74, pode-se evidenciar a diferença dos erros quando controlado por SDRE e ESN. Quantitativamente falando ao adicionar distúrbios o Erro médio Quadrático para o SDRE passou de 0,0058705 (sem distúrbios), para 0,006022 (com distúrbios), e da ESN passou de 0,0054752 para 0,0059416. Com esse resultado podemos afirmar que tanto a rede neural quanto o SDRE foram capazes de realizar o controle desejado do sistema MEMs. Vale dizer que quando o distúrbio é aumentado em maiores níveis o controle do sistema se torna cada vez mais ineficiente.

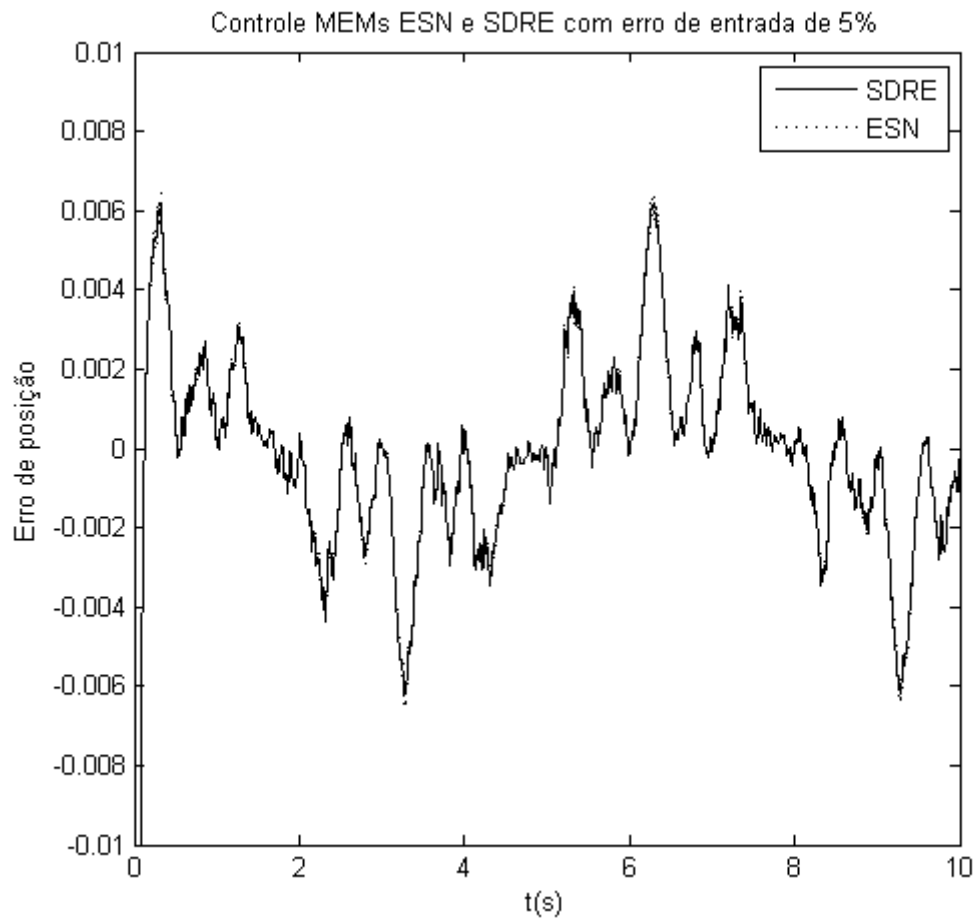


Figura 27 – Erro com distúrbio de 5% na entrada do sistema MEMs (Autoria própria)

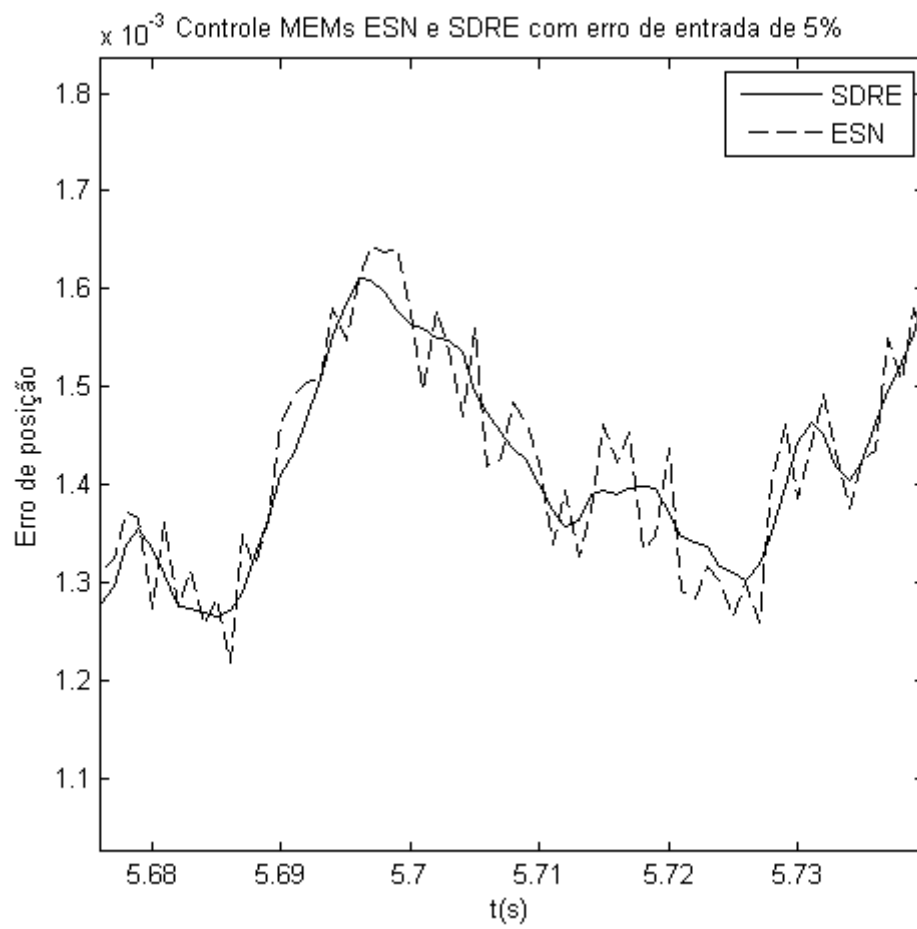


Figura 28 – Erro com distúrbio de 5% na entrada do sistema MEMs zoom (Autoria própria)

5 USANDO CONTROLE SDRE PARA TREINAMENTO DA ESN E SEU CONTROLE DE UM SISTEMA DE MEIO CARRO (*HALF-CAR*)

5.1 SISTEMA *HALF-CAR* (Meio Carro)

No presente capítulo será realizado o mesmo procedimento utilizado no Capítulo 4 para treinar uma rede neural ESN a partir do dados de controle SDRE, porém para um sistema de amortecimento “half-car”, importante para manter estabilidade, segurança e o conforto do carro.

Este outro projeto de controle se refere a sistemas para redução de vibrações da suspensão de veículos. Nos últimos anos muita atenção tem sido dada às técnicas para reduzir as vibrações das suspensão veicular. A suspensão veicular é um dos elementos mais importantes que compõem um veículo, sendo responsável por conectar as rodas a ele de tal maneira que os impactos causados pelas perturbações do local onde o carro está passando sejam absorvidos e percebidos de maneira a reduzir as perturbações dentro do veículo (LIU; NONAMI; HAGIWARA, 2008).

A Figura 29 ilustra o sistema que será considerado neste trabalho. Esse sistema foi aprofundadamente estudado em (TUSSET et al., 2012b). É importante destacar que nestes estudos foram aplicados diferentes controladores ao sistema half-car e dentre eles o único mecanismo de controle bem sucedido neste cenário foi o SDRE.

O modelo é composto por 3 massas m_s , m_{uf} e m_{ur} , onde m_s é a massa do chassi, m_{uf} a massa da roda da frente e m_{ur} a massa da roda traseira. Além disso o sistema pode ser descrito com quatro graus de liberdade. Considerando que a massa do chassi é um corpo rígido com movimento vertical e rotação, x_1 é definido como o centro de gravidade do chassi, e o θ o ângulo de rotação de m_s . O deslocamento vertical da parte da frente de m_s é demonstrado por x_{1f} e o traseiro por x_{1r} . O movimento vertical da massa m_{uf} é denotado por x_{2f} , enquanto da massa m_{ur} é chamado de x_{2r} . O deslocamento é definido por $w_f = \alpha \text{sen}(2\pi\omega t)$ na frente e $w_r = \alpha \text{sen}(2\pi\omega t + \pi)$ atrás causado por ruas irregulares. A força de controle é composta F_f na frente e F_r atrás. As molas dianteiras e traseiras são representadas por K_{sf} e K_{sr} respectivamente, os amortecedores dianteiro e traseiro por b_{sf} e b_{sr} e os pneus representados por K_{tf} e K_{tr} .

As forças aplicadas pelos amortecedores são representadas pela equação:

$$\begin{aligned} F_{bsf} &= b_f^l(\dot{\Delta}_f) - b_{sf}^y |\dot{\Delta}_f| + b_{sf}^{nl} \sqrt{|\dot{\Delta}_f|} \text{sgn}(\dot{\Delta}_f) \\ F_{bsr} &= b_r^l(\dot{\Delta}_r) - b_{sr}^y |\dot{\Delta}_r| + b_{sr}^{nl} \sqrt{|\dot{\Delta}_r|} \text{sgn}(\dot{\Delta}_r) \end{aligned}, \quad (5.1)$$

onde: $\dot{\Delta}_f = \dot{x} - \dot{x}_{2f} - l_f \dot{\theta} \cos\theta$, $\dot{\Delta}_r = \dot{x} - \dot{x}_{2r} - l_r \dot{\theta} \cos\theta$

O coeficiente de amortecimento $b_{s_}$ (b_{sf} ou b_{sb}) é composto por $b_{s_}^l$ representando a parte linear, $b_{s_}^{nl}$ as características não-lineares, e $b_{s_}^y$ representando o comportamento de

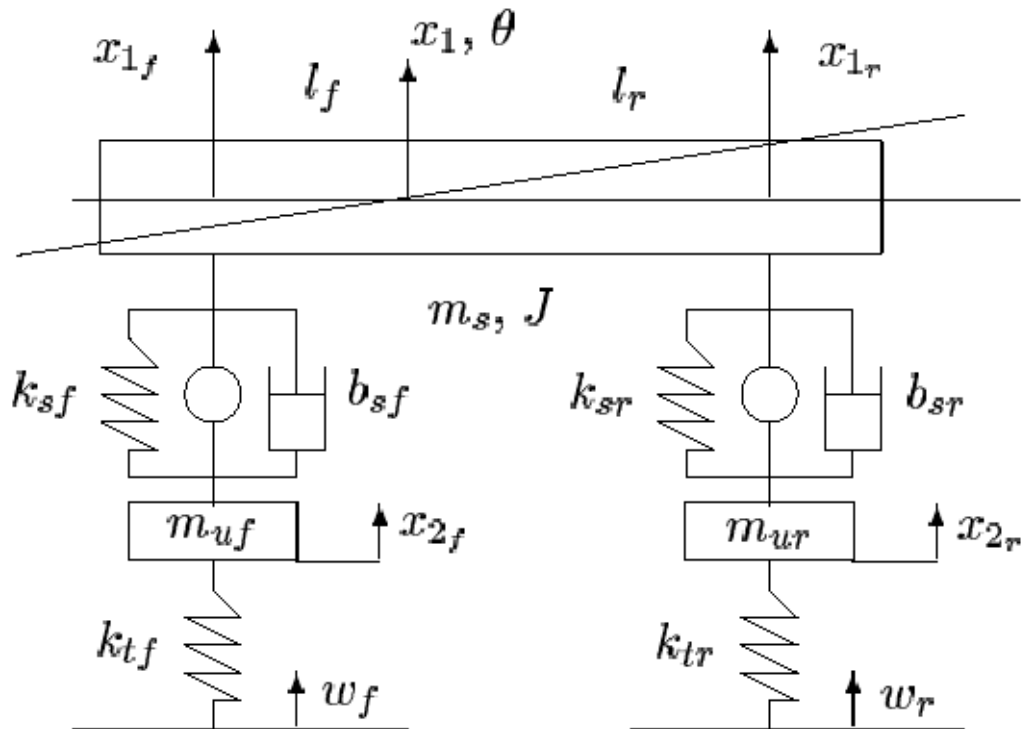


Figura 29 – Modelo de meio carro (TUSSET et al., 2012b)

amortecimento assimétrico.

As forças das molas acontecem tanto na parte dianteira quanto na parte traseira e podem ser representadas pelas equações:

$$\begin{aligned} F_{ksf} &= k_{sf}^l(\Delta_f) + k_{sf}^{nl}(\Delta_f)^3 \\ F_{ksr} &= k_{sr}^l(\Delta_r) + k_{sr}^{nl}(\Delta_r)^3 \end{aligned} \quad (5.2)$$

onde $\Delta_f = x - x_{2f} - l_f \sin\theta$ e $\Delta_r = x - x_{2r} - l_r \sin\theta$.

O coeficiente de mola $k_{s_}$ é composto por: $k_{s_}^l$ representando a parte linear, e $k_{s_}^{nl}$ representando a parte não-linear.

As forças dos pneus, representados por molas lineares, podem ser representadas assim:

$$\begin{aligned} F_{ktf} &= k_{tf}(x_{2f} - w_f) \\ F_{ktr} &= k_{tr}(x_{2r} - w_r) \end{aligned} \quad (5.3)$$

A força vertical, a força de ação de rotação e a força que atua sobre o deslocamento

do chassis podem ser representadas por :

$$\begin{aligned}
 F_{ms} &= m_s \ddot{x}_1 \\
 F_\theta &= J \ddot{\theta} \\
 F_{muf} &= m_{uf} \ddot{x}_{2f} \\
 F_{mur} &= m_{ur} \ddot{x}_{2r}
 \end{aligned} \tag{5.4}$$

Considerando as características de molas e amortecedores, a equação que representa as forças da suspensão podem ser representadas da seguinte forma:

$$\begin{aligned}
 F_{ms} &= -F_{ksf} - F_{bsf} - F_{ksr} - F_{bsr} - F_f - F_r \\
 F_\theta &= l_f \cos(\theta)(F_{ksf} + F_{bsf} + F_f) - l_r \cos(\theta)(F_{ksr} + F_{bsr} + F_r) \\
 F_{muf} &= F_{ksf} + F_{bsf} - F_{ktf} + F_f \\
 F_{mur} &= F_{ksr} + F_{bsr} - F_{ktr} + F_r
 \end{aligned} \tag{5.5}$$

Substituindo (5.5) em (5.4):

$$\left\{ \begin{aligned}
 \dot{x}_1 &= x_2 \\
 \dot{x}_2 &= \frac{1}{m_s}(-F_{ksf} - F_{bsf} - F_{ksr} - F_{bsr} - F_f - F_r) \\
 \dot{x}_3 &= x_4 \\
 \dot{x}_4 &= \frac{1}{J}((l_f \cos(\theta)(F_{ksf} + F_{bsf} + F_f) - l_r \cos(\theta)(F_{ksr} + F_{bsr} + F_r)) \\
 \dot{x}_5 &= x_6 \\
 \dot{x}_6 &= \frac{1}{m_{uf}}(F_{ksf} + F_{bsf} - F_{ktf} + F_f) \\
 \dot{x}_7 &= x_8 \\
 \dot{x}_8 &= \frac{1}{m_{ur}}(F_{ksr} + F_{bsr} - F_{ktr} + F_r)
 \end{aligned} \right. \tag{5.6}$$

onde: $x_1 = x$; $x_2 = \dot{x}$; $x_3 = \theta$; $x_4 = \dot{\theta}$; $x_5 = x_{2f}$; $x_6 = \dot{x}_{2f}$; $x_7 = x_{2r}$; $x_8 = \dot{x}_{2r}$. Na prática x_1 é o deslocamento do chassi em metros, x_3 o ângulo do chassi em Radianos (movimento de *pitch*), x_5 o deslocamento da roda dianteira em metros e x_7 o deslocamento da roda dianteira em metros.

Para a simulação numérica os seguintes parâmetros foram considerados por (TUSSET et al., 2012b): $m_s = 580[kg]$; $J = 1100[kgm^2]$; $b_{sr}^l = 400[Ns/m]$; $m_{uf} = 40[kg]$; $m_{ur} = 40[kg]$; $l_f = 1[m]$; $l_r = 1,5[m]$; $k_{sf}^l = 150(10^2)[N/m]$; $k_{sr}^l = 150(10^2)[N/m]$; $b_{sf}^l = 400[Ns/m]$; $k_{sf}^{nl} = 150(10^4)[N/m]$; $k_{sr}^{nl} = 150(10^4)[N/m]$; $k_{tf} = 190(10^2)[N/m]$; $k_{tr} = 190(10^2)[N/m]$; $b_{sf}^y = 288[Ns/m]$; $b_{sr}^y = 288[Ns/m]$; $b_{sf}^{nl} = 288[Ns/m]$; $b_{sr}^{nl} = 288[Ns/m]$. Os distúrbios da entradas, ou seja, o terreno onde o half-car será submetido, serão senoidais $w_f = \alpha \text{sen}(2\pi\omega)$ e $w_r = \alpha \text{sen}(2\pi\omega)$

O controle SDRE é realizado em cima desse sistema. O objetivo é determinar um controle que leve o estado x_1 de qualquer estado inicial para um predeterminado estado final, portando:

$$e(\infty) = 0, \quad (5.7)$$

onde: $e = [x_1 \quad -\tilde{x}_1 \quad x_2 \quad -\tilde{x}_1]^T$, e \tilde{x} é o vetor de estados desejados. Introduzindo o controle proposto, nas primeiros 2 sistemas de equação, então temos:

$$\dot{x} = A(x)x + \Gamma - U_1 - U_2, \quad (5.8)$$

onde o vetor $x = [x_1 + x_2]^T$ representa o estado do sistema, Γ é o vetor com os termos que não podem ser representados pelos estados da matriz $A(x)$. E então, U_1 e U_2 os controles propostos.

$$\begin{aligned} U_1 &= u_f + \tilde{u}_f \\ U_2 &= u_r + \tilde{u}_r \end{aligned} \quad (5.9)$$

O controle *feedforward* é dado por:

$$\tilde{u} = -\Gamma. \quad (5.10)$$

O controle *feedback* o obtido por:

$$u = -R^{-1}B^T P(x), \quad (5.11)$$

onde $P(x)$ é a solução da equação de *Riccati*:

$$P(x)A(x) + A^T(x)P(x) - P(x)BR^{-1}B^T P(x) + Q = 0. \quad (5.12)$$

Então serão introduzido os controles:

$$\begin{aligned} \tilde{u}_f &= \frac{1}{m_s}(k_{sf}^{nl}\Delta_f^3 - k_{sf}^l l_f \text{sen}(x_3) - b_{sf}^y |\dot{\Delta}_f|, \\ &+ b_{sf}^{nl} \sqrt{|\dot{\Delta}_f|} \text{sign}(\dot{\Delta}_f) - b_{sf}^l l_f x_4 \cos(x_3) - k_{sf}^l x_5 - b_{sf}^l x_6), \\ \tilde{u}_r &= \frac{1}{m_s}(k_{sr}^{nl}\Delta_r^3 - k_{sr}^l l_r \text{sen}(x_3) - b_{sr}^y |\dot{\Delta}_r| e \\ &+ b_{sr}^{nl} \sqrt{|\dot{\Delta}_r|} \text{sign}(\dot{\Delta}_r) - b_{sr}^l l_r x_4 \cos(x_3) - k_{sr}^l x_7 - b_{sr}^l x_8), \end{aligned} \quad (5.13)$$

no sistema , e reescrevendo as duas primeiras equações do sistema como a seguir:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{1}{ms}(k_{sf}^l + k_{sr}^l) + x_3^3 x_2 & -\frac{1}{ms}(b_{sf}^l + b_{sr}^l) + x_3^3 x_1 \end{bmatrix}, \quad (5.14)$$

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ -\frac{1}{ms} & -\frac{1}{ms} \end{bmatrix} \begin{bmatrix} u_f \\ u_r \end{bmatrix}.$$

Os controles u_f e u_r são obtidos através do controle SDRE, no qual é necessário minimizar o custo funcional:

$$J = \int_0^{\infty} (e^T Q e + u^T R u) dt. \quad (5.15)$$

Deseja-se determinar o controle *feedback* considerando as condições iniciais: $x(t(0)) = x_0$, onde a matriz é:

$$Q = 10^3 \begin{bmatrix} 500 & 0 \\ 0 & 1 \end{bmatrix}, \quad (5.16)$$

$$R = 10^{-4} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

Substituindo (5.7) no sistema (5.6) e determinando a corrente elétrica (i) pela equação (5.8), enquanto $F_{f,r} = U_{1,2}$. U_1 e U_2 são determinados por (5.12). A Figura 30 demonstra o comportamento obtido através do controle baseado em SDRE.

5.1.1 Aquisição de Dados Para o Treinamento da Rede

Neste sistema existem dois sinais de controle: um no sistema de amortecimento traseiro, e outro no dianteiro do veículo, portanto diferentemente do sistema MEMs, serão necessários dois cálculos, um para cada saída. O sistema é representado por 8 estados e todos eles são usados para os cálculos de atuação através das SDRE porém, alguns destes dados são de difícil aquisição, então, na rede usou-se apenas os estados x_2 , x_3 , x_6 e x_8 que representam valores de fácil sensoriamento no sistema. Esses dados de entrada para o treinamento da rede neural podem ser visualizados na Figura 30.

Para que esses estados sejam obtidos o controlador baseado em SDRE agiu da forma demonstrada nos gráficos da Figura 32, onde cada gráfico representa sinal de controle do SDRE no domínio do tempo, um para cada roda. Tendo esses dados o treinamento da rede neural é realizado de forma que as entradas da rede neural sejam os estados demonstrados na Figura 31, e que a rede neural responda de acordo com os dados da Figura 32. Foram utilizadas duas redes neurais uma para cálculo de cada saída.

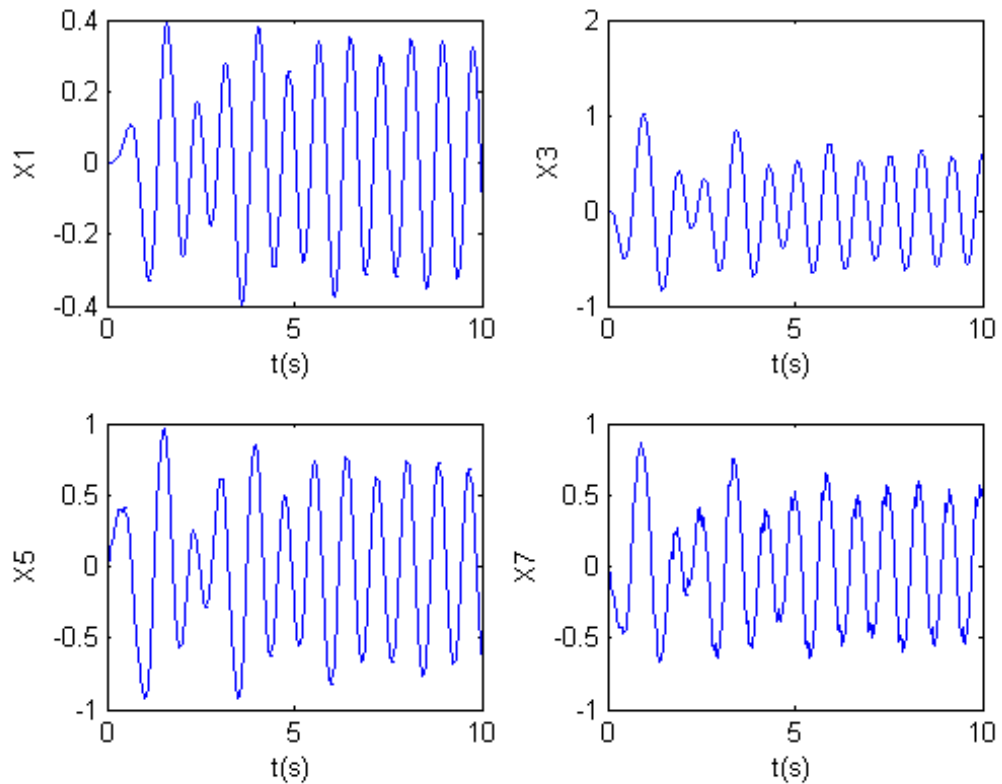


Figura 30 – Estados SDRE do sistema 2 (Autoria própria)

Assim como no sistema 1, esse também possui 100 segundos de simulação armazenados em vetores com cada um dos estados com suas respectivas atuações. A precisão de cada segundo é de 1.000 amostras no segundo, ou seja, uma coleta de dados com passo 0,001.

5.1.2 Treinamento, Ajuste e Teste da Rede Neural

O primeiro resultado pertinente foi obtido com as seguintes configurações para as duas redes neurais utilizadas:

- Quantidade de neurônios no reservatório: 120
- Raio Espectral da matriz \mathbf{W} : 0,4
- Grau de esparsidade da matriz \mathbf{W} : 5%
- Quantidade de amostras utilizadas para o treino: 40.000

O resultado obtido para esse experimento pode ser demonstrado nas Figuras 33 e 34, na qual a primeira mostra o comparativo das atuações para os dados do amortecedor dianteiro, SDRE (pontilhado) e a atuação da ESN (linha contínua), considerando o conjunto de dados de treinamento. Nota-se uma atuação muito parecida obtendo um erro médio quadrático de 0,011808

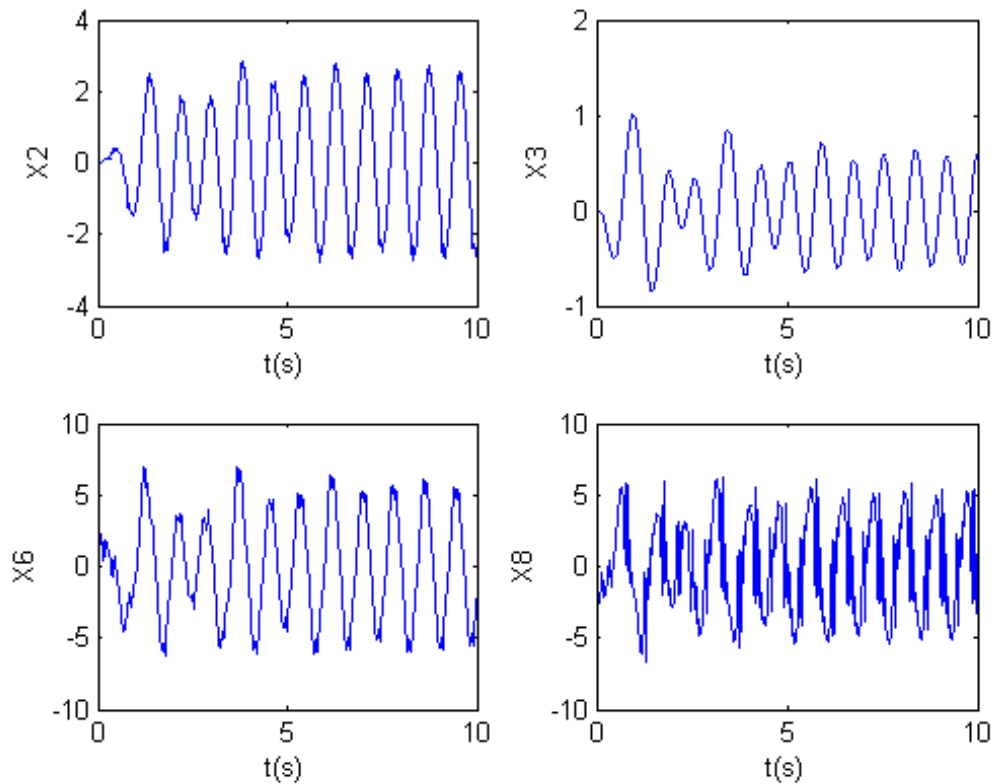


Figura 31 – Entradas da rede neural para treinamento (Autoria própria)

para este teste entre o SDRE (desejado) e ESN (calculado). A Figura 34 representa a mesma coisa, porém com a outra ESN treinada para a roda traseira gerando um erro médio quadrático de 0,0092385. Estes testes foram realizados com os dados de treinamento, não com a rede operando diretamente no sistema *half-car*. A seguir serão realizados testes com as redes neurais atuando na simulação do *half-car*.

5.1.3 Atuação da Rede de Estado de Eco Diretamente na Planta do Sistema *Half-car*

De forma parecida ao sistema MEMs a rede neural foi colocada como controladora da planta simulada do sistema *half-car*, a diferença aqui é que existem duas variáveis controladas no sistema onde cada uma é controlada por uma rede neural distinta treinada. Na Figura 35 pode ser observado o comparativo entre resultado obtido pela rede neural (linha contínua) e o controle através do SDRE (linha pontilhada). Para os quatro estados analisados pode-se observar que poucas são as diferenças entre o comportamento do *half-car* quando controlado por SDRE e ESN.

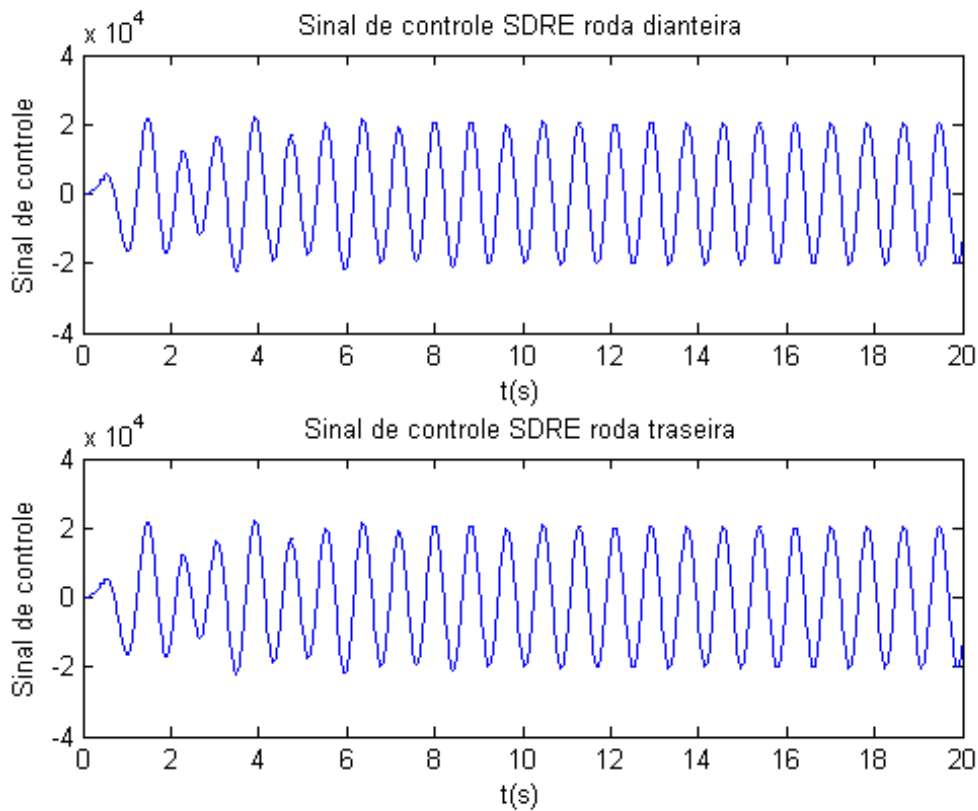


Figura 32 – Atuação SDRE sistema 2 (Autoria própria)

5.1.4 Comparativo Entre a ESN e SDRE Quanto ao Tempo de Processamento Para o Sistema *Half-car*

Da mesma forma para o sistema MEMs, foi realizada a comparação de tempo para cada um dos sistemas de controle. Ao longo dos 20 segundos de simulação foram capturados 20.000 dados para o cálculo do sinal de controle, isso para as redes neurais de 120 neurônios.

Como visualizado na Figura 36, pode-se observar que os tempos de cálculo do sinal de controle é muito maior para o SDRE do que os tempos levados pela rede neural de 120 neurônios. Na Figura 37 é mostrado o histograma dos erros, no qual também pode ser verificado que os tempos para cálculo de atuação são menores, e variam menos para a ESN em relação ao SDRE.

Também como no sistema MEMs, foram efetuados outros testes com ESNs com diferentes quantidades de neurônios. Os dados obtidos podem ser observados na tabela 2, na qual se percebe que a eficiência da rede neural se manteve independente da quantidade de neurônios utilizados. Porém, com menos de 60 neurônios a rede começou a não se comportar de forma eficiente divergindo totalmente do comportamento do SDRE. Por se tratar de um sistema cuja perturbação é uma senoide, os estados acabam sendo sinais periódicos e a média aritmética é muito próxima de zero. Portanto, para expressar o sistema numericamente foi utilizada a função RMS (*Root mean square*), que é o valor quadrático médio. Outro valor calculado

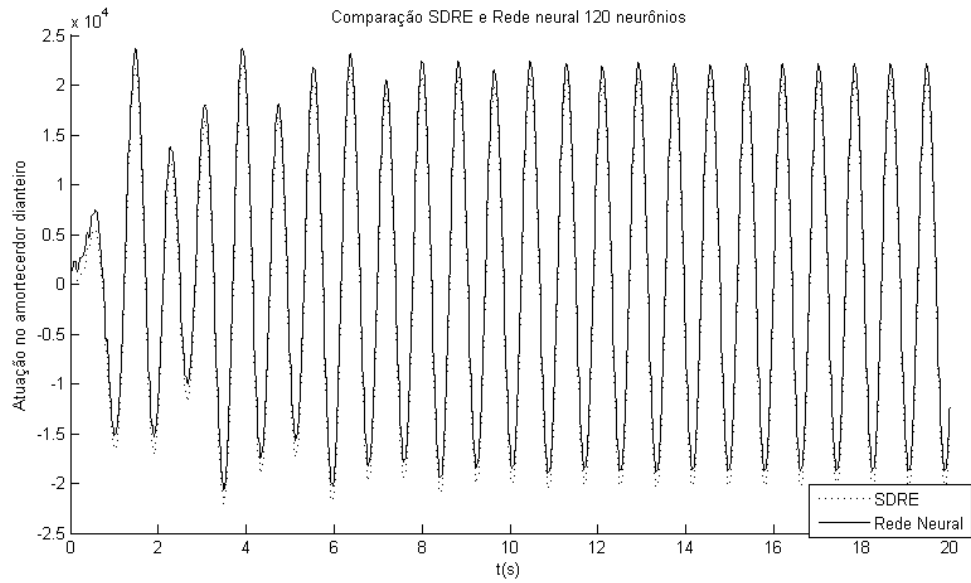


Figura 33 – Teste resposta da rede neural em relação ao SDRE sistema *half-car* amortecedor dianteiro (Autoria própria)

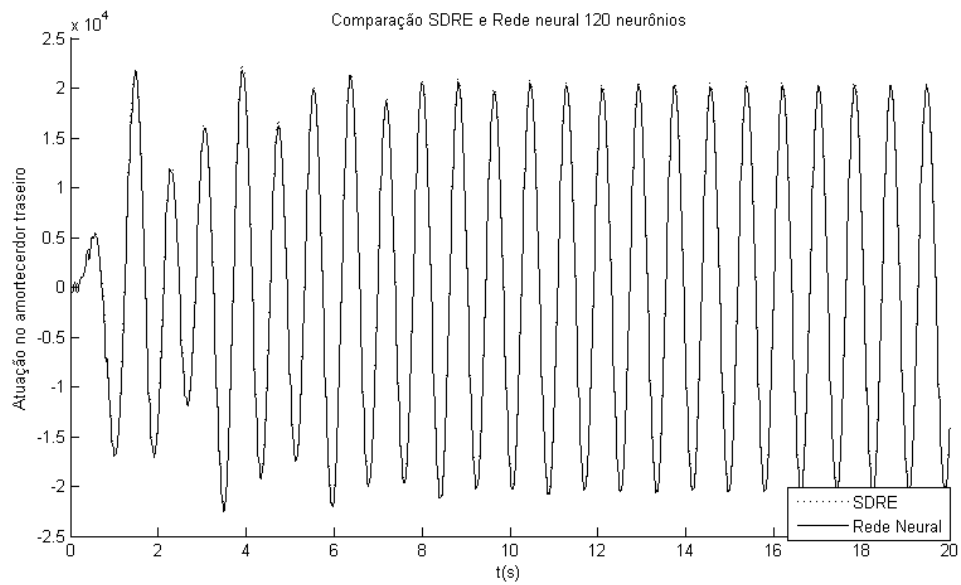


Figura 34 – Teste resposta da rede neural em relação ao SDRE sistema *half-car* amortecedor traseiro (Autoria própria)

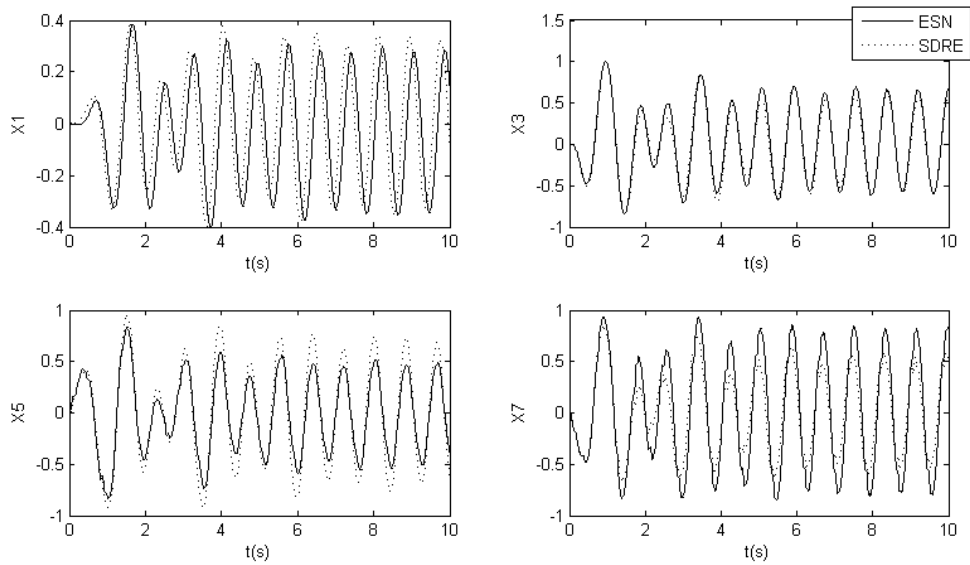


Figura 35 – Estados do sistema *half-car* com atuação ESN (Autoria própria)

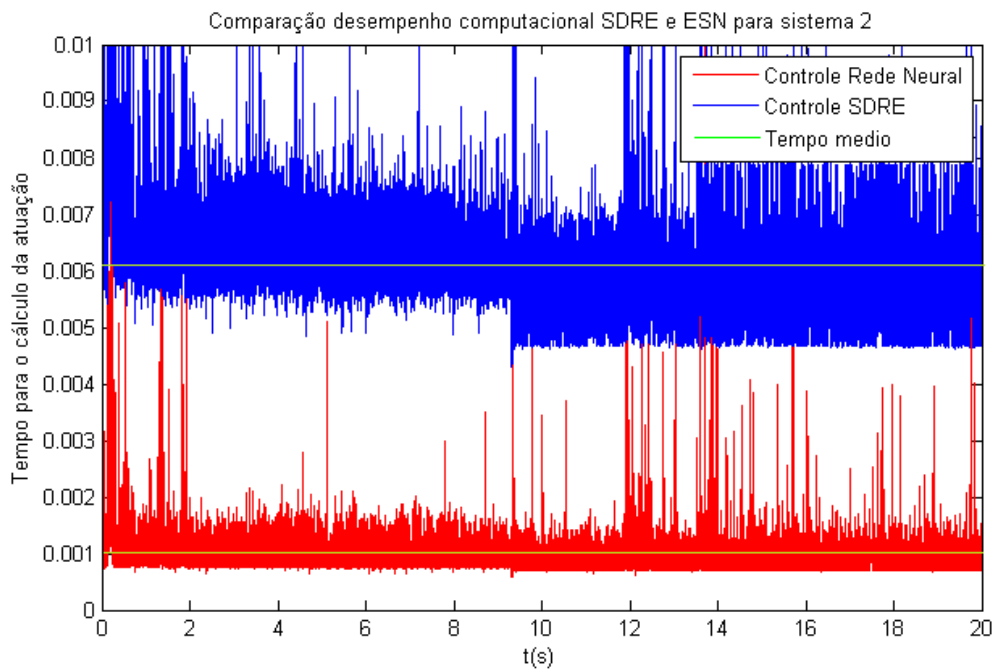


Figura 36 – Tempos para cálculo de atuação para SDRE e ESN com 120 neurônios no reservatório dinâmico sistema *half-car* (Autoria própria)

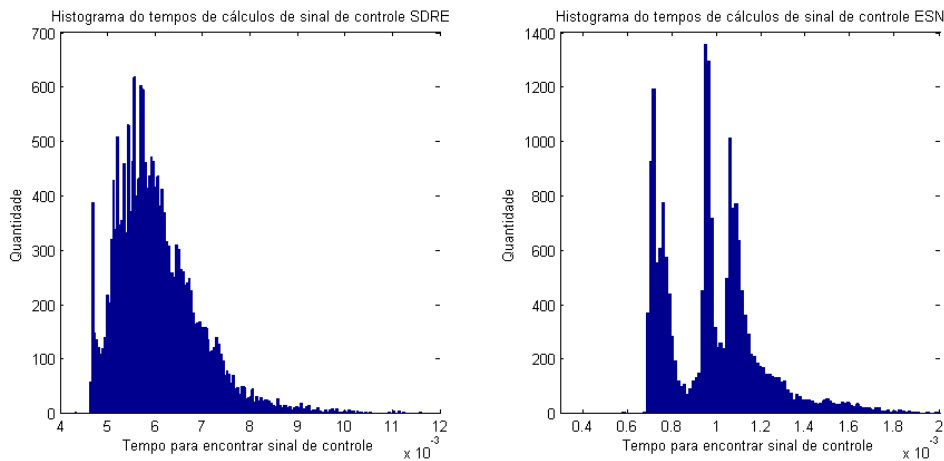


Figura 37 – Histograma ESN x SDRE para controle do sistema half-car (Autoria própria)

	SDRE	ESN 60	ESN 120	ESN 200	ESN 500
RMS X1	0,2348	0,2185	0,2183	0,2151	0,2259
max(mod(X1))	0,3961	0,4168	0,3993	0,4106	0,4220
RMS X3	0,4226	0,4535	0,4574	0,4592	0,4572
max(mod(X3))	1,0031	1,0684	1,0044	1,0540	0,9949
RMS X5	0,5097	0,4157	0,3687	0,3781	0,3653
max(mod(X5))	0,9575	0,9298	0,8342	0,8851	0,8163
RMS X7	0,3964	0,5096	0,7759	0,5499	0,5712
max(mod(X7))	0,8615	0,9292	0,9362	0,9825	0,9769
Tempo médio	0,0061	9,1041e-04	0,0010	0,0015	0,0064
Desv. Pad. Tempo	0,0011	4,5331e-04	6,9949e-04	8,2536e-04	5,5698e-04

Tabela 2 – Erros para SDRE e ESN com 60 neurônios sistema Half-car

é o valor máximo em módulo, representando a maior perturbação que houve em cada estado usando determinado controle.

A tabela 2 também mostra o desvio padrão e a média dos tempos necessários para fazer o cálculo de sinal de controle. Nota-se que a rede neural obteve um desvio padrão muito pequeno em relação ao desvio padrão do SDRE, representando que o tempo gasto para cálculo do sinal de controle difere menos da média que o SDRE. Com uma rede neural de 500 neurônios, a rede passa a ter um tempo médio maior do que o SDRE porém, com uma rede neural de 60 neurônios consegue-se ótimos resultados com valor médio de aproximadamente 6,7 vezes menor que o SDRE.

Como realizado para o sistema MEMs, para o *half-car* também foi verificada a influência de diferentes pesos da camada de entrada \mathbf{W}^{in} e do reservatório de dinâmicas \mathbf{W} , os quais são gerados aleatoriamente. Foi verificado o erro médio quadrático em relação ao comportamento desejado, ou seja, o comportamento da ESN em relação ao SDRE tanto para a roda dianteira quanto para a roda traseira. Foram gerados 130 redes neurais para a roda traseira e roda dianteira, treinadas com diferentes pesos do reservatório de dinâmicas (gerados aleatoriamente). Para cada rede foi verificado o erro em relação à atuação SDRE. A forma de apresentar

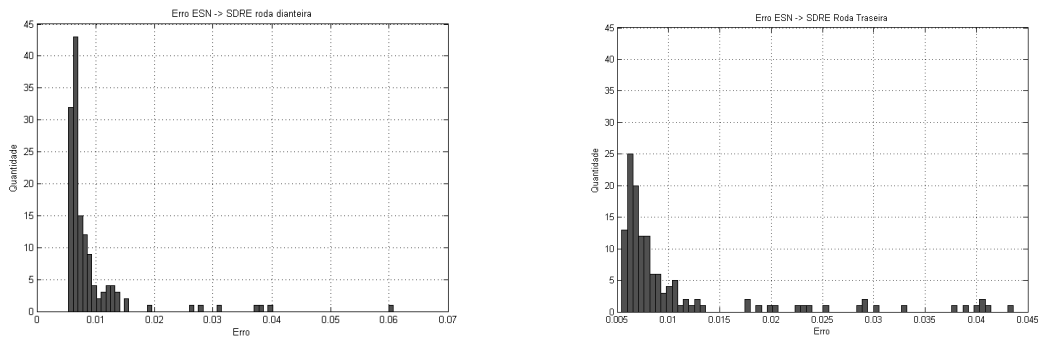


Figura 38 – Histogramas de erros da ESN em relação ao SDRE roda dianteira e roda traseira respectivamente (Autoria própria)

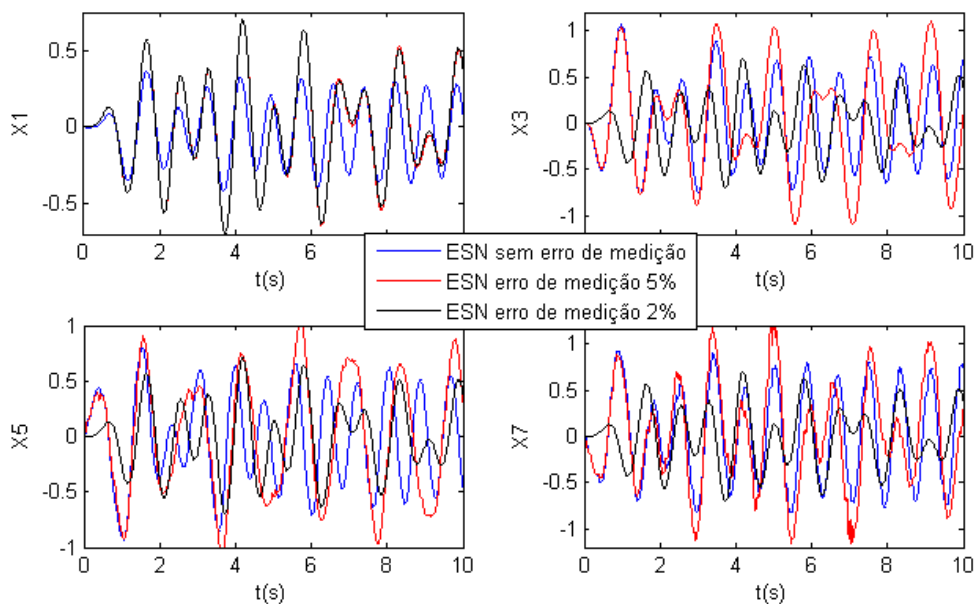


Figura 39 – Simulação com erro de leitura dos sensores de 5%, 2% e sem erro (Autoria própria)

estes erros foi através dos histogramas da Figura 38 tanto para a roda dianteira, quanto para a roda traseira. Como pode ser observado, a variação dos pesos do reservatório de dinâmicas não influenciam de forma considerável no comportamento da rede neural.

5.1.5 Sensibilidades a Erros Paramétricos

Para a sensibilidade a erros paramétricos do sistema *half-car* foi realizado o controle do sistema inserindo um ruído de 2% e de 5% na entrada da rede neural, simulando assim falhas nos sensores de aquisição dos dados do carro. Assim como para o sistema MEMs, a rede neural ainda conseguiu realizar o controle adequado com algumas diferenças notáveis na saída do sistema a partir da Figura 39, no qual é mostrado o resultado do controle da ESN sem perturbação, 2% de perturbação e 5% de perturbação. A Tabela 3 demonstra o crescimento do erro RMS para todas as variáveis, ao passo que o erro de medição aumenta. Portanto, a ESN, assim como o SDRE, são robustos em relação à presença de erros paramétricos. Com isso,

	ESN 60	ESN 60 ruído 2%	ESN 60 ruído 5%
RMS X1	0,2185	0,3215	0,3232
max(mod(X1))	0,4168	0,7057	0,7063
RMS X3	0,4535	0,5720	0,5743
max(mod(X3))	1,0684	1,1300	1,0982
RMS X5	0,4157	2,7599	0,5404
max(mod(X5))	0,9298	1,0805	1,0785
RMS X7	0,5096	0,5657	0,5673
max(mod(X7))	0,9292	1,4444	1,4308

Tabela 3 – Erros para SDRE e ESN com 60 neurônios com ruído sistema *Half-car*

percebe-se ainda que a rede neural conseguiu encontrar uma solução generalizada.

6 CONSIDERAÇÕES FINAIS

SOBRE AS CONTRIBUIÇÕES DO TRABALHO

No presente trabalho foram realizados estudos referentes às redes neurais, mais especificamente sobre a rede neural com estados de eco (ESN), sistemas de controle, principalmente usando a equação de Riccati dependente de estados (SDRE), e a elaboração de controle de dois sistemas: o primeiro um sistema micro eletromecânico (MEMS) e o outro um sistema de amortecimento de meio carro (*half-car*). O intuito do trabalho foi controlar os sistemas com a ESN a partir de amostras de comportamentos do controle SDRE a fim de produzir um controlador com as mesmas qualidades do controlador SDRE, porém, com economia de recursos computacionais.

Controladores SDRE são sistemas não-lineares que têm uma grande eficiência, exemplo disso é em (TUSSET et al., 2012b), onde foi demonstrado que apenas um controle SDRE, entre todas as técnicas de controle usadas no artigo, foi capaz de controlar o sistema *half-car* com sucesso. O uso do controlador SDRE em sistemas reais se torna custoso em relação ao uso de muito processamento e dificuldades de implementação, principalmente em sistemas embarcados, onde o processamento normalmente é limitado pelo dispositivo e pela falta de recursos de linguagem de programação, pois nem todos os microcontroladores possuem recursos avançados de programação comparando com softwares matemáticos como o MatLab.

As redes neurais são sistemas computacionais que fornecem a capacidade de reconhecer padrões, reproduzir séries e comportamentos se utilizando de arquiteturas baseadas no funcionamento do cérebro biológico. Dentre várias novas propostas de redes neurais que foram criadas recentemente, se tem a rede neural com estado de eco ESN (LUKOSEVICIUS; JAEGER, 2009), a qual se caracteriza por ser uma rede neural recorrente de grande eficiência. O fato dela ser recorrente faz com que seja uma boa ferramenta para manipular dados de sistemas que dependam do tempo para operação. Além disso, diferentemente de outras redes recorrentes a ESN, é de fácil treinamento, pois somente a parte sem recorrência da rede é treinada (camada de saída). Daqui, se parte para o propósito do trabalho, que foi a utilização da ESN para se comportar como um controlador SDRE porém, utilizando menos recursos computacionais e mais facilidade de implementação.

A tarefa se baseou em dois artigos científicos, nos quais se demonstram sistemas e seus respectivos controles utilizando o SDRE, através de simulações dos estudos realizados nesses. Com estas simulações dos sistemas se obteve um histórico de estados e sinal de controle calculados pelo controle, e utilizando esses dados foi realizado o treinamento da rede neural de estado de eco de diferentes formas. Com a rede neural treinada foi feito o controle dos dois sistemas. Analisando os resultados obtidos por meio do controle com a rede neural e os dados controle baseado em SDRE, foram verificados através de gráficos e tabelas com análises de erros: erro médio quadrático, erro médio, RMS e erro absoluto. Pode-se observar

que a rede neural conseguiu fazer um controle muito parecido com a forma de controle SDRE, o que garantiu assim, com esses resultados, que a rede neural é suficiente para realizar controle similar ao controle baseado em SDRE.

Tendo a rede neural controlado os dois sistemas com eficácia, foram realizados testes de custos computacionais através de medições de tempos. Essas medições foram realizadas no momento dos cálculos de atuações das duas técnicas de controle, nas quais o tempo para o cálculo de cada atuação foi guardado em uma estrutura de dados. Não somente uma configuração de ESN foi utilizada; variações da quantidade de neurônios e outras configurações foram realizadas para firmar que a rede é suficiente para realizar o controle de ambas as plantas. Para os dois casos de controle foram verificados esforços computacionais inferiores aos esforços capturados quando usado controle SDRE.

Foram realizados também testes em relação a robustez dos dois controladores. Foram inseridos erros na leitura das variáveis da planta, e então foi verificado o comportamento do controlador. Para esse caso o resultado também foi satisfatório por a ESN e SDRE trazerem resultados aceitável no controle.

SOBRE TRABALHOS FUTUROS

Outras propostas de controles podem ser usadas para a geração de dados para o treinamento da rede neural, principalmente outras técnicas que fazem o uso de grande processamento. Uma sequência para este trabalho seria: classificar a ESN como uma rede neural polimórfica que venha a assumir o comportamento de controles diferentes, onde se obtém o comportamento desejado do controlador a partir de amostras. E então treina-se uma ESN para se obter o mesmo comportamento do controlador.

O próximo passo deste trabalho é a implementação da rede neural em sistemas embarcados em que se constrói uma planta real de controle usando ESN através de microcontroladores muito difundidos no mercado como da Atmel, Microchips e outros. Apontando assim a possibilidade da rede neural ser colocada em sistemas reais, implementada em microcontroladores de relativo baixo custo, com o intuito de realização de controles eficientes sem o uso de microprocessadores de alto desempenho.

Outras redes neurais podem ser utilizadas para o mesmo fim, as quais possam se comportar de forma parecida com técnicas de controles existentes. Entre essas redes podem ser usadas interessantes propostas como as máquinas de aprendizado extremo (ELM), novas propostas de ESN com realimentação da camada de saída para o reservatório demonstradas em (JAEGER, 2008), entre outras redes neurais.

A ESN também pode ser utilizada como outra ferramenta de controle, como por exemplo a realização de um observador de estados. O que serviria para estimar estados do sistema, os quais não podem ser facilmente medidos por sensores. Isso poderia ser realizado através da estimação de um estado através de outros estados do sistema a ser controlado.

REFERÊNCIAS

- ABDELRAHMAN, M.; CHANG, I.; PARK, S.-Y. Magnetic torque attitude control of a satellite using the state-dependent riccati equation technique. **International Journal of Non-Linear Mechanics**, 2011.
- BANK, H. T.; LEWIS, B. M.; TRAN, H. T. Nonlinear feedback controllers and compensators. **A State Dependent Riccati Equation Approach Comput Optim Appl**, 2007.
- BANKS B. M. LEWIS, H. T. T. H. T. Nonlinear feedback controllers and compensators: a state-dependent riccati equation approach. **Computational Optimization and Applications**, 2007.
- BANKS, H. T. et al. An sdre-based approach for hiv feedback control and control of thin film growth in a cvd reactor. **Preprints of the 18th IFAC World Congress**, 2011.
- BANKS, H. T.; LEWIS, B. M.; TRAM, H. T. Nonlinear feedback controllers and compensators: a state-dependent riccati equation approach. **Comput. Optim. Appl.**, 2007.
- BARRETO, J. M. **Introdução às Redes Neurais Artificiais**. [S.l.: s.n.], 1997.
- BASSINELLO, D. G. **Comportamento Dinâmico Não Linear e Controle de Sistemas Eletromecânicos em Macro e Micro Escalas**. Dissertação (Mestrado) — Universidade Estadual Paulista Júlio de Mesquita Filho Faculdade De Engenharia De Bauru, 2011.
- BOCCATO, L. **Novas Propostas e Aplicações de Redes Neurais com Estados de Eco**. Tese (Doutorado) — UNIVERSIDADE ESTADUAL DE CAMPINAS FACULDADE DE ENGENHARIA ELETRICA E DE COMPUTAÇÃO, 2013.
- BOLTON, W. **Engenharia de Controle**. [S.l.]: Makron Books, 1993.
- BUPP, R. T.; BERNSTEIN, D. S.; COPPULA, V. T. A benchmark problem for nonlinear control design. **International Journal of Robust and Nonlinear Control**, 1998.
- BURGHART, J. A. A technique for suboptimal control of nonlinear systems. **IEEE Transactions on Automatic Control**, 1969.
- CHANG, S. C. I. Exponential stability region estimates for the state-dependent riccati equation controllers. **Decision and Control**, 2009.
- CLOUTIER, J. R.; D'SOUZA, C.; MRACEK, C. Nonlinear regulation and nonlinear control via the state-dependent riccati equation technique. **First International Conference on Nonlinear Problems in Aviation and Aerospace**, 1996.
- CYBENKO, G. Approximation by superpositions of a sigmoidal function, mathematics and control. **Signals, and Systems**, 1989.
- DORF, R. C. **Modern control systems**. [S.l.]: LTC, 2009.
- DUTOIT, X. et al. Pruning and regularization in reservoir computing. **Elsevier**, 2009.
- ERHAN, D. **Exploration of combining ESN learning with gradient-descent RNN learning techniques**. Tese (Doutorado) — School of Engineering & Science of the INTERNATIONAL UNIVERSITY BREMEN, 2004.

- FERREIRA, A. A.; LUDERMIR, T. B.; AQUINO, R. R. de. An approach to reservoir computing design and training. **Expert Systems with Applications**, 2013.
- FRANK, S. L. Generalization and systematicity in echo state networks. In: **FNWI: Institute for Logic, Language and Computation (ILLC)**. [S.l.: s.n.], 2008.
- GARRARD, W. L.; MCCLAMROCH, N. H.; CLARK, L. G. An approach to sub-optimal feedback control of nonlinear systems. **Internat. J. Control**, 1967.
- HAYKIN, S. **Neural Networks - A Comprehensive Foundation**. [S.l.]: Macmillan College Publishing Company, 1994.
- HAYKIN, S. **Redes Neurais Princípios e prática 2ª edição**. [S.l.]: Bookman, 2001.
- HOPFIELD, J. J. Neural networks and physical systems with emergent collective computational properties. **Proceedings of the National Academy of Sciences of the USA**, 1982.
- HUANG, G. B.; ZHU, Q. Y.; SIEW, C. K. Extreme learning machine: A new learning scheme of feedforward neural networks. **Proc. of Intern. Joint Conf. on Neural Networks**, 2004.
- ÇIMEN, T. State-dependent riccati equation (sdre) control. **A Survey World Congress**, 2008.
- ÇIMEN, T. Systematic and effective design of nonlinear feedback controllers via the state-dependent riccati equation (sdre) method. **Annual Reviews in Control**, 2010.
- JAEGER, H. The echo state approach to analysing and training recurrent neural networks. **Fraunhofer Institute for Autonomous Intelligent Systems**, 2001.
- JAEGER, H. A tutorial on training recurrent neural networks, covering bppt, rtrl, ekf and the echo state network approach. **Fraunhofer Institute for Autonomous Intelligent Systems (AIS)**, 2008.
- JUNG, J. et al. Hardware-in-the-loop simulations of spacecraft attitude synchronization using the state-dependent riccati equation technique.
- KEITH, T. K. **A General-Purpose GPU Reservoir Computer**. Tese (Doutorado) — University of Canterbury, Christchurch, New Zealand, April 2013.
- LIU, H.; NONAMI, K.; HAGIWARA, T. Active following fuzzy output feedback sliding mode control of real-vehicle semi-active suspensions. **Journal of Sound and Vibration** **314**, 2008.
- LUKOSEVICIUS, M.; JAEGER, H. Reservoir computing approaches to recurrent neural network training. **Computer Science Review**, 2009.
- LUO, A. C. J.; WANG, F. Y. Chaotic motion in a micro-electro-mechanical system with non-linear form capacitors. **Communications in Nonlinear Science and Numerical Simulation** **7 (31-49)**, 2002.
- MAASS, W.; NATSCHLÄGER, T.; MARKRAM, H. Real-time computing without stable states: a new framework for neural computation based on perturbations. **Neural Comput**, 2002.
- MACHADO, A. B. M. **Neuroanatomia Funcional**. [S.l.]: 2ª Ed - São Paulo, 2006.
- MASSAR, M.; MASSAR, S. Mean-field theory of echo state networks. **PHYSICAL REVIEW**, 2013.

MAURO, G. D. et al. Experimental implementation of sdre method for autonomous rendezvous and docking maneuvering. **5th International Conference on Spacecraft Formation Flying Missions and Technologies**, 2013.

MCCULLOCH, W.; PITTS, W. A logical calculus of the ideas immanent in nervous activity. **Bulletin of Mathematical Biophysics**, 1943.

MENON, P. K.; LAM, T.; CHENG, L. S. C. V. H. L. Real-time computational methods for sdre nonlinear control of missiles. **American Control Conference, 2002**, 2002.

MINSKY, M. L.; PAPERT, S. A. Perceptrons. **MIT Press**, 1969.

MOLTER, A. **Controle de Manipuladores Robóticos Flexíveis Usando Atuadores e Sensorres Piezelétricos Otimizados**. Tese (Doutorado) — Universidade Federal do Rio Grande do Sul, 2008.

MRACEK, C.; CLOUTIER, J. Full envelope missile longitudinal autopilot design using the state-dependent riccati equation method. **Proc. of the AIAA Guidance, Navigation, and Control Conference**, 1997.

MRACEK, P. C.; CLOUTIER, J. R. Control designs for the nonlinear benchmark problem via the state-dependent riccati equation method. **International Journal of robust and nonlinear control**, 1998.

NISE, N. S. **Control systems engineering - 3rd ed.** [S.l.]: New York: John Wiley & Sons, 2000.

OGATA, K. **Engenharia de controle Moderno**. [S.l.]: Pearson, 2011.

OUBBATI, M. **Neural Dynamics for Mobile Robot Adaptive Control**. Tese (Doutorado) — Institut fur Parallele und Verteilte Systeme der Universitat Stuttgart, 2006.

PEARSON, J. E. Approximation methods in optimal control i. sub-optimal control. **International Journal of Electronics**, 1962.

ROSENBLATT, F. The perceptron: A probabilistic model for information storage and organization in the brain. **Psychological Review**, 1958.

RUMELHART, D.; HINTON, G.; WILLIAMS, R. Learning representations by backpropagation errors. **Nature**, 1986.

SALMEN, M.; PLOGER, P. G. Echo state networks used for motor control. **International Conference on Robotics and Automation**, 2005.

SCHMID, A. B.; RAFIKOV, M. Síntese do controle linear aplicado em sistemas populacionais. **Tema - Tendencias em Matemática Aplicada e Computacional**, 2005.

SHAWKY, A. M. et al. Position control of flexible manipulator using nonlinear h with state-dependent riccati equation. **Jornal of Systems and Control Engineering**, 2007.

SILVA, I. N. da; SPATTI, D. H.; FLAUZINO, R. A. **Redes Neurais Artificiais**. [S.l.]: Artliber Editora, 2010.

SILVA, J. V. **Fundamentos das Redes Neurais**. [S.l.]: Livro Rápido, 2011.

SILVA, S. **Estudo da Fusão dos Métodos LQG/LTR E SDRE Para o Controle de uma Classe de Sistemas Não Lineares**. Tese (Doutorado) — Instituto Nacional de Pesquisas Espaciais, 2014.

SIQUEIRA, H. V. **Máquinas Desorganizadas para Previsão de Séries de Vazões**. Tese (Doutorado) — FACULDADE DE ENGENHARIA ELÉTRICA E DE COMPUTAÇÃO, 2013.

SOUZA, L.; GONZÁLES, R. Application of the state-dependent riccati equation and kalman filter techniques to the design of a satellite control system. **Shock and Vibration**, 2012.

SOUZA, L. C. G. de; ARENA, V. M. R. Design of satellite attitude control algorithm based on the sdre method using gas jets and reaction wheels. **Journal of Engineering**, 2013.

TURING, A. M. Intelligent machinery. **Cibernetis: Key Papers**, 1968.

TUSSET, A. M. et al. Chaos suppression in nems resonators by using nonlinear control design. **9th International Conference on Mathematical Problems in Engineering, Aerospace and Sciences**, 2012.

TUSSET Ângelo M. et al. Suppression of vibrations in a nonlinear half-car model by magneto-rheological damper. **International Conference on Mathematical Problems in Engineering, Aerospace and Sciences**, 2012.

VERSTRAETEN, D. **Reservoir Computing: computing with dynamical systems**. Tese (Doutorado) — Ghent University, 2009.

WERBOS, P. J. **Beyond regression: New tools for prediction and analysis in the behavioral ciences**. Tese (Doutorado) — Harvard University, 1974.

WERNLI, A.; COOK, G. Suboptimal control for the nonlinear quadraticregulator problem. **Automatica**, 1975.

WIDROW, B.; HOFF, M. Adaptative switching circuits. **IRE WESCON Convention Record**, 1960.

WIENER, N. **Cybernetics or Control and Communication in the Animal and the Machine**. [S.l.]: Mit Press, 1948.