

Universidade Federal do Maranhão
Centro de Ciências Exatas e Tecnologia
Programa de Pós-Graduação em Engenharia de
Eletricidade

*Métodos Neurais para a Solução da Equação
Algébrica de Riccati e o LQR*

Fábio Nogueira da Silva

São Luís
Junho 2008

Universidade Federal do Maranhão
Centro de Ciências Exatas e Tecnologia
Programa de Pós-Graduação em Engenharia de
Eletricidade

*Métodos Neurais para a Solução da Equação
Algébrica de Riccati e o LQR*

Fábio Nogueira da Silva

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia de Eletricidade da UFMA como parte dos requisitos necessários para obtenção do grau de Mestre em Engenharia Elétrica.

São Luís
Junho 2008

Silva, Fábio Nogueira da

Métodos Neurais para a solução da Equação Algébrica de Riccati e o LQR / Fábio Nogueira da Silva. – São Luís, 2008.

108f.

Orientador: João Viana da Fonseca Neto.

Impresso por computador (Fotocópia).

Dissertação (Mestrado) – Universidade Federal do Maranhão, Programa de Pós-Graduação em Engenharia de Eletricidade. São Luís, 2008.

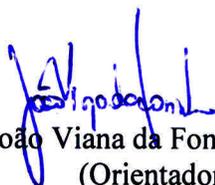
1. Equação algébrica de Riccati 2. Regulador linear quadrático 3. Redes neurais I. Fonseca Neto, João Viana da orientador II. Título.

CDU 681.511.2:004.032.26

MÉTODOS NEURONAIS PARA A SOLUÇÃO DA EQUAÇÃO ALGÉBRICA DE RICCATI E O LQR

Fábio Nogueira da Silva

Dissertação aprovada em 20 de junho de 2008.



Prof. João Viana da Fonseca Neto, Dr.
(Orientador)



Prof. Allan Kardec Duailibe Barros Filho, Ph. D.
(Co-orientador)



Prof. José Carlos Campos Teles, Dr.
(Membro da Banca Examinadora)



Profa. Valeska Martins de Souza, Dra.
(Membro da Banca Examinadora)

”O que mais te impressiona nesta vida?

Os homens. Eles perdem a saúde para juntar dinheiro, depois perdem o dinheiro para recuperar a saúde. E por pensarem ansiosamente no futuro esquecem do presente de forma que acabam por não viver nem no presente nem no futuro. E vivem como se nunca fossem morrer... e morrem como se nunca tivessem vivido.”

Dalai Lama, líder do Gelug uma divisão do Budismo Tibetano

Dedicatória

Aos meus pais Maria do Socorro Nogueira da Silva e Bonifácio José da Silva, pela compreensão e paciência; a minha irmã Karine Nogueira de Abreu, pelo incentivo e apoio; aos colegas de trabalho do laboratório de controle de processos *LCP*, que tornaram o dia à dia mais agradável para o trabalho e pelas mais diversas discussões sempre de algum modo produtivas e pela receptividade; ao professor João Viana por toda a confiança em mim depositada e incentivo, assim como as diversas conversas, orientações e discussões e finalmente a Deus.

Agradecimentos

Ao Programa de Pós-Graduação em Engenharia de Eletricidade (*PGEE*) pelo apoio a produção e desenvolvimento científico; ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (*CNPQ*), pelo apoio financeiro a pesquisa científica o que sem ela não seria possível concluir esse trabalho; a Fundação de Amparo à Pesquisa e ao Desenvolvimento Científico e Tecnológico do Estado do Maranhão (*FAPEMA*), pelo apoio financeiro para a divulgação de resultados deste trabalho em congressos científicos; a meus colegas de trabalho do laboratório de controle de processos, pelos mais diversos tipos de apoios, conselhos e discussões, para o amigo Ivanildo da Silva Abreu, pelos dias de trabalho e conselhos dia à dia e pelas sábias observações. A João Inácio, pela paciência transparente e pelo modo como passa tranquilidade e segurança às pessoas; a Renan, Samy, Aline, pelo modo ativo de incentivar a todo o laboratório e inclusive a min; a Patricia Rego pelas discussões científicas e apontamentos sobre os trabalhos de pesquisa e suas colocações bem apropriadas sobre rigor e aos colegas, Jorge Henrique, Jorge Farid, Marcio Mendes, pelas discussões sobre projetos e planos de vida futuros.

RESUMO

Apresenta-se nesta dissertação os resultados a respeito de dois métodos neuronais para a resolução da equação algébrica de *Riccati*(*EAR*), que tem varias aplicações, sendo principalmente usada pelos Regulador Linear Quadrático(*LQR*), controle H_2 e controle H_∞ . É apresentado a *EAR* real e simétrica e dois métodos baseados em uma rede neuronal direta (*RND*) que tem a função de erro associada a *EAR* e uma rede neuronal recorrente (*RNR*) que converte um problema de otimização restrita ao modelo de espaço de estados em outro de otimização convexa em função da *EAR* e do fator de *Cholesky* de modo a usufruir das propriedades de convexidade e condições de otimalidade.

Uma proposta para a escolha dos parâmetros da *RNR* usada para solucionar a *EAR* por meio da geração de superfícies com a variação paramétrica da *RNR*, podendo assim melhor sintonizar a rede neuronal para um melhor desempenho.

Experimentos computacionais relacionados a perturbações nos sistemas foram realizados para analisar o comportamento das metodologias apresentadas, tendo como base o princípio dos métodos homotópicos, com uma boa condição inicial, a partir de uma ponto de operação estável e comparamos os resultados com o método de *Schur*. Foram usadas as plantas de dois sistemas: uma representando a dinâmica de uma aeronave e outra de um motor de indução eólico duplamente alimentado(*DFIG*), ambos sistemas de 6^a ordem. Os resultados mostram que a *RNR* é uma boa alternativa se comparado com a *RND* e com o método de *Schur*.

Palavras-Chave: Equação Algébrica de Riccati, Regulador Linear Quadrático, Redes Neuronais, Redes Neuronais Recorrentes, Redes Neuronais Diretas, Otimização, Controle.

ABSTRACT

We present in this work the results about two neural networks methods to solve the algebraic *Riccati*(*ARE*), what are used in many applications, mainly in the Linear Quadratic Regulator (*LQR*), H_2 and H_∞ controls. First is showed the real symmetric form of the *ARE* and two methods based on neural computation. One feedforward neural network (*FNN*), that defines an error as function of the *ARE* and a recurrent neural network (*RNN*), which converts a constrain optimization problem, restricted to the state space model, into an unconstrained convex optimization problem defining an energy as function of the *ARE* and *Cholesky* factor. A proposal to chose the learning parameters of the *RNN* used to solve the *ARE*, by making a surface of the parameters variations, thus we can tune the neural network for a better performance.

Computational experiments related with the plant matrices perturbations of the tested systems in order to perform an analysis of the behavior of the presented methodologies, that are based on homotopies methods, where we chose a good initial condition and compare the results to the *Schur* method. Two 6th order systems were used, a Doubly Fed Induction Generator(*DFIG*) and an aircraft plant. The results showed the *RNN* a good alternative compared with the *FNN* and *Schur* methods.

Keywords: Algebraic Riccati Equation, Linear Quadratic Regulator, Neural Networks, Recurrent Neural Networks, Feedforward Neural Networks, Optimization, Control.

Lista de Tabelas

5.1	Tabela com as variações feitas as matrizes A , B e a ambas simultaneamente	56
5.2	Norma de <i>Frobenius</i> para a solução da <i>EAR</i> para variações nos parâmetros da planta	57

Lista de Figuras

4.1	Diagrama da Rede Direta	34
4.2	Diagrama da Rede Recorrente	40
5.1	Função de Ativação usada pelos neurônios	45
5.2	$norma_{\max}$ da solução P para um sistema de 6 ^a ordem representando a planta da aeronave.	46
5.3	$norma_{\infty}$ da solução P para um sistema de 6 ^a ordem representando a planta da aeronave	47
5.4	$norma_{\max}$ e $norma_{\infty}$ da solução P para a variações do N_{Sample} para um sistema de 3 ^a ordem.	48
5.5	n_{\max} da solução V para a variação η_v : 1-11000 e η_z :1000-40000, para um sistema de 5 ^a ordem	49
5.6	Superfície da Energia para variação dos Parâmetros - $\eta_v = 1450-1540$; $\eta_z = 1000000-1600000$	50
5.7	$norma_{\infty}$ das soluções da EAR para variação parâmetros - $\eta_v = 1000-1500$; $\eta_z = 10000-200000$	51
5.8	Norma Infinito da ARE para variação parâmetros - $\eta_v = 1200-1550$; $\eta_z = 100000-500000$	52
5.9	Energia em função do número de iterações.	53
5.10	Norma Infinito da solução da EAR em função do número de iterações.	54
5.11	Erro da Rede Direta para a Planta Perturbada.	58
5.12	Erro Rede Direta para a Matriz B de Entrada Perturbada.	59
5.13	Erro Rede Direta para a Matriz B e Planta A Perturbada.	59
5.14	Norma de <i>Frobenius</i> da Solução da RNR para Perturbações na Planta.	60

5.15	Norma de <i>Frobenius</i> da Solução da <i>RNR</i> para Perturbações na Matriz <i>B</i> de Entrada do Sistema.	61
5.16	Norma de <i>Frobenius</i> da Solução da <i>RNR</i> para Perturbações na Planta e Matriz <i>B</i> de Entrada do Sistema.	61
5.17	Energia para Perturbações na Planta do Sistema.	62
5.18	Energia para Perturbações na Matriz <i>B</i> de Entrada.	63
5.19	Energia para Perturbações na Planta e Matriz <i>B</i> de Entrada.	63
5.20	Norma de <i>Frobenius</i> da Diferença Entre as Soluções de <i>Schur</i> e <i>RNR</i> para Perturbações na Planta do Sistema.	64
5.21	Norma de <i>Frobenius</i> da Diferença Entre as Soluções de <i>Schur</i> e <i>RNR</i> para Perturbações na Matriz <i>B</i> de Entrada.	65
5.22	Norma de <i>Frobenius</i> da Diferença Entre as Soluções de <i>Schur</i> e <i>RNR</i> para Perturbações na Planta e Matriz <i>B</i> de Entrada.	65
E.1	Diferença da Norma de <i>Frobenius</i> para as Soluções da <i>EAR</i> usando o método de <i>Schur</i> e <i>RND</i> para Planta Perturbada	90
E.2	Diferença da Norma de <i>Frobenius</i> das Soluções da <i>EAR</i> usando o método de <i>Schur</i> e <i>RND</i> para a Matriz <i>B</i> Perturbada	91
E.3	Diferença da Norma de <i>Frobenius</i> das Soluções da <i>EAR</i> usando o método de <i>Schur</i> e <i>RND</i> para Planta e Matriz <i>B</i> Perturbadas	91
E.4	Norma de <i>Frobenius</i> da Solução da <i>RND</i> para Perturbações na Planta	92
E.5	Norma de <i>Frobenius</i> da Solução da <i>RND</i> para Perturbações na Matriz de Entrada <i>B</i>	93
E.6	Norma de <i>Frobenius</i> da Solução da <i>RND</i> para Perturbações na Planta e Matriz <i>B</i> de Entrada do Sistema	93
F.1	n_{\max} da solução <i>V</i> para a variação η_v : 1-600 e η_z :500-2500 para um sistema de 6 ^a ordem da aeronave.	95
F.2	n_{\max} da solução <i>V</i> para a variação η_v : 1-120 e η_z :500-2500 para um sistema de 6 ^a ordem da aeronave.	96
F.3	n_{\max} da solução <i>V</i> para a variação η_v : 1-120 e η_z :100-1000 para um sistema de 6 ^a ordem da aeronave.	96
G.1	Decomposição de Valores Singulares para o Planta do DFIG	98

G.2 Decomposição de Valores Singulares para o Planta da Aeronave . 99

Lista de Abreviaturas e Siglas

<i>EARC</i>	Equação Algébrica de Riccati Contínua
<i>EAR</i>	Equação Algébrica de Riccati
<i>ARE</i>	Algebraic Riccati Equation
<i>LQR</i>	Linear Quadratic Regulator (Regulador Linear Quadrático)
<i>RNR</i>	Rede Neural Recorrente
<i>RNN</i>	Recurrent Neural Network
<i>RND</i>	Rede Neural Direta
<i>FNN</i>	Feedforward Neural Network
<i>EDR</i>	Equação Diferencial de Riccati
<i>EMQ</i>	Erro Médio Quadrático
<i>DFIG</i>	Doubly Fed Induction Generator
<i>SVD</i>	Singular Values Decomposition
<i>MIMO</i>	Multiple Input/Multiple Output.

Lista de Símbolos

Q	Matriz de Ponderação Associada ao Vetor de Estado
R	Matriz de Ponderação Associada ao Vetor de Controle
P	Solução da Equação Algébrica de Riccati
η_v	Parâmetro de Aprendizagem da RNR
η_z	Parâmetro de Aprendizagem da RNR
μ	Taxa de Aprendizagem da RND
$G(P)$	Subespaço Grafo de P
$\nu(X)$	= min(Nº de Autovalores > 0 de X , Nº de Autovalores < 0 de X)
ϵ_j	Parâmetro de Característica do Sinal de Um Par de Matrizes
\mathfrak{M}	Conjunto dos Subespaços H-positivo M-invariante
$\mathfrak{R}_{-\alpha \pm i\beta}(A)$	O subespaço das Raízes de uma Matriz Real $A \in \mathbb{R}^{(n \times n)}$
$\mathfrak{C}_{A,B}$	Subespaço Controlabilidade do Par de Matrizes (A, B)
J	Índice de Desempenho Quadrático
$H(x(t), u(t), p(t), t)$	Hamiltoniano do Processo
u	Lei de Controle
e	Erro da Rede Neuronal Direta
K	Matriz de Ganho de Realimentação de Estado
L	Fator de <i>Cholesky</i>
ξ	Função Energia da Rede Neuronal Recorrente
P	Camada de Saída da Rede Neuronal Recorrente Solução da EAR
U	Camada de Entrada da Rede Neuronal Recorrente
Y,L	Camadas Ocultas da Rede Neuronal Recorrente

Sumário

1	Introdução	9
1.1	Objetivo	10
1.2	Motivação	11
1.3	Justificativa	12
1.4	Organização da Dissertação	13
2	Equações de Riccati	16
2.1	Estrutura Algébrica da EAR	17
2.2	Perturbação da Equação Algébrica de Riccati	22
2.3	Conclusão	23
3	Aplicações da EAR no LQR	24
3.1	Controle Ótimo	25
3.2	Princípio Mínimo de Pontryagin	28
3.3	A EAR e o LQR	29
3.4	Conclusão	31
4	Redes Neurais para a Solução da EAR	32
4.1	Rede Neural Direta	33
4.1.1	Formulação da Rede Direta	33
4.2	Rede Neural Recorrente	37
4.2.1	Formulação do problema para a solução da EAR	37
4.2.2	Estrutura da Rede Neuronal e Equações Dinâmicas	38
4.3	Conclusão	41

5	Estabilidade e Solvabilidade	42
5.1	Determinação dos Parâmetros η_v e η_z	43
5.1.1	Função de Ativação e Normas	44
5.1.2	Variação da Ordem do Sistema	48
5.1.3	Experimentos Planta DFIG	50
5.2	Perturbações Paramétricas no Sistema(DFIG)	54
5.3	Conclusão	66
6	Conclusão	67
6.1	Propostas Futuras	68
	Referências Bibliográficas	69
A	Método de Runge Kutta	74
B	Resultados Analíticos da Rede Neural	76
C	Plantas dos Sistemas	81
D	Método de Schur	85
E	Rede Neuronal Direta	88
E.1	Comportamento do Treinamento e Convergência da RND com Perturbações	90
F	Aeronave	94
G	Decomposição de Valores Singulares	97

CAPÍTULO 1

Introdução

Com o processo de globalização e abertura dos mercados de todo o mundo, ocorreu de forma direta e indiretamente que os países buscassem adequar seus processos industriais de modo a acompanhar os avanços dos seus concorrentes e com isso poder concorrer de modo mais competitivo. Investimentos em pesquisa, ciência e tecnologia se tornaram cada vez mais freqüentes, sejam por interesses industriais, sociais, esportivos, ecológicos, espaciais, militares, médicos ou mesmo para a melhoria da qualidade de vida e bem estar do ser humano.

Diversas teorias têm sido desenvolvidas nas últimas décadas visando compreender melhor os fenômenos, processos e padrões envolvidos nos processos industriais. Compreensão de sistemas lineares, não-lineares, dinâmicos, estáticos, caóticos, contínuos, discretos, estocásticos e outros padrões até mesmo aleatórios sem uma distribuição clara para a sua representação. Várias técnicas para a identificação desses sistemas foram desenvolvidas, muitas delas empregadas até hoje e muitas outras em desenvolvimento contínuo.

Processos industriais, equipamentos tecnológicos, mecânicos, elétricos, químicos e vários outros, para um bom funcionamento possuem restrições operacionais e questões que necessitam de algum tipo de controle que garanta que o processo permaneça em funcionamento. É necessário entender como se aplicar o controle e ter um controle continuamente supervisionado. A teoria de controle visa apresentar uma abordagem concisa e robusta para esse problemas. Com o desenvolvimento da teoria de controle, várias vertentes foram se solidificando, pois, em muitos casos se deseja controlar de modo ótimo, de modo automático e que para algumas

interferências o sistema possa retornar para uma condição de operação estável.

Uma série dos problemas acima mencionados estão relacionados ao controle e tem ligações diretas sendo estas possíveis de serem solucionadas por meio da resolução da equação algébrica de *Riccati*. Diversos métodos para a solução da *EAR* em diversas áreas podem ser facilmente encontradas. Referências sobre os métodos numéricos, variações dos métodos clássicos como o de *Schur* dentre artigos com questões relacionadas a estrutura interna dos métodos e sobre convergência, estabilidade, desempenho podem ser encontradas nas referências seguintes (Laub e Gahinet 1997), (Nils R. Sandell 1974), (Charles Kenney e Wette 1989), (Kleiman 1968), (Martensson 1972), (Peter Benner 1998), (Gülsu e Sezer 2006).

O estudo relacionado ao condicionamento numérico, quase singularidade das constantes da *EAR* e precisão numérica foram abordados em diversos trabalhos, (Byers 1985), (Golub e Wilkinson 1976) que visa verificar o comportamento numérico da solução da *EAR*. Métodos clássicos usando álgebra simplética, função sinal e outras formulações estruturais (Trapp 1985), (Byers 1987), (Gardiner e Laub 1986), (Rodrigues-Canabal 1973), (Faibusovich 1986), (Ammar e Mehrmann 1991), (Loan 1984).

1.1 Objetivo

O objetivo deste trabalho é abordar uma metodologia baseada em inteligência computacional, mais especificamente na descrição de duas redes neuronais para a resolução da *EAR* sendo uma rede direta e outra recorrente. Apresentar uma alternativa a sintonia da rede recorrente a partir da geração de superfícies com variações paramétricas da *RNR* que influenciam diretamente na solução da Equação Algébrica de *Riccati*, verificar a sensibilidade paramétrica dos modelos em relação à convergência e viabilidade do modelo. Comparar as metodologias com o método de *Schur* que é o mais usado. Finalmente, verificar a robustez em relação a perturbações nas plantas dos sistemas levando em consideração um ponto de operação anterior estável.

1.2 Motivação

Os mercados dos mais diversos setores espalhados pelo mundo estão interessados em aumentar a sua lucratividade em um tempo menor, com isso o processo de automação se expandiu cada vez mais e outro ramo de pesquisa que cresceu largamente foi o de otimização, que tem como objetivo poder garantir e solucionar problemas de maximização e minimização de funções, sendo chamadas de função objetivo ou multi-objetivo. A Otimização é aplicada nos mais diversos setores da ciência, tecnologia, indústria, comércio e mercados financeiros.

A busca da solução de problemas em muitos casos existem inúmeras soluções e sempre estamos interessados nas melhores soluções, que resolvem o problema mais próximo do ótimo. Deste modo a proposta do controle e filtragem ótimo fica claro, associando a teoria de controle e filtragem juntamente com suas restrições com a teoria de otimização.

Vários processos e aplicações tem como restrição o tempo sendo este em muitos caso pequeno. Esses problemas são freqüentemente vistos em aplicações em tempo real, deste modo um grande esforço por conta dos pesquisadores tem sido feito para solucionar este problema. Muitos processos não podem parar, tem que ser continuamente executados e sobre quaisquer circunstâncias o sistema tem que retornar para uma condição de equilíbrio num menor tempo possível, ou em um menor tempo que o de resposta do processo. A formulação destes problemas não é uma tarefa fácil em muitos casos, pois, existem muitas variáveis envolvidas no processo o que torna o problema de modelagem e construção da função objetivo ainda mais complexo.

A equação algébrica de *Riccati* está associada à minimização entre a diferença absoluta do valor de saída do sistema e o valor desejado, de modo que um índice de desempenho ou função de custo é associada a esse erro e sendo este índice a função a ser minimizada. Várias técnicas foram propostas para a solução da *EAR*, sendo que a solução pelo método de *Schur* uma das mais usadas e difundidas (Athans e Falb 1966).

Com a crescente complexidade dos problemas, modelos mais realistas são necessários para a descrição dos mesmos, com isso as técnicas usadas para o cálculo e obtenção da solução deveriam evoluir ao mesmo passo. A complexidade de diversos sistemas fez com que muitos dos métodos clássicos que buscavam por soluções

analíticas tornaram a busca das soluções muito trabalhosas e em muitos casos até mesmo inviável. Com o advento dos computadores digitais, hoje é possível fazer a implementação de algoritmos e buscar por soluções aproximadas e com isso usar o computador para fazer essas aproximações e cálculos das soluções desse problema, que até então não eram viáveis de se solucionar por métodos analíticos. Os processos interativos e métodos numéricos se difundiram e a análise numérica fundamenta matematicamente essas buscas, a fim de garantir soluções estáveis ou regulares e até mesmo poder garantir se certa heurística ou algoritmo alcançará uma solução ou não.

1.3 Justificativa

Com a grande computabilidade proporcionada pelos computadores digitais, pode-se implementar vários métodos numéricos e outras técnicas de processamento paralelo, distribuído e técnicas evolutivas. Com isso a teoria de controle passou a ter uma poderosa ferramenta e o controle digital se tornou uma ferramenta viável de baixo custo em muitos casos.

Para certos tipos de controle é necessário uma resposta rápida para que o sistema continue com seu regime estável, tendo transitórios mais amortecidos e *overshootings* mais adequados às condições de operação desejáveis. O tempo para se estabilizar é de extrema importância que seja mínimo em várias aplicações, especialmente em tempo real. Assim intensificando uma crescente busca por técnicas que possam garantir a busca de uma solução de modo mais rápido e que as mesmas sejam estáveis.

Uma problemática muito encontrada nestes métodos é a busca de solução global, o que nem sempre é garantida. Problemas relacionados à convergência são também observados. Algoritmos paralelos e distribuídos tem sido usados de modo a minimizar os tempos de processamento e para aumentar o poder de processamento das informações, dos cálculos e com isso melhorar a resposta do sistema.

Uma abordagem recente no desenvolvimento dos algoritmos e dos processos de modelagem de sistemas, estão relacionados com as observações dos comportamentos dos processos biológicos, evolutivos, assim como à aprendizagem do ser

humano e do processo de evolução das espécies. A modelagem do conhecimento, da inteligência e discussões sobre as variáveis envolvidas nesse processo tem sido amplamente estudadas nas últimas décadas.

A inteligência artificial é a grande área que engloba as mais diversas técnicas e métodos para esses estudos. Técnicas de inteligência artificial usadas em controle são frequentemente chamados de controle inteligente. Lógica fuzzy, algoritmos genéticos, redes neurais são as principais vertentes da inteligência computacional aplicada em controle e filtragem.

A escolha de uma técnica de inteligência artificial é decisiva, pelo modo como é feito o processamento da informação, baseado nos sistemas biológicos de uma rede neuronal, que possui processamento, paralelo e distribuído (Haykin 1999), com o atrativo de aprendizagem ao longo do processo iterativo.

Ganhos significativos no desempenho dos algoritmos e na qualidade da solução já são comprovados, (Zhang 2000), (Jain e Medsker 1999), (Kosko 1992) e (Michel e Liu 2002), fato determinante para a execução de um estudo mais detalhado visando o beneficiamento de implementações em sistemas de controle em tempo real.

O processamento paralelo e distribuído das redes neurais apresenta-se como uma alternativa viável para implementação em sistemas de controle e filtragem com necessidades de funcionamento em tempo real e mesmo para sistemas com constante de tempo maiores, pois, estas podem usufruir do processo de aprendizagem e tratamento de informação que é vital para um melhor funcionamento e sintonia no controle do processo.

1.4 Organização da Dissertação

Este trabalho é organizado pela caracterização e formulação do problema e solução da *EAR*. Discussões associadas aos problemas de controle, otimização e às técnicas usadas para a solução deste problema.

O Capítulo 2 discute sobre a *EAR* e sua estrutura, assim como uma breve discussão das várias formas da *EAR*, sobre a forma simétrica que é a forma usada neste trabalho assim como a abordagem geométrica da *EAR* que busca um solução real e simétrica em um sub-espaco invariante. A apresentação de lemas,

corolários, teoremas e definições que dão suporte para a existência de solução da *EAR*, assim como a apresentação da inserção de perturbação na *EAR*.

No Capítulo 3 apresentamos a teoria de controle, o problema *LQR*, a estrutura de otimização e a definição do índice de desempenho quadrático, sob o ponto de vista do Controle Ótimo. O *Princípio Mínimo de Pontryagin* que apresenta o surgimento da origem da *EAR* como parte da formulação do *LQR* e trata das condições para a existência de solução da *EAR* e a definição do *Hamiltoniano* associado à lei de controle.

No Capítulo 4 é feita a formulação dos métodos neuronais usados para o cálculo da *EAR*, sendo uma rede neuronal direta que define uma função de erro a partir da *EAR* e uma rede recorrente para representação de um sistema dinâmico associado ao problema de Controle Ótimo. O *LQR* converte o problema com restrições em um problema irrestrito a partir da definição de uma função energia convexa, que é função da *EAR* e da fatoração de *Cholesky*.

O desempenho das redes neuronais na solução da *EAR* é amplamente avaliado e discutido no Capítulo 5, tratando da estabilidade e solvabilidade das redes neuronais. Dois sistemas foram usados: um representando um *Gerador de Indução Eólico Duplamente Alimentado (DFIG)* e outro uma aeronave para avaliar por meio da escolha dos parâmetros da *RNR* a partir de uma heurística. Foram executadas uma análise das métricas usadas assim como uma comparação entre as mesmas e uma análise de robustez dos métodos em relação ao método de *Schur* com a inserção de perturbações paramétricas na *EAR*.

O Capítulo 6 é composto da Conclusão e discussões sobre trabalhos futuros, seguidos de 7 Apêndices, sendo o Apêndice *A* o método de *Runge – Kutta* de 4ª ordem usado no processo de integração da *RNR*.

No Apêndice *B* com resultados analíticos da *RNR* com a aplicação do segundo método de *Lyapunov* para a análise de estabilidade de sistemas dinâmicos.

O Apêndice *C* contendo as matrizes usadas por algumas simulações apresentadas no Capítulo 5. No Apêndice *D* a formulação do método de *Schur* para a solução da *EAR*.

No Apêndice *E* apresentamos uma demonstração sobre o desenvolvimento da *RND* com alguns resultados sobre o treinamento e convergência da rede direta.

No Apêndice *F* com análise de simulações e discussões relacionadas ao sistema

da aeronave e o Apêndice G uma análise do desempenho e estabilidade do planta $DFIG$ e da aeronave por meio da decomposição em valores singulares (SVD).

CAPÍTULO 2

Equações de Riccati

Neste Capítulo apresentaremos algumas das formas que a equação de *Riccati* assume, mais especificamente a de nosso interesse que está associada ao problema do Regulador Linear Quadrático (*LQR*), que é a forma real e simétrica da *EAR*.

Várias são as formas que a equação de *Riccati* assume. A forma diferencial e algébrica são as mais comuns, referências sobre trabalhos matemáticos sobre as estruturas, álgebra matricial, subespaços invariantes e decomposições da *EAR* podem ser vistas em: (Sergio Bittanti e Willems 1991), (Levin 1959), (Potter 1966), (Lewis 1981), (Kwon e Pearson 1977).

Na Seção 2.1 apresentaremos teoremas, corolários e lemas que são necessários para a compreensão da estrutura da *EAR*, de forma que as mesmas sirvam de suporte para a estruturação das redes neuronais discutidas nos próximos Capítulos. Na Seção 2.2 apresenta-se uma breve discussão sobre perturbações na equação algébrica de *Riccati*, seguidos da conclusão do Capítulo.

O estudo da *EAR* é de extrema necessidade para tornar possível a formulação e estruturação das redes neuronais, pois, com o conhecimento da estrutura algébrica da *EAR*, pode-se ajustar a rede neuronal de modo a garantir a convergência, estabilidade, desempenho e satisfazer as restrições associadas intrínsecas a *EAR*. Estaremos adotando como convenção que a sigla *EAR* refere-se a equação algébrica de *Riccati* contínua (*EARC*).

2.1 Estrutura Algébrica da EAR

São extensas as pesquisas e desenvolvimentos à cerca da *EAR*, nos focaremos um certo grupo de *EAR*. Uma das formas mais gerais que a *EAR* assume é a forma não simétrica e complexa, apresentada abaixo:

$$XDX + XA + BX + C = 0 \quad (2.1)$$

sendo os coeficientes A, B, C, D , matrizes quadradas de ordem n complexas ou reais e X é uma matriz ($n \times n$) que soluciona a Eq.(2.1). Temos, para o nosso problema o caso simétrico e real. Deste modo a equação assume a seguinte forma:

$$XDX + XA + A^*X + C = 0 \quad (2.2)$$

sendo C, D matrizes hermitinas e A^* é o complexo conjugado da matriz A^T a transposta da matriz A . Este tipo de equação é bastante usada em problemas relacionados à solução de sistemas de equações diferenciais lineares do tipo $\dot{x}(t) = Ax(t) + Bu(t)$ que são ótimas no sentido de maximizar algumas equações quadráticas para $x(t)$ e $u(t)$.

Apresentaremos a seguir uma abordagem relacionada a subespaços invariantes. Alguns resultados e teoremas importantes que dão suporte para a busca da solução da *EAR* são apresentados em linhas gerais. Para informações mais detalhadas, uma abordagem mais completa pode ser encontrada em (Lancaster e Rodman 1995):

Sejam $A, Q, S \in \mathbb{R}^{n \times n}$ e $Q^T = Q, S^T = S$. Considere a equação algébrica de *Riccati* contínua,

$$A^T P + PA + PSP - Q = 0 \quad (2.3)$$

e as matrizes reais ($2n \times 2n$)

$$M = \begin{bmatrix} A & S \\ Q & -A^T \end{bmatrix}, \hat{H} = \begin{bmatrix} 0 & I \\ -I & 0 \end{bmatrix}, H = \begin{bmatrix} -Q & A^T \\ A & S \end{bmatrix}, \quad (2.4)$$

portanto $\hat{H}M = -M^T \hat{H}$, $HM = -M^T H$ e $H = -\hat{H}M$. Desde que $H^T = -H$ e $\hat{H}M = -M^T H$, M é \hat{H} -anti-simétrica. Além do mais, $H^T = H$ e $HM = -M^T H$ implicam que M é H -anti-simétrica, sendo que H é não singular.

A noção de subespaço de um grafo é de grande importância para a construção do espaço de solução da Eq.(2.3), pois, usaremos resultados baseados nos grafos, para a apresentação do espaço solução da EAR . Estamos interessados nas soluções $P \in \mathbb{R}^{2n}$, em particular, P é solução real para a Eq.(2.3), se e somente se, o grafo $G(P)$

$$G(P) := \text{Im} \begin{bmatrix} I \\ P \end{bmatrix} \subseteq \mathbb{R}^{2n} \quad (2.5)$$

definido acima for M – *invariante*

Para a distinção entre as soluções simétricas e não simétricas as matrizes H e \hat{H} são fundamentais.

Proposição 2.1.1. *Seja P uma solução real de Eq.(2.3). Então,*

(α) *P é simétrica se e somente se $G(P)$ for \hat{H} – neutro*

(β) *O subespaço $G(P)$ é H - não positivo(respectivamente H - não negativo), se e somente se,*

$$(P^T - P)(A + SP) \leq 0 \quad (2.6)$$

respectivamente, $(P^T - P)(A + SP) \geq 0$

A inequação Eq.(2.6) significa que a matriz real $(P^T - P)(A + SP)$ é simétrica e semi-definida negativa.

Lema 2.1.1. *Entre as seguintes declarações, (ii) e (iii) são equivalentes, e cada uma delas implica em (i):*

(i) *Existe um subespaço n -dimensional H –neutro M –invariante;*

(ii) *Existe um subespaço n -dimensional \hat{H} –neutro M –invariante;*

(iii) *Para cada autovalor imaginário puro não nulo ib de M (se existir algum) teremos $\nu([z_i^T H z_j]_{i,j=1}^p) = \frac{p}{2}$, sendo z_1, \dots, z_p é uma base do subespaço par de $\mathfrak{R} \pm ib(M)$.*

Se, em adição, M for invertível, então todas as três declarações (i) – (iii) são equivalentes,(de fato, um subespaço M –invariante é H –neutro se e somente se, ele for \hat{H} –neutro)

Para a apresentação da próxima proposição, definiremos um parâmetro α , por conveniência como segue

$$\alpha(M, H) = \sum \epsilon_j \quad (2.7)$$

sendo ϵ_j os membros da característica do sinal de (M, H) associados as multiplicidades parciais ímpares (correspondendo aos autovalores imaginários puro da matriz M , se existir algum).

Lema 2.1.2. *Se M é invertível então a máxima dimensão de um subespaço M – invariante e H -não-positivo é*

$$n - \alpha(M, H) \quad (2.8)$$

Enunciaremos um teorema abaixo sobre um importante resultado sobre o número de autovalores máximos de um subespaço.

Teorema 2.1.1. *A máxima dimensão de um subespaço positivo ou não-negativo em relação ao produto escalar $[x, y] = \langle Hx, y \rangle$ coincide com o número de autovalores positivos de H (como multiplicidades algébricas).*

Para o caso quando H tem n autovalores positivos e negativos em vista do Teorema 2.1.1 e Lemma 2.1.2, acima apresentados isso só acontecerá, se e somente se, $\alpha(M, H) = 0$, ou, se e somente se existe um subespaço M – invariante H – não – positivo.

Outra possível contração, é uma condição que assegure que esta propriedade segue a definição da Eq.(2.4), quando A for razoavelmente grande se comparadas a Q e S . Para tornar esta declaração mais precisa usaremos o seguinte fato. Na declaração desta proposição a norma da matriz deve ser uma "potência submultiplicativa", ou seja, para qualquer matriz W , $n \times n$, e qualquer inteiro positivo m , a desigualdade $\|W^m\| \leq \|W\|^m$ é satisfeita.

Proposição 2.1.2. *Seja*

$$Y = \begin{bmatrix} -Q & A^T \\ A & S \end{bmatrix}$$

uma matriz real simétrica $2n \times 2n$, particionada em blocos $n \times n$. Assuma que A seja invertível e

$$\min(\|QA^{-1}S(A^T)^{-1}\|, \|(A^T)^{-1}QA^{-1}S\|) < 1 \quad (2.9)$$

Então Y é invertível e possui n autovalores positivos e n autovalores negativos

Teorema 2.1.2. *Assuma que $S \geq 0$, o par (A, S) controlável e a matriz M invertível. Então a Eq.(2.3) admite solução X , tal que*

$$(P^T - P)(A + SP) \leq 0 \quad (2.10)$$

se e somente se, $\alpha(H, M) = 0$. Se esta condição é satisfeita, então a fórmula

$$\mathfrak{M} = \begin{bmatrix} I \\ X \end{bmatrix} \quad (2.11)$$

estabelece uma correspondência um a um entre o conjunto de soluções X com a propriedade (2.10) e o conjunto \mathfrak{M} dos subespaços H -positivo M -invariante máximos.

Corolário 2.1.1. *Assuma a hipótese do teorema e assumo que ainda que A seja invertível e que o $\min(\|QA^{-1}S(A^T)^{-1}\|, \|(A^T)^{-1}QA^{-1}S\|) < 1$ para alguma norma de potência multiplicativa $\|\cdot\|$; portanto Eq.(2.3) admite solução P com a propriedade 2.10, e o conjunto de todos os P é descrito pela fórmula 2.11, sendo \mathfrak{M} subespaço n -dimensional H -positivo M -invariante qualquer.*

Este corolário é obtido usando a Proposição 2.1

Teorema 2.1.3. *Assuma que $S \geq 0$, e o par (A, S) controlável. Então a Eq.(2.3) admite uma solução real e simétrica, se e somente se, para cada par de autovalores puros imaginários não nulos $\pm ib$ de M , a igualdade*

$$\nu([z_i^T H z_j]_{i,j=1}^{2p}) = p \quad (2.12)$$

é satisfeita, sendo z_1, \dots, z_{2p} é uma base de $\mathfrak{R}_{\pm ib}(M)$. Neste caso a mesma fórmula, Eq.(2.11), estabelece uma correspondência um a um entre o conjunto das soluções reais e simétricas, P da Eq.(2.3) e o conjunto dos subespaços \mathfrak{M} , \hat{H} -neutro M -invariante máximos.

Para o próximo teorema definiremos o conceito relacionado ao sinal de controlabilidade de um par de matrizes.

Definição 2.1.1. Dizemos que um par de matrizes reais (A, B) , $A \in \mathbb{R}^{n \times n}$ e $B \in \mathbb{R}^{n \times m}$ é dita ser sinal controlável, se para cada número complexo $\alpha + i\beta$ ou

$$\mathfrak{R}_{\alpha \pm i\beta}(A) \subseteq \mathfrak{C}_{A,B} \quad (2.13)$$

ou

$$\mathfrak{R}_{-\alpha \pm i\beta}(A) \subseteq \mathfrak{C}_{A,B} \quad (2.14)$$

Usualmente, interpretamos $\mathfrak{R}_{\alpha \pm i\beta}(A)$ como o subespaço nulo se $\alpha \pm i\beta$ não são autovalores de A .

Teorema 2.1.4. *Assuma que $S \geq 0$, e que o par (A, S) é sinal controlável e M invertível. Então as condições (i)-(iii) abaixo são equivalentes.*

(i) *A Eq.(2.3) admite solução real P , tal que $(P^T - P)(A + SP) \leq 0$.*

(ii) *Existe um subespaço n -dimensional H -não-positivo M -invariante*

(iii) *A igualdade $\alpha(M, H) = 0$ é mantida.*

E se, A for invertível a condição Eq.(2.11) for satisfeita, então as condições (i)-(iii) são mantidas.

Lema 2.1.3. *Assuma a hipótese do teorema 2.1. Se existir um subespaço n -dimensional M -invariante H -positivo, então existe um subespaço, tal que o mesmo é também um subespaço grafo.*

Corolário 2.1.2. *Assuma que $S \geq 0, Q \geq 0, (A, S)$ estabilizável e (Q, A) detectável. Então a EARC Eq.(2.3) admite uma solução real simétrica. Além disso existe uma solução real simétrica P_0 em Eq.(2.3) que é estabilizável, isto é, de modo que $A + SP_0$ é c -estabilizável.*

Teorema 2.1.5. *Assuma que $S \geq 0$. Se EARC Eq.(2.3) tem uma solução P real e simétrica, e*

$$\mathfrak{R}_{-\alpha \pm i\beta}(A + SP) \subseteq \mathfrak{C}_{A,S} \quad (2.15)$$

para cada par de autovalores puros imaginários ou autovalores nulos $\pm i\beta$ de $A + SP$, então todas as multiplicidades parciais correspondendo aos autovalores puros imaginários e nulos de M se existirem, são pares. Além do mais a característica do sinal do par (M, H) consiste de -1 's para os autovalores imaginários puro e nulos, e o sinal associado a qualquer multiplicidade parcial par correspondendo a um autovalor nulo de M será $(-1)^{m-1}$, sendo $2m$ uma multiplicidade parcial.

2.2 Perturbação da Equação Algébrica de Riccati

Para o problema do regulador linear quadrático existe uma série de exigências sobre a positividade da solução da equação diferencial de *Riccati* matricial, bem como as condições necessárias e suficientes que garantam a existência de uma solução para o problema de minimização de um índice de desempenho quadrático associado ao *LQR*.

Perturbações na dinâmica do processo das variáveis relacionadas a *EAR*, produziriam perturbações na estrutura da equação em si, de modo que um estudo relacionado a perturbação da *EAR* é de extrema importância, assim como a verificação dos métodos para solucionar a *EAR*. Neste ponto os métodos neuronais são muito úteis, pois, os mesmos possuem relações de aprendizagem, o que torna possível lidar com essas variações na formulação perturbada da *EAR*.

No processo de entrada e saída do sistema pressupõe-se que o sistema seja linear e tanto as entradas quanto as saídas não possuam ruídos, pois, deste modo existe uma covariância associada ao comportamento dinâmico do sistema e com isso a própria equação diferencial de *Riccati* matricial, o que não é escopo deste trabalho, mas informações sobre a inserção de ruídos ao modelo e de observadores de estado, podem ser encontradas em (Goodwin e Sin 1984) e (Caines 1988).

A equação diferencial de *Riccati* é bastante encontrada em problemas de controle estocásticos, em processos que possuem em dinâmica que podem ser representadas por modelos lineares estocásticos, pois, o filtro de *Kalman* pode então ser aplicado para predição, filtragem e suavização dos estados desses sistemas, embora não seja apresentado neste trabalho o filtro de *Kalman* tem ligação direta com a equação algébrica de *Riccati* discreta e conseqüentemente com a forma contínua da *EAR*.

Para a resolução do problema associado ao *LQR*, é necessário determinar uma lei de controle ótimo que minimize o índice de desempenho quadrático. Esta lei de controle é função do ganho do controlador, sendo esta função da solução da *EAR*, (Moore 1969).

Métodos usando computação paralela, algoritmos recursivos e equação algébrica de *Riccati* discreta, para implementações em controle e aplicações também são de

grande interesse e muito usados. Alguns trabalhos foram decorridos por (Vaughan 1970), (Staffans 1996), (P. Lancaster e Rodman 1986), (Jonckheere 1981), (Gardiner e Laub 1991).

Essas técnicas de processamento são interessante de modo que com a inserção de perturbações ao sistema, implica que mais cálculos e computações serão necessárias para o cálculo da solução e neste ponto as redes neuronais com seu poder de adaptação e processamento paralelo se mostram interessantes.

Uma forma perturbada da *EAR*, Eq.(2.3) pode ser representada do seguinte modo:

$$(A + \Delta A)^T P + P(A + \Delta A) - P(S + \Delta S)P + Q = 0 \quad (2.16)$$

sendo $S + \Delta S = (B + \Delta B) R^{-1} (B + \Delta B)^T$, ΔA e ΔB são as matrizes de perturbação. Procurar métodos que solucionem a *EAR* a partir de uma condição anterior, de um estado estável e que retornem uma solução rápida é de grande valia para aplicações em tempo real.

A norma da perturbação relativa deve satisfazer as seguintes condições:

$$\|\Delta A\| \leq \epsilon \|A\|, \quad \|\Delta B\| \leq \epsilon \|B\| \quad (2.17)$$

Mais questões sobre teoria da perturbação podem ser encontradas em (Lancaster e Rodman 1995), abordando questões relacionadas às equações algébricas de *Riccati* contínuas e discretas.

2.3 Conclusão

Neste Capítulo foram apresentadas as estruturas de algumas equações algébricas de *Riccati*, da forma mais geral que é a forma complexa e não simétrica até a forma real e simétrica que é de nosso maior interesse.

A formulação apresentada na Seção 2.1, nos mostram as estruturas, lemas, teoremas e corolários referentes as condições para a existência e construção do conjunto solução para a *EAR*.

Perturbações na estrutura paramétrica da *EAR* apresentada serão usadas para avaliação do desempenho das redes neuronais, da robustez dos métodos por meio da análise de estabilidade e solvabilidade.

Aplicações da EAR no LQR

Na atualidade, diversos problemas são encontrados nos mais diversos tipos de setores ligados a indústria, ciências, tecnologia, militar, aeroespacial que necessitam de algum tipo de controle. A teoria de controle já é bastante utilizada, nesses diversos ramos.

Diversas aplicações podem ser vistas freqüentemente no nosso dia à dia, seja no controle de temperatura de um ar condicionado, nos celulares, no controle de combustível dos carros, no microondas, no controle dos mais diversos tipos de motores usados nos mais variados setores da indústria ou mesmo no processo de fabricação, acabamento, extração, mineração, automotivo, brinquedos, jogos e indústria eletrônica.

Neste Capítulo é feita uma introdução da importância do controle, mais especificamente do Controle Ótimo. A Seção 3.1 são apresentados alguns pontos sobre utilização e a importância do controle para diversas áreas, sendo as técnicas de otimização essenciais para a desenvolvimento e garantia de soluções ótimas.

O *Princípio Mínimo de Pontryagin* é apresentado com a finalidade de fornecer as condições necessárias e suficientes mínimas para a existência da solução do sistema de equações diferenciais associado ao problema de controle, apresentado na Seção 3.2. O problema do *LQR* e a ligação com a *EAR* é discutida na Seção 3.3 e finalmente Seção 3.4 uma breve conclusão das abordagens apresentadas no Capítulo.

3.1 Controle Ótimo

No controle ótimo as técnicas de otimização são de extrema importância, pois, as mesmas são usadas para se determinar as condições necessárias ou suficientes que possam garantir a existência de uma solução ótima para o controle, se ela existir.

A compreensão do comportamento das dinâmicas dos sistemas é uma das etapas fundamentais, pois, é necessário modelar o processo ou fenômeno de modo a se construir a função objetivo que será otimizada e quais as possíveis restrições relacionadas ao problema. Uma vez construída a função objetivo e compreendido o comportamento da mesma, pode-se então definir quais critérios e métodos seriam mais adequados para a minimização ou maximização da função em questão.

Técnicas como o gradiente descendente, programação dinâmica, métodos mais recentes como redes neurais e algoritmos genéticos têm sido usados com grande sucesso no processo de minimização ou maximização dessas funções dependendo do objetivo que se deseja com o processo de otimização.

Para se realizar o controle é necessário o conhecimento das leis que regem o sistema, deste modo uma das etapas seria a modelagem matemática, pois, um modelo matemático que descreva a dinâmica do sistema torna possível a aplicação do controle de modo mais eficiente.

Nas metodologias de projeto de controle, existe uma série de figuras de méritos que representam uma série de questões desejáveis e que nos dão mais informação a cerca do processo de controle e resposta dinâmica do sistema. Algumas dessas figuras de mérito como: tempo de subida, tempo de resposta, tempo de acomodação, ganho de margem e ganho de fase e largura de banda, são ferramentas que auxiliam no processo de controle e fornecem várias informações sobre o comportamento do processo e como o controle está atuando.

Métodos clássicos são usados e foram formulados com o propósito de solucionar problemas de controle de sistemas lineares com uma entrada e uma saída. Para o controle de sistemas mais complexos, não lineares e com múltiplas entradas e saídas as metodologias clássicas não são adequadas. A teoria de controle ótimo tem sido usada para o controle sistemas mais complexos lineares, não-lineares e multivariáveis. A indústria bélica e aeroespacial teve grande contribuição nas últimas décadas.

A aplicação das técnicas de controle ótimo necessitam de uma clara descrição dos índices de desempenho a serem alcançados pelo sistema, de um modelo matemático bem descrito que é representado freqüentemente por equações diferenciais ordinárias, da compreensão e determinação das restrições do sistema e do projeto de controle completo. Na prática, a dinâmica do processo de controle ótimo é freqüentemente um processo iterativo, em que o índice de desempenho é alterado a cada observação atual e avaliado e então atua na planta do sistema de modo a restabelecer ou mesmo manter a estabilidade do sistema, para um estado de operação próximo do especificado pelo projetista.

Estudos relacionados a robustez, projetos de controle, perturbações no sistema e o quão flexível a metodologia é em relação a inserção desses ruídos, a garantia das propriedades da alocação de auto estrutura relacionada a estabilidade e desempenho estão relacionados a equação de *Riccati*, (Kuriyama e Mita 1997), (V 1981), (J. J. Hench e Mehrmann 1998), (Cloutier 1997), (Carro e Torres 1977) e (Leden 1976).

Os problemas de otimização apresentados em cálculo visam a minimização ou maximização de uma função no espaço vetorial de dimensão finita. Para o problema de controle ótimo, as variáveis independentes de interesse são as trajetórias de controle e estado, sendo estas funções do tempo, portanto o problema de encontrar uma trajetória de controle que minimize um índice de desempenho, implica em encontrar o mínimo de um funcional, ou seja, uma função no espaço de funções, o que é um problema do cálculo de variações.

O caso mais clássico para um índice de desempenho quadrático instantâneo, assume a forma de uma integral no tempo, em que dado um sistema linear variante no tempo relacionadas a seguir:

$$\dot{x}(t) = Ax(t) + Bu(t), \quad x(0) = x_0, \quad (3.1)$$

$$y(t) = Cx(t) \quad (3.2)$$

sendo $x(t) \in \mathbb{R}^n$ o vetor de estado do sistema, $u(t) \in \mathbb{R}^m$ o vetor controle, $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$ são matrizes associadas ao vetor de estado $x(t)$ e ao vetor de entrada $u(t)$ respectivamente, x_0 representa o vetor com o estado inicial com $m \leq n$. A forma do índice de desempenho quadrático pode ser expressa da seguinte forma:

$$J = h(x(t_f), t_f) + \int_{t_0}^{t_f} g(x(t), u(t), t) dt \quad (3.3)$$

sendo t_0 e t_f , o tempo inicial e final, respectivamente. A função h é usada no problema de horizonte finito associado ao ajuste da ponderação do estado no tempo final e a função g calcula o valor instantâneo do índice de desempenho para um sinal de controle específico e particular estado.

Como os problemas de controle ótimo possuem a relação apresentada pela Eq.(3.1) que é vista como uma restrição para o problema de otimização, deste modo, para que possamos lidar com a relação do estado adiciona-se uma variável $p(t)$ representando o coestado. O coestado assume papel semelhante aos multiplicadores de *Lagrange* no cálculo de funções, ou seja, converte o problema de otimização restrita em um de otimização irrestrita. Deste modo temos que o índice de desempenho agora será representado do seguinte modo:

$$J = h(x(t_f), t_f) + \int_{t_0}^{t_f} g(x(t), u(t), t) + p(t)^T [f(x(t), u(t), t) - \dot{x}(t)] dt \quad (3.4)$$

Solucionar o índice de desempenho Eq.(3.4), em relação as variações independentes de $x(t)$, $p(t)$, $u(t)$, é semelhante a minimizar o índice de desempenho original, Eq.(3.3) em relação a $u(t)$. Para extremizar o índice de desempenho aumentado, Eq.(3.4), basta aplicar as condições de otimalidade:

$$\dot{x}^*(t) = f(x^*(t), u^*(t), t) \quad (3.5)$$

$$\dot{p}^*(t) = \left[\frac{\partial f}{\partial x}(x^*(t), u^*(t), t) \right]^T p^*(t) - \frac{\partial g}{\partial x}(x^*(t), u^*(t), t) \quad (3.6)$$

e

$$\frac{\partial g}{\partial u}(x^*(t), u^*(t), t) + \left[\frac{\partial f}{\partial u}(x^*(t), u^*(t), t) \right]^T p^*(t) = 0 \quad (3.7)$$

sendo que $*$, representa os valores ótimos. Uma equação adicional fornece os valores de contorno para as equações diferenciais, (Kirk 1970). Em geral, a equação que fornece os valores iniciais para a equação de estado Eq.(3.5) e as condições

finais para a equação do coestado Eq.(3.6). Solucionar o problema de controle ótimo se resume a solucionar um problema de contorno em torno de dois valores limites.

As equações diferenciais, Eq.(3.5) e Eq.(3.6) são geralmente equações de segunda ordem ou de ordem maior e em sua maioria equações não lineares. Soluções analíticas para equações diferenciais deste tipo geralmente não estão disponíveis. Técnicas de integração numérica como o método de *Runge – Kutta* não são aplicáveis, por não contornar o problema de descontinuidade nos limites das condições de contorno.

Métodos tentativa e erro podem ser mais demorados de retornar uma solução satisfatória ou até mesmo podem não retornar um solução viável, porque a equação do coestado é o oposto da adjunta da equação de estado, ou seja, a estabilidade da equação de estado implica em instabilidade na equação do coestado. Deste modo a busca de métodos alternativos para contornar esse problema são de extrema importância e de grande interesse para aplicações em tempo real e em Controle Ótimo. Técnicas baseadas em inteligência computacional são uma alternativa para a busca de solução deste tipo de problema, pois, nessas técnicas é possível trabalhar com a não linearidade intrínseca a dinâmica do sistema.

3.2 Princípio Mínimo de Pontryagin

Em vários problemas de controle, especialmente no Controle Ótimo em tempo real, muitas restrições de desigualdade e outras restrições relacionadas diretamente com o controle ou estados do processo são encontradas nos setores da indústria ciência e tecnologia. Essas restrições implicam em modelos e equações mais complexas para a descrição do comportamento e para se realizar o controle, pois, são encontradas dinâmicas irregulares e até mesmo descontínuas.

Determinar uma lei que garanta a solução do problema de controle geral com todas as restrições é um problema que não podia ser solucionado somente com as técnicas de cálculo de variações, deste modo para poder lidar com as restrições de desigualdades, com uma generalização do teorema fundamental do cálculo de variações, essa generalização é conhecida como *Princípio Mínimo de Pontryagin*.

Para definirmos o princípio é necessário apresentar as condições necessárias

para garantir a existência do mínimo do problema de otimização com restrição. O *Hamiltoniano* nos apresenta uma notação mais compacta para as condições necessárias de otimalidade (Kirk 1970), apresentadas pelas equações a seguir.

Inicialmente apresentamos a função escalar conhecida como *Hamiltoniano* do processo:

$$H(x(t), u(t), p(t), t) = g(x(t), u(t), t) + p^T(t)f(x(t), u(t), t) \quad (3.8)$$

$$\dot{x}^*(t) = \frac{\partial H}{\partial p}(x^*(t), u^*(t), p^*(t), t) \quad (3.9)$$

$$\dot{p}^*(t) = \frac{\partial H}{\partial x}(x^*(t), u^*(t), p^*(t), t) \quad (3.10)$$

$$\frac{\partial H}{\partial x}(x^*(t), u^*(t), p^*(t), t) \leq \frac{\partial H}{\partial x}(x^*(t), u(t), p^*(t), t) \quad (3.11)$$

para todos os $u(t)$ admissíveis. O Princípio mínimo de *Pontryagin* exige que a lei de controle ótimo \dot{u}^* deve minimizar o *Hamiltoniano*. Nós podemos também apresentar a lei de controle ótimo explicitamente como:

$$\dot{u}^*(t) = \arg \min_{u(t)} g(x(t), u(t), t) + p^T(t)f(x(t), u(t), t) \quad (3.12)$$

Um dos mais interessantes aspectos do Princípio mínimo de *Pontryagin* é que ele pode ser usado para estabelecer as condições na forma de uma solução ótima, mesmo quando ele não fornece a solução ótima diretamente, de modo que estas condições podem reduzir o problema de encontrar o controle ótimo para um problema mais simples que pode ser solucionado mais simplesmente por métodos numéricos.

3.3 A EAR e o LQR

Apresentamos a seguir o problema de controle de um sistema linear sem a presença de ruído e a importância que a *EAR* possui para a garantia da determinação do ganho do sistema realimentado associado ao problema do *LQR*.

Considere abaixo um sistema linear de estado completamente controlável:

$$\dot{x}(t) = Ax(t) + Bu(t), \quad x(0) = x_0, \quad (3.13)$$

Se todas as variáveis de estado forem mensuráveis, podemos então determinar uma lei de controle de realimentação de estado linear que regula a saída do sistema mantendo-a controlada. Essa lei de controle é expressa por:

$$u(t) = -Kx(t) \quad (3.14)$$

pode ser aplicada ao sistema descrito pela Eq.(3.13). Deste modo a nova matriz dinâmica de estado realimentado é representada como segue,

$$\dot{x}(t) = (A - BK)x(t), \quad x(0) = x_0, \quad (3.15)$$

sendo $K \in \mathbb{R}^{m \times n}$ a matriz de ganho de realimentação de estado.

Para minimizarmos o erro entre os sinais de entrada e a saída com uma realimentação, em geral minimizar um índice de desempenho quadrático associado ao comportamento do erro, pode ser representado por uma estrutura de otimização, como segue:

$$\min_{x,u} J = \int_0^{\infty} [x(t)^T Q x(t) + u(t)^T R u(t)] dt \quad (3.16)$$

sendo o sobrescrito T o operador de transposição, $Q \geq 0$, $Q \in \mathbb{R}^{n \times n}$ é a matriz de ponderação simétrica e definida positivas(ou semidefinida positiva) associada ao vetor de estado x e $R > 0$, $R \in \mathbb{R}^{m \times m}$ é a matriz de ponderação simétrica e definida positiva associada ao vetor de controle $u(t)$.

A matriz ganho ótimo do sistema realimentado pode ser deduzida a partir da expressão:

$$K^* = R^{-1} B^T P \quad (3.17)$$

com $K^* \in \mathbb{R}^{m \times n}$ e P uma matriz simétrica e definida positiva que pode ser determinada através da resolução da equação algébrica de *Riccati* contínua no tempo, apresentada abaixo.

$$A^T P + PA - PBR^{-1}B^T P + Q = 0 \quad (3.18)$$

sendo A e B matrizes do sistema e R uma matriz não singular simétrica e definida positiva, (Wang e Wu 1998).

3.4 Conclusão

Neste Capítulo foram discutidos na Seção 3.1, questões relacionadas à teoria de controle especificamente ao Controle Ótimo. A definição do índice de desempenho quadrático que é um problema de otimização restrito e como torná-lo em um problema irrestrito por meio da introdução de uma variável representando coestado, que funciona como um multiplicador de *Lagrange* no cálculo de funções.

Para este novo problema aplica-se as condições de otimalidade que retornam um sistema de equações diferenciais o que soluciona-se em partes pelo uso de métodos numéricos, mas em muitos casos nem mesmo os métodos como o de *RungeKutta* são aplicáveis, para isso a busca de métodos alternativos, como os relacionados a inteligência artificial e heurísticas são de extrema importância para a solução deste grupo de equações diferenciais associados ao Controle Ótimo.

Na Seção 3.2 foi apresentado o *Princípio Mínimo de Pontryagin* que é uma proposta para a solução do problema geral de controle. Foi feita a definição da função escalar, chamada de *Hamiltoniano* do processo, que é dependente da lei de controle e com a aplicação do *Princípio Mínimo de Pontryagin*, estabelece-se as condições de uma solução ótima para o problema de controle em função da lei de Controle Ótimo.

A ligação da *EAR* com o problema *LQR* é apresentado na Seção 3.3, onde apresentamos a importância da *EAR* no contexto do problema de otimização. Sendo que para a garantia de existência de solução ótima deste problema de controle, foi necessário a determinação do ganho de realimentação de estado, sendo que isso somente é possível por meio do cálculo da solução da *EAR*.

O fundamento das redes neurais para a solução de equações algébricas está alicerçado na teoria clássica de otimização que também é usada no Controle Ótimo. A união das metodologias fornecem suporte para o *LQR* e as redes neurais compõem os fundamentos da teoria de controle inteligente (Reinhold 1992).

Redes Neurais para a Solução da EAR

Nos Capítulos anteriores discutiu-se a equação de *Riccati* sobre o ponto de vista dos métodos clássicos. Inicialmente apresentou-se as estruturas algébricas da *EAR*, sua forma simétrica e as propriedades do seu espaço de solução. Já no Capítulo 3 discutiu-se a gênese da *EAR* por meio do *Regulador Linear Quadrático*. Temos assim dois enfoques da *EAR*: as formas algébricas e a forma no contexto de uma problema, o *LQR*. Agora abordaremos as discussões a respeito da solução, por meio das redes neurais.

A aplicação da forma perturbada da *EAR* apresentada na Seção (2.2) será posteriormente aplicada para a análise do desempenho e robustez das redes neurais.

Apresentamos neste Capítulo dois métodos de solução baseados em Inteligência Computacional. Duas redes neurais, uma rede direta multicamada na Seção 4.1 que usa o gradiente descendente para a minimização do erro que é definido em função da equação de *Riccati* e tendo como condição inicial a solução de *Schur*.

Na Seção 4.2 uma rede recorrente é usada para representar o comportamento de um sistema dinâmico que está associado à conversão de um problema de otimização restrita para um irrestrito, facilitando deste modo a busca pelo mínimo. No processo de conversão, propõe-se a minimização de uma função energia que é uma função de *Lyapunov*, quadrática e convexa. Finalmente na Seção 4.3 uma breve conclusão dos tópicos apresentados neste capítulo.

4.1 Rede Neural Direta

Dos mais diversos métodos usados para o cálculo da *EAR*, apresentaremos uma abordagem usando uma rede neuronal direta, multicamada. Este algoritmo se baseia em uma abordagem de homotopia de algoritmos, que tem como principais características o poder de se utilizar da informação contida na interação anterior, funciona como um algoritmo recursivo, que a partir de uma condição inicial boa é possível uma convergência mais rápida com um menor número de iterações (Ham e Collins 1996).

Alguns trabalhos sobre essas abordagens e trabalhos anteriores, bem como aplicações podem ser encontradas em (Oettli 1995). Na formulação dessa rede neuronal, desejamos minimizar um erro usando o método do gradiente descendente para a diferenciação do erro que é definido em função da equação algébrica de *Riccati*, portanto, ao minimizarmos o erro estaremos minimizando a *EAR* e com isso temos a solução da *EAR*.

4.1.1 Formulação da Rede Direta

Apresentaremos uma metodologia para a formulação da rede neuronal direta, em que desejamos minimizar um erro associado a *EAR*. Definimos o erro em função dos parâmetros da equação algébrica de *Riccati*, de modo que possamos aplicar o gradiente descendente à função do erro definida.

Definimos abaixo, por questões de linearidade e clareza de leitura a Eq.(3.18) a *EAR* em sua forma padrão:

$$A^T P + PA - PSP + Q = 0 \quad (4.1)$$

sendo $A \in \mathbb{R}^{n \times n}$ a matriz dinâmica do sistema, $B \in \mathbb{R}^{n \times n}$ é a matriz associada ao vetor de entradas do sistema, $Q \in \mathbb{R}^{n \times n}$, $Q^T = Q$, $Q \geq 0$ e $R \in \mathbb{R}^{n \times n}$, $R^T = R$ e $R > 0$.

Para facilitar a visualização das demonstrações que seguem, definimos a variável $S \in \mathbb{R}^{n \times n}$, sendo $S = BR^{-1}B^T$, $S^T = S$. Diversas soluções para a *EAR* podem ser encontradas, mas temos interesse em uma solução $P \in \mathbb{R}^{n \times n}$, $P^T = P$ e $P \geq 0$.

Desejamos minimizar o erro instantâneo que é definido como,

$$e = b - y = PRv - A^T Pz - PAz - Qz \quad (4.2)$$

observamos da Figura 4.1, que $v = Pz$, deste modo podemos escrever Eq.(4.2) como segue,

$$e = [PRP - A^T P - PA - Q]z \quad (4.3)$$

A Figura 4.1 apresenta a estrutura neuronal da rede direta.

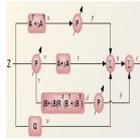


Figura 4.1: Diagrama da Rede Direta

A parte interior aos colchetes representa a equação de *Riccati*, Eq.(4.1) e desde que tenhamos um vetor P que minimize a Eq.(4.1) o erro instantâneo Eq.(4.3) convergirá também para zero, dado um vetor de entrada satisfatório $z(t)$. Podemos definir uma função de custo a ser minimizada do erro usando a Eq.(4.3), a estrutura de otimização é apresentada a seguir:

$$\min_P E(P) = \frac{1}{2} \|[PRP - A^T P - PA - Q]z\|_2^2 \quad (4.4)$$

sendo $\|\cdot\|_2$, representa a norma 2 (ou Euclidiana).

Logo, desejamos minimizar a função de custo do erro Eq.(4.4) fazendo uso do método da descida mais íngreme, (Luenberger 1984), o nosso problema resultante é o de solucionar uma equação diferencial matricial de primeira ordem, ou seja,

$$\frac{dP(t)}{dt} = -\mu \nabla_P E(P) = -\mu \frac{\partial E(P)}{\partial P} \quad (4.5)$$

em que $\mu > 0$, é um parâmetro relacionado a taxa de aprendizagem e velocidade de convergência.

Para calcularmos a Eq.(4.5), devemos calcular o gradiente da função de custo do erro, $\nabla_P E(P)$. Substituindo o erro definido em Eq.(4.3) na Eq.(4.4), temos,

$$\begin{aligned}
 E(P) &= \frac{1}{2} \|e\|_2^2 = \frac{1}{2} e^T e & (4.6) \\
 &= \frac{1}{2} (PRv - A^T Pz - PAz - Qz)^T (PRv - A^T Pz - PAz - Qz) \\
 &= \frac{1}{2} (v^T R^T P^T PRv - v^T R^T P^T A^T Pz - v^T R^T P^T PAz - v^T R^T P^T Qz - \\
 &\quad z^T P^T APRv + z^T P^T AA^T Pz + z^T P^T APAz + z^T P^T AQz - \\
 &\quad z^T A^T P^T PRv + z^T A^T P^T A^T Pz + z^T A^T P^T PAz + z^T A^T P^T Qz - \\
 &\quad z^T Q^T PRv + z^T Q^T A^T Pz + z^T Q^T PAz + z^T Q^T Qz)
 \end{aligned}$$

Desejamos portanto calcular o gradiente da Eq.(4.6). Para isso é necessário usarmos dois resultados do cálculo de matrizes (LEWIS e Syrmos 1995), isto é,

$$\frac{\delta \text{traco}(BAC)}{\delta A} = B^T C^T, \quad \frac{\delta \text{traco}(BA^T C)}{\delta A} = CB \quad (4.7)$$

Usando o resultado acima podemos então calcular o gradiente de cada um dos termos da Eq.(4.6). A demonstração encontra-se no Apêndice E.

Uma vez calculado os gradientes Eq.(E-1) e Eq.(E-14) dos termos da Eq.(4.6) e substituindo temos,

$$\begin{aligned}
 \frac{1}{2} \nabla_P E(P) &= \frac{1}{2} [2(PRvv^T R^T) - (A^T Pzv^T R^T) - (APRvz^T) - \\
 &\quad (PAzv^T R^T) - (PRvz^T A^T) - (Qzv^T R^T) - (APRvz^T) - \\
 &\quad (A^T Pzv^T R^T) + (A^T Pzz^T A^T) + (APAz z^T) + 2(PAzz^T A^T) + \\
 &\quad (Qzz^T A^T) - (Qzv^T R^T) + (AQzz^T) + (Qzz^T A^T)] & (4.8)
 \end{aligned}$$

Agrupando os termos iguais e fazendo alguns cancelamentos pondo em evidência os termos $v^T R^T$ e $z^T A^T$ obtemos,

$$\begin{aligned} \frac{1}{2}\nabla_P E(P) = \frac{1}{2}[(2PRv - 2A^T Pz - 2PAz - 2Qz)v^T R^T - \\ (2PRv - 2A^T Pz - 2PAz - 2Qz)z^T A^T - \\ A(2PRv - 2A^T Pz - 2PAz - 2Qz)z^T] \end{aligned} \quad (4.9)$$

efetuando a multiplicação da constante $\frac{1}{2}$, a expressão Eq.(4.9) temos,

$$\begin{aligned} \frac{1}{2}\nabla_P E(P) = [(PRv - A^T Pz - PAz - Qz)v^T R^T - \\ (PRv - A^T Pz - PAz - Qz)z^T A^T - \\ A(PRv - A^T Pz - PAz - Qz)z^T] \end{aligned} \quad (4.10)$$

como o erro, $e = PRv - A^T Pz - PAz - Qz$, Eq.(4.2), substituindo na Eq.(4.10), teremos

$$\nabla_P E(P) = ev^T R^T - ez^T A^T - Aez^T \quad (4.11)$$

como $R = R^T$, o gradiente do erro será,

$$\nabla_P E(P) = ev^T R - ez^T A^T - Aez^T \quad (4.12)$$

deste modo podemos agora escrever a forma continuá da regra de aprendizagem, Eq.(4.5),

$$\frac{dP(t)}{dt} = \mu \nabla_P E(P) = \mu [Ae(t)z^T(t) + e(t)z^T(t)A^T - e(t)v^T(t)R] \quad (4.13)$$

A regra de aprendizagem Eq.(4.13), não satisfaz às necessidades de simetria para a solução P , ou seja, $P = P^T$. Em (Cochocki e Unbehauen 1993), abordagem de neurocomputação para solucionar a equação algébrica de *Lyapunov*, se a transposta de cada termo do gradiente da Eq.(4.12) for adicionado ao longo dos termos e se escolhermos o valor de $\mu = \frac{\mu}{2}$, a simetria para P é satisfeita. Deste modo a regra de aprendizagem assume a seguinte forma,

$$\begin{aligned} \frac{dP(t)}{dt} = \frac{\mu}{2}\nabla_P E(P) = \frac{\mu}{2}[Ae(t)z^T(t) + z(t)e^T(t)A^T + e(t)z^T(t)A^T + \\ Az(t)e^T(t) - e(t)v^T(t)R - Rv(t)e^T(t)] \end{aligned} \quad (4.14)$$

a regra de aprendizagem Eq.(4.14), não possui divisões e pode ser escrita na forma discreta como,

$$P(k+1) = P(k) + \frac{\mu}{2}[Ae(k)z^T(k) + z(k)e^T(k)A^T + e(k)z^T(k)A^T + Az(k)e^T(k) - e(k)v^T(k)R - Rv(k)e^T(k)] \quad (4.15)$$

sendo k o índice que representa o tempo, o erro é dado em Eq.(4.2), $v(k) = P(k)z(k)$.

4.2 Rede Neural Recorrente

Apresentaremos nesta Seção, a arquitetura e o modelo da rede neuronal recorrente e as estratégias usadas para a conversão do problema inicial de otimização restrita em um problema de otimização irrestrita.

Outra questão importante é a garantia da positividade e simetria da solução da equação algébrica de *Riccati*. Para garantir essa positividade será necessário fazer uma fatoração *LU*, mais especificamente a fatoração de *Cholesky*, (Wang e Wu 1998).

4.2.1 Formulação do problema para a solução da EAR

A seguir apresentaremos a formulação do problema a ser solucionado com a resolução da *EAR*, definida na Eq.(3.18).

Na estrutura matricial da *EAR* existem múltiplas soluções, uma restrição adicional será incluída para obter uma única matriz P simétrica e definida positiva. Utilizaremos uma fatoração *LU*, a fatoração de *Cholesky*, em que qualquer matriz real, simétrica e definida positiva possui uma fatoração de *Cholesky*, deste modo podemos reescrever a solução da *EAR*, como segue.

$$L^T L = P \quad (4.16)$$

sendo $L \in \mathbb{R}^{n \times n}$ uma matriz triangular inferior, possuindo diagonal positiva (Fator de *Cholesky*).

Definindo as matrizes de blocos \tilde{P} e Λ , podemos reescrever a equação de *Riccati*, Eq.(3.18) da seguinte forma

$$\tilde{P}\Lambda\tilde{P}^T = Q + I \quad (4.17)$$

sendo I a matriz identidade,

$$\tilde{P} \triangleq (P, I), \quad \Lambda \triangleq \begin{bmatrix} S & -A \\ -A^T & I \end{bmatrix}.$$

Visto que a matriz Q é simétrica e definida positiva (ou semidefinida positiva), a matriz $Q + I$ também será simétrica e definida positiva, e a matriz A também é simétrica. Portanto, o problema de síntese do LQR pode ser reformulado como sendo a resolução das equações matriciais como segue:

$$\tilde{P}A\tilde{P}^T = Q + I \quad (4.18)$$

$$LL^T = P \quad (4.19)$$

Note que as Eqs. (4.18) e (4.19) tem o mesmo formato.

4.2.2 Estrutura da Rede Neuronal e Equações Dinâmicas

O problema reformulado para a síntese do LQR pode ser considerado como um problema de otimização irrestrita. De modo que dadas as matrizes A , B , Q , e R , queremos achar as matrizes P e L de modo a minimizar uma função energia, como segue abaixo.

$$\min_{P,L} \xi(PSP - A^T P - PA - Q + LL^T - P) \quad (4.20)$$

Note que se $S = BR^{-1}B^T$ e $\xi_{ij} : \mathbb{R} \rightarrow \mathbb{R}$ uma função convexa limitada inferiormente. A expressão interior da função energia verifica-se a soma da EAR com a fatoração de *Cholesky* que é expressa na forma escalar como segue:

$$\left[\sum_{k=1}^n \sum_{l=1}^n p_{ik} s_{kl} p_{lj} - \sum_{k=1}^n (a_{ki} p_{kj} + p_{ik} a_{kj}) - q_{ij} \right] + \left[\sum_{k=1}^{\min\{i,j\}} l_{ik} l_{jk} - p_{ij} \right]. \quad (4.21)$$

As equações dinâmicas da rede neuronal recorrente para solução da EAR são para $i, j = 1, \dots, n$, descritas a seguir:

$$\begin{aligned} \frac{dp_{ij}(t)}{dt} = & -\eta_v \left\{ \sum_{k=1}^n \sum_{l=1}^n [p_{ik}(t)s_{kl}u_{lj}(t) + u_{ik}(t)s_{kl}p_{lj}(t)] \right. \\ & \left. - \sum_{k=1}^n [a_{ik}u_{kj}(t) + u_{ik}(t)a_{jk}] - y_{ij}(t) \right\} \end{aligned} \quad (4.22)$$

$$\frac{dl_{ij}(t)}{dt} = -\eta_z \sum_{k=j}^n y_{ik}(t)l_{kj}(t), \quad i \geq j \quad (4.23)$$

$$\begin{aligned} u_{ij}(t) = & f_{ij} \left[\sum_{k=1}^n \sum_{l=1}^n p_{ik}(t)s_{kl}p_{lj}(t) - \sum_{k=1}^n [a_{ki}p_{kj}(t) + p_{ik}(t)a_{kj}] + q_{ij} \right] \\ y_{ij}(t) = & f_{ij} \left[\sum_{k=1}^{\min\{i,j\}} l_{ik}(t)l_{jk}(t) - p_{ij}(t) \right], \end{aligned} \quad (4.24)$$

Escritas na forma matricial,

$$\frac{dP(t)}{dt} = -\eta_v [P(t)SU(t) + U(t)SP(t) - AU(t) - U(t)A^T - Y(t)] \quad (4.25)$$

$$\frac{dL(t)}{dt} = -\eta_z Y(t)L(t) \quad (4.26)$$

$$U(t) = F[P(t)SP(t) - A^T P(t) - P(t)A + Q] \quad (4.27)$$

$$Y(t) = F[L(t)L(t)^T - P(t)] \quad (4.28)$$

sendo $U(t) = [u_{ij}(t)] \in \mathbb{R}^{n \times n}$, $P(t) = [p_{ij}(t)] \in \mathbb{R}^{n \times n}$ e $Y(t) = [y_{ij}(t)] \in \mathbb{R}^{n \times n}$, são matrizes quadradas dos estados de ativação, $L(t) = [l_{ij}(t)] \in \mathbb{R}^{n \times n}$ uma matriz triangular inferior dos estados de ativação. $P(0) = P(0)^T$, $P(0) \neq 0$, η_v e η_z são constantes de escalonamento positivas, parâmetros do projeto. $F = f_{ij}$ é uma matriz simétrica dos estados de ativação, logo se F é simétrica, $P(t)$, $U(t)$ e $Y(t)$ também serão simétricas. Essas condições de simetria são utilizadas para a redução do número de neurônios usados para a construção da rede.

A Figura 4.2 apresenta um diagrama para ilustrar a arquitetura da *RNR*.



Figura 4.2: Diagrama da Rede Recorrente

A arquitetura da rede neuronal recorrente usada para a resolução da *EAR* é composta de quatro camadas conectadas bidirecionalmente, sendo $P(t)$ a camada de saída, $U(t)$ a camada de entrada e duas camadas ocultas, $Y(t)$ e $L(t)$ que fazem a interconexão entre as camadas de saída $P(t)$ e entrada $U(t)$. As camadas $P(t)$, $U(t)$ e $Y(t)$, consistem de matrizes quadradas de neurônios ($n \times n$) e $L(t)$ é uma matriz triangular. A matriz P representa a solução da *EAR* apresentada na Eq.(3.18) e a matriz $L(t)$ representa o fator de *Cholesky* Eq.(4.16).

Resultados anteriores obtidos em trabalhos de (Wang e Wu 1993) indicam que qualquer função de ativação F não decrescente, pode ser escolhida. Exemplos típicos de funções de ativação: função linear, quadrática e funções trigonométricas. A escolha de uma função não decrescente tem relação direta com a função energia que é uma função convexa, pela relação, $\frac{de_{ij}(\xi)}{d\xi} = f_{ij}(\xi)$.

Os parâmetros (η_z, η_v) , são responsáveis pela velocidade da convergência e também no refinamento em torno da solução, logo que a diferença entre duas iterações subseqüentes seja cada vez menor. Esses parâmetros de projeto devem ser especificados e existem uma séria problemática, entre velocidade de convergência e exatidão na solução.

Algumas heurísticas sobre quais os valores iniciais para a escolha desses parâmetros, se inicialmente se escolheria um passo fixo ou um passo variável, com o intuito de melhorar o desempenho, podemos piorar a proximidade da solução.

Quanto melhor a minha solução, maior o custo computacional do processo de busca e com isso piora o desempenho e conseqüentemente a resposta global do algoritmo, pois é exigido um maior tempo de treinamento da rede.

Independente da heurística utilizada é necessário que os valores dos parâmetros (η_z, η_v) sejam suficientemente grandes para garantir a convergência da rede em regime permanente. Além do mais, devemos garantir a positividade da solução em regime permanente, deste modo a camada oculta L deve convergir mais rapidamente, pois, a mesma fornece informação usada no cálculo da camada de saída P , a fim de garantir a restrição de convergência acima citada devemos escolher os valores de (η_z, η_v) de modo que $\eta_z \geq \eta_v$.

4.3 Conclusão

Neste Capítulo foram apresentados os aspectos relacionados aos métodos usados para o cálculo da *EAR* por duas redes neuronais. Na Seção 4.1.1 foram discutidas as estruturas necessárias para a formulação e desenvolvimento matemático da rede neuronal direta necessários para a sua compreensão, assim como o desenvolvimento da lei de aprendizagem para a implementação da mesma.

Na Seção 4.2.1 foi apresentada a formulação da rede neuronal recorrente juntamente com a estrutura do problema de otimização com restrições, assim como a conversão do problema em uma estrutura de otimização irrestrita e convexa, em que a rede recorrente executara a busca da solução da *EAR* e do fator de *Cholesky*. Na Seção 4.2.2 foram apresentadas a estrutura da rede recorrente e as equações dinâmicas, representando o comportamento do sistema dinâmico associado ao problema de otimização que tem por objetivo solucionar a *EAR*.

Estabilidade e Solvabilidade

Apresentaremos neste Capítulo o comportamento numérico das redes neuronais dedicadas a solução da *EAR* para o *LQR*, com o propósito de auxiliar na análise de estabilidade e solvabilidade das redes neuronais apresentadas.

As matrizes usadas nos testes de desempenho podem ser encontradas no Apêndice C, contendo as matrizes do sistema, matrizes de entrada, matriz Q e R usadas nos experimentos computacionais relativos as redes neuronais recorrente e direta.

Dois sistemas multivariáveis de 6^a ordem foram usados, um representando a dinâmica de uma aeronave e um outro sistema representando um gerador de indução eólico duplamente alimentado(*DFIG*).

Para o cálculo da Equação Algébrica de *Riccati* necessita-se das matrizes de ponderação Q e R , sendo as mesmas fornecidas por um algoritmo genético responsável pela a realização da busca e seleção das matrizes de ponderação que melhor alocam a autoestrutura definida pelo projetista, (Ferreira e Fonseca Neto 2003) e (Ferreira e Torrico 2003).

Para a rede neuronal recorrente temos alguns problemas relacionados ao ajuste dos parâmetros η_z e η_v , Eq.(4.22) e Eq.(4.23), sendo estes responsáveis pela velocidade de convergência da rede neuronal, estabilidade e qualidade da solução. Portanto a escolha desses parâmetros é de extrema importância para a sintonia da rede neuronal recorrente.

Na Seção 5.1 é feita a determinação dos parâmetros η_v e η_z , através da avaliação do comportamento da rede neuronal para uma faixa de variação dos dois parâmetros. Assim um dos nossos experimentos é verificar o quanto esses

parâmetros influenciam no comportamento de nossa solução.

Apresentaremos uma variação desses parâmetros em relação a norma infinito da solução da *EAR* e a energia. A Seção 5.1 subdivide-se na Seção 5.1.1 referente a uma breve discussão sobre normas e funções de ativação usadas para se realizar as simulações e computações necessárias para determinação dos parâmetros η_z e η_v da *RNR*, na Seção 5.1.2 discute sobre a influência da ordem do sistema na rede recorrente e a Seção 5.1.3 referente aos experimentos da planta *DFIG*. Análise da planta da aeronave podem ser encontradas no Apêndice *F*.

Outro importante experimento computacional é a relação de robustez dos algoritmos em frente a perturbações paramétricas, vistos na Seção 5.2, onde apresentamos diversos aspectos e experimentos computacionais com perturbações nas plantas, ou seja, temos interesse de saber o quanto que estes dois métodos podem seguir a solução perante variações externas as dinâmicas dos sistemas, assim como nas entradas, a partir de uma condição de operação estacionária anterior dos sistemas.

Uma das vantagens de se utilizar as técnicas das redes neurais é que elas podem aproveitar de uma solução anterior para calcular a próxima solução e deste modo convergem mais rapidamente que métodos tradicionais, que necessitariam calcular novamente a solução.

O método de *Schur* que é apresentado no Apêndice *D* ou mesmo outros métodos como o dos *Autovetores* são exemplos de métodos, onde a cada perturbação é necessário recalcular a solução, calcular a inversa de matrizes, ordenação e transformações das matrizes associadas a busca da solução, o que aumentaria o custo computacional a cada nova perturbação.

5.1 Determinação dos Parâmetros η_v e η_z

Nas próximas seções discuti-se o comportamento da *RNR* com o auxílio de superfícies com variações dos parâmetros η_v e η_z , com o propósito de se ter uma visão geral da influência dos parâmetros η_v e η_z , Eq.(4.22) e Eq.(4.23), no treinamento, convergência, estabilidade e desempenho da rede recorrente, para cada uma das plantas dos sistemas apresentados no Apêndice *C*.

Inicialmente apresentamos na Seção 5.1.1, com as normas usadas e função

de ativação usada pela *RNR*, experimentos relacionados a planta da aeronave e outras plantas de sistemas de 3ª e 5ª, que se encontram no Apêndice *C*. Na Seção 5.1.2 a influência da ordem do sistema no cálculo da *EAR* e na Seção 5.1.3 mostramos os experimentos e a determinação dos parâmetros η_v e η_z para o gerador eólico *DFIG*.

5.1.1 Função de Ativação e Normas

Para a determinação dos parâmetros de convergência da rede neuronal para a resolução do problema de otimização associado a *EAR* e fatoração de *Cholesky*, iremos inicialmente variar os parâmetros η_z e η_v de modo a verificar a influência desses parâmetros em relação a busca do mínimo da função energia e conseqüentemente da norma infinito do *mismatch* da *EAR*.

Para o uso da rede neuronal recorrente apresentada na Seção (4.2), é necessário especificar as condições iniciais usadas para a realização dos experimentos computacionais, possibilitando a posterior reprodução dos resultados e para que seja possível uma análise mais concreta sobre o comportamento da metodologia.

No Apêndice *C* encontram-se as matrizes da planta da aeronave usada nas simulações. A função energia usada, foi o erro médio quadrático (*EMQ*). A função de ativação usada pelos elementos de processamento, neurônios, foi a função de saturação linear, Fig. 5.1, apresentada a seguir,

$$f(x) = \begin{cases} -1, & \text{se } x \leq -1 \\ x, & \text{se } -1 \leq x \leq 1 \\ 1, & \text{se } x \geq 1 \end{cases} \quad (5.1)$$

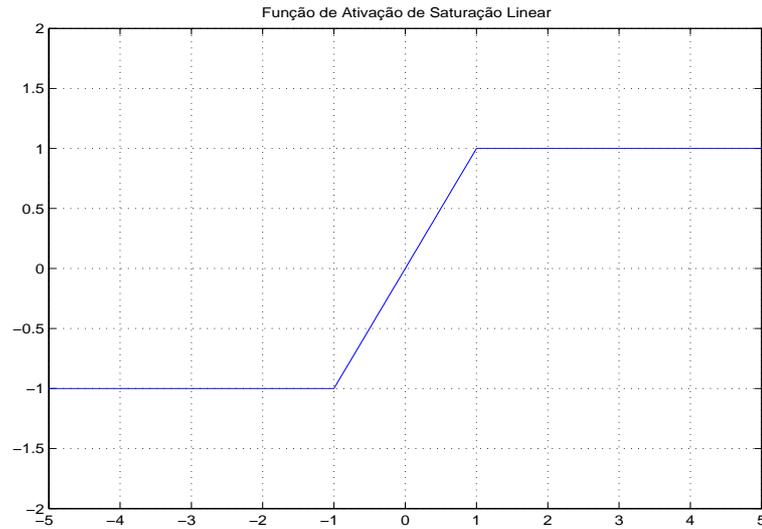


Figura 5.1: Função de Ativação usada pelos neurônios

Para ser feito o uso da metodologia apresentada para a rede neuronal recorrente, algumas etapas são necessárias, tais como a inicialização das camadas de entrada, ocultas, saída e posteriormente a integração de uma camada oculta, relacionada com a dinâmica da fatoração de *Cholesky*, Eq.(4.26) e da solução da *EAR*, Eq.(4.25).

Para tal integração usaremos o método de *Runge – Kutta* de quarta ordem de modo a termos como retorno as matrizes que possibilitarão o cálculo da camada de saída representado pela solução P da *EAR*, Eq.(3.18) e pelo fator L de *Cholesky*, Eq.(4.16). O método de integração de *Runge – Kutta* usado, pode ser encontrado no Apêndice A.

Para o sistema de 6^a ordem representado pela aeronave foi usada a seguinte configuração; número de amostras $N_{sample} = 1600$ com tempo de processamento $T_p = 0.0003$ segundos, sendo estes parâmetros usados pelo método de *Runge – Kutta* de 4^a ordem.

Como a nossa solução é uma matriz, devemos especificar uma métrica, pois, deste modo poderemos comparar as diversas soluções que o algoritmo nos retorna a cada iteração. Uma das métricas usadas foi a norma do máximo, mas outras métricas como a norma infinito e a norma de *Frobenius*, também foram utilizadas, mas em outros experimentos de modo a verificar as métricas usadas se comportavam para este nosso problema especificamente.

A definição matematicamente das normas é apresentada a seguir:

$$\|A\|_{\infty} = \max_{i=1,\dots,n} \sum_{j=1}^n |a_{ij}|; \quad (5.2)$$

$$\|A\|_{max} = \max_{j=1,\dots,n} \sum_{i=1}^n |a_{ij}|; \quad (5.3)$$

$$\|A\|_{fro} = \left(\sum_{i,j=1}^n |a_{ij}|^2 \right)^{\frac{1}{2}} \quad (5.4)$$

sendo a_{ij} , os elementos da matriz A .

Inicialmente duas normas foram usadas para a planta da aeronave a norma do máximo, já para a planta representando o *DFIG*, foi usada a norma infinito, embora alguns testes tenham sido feitos na aeronave com as duas normas para se verificar o comportamento de cada uma das normas em um mesmo experimento.

Para melhor visualizarmos a norma infinito apresentamos em mais detalhes o comportamento da solução da *EAR*, para cada uma das normas, como segue nas Figuras 5.2 e 5.3.

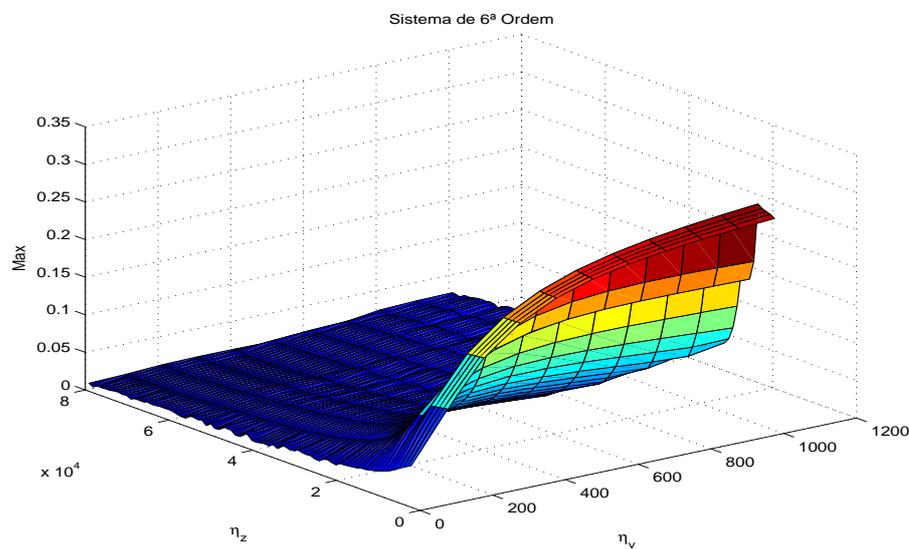


Figura 5.2: $norma_{max}$ da solução P para um sistema de 6ª ordem representando a planta da aeronave.

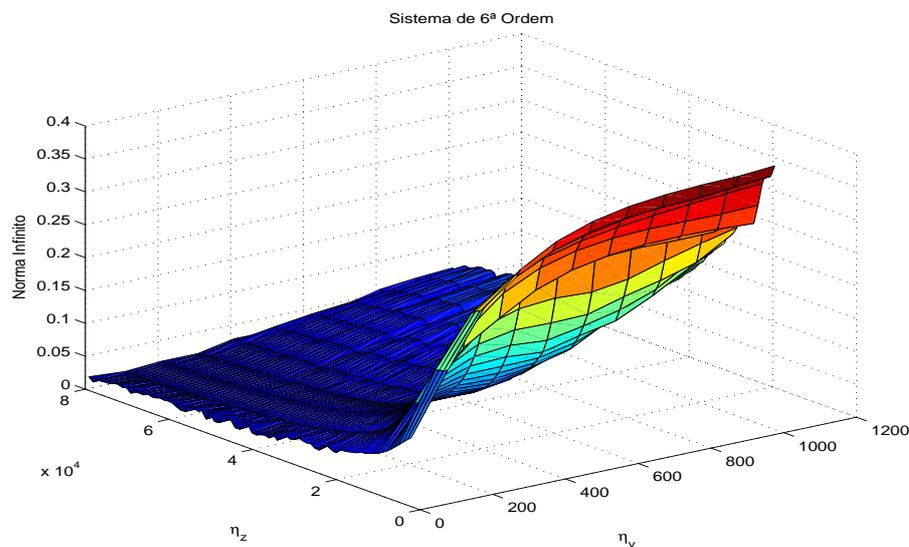


Figura 5.3: $norma_{\infty}$ da solução P para um sistema de 6^a ordem representando a planta da aeronave

A norma do máximo apresenta uma maior uniformidade na reprodução das superfícies, ao passo que a norma do infinito nos mostra mais detalhes, pois, fica mais claro a observação das oscilações. Não existe uma diferença substancial entre as duas normas apresentadas, de acordo com o problema o projetista pode escolher-se qualquer umas das duas.

Apresentamos na Figura 5.4 a solução da EAR para um sistema teste de 3^a ordem, as matrizes relacionadas ao sistema podem ser encontradas no Apêndice C, para as matrizes Q e R foram usadas as matrizes identidade. Foi escolhido um sistema de 3^a ordem, pois, a rede retorna uma boa solução, não tivemos maiores problemas de estabilidade, desempenho e a rede recorrente mostrou não ter tanta dependência em relação aos parâmetros η_v e η_z .

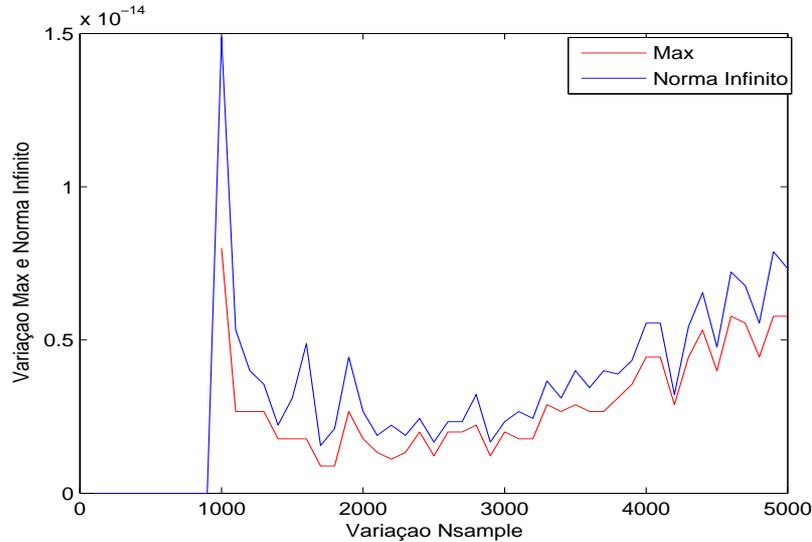


Figura 5.4: $norma_{\max}$ e $norma_{\infty}$ da solução P para as variações do N_{Sample} para um sistema de 3ª ordem.

Visualizando o comportamento na Figura 5.4, verificamos que não existe grande influência do N_{Sample} em relação às normas do infinito ou máximo, ainda podemos verificar que os valores que as duas normas retornam possuem um comportamento semelhante, sendo que a norma do máximo apresenta menos variações em relação à norma infinito.

5.1.2 Variação da Ordem do Sistema

Testes em outros sistemas foram realizados, mas a rede neuronal não teve problemas para solucionar a EAR . Apresentamos na Figura 5.5 o comportamento da solução da EAR para variações dos parâmetros de aprendizagem da rede para um sistema de 5ª ordem.

Outros sistemas como de 3ª e 4ª, também foram testados, mas não foi verificada uma grande influência na solução retornada pela rede neuronal, apenas a influência para a velocidade de convergência, mas essa velocidade também não é percebida com a observação, somente após análise de variáveis que armazenavam o tempo global de execução do algoritmo.

À medida que a ordem do sistema aumenta, maior o número de variáveis a serem observadas e com isso, maior se torna a complexidade do sistema a ser estudado. Comportamentos irregulares, mais oscilatórios e instáveis se tornam mais comuns. Somente o sistema de 5ª ordem que apresentou um comportamento contendo informação relativa ao comportamento mais oscilatório da solução da *EAR* com relação a influência dos parâmetros em relação a ordem do sistema, como segue na Figura 5.5.

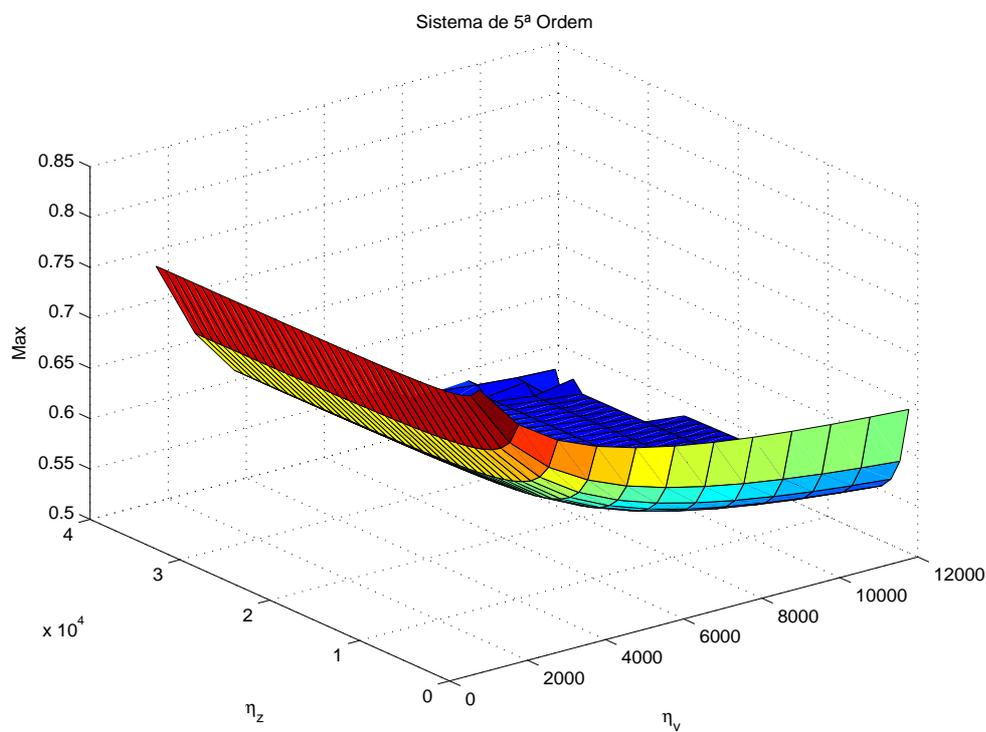


Figura 5.5: n_{\max} da solução V para a variação η_v : 1-11000 e η_z :1000-40000, para um sistema de 5ª ordem

Com o aumento do número de variáveis e com o aumento da ordem do sistema, maior a complexidade e mais difícil se torna a análise tanto do comportamento do algoritmo quanto para a compreensão da dinâmica do sistema, que estão direta e indiretamente associados às variáveis responsáveis pelo ajuste e sintonia relacionados ao treinamento da rede neuronal.

5.1.3 Experimentos Planta DFIG

Apresentamos aqui os resultados associados à planta do *DFIG*, cujas matrizes encontram-se no Apêndice *C*. As matrizes iniciais usadas e os valores das variáveis usadas para este experimento foram, $\Delta T = 9.0 \cdot 10^{-4}$, para o tempo de processamento e $N_{sample} = 200$. As condições iniciais para as camadas foram as matrizes identidades, $L = P = I_{6 \times 6}$. As matrizes Q e R usadas foram (C-8) e (C-9) contidas no Apêndice *C*.

Na Figura 5.6, apresentamos o comportamento global da energia total do sistema para as variações dos parâmetros η_v e η_z , de tal modo que possamos constatar a influência dos parâmetros em relação a qualidade de solução, estabilidade e desempenho da rede neuronal recorrente.

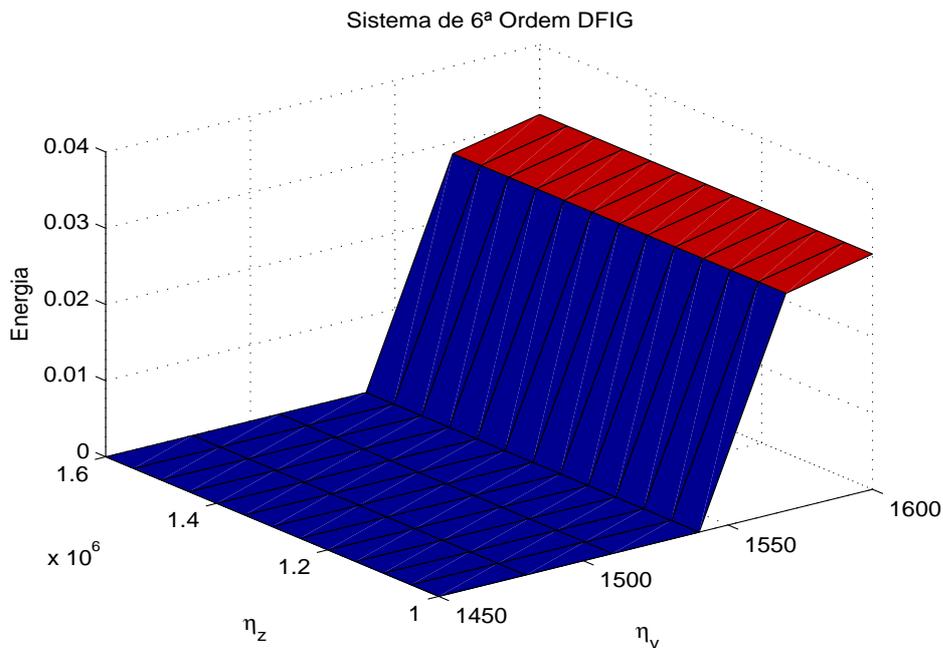


Figura 5.6: Superfície da Energia para variação dos Parâmetros - $\eta_v = 1450-1540$; $\eta_z = 1000000-1600000$.

Outro fator que devemos frisar é que a faixa de variação dos parâmetros apresentadas nesta Figura 5.6, na realidade é uma faixa considerada boa, pois, anteri-

ormente foi realizada uma maior variação dos parâmetros e a região apresentada representa os melhores resultados.

Na Figura 5.7, segue a norma do infinito para variações dos parâmetros η_v e η_z . Para esse sistema não foi necessário uma boa condição inicial para alcançarmos uma solução satisfatória. Bastando somente as matrizes identidade como condições iniciais, podemos verificar o quanto os parâmetros influenciam diretamente na solução da *EAR*.

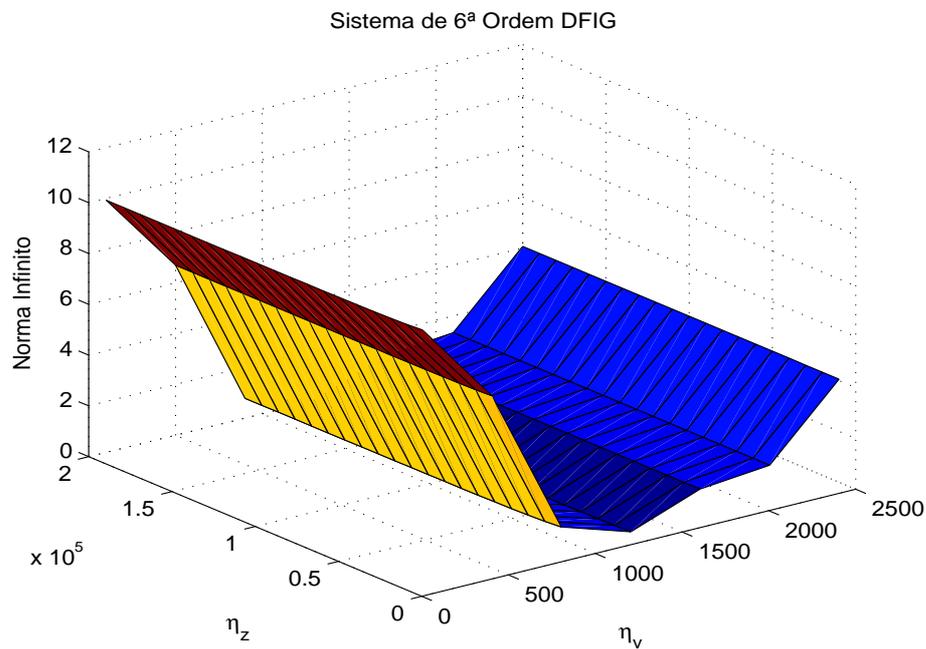


Figura 5.7: $norma_{\infty}$ das soluções da *EAR* para variação parâmetros - $\eta_v = 1000-1500$; $\eta_z = 10000-200000$.

Escolhendo uma faixa de variação menor para os valores de η_v de modo a refinar e melhorar o resultado retornado pela rede, aumentamos a faixa de variação para η_z , pois, para tentarmos verificar se existe uma influência maior, para grandes valores de η_z em relação ao desempenho do treinamento em relação a convergência e estabilidade da rede.

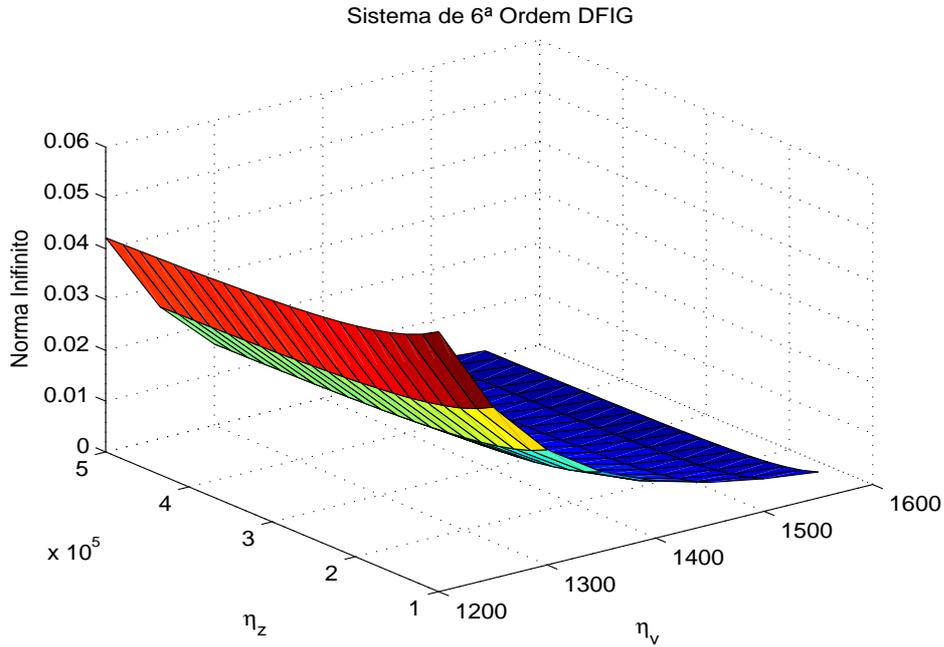


Figura 5.8: Norma Infinito da ARE para variação parâmetros - $\eta_v = 1200-1550$; $\eta_z = 100000-500000$.

Outro importante experimento é o comportamento da rede neuronal em relação ao número de iterações. O treinamento da rede neuronal pode ser feito por meio de um critério de parada do algoritmo ou se estipulando um número de iterações fixo, ou até mesmo de modo híbrido, com vários passos.

Nesses experimentos foi determinado um número de iterações e à medida que o treinamento ocorre, o valor da energia era mostrada na tela do computador assim como a norma do infinito da solução da *EAR*, pois, deste modo pode-se verificar os pontos usados para a construção dos gráficos e então verificar a precisão numérica do algoritmo. Uma outra variável armazena todos os valores de energia e solução da *EAR* e variações dos parâmetros η_v e η_z para que se possa olhar pontualmente quais os valores de η_v e η_z que retornam uma melhor solução ao longo da superfície.

Para as Figuras 5.9 e 5.10, foi escolhido um intervalo conveniente que resultou em uma boa solução, $\eta_v = 10000$, $\eta_z = 100000$, $N_{Sample} = 100$ e $\Delta T = 10^{-5}$ e as

matrizes Q e R matrizes identidades.

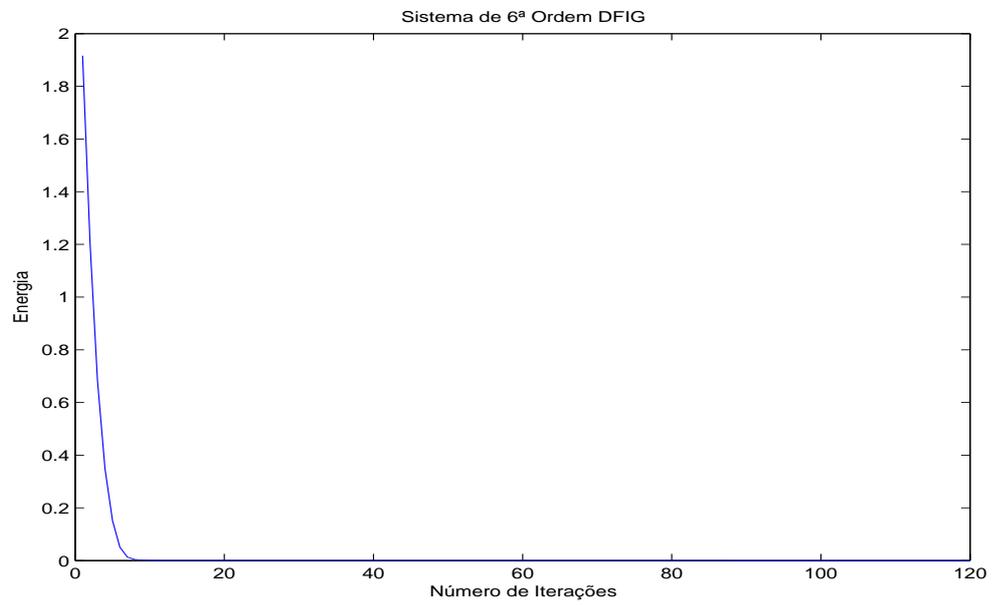


Figura 5.9: Energia em função do número de iterações.

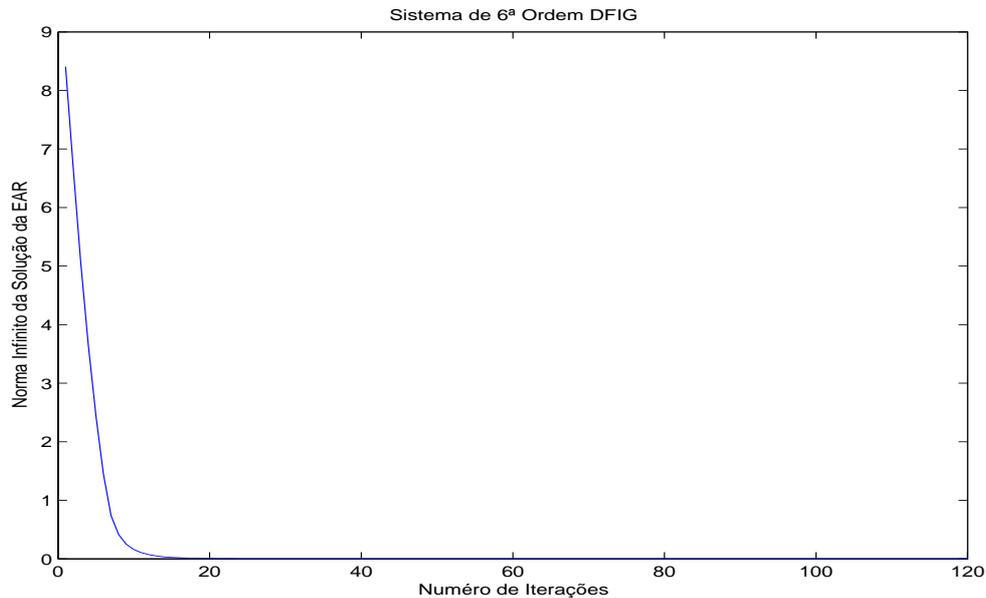


Figura 5.10: Norma Infinito da solução da *EAR* em função do número de iterações.

Tanto na Figura 5.9 que representa a energia, quanto na Figura 5.10 que representa o impacto do treinamento no comportamento da solução da *EAR*, a *RNR* converge em torno de 40 iterações. O treinamento retorna uma solução com precisão de 6 casas decimais, a partir de 70 iterações a parcela de decremento se torna bastante pequena, não sendo significativa.

5.2 Perturbações Paramétricas no Sistema(DFIG)

Nesta seção apresentamos os resultados relacionados às variações da planta *DFIG*, em relação aos métodos apresentados e os comparamos com o método de *Schur*, que se encontra no Apêndice *D*.

Uma das condições que temos interesse de verificar é o comportamento dos métodos apresentados em relação a uma condição de operação anterior estável. Tomamos como condição inicial para estes experimentos a solução de *Schur* e então perturbamos um dos elementos da matriz *A* da planta, para verificar o comportamento da rede recorrente e da rede direta em relação a essas perturbações

e comparamos todos os resultados.

Em outros experimentos foram variadas as matrizes relacionadas à dinâmica, à entrada e as duas ao mesmo tempo. Os percentuais de variação para os mesmos foram de 1%, 5%, 10% e 15%.

Apresentamos a solução da *EAR* pelo método de *Schur* e então perturbamos essa solução e aplicamos a rede direta e a rede recorrente. Para a rede recorrente usamos a seguinte configuração: $\eta_v = 1540$, $\eta_z = 1600000$, $N_{Sample} = 100$ e $\Delta T = 10^{-5}$. A norma de *Frobenius* foi aplicada à solução P da *EAR* e à própria *EAR* em si, como mostramos a seguir os resultados.

O elemento $i_{4,4}$ da matriz do sistema *DFIG*, Eq.(C-6), foi modificado de -24.5578 para -23.3299 , uma redução de 5%. A matriz P_{Schur} representa a solução perturbada de *Schur*.

$$P_{Schur} = \begin{bmatrix} 0.032606 & 0.0035979 & 0.01818 & 0.0037885 & 0.0057903 & 0.044966 \\ 0.0035979 & 0.039141 & 0.018282 & -0.0031616 & 0.02099 & 0.077495 \\ 0.01818 & 0.018282 & 0.078654 & 0.093415 & 0.010466 & 0.096516 \\ 0.0037885 & -0.0031616 & 0.093415 & 0.29229 & 0.0010405 & -0.20485 \\ 0.0057903 & 0.02099 & 0.010466 & 0.0010405 & 0.33153 & 0.011495 \\ 0.044966 & 0.077495 & 0.096516 & -0.20485 & 0.011495 & 1.6978 \end{bmatrix} \quad (5.5)$$

A norma de *Frobenius* da solução de *Schur* foi $N_{Schur} = 7.7345 \times 10^{-14}$

P_{RND} sendo a matriz representando a solução da rede direta e P_{RNR} para a rede recorrente.

$$P_{RND} = \begin{bmatrix} 0.032486 & 0.0036279 & 0.01792 & 0.0024806 & 0.0057843 & 0.045209 \\ 0.0036279 & 0.039134 & 0.018336 & -0.0028678 & 0.020989 & 0.077436 \\ 0.01792 & 0.018336 & 0.078741 & 0.091525 & 0.010472 & 0.094686 \\ 0.0024806 & -0.0028678 & 0.091525 & 0.27861 & 0.00099597 & -0.20515 \\ 0.0057843 & 0.020989 & 0.010472 & 0.00099597 & 0.33154 & 0.011289 \\ 0.045209 & 0.077436 & 0.094686 & -0.20515 & 0.011289 & 1.6931 \end{bmatrix} \quad (5.6)$$

A norma de *Frobenius* da solução de *RND* foi $N_{RND} = 0.78553$

$$P_{RNR} = \begin{bmatrix} 0.032605 & 0.003598 & 0.018181 & 0.0037884 & 0.0057903 & 0.044966 \\ 0.003598 & 0.03914 & 0.018283 & -0.0031617 & 0.02099 & 0.077495 \\ 0.018181 & 0.018283 & 0.078652 & 0.093415 & 0.010466 & 0.096516 \\ 0.0037884 & -0.0031617 & 0.093415 & 0.29229 & 0.0010405 & -0.20485 \\ 0.0057903 & 0.02099 & 0.010466 & 0.0010405 & 0.33153 & 0.011495 \\ 0.044966 & 0.077495 & 0.096516 & -0.20485 & 0.011495 & 1.6978 \end{bmatrix} \quad (5.7)$$

A norma de *Frobenius* da solução de *RNR* foi $N_{RNR} = 9.4476 \times 10^{-14}$

Na tabela 5.1 apresentamos as perturbações que serão inseridas de modo a verificar o comportamento das metodologias citadas nos capítulos anteriores em relação ao cálculo da solução da *EAR*, um amplo espectro de variação foi submetido neste caso para verificar o comportamento dos algoritmos e se ocorre a divergência em algum caso.

Tabela 5.1: Tabela com as variações feitas as matrizes A , B e a ambas simultaneamente

Caso	<i>Normas</i>	
	$\ \Delta A\ $	$\ \Delta B\ $
1	0	0
2	0.99%	0
3	0.95%	0
4	0.9%	0
5	0.85%	0
6	0	0.99%
7	0	0.95%
8	0	0.9%
9	0	0.85%
10	0.99%	0.99%
11	0.95%	0.95%
12	0.9%	0.9%
13	0.85%	0.85%

Tabela 5.2: Norma de *Frobenius* para a solução da *EAR* para variações nos parâmetros da planta

Caso	Norma de <i>Frobenius</i>		
	$\ \Delta P_{schur}\ $	$\ P_{RND}\ $	$\ P_{RNR}\ $
1	4.6861×10^{-13}	0.00066032	9.3255×10^{-14}
2	4.2381×10^{-13}	0.25034	9.5007×10^{-14}
3	4.7789×10^{-13}	1.2514	9.9263×10^{-14}
4	3.8887×10^{-13}	2.5025	1.0578×10^{-13}
5	5.6341×10^{-13}	3.7531	1.1584×10^{-13}
6	1.9545×10^{-13}	0.021819	9.3018×10^{-14}
7	2.5744×10^{-13}	0.10805	9.2247×10^{-14}
8	3.5081×10^{-13}	0.21085	9.0953×10^{-14}
9	4.5302×10^{-13}	0.30808	9.0675×10^{-14}
10	5.5341×10^{-13}	0.26709	9.4898×10^{-14}
11	3.093×10^{-13}	1.3334	9.7771×10^{-14}
12	2.461×10^{-13}	2.6619	1.0453×10^{-13}
13	2.9764×10^{-13}	3.9856	1.1263×10^{-13}

O valor inicial usado para o erro da rede direta foi $e = 1$ e o valor da variável $\mu = 0.000001$. Pelas Figuras 5.11–5.13, vemos o comportamento do erro à medida que o treinamento da rede é iniciado até que um intervalo de convergência seja alcançado. O critério de parada para o treinamento da rede direta foi que ocorresse quanto o erro estivesse na ordem de 10^{-10} .

Por meio das Figuras que serão no decorrer desta Seção apresentados. Verifica-se que existe uma maior dependência da planta do sistema, pois, o mesmo influencia diretamente na qualidade da solução retornada pelo algoritmo, ao passo que a matriz de entrada do sistema não dificulta tanto o processo de convergência da *RND*, o que para este caso significa uma menor dependência sobre a matriz *B* de entrada no sistema.

O fato da matriz *B* de entrada do sistema não influenciar tanto quanto a matriz *A*, não implica que a mesma não exerça influência sobre a dinâmica global do comportamento da *RND*, simplesmente não ocorreu neste caso, entretanto,

outros testes podem ser realizados para verificar se existe ou não maior influência da matriz no comportamento global, sendo que para outros sistemas o comportamento pode ser outro.

As perturbações das duas matrizes foram feitas com o intuito de verificar se existia correlação entre as matrizes da planta e de entrada no processo de treinamento da rede, para este sistema em específico.

Como podemos perceber na Figura 5.11, à medida que a matriz perturbada se afasta da solução inicial, mais demorado se torna o treinamento e conseqüentemente a convergência do algoritmo. Mesmo assim a *RND* conseguiu convergir para zero, ocorrendo também na Figura 5.13, que apresenta as perturbações das duas matrizes combinadas sendo que a maior influência é relativa à matriz *A* relacionada a planta do sistema.

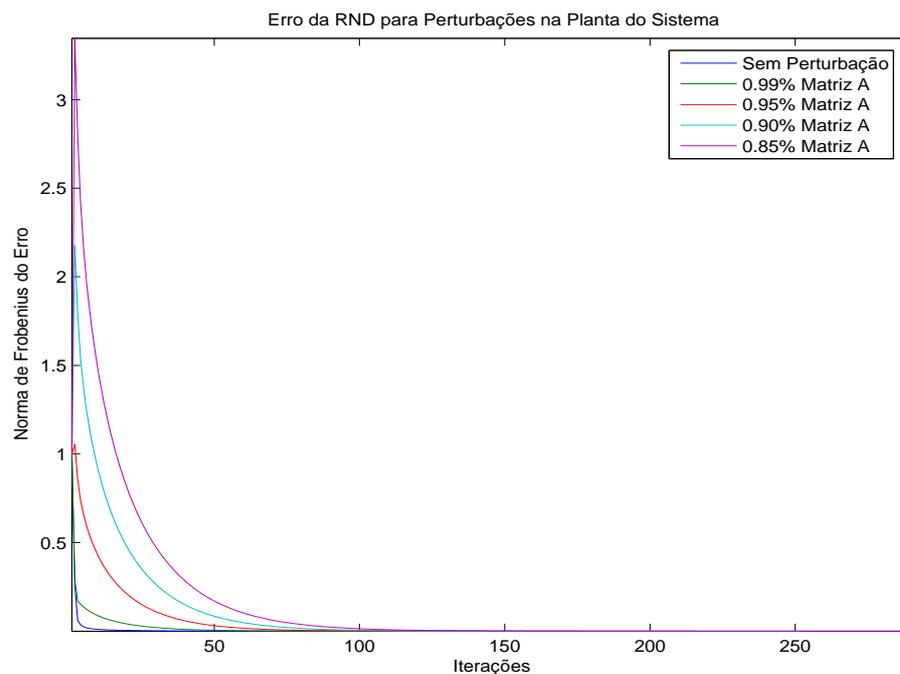
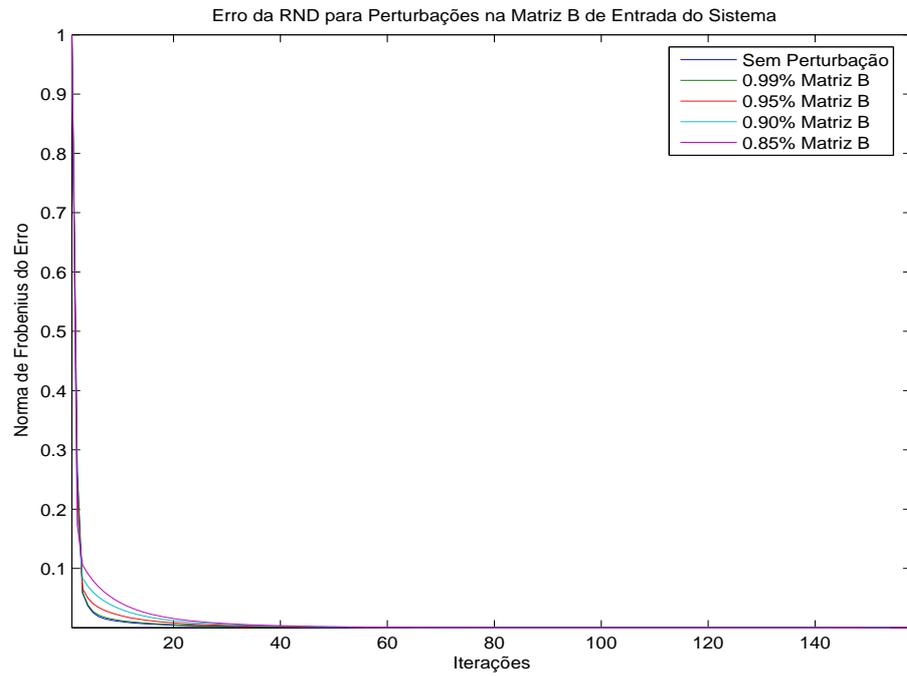
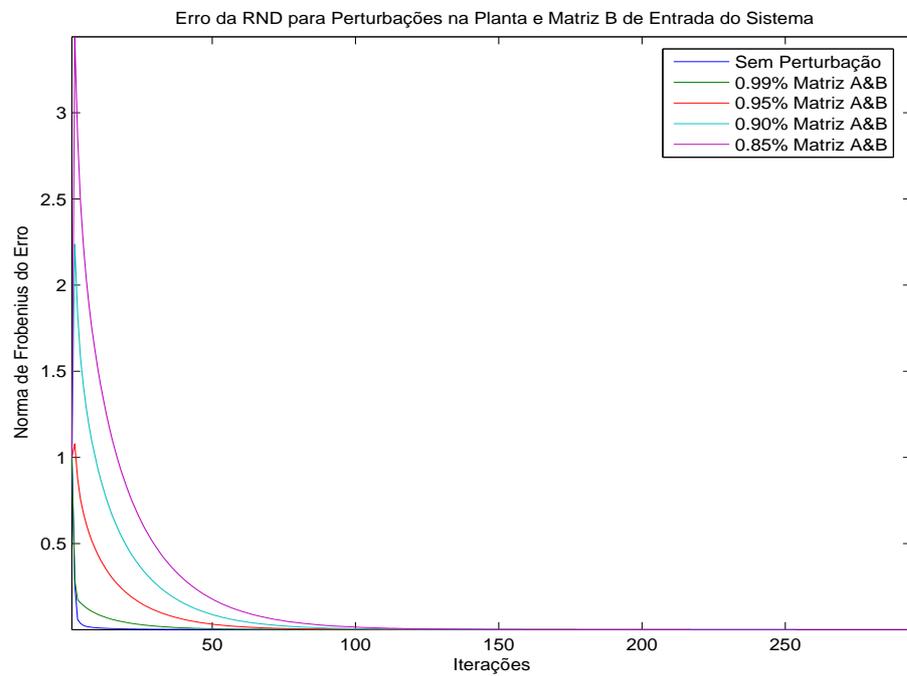


Figura 5.11: Erro da Rede Direta para a Planta Perturbada.

Figura 5.12: Erro Rede Direta para a Matriz B de Entrada Perturbada.Figura 5.13: Erro Rede Direta para a Matriz B e Planta A Perturbada.

A rede recorrente retornou melhores resultados se comparados com a rede direta e retornou uma solução semelhante ao método de *Schur*, tendo variações somente nas últimas casas decimais de modo que não representa diferenças substanciais associados a precisão numérica.

A rede direta retorna uma maior erro com o aumento das perturbações e conseqüentemente um maior erro na solução da *EAR*. Uma vantagem que ocorre com o uso da rede recorrente é que a mesma converge mais rápido que o método de *Schur*, pois, o mesmo tem que calcular a inversa novamente ao passo que a rede recorrente com o processo de aprendizagem necessita de um número de iterações menor para a convergência.

As Figuras 5.14-5.16, representam o comportamento da *RNR* ao longo do processo de treinamento. As variações na planta acarretaram em um atraso na convergência, de modo que quanto mais afastado da condição inicial do problema maior será o tempo para o treinamento.

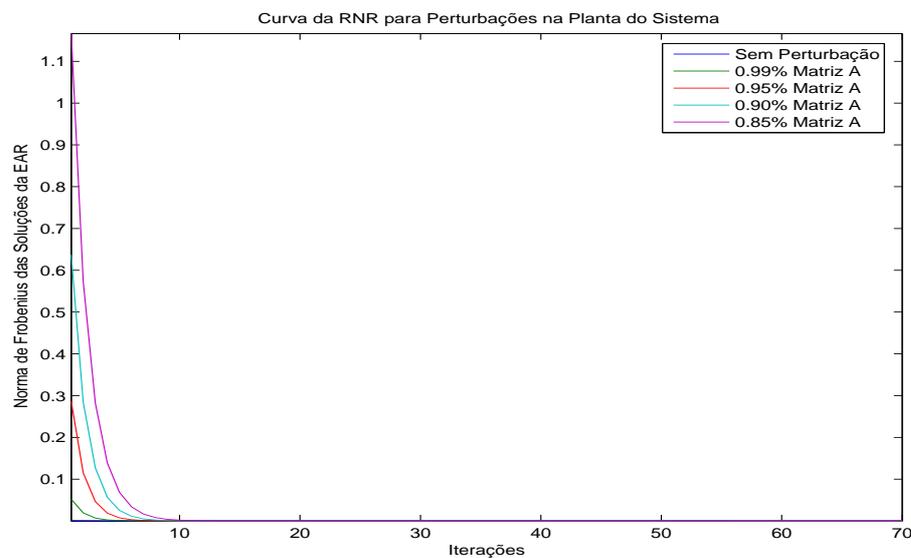


Figura 5.14: Norma de *Frobenius* da Solução da *RNR* para Perturbações na Planta.

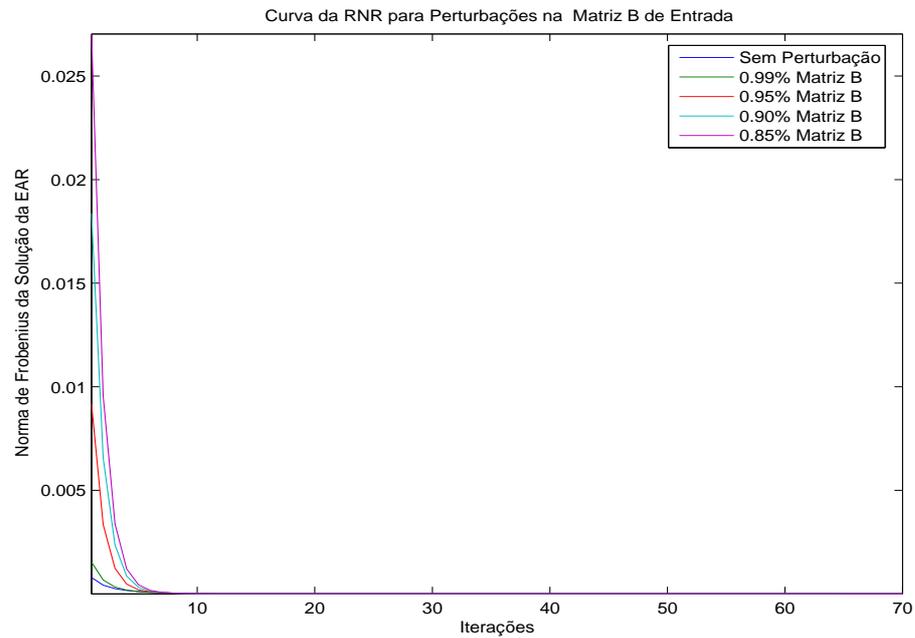


Figura 5.15: Norma de *Frobenius* da Solução da *RNR* para Perturbações na Matriz *B* de Entrada do Sistema.

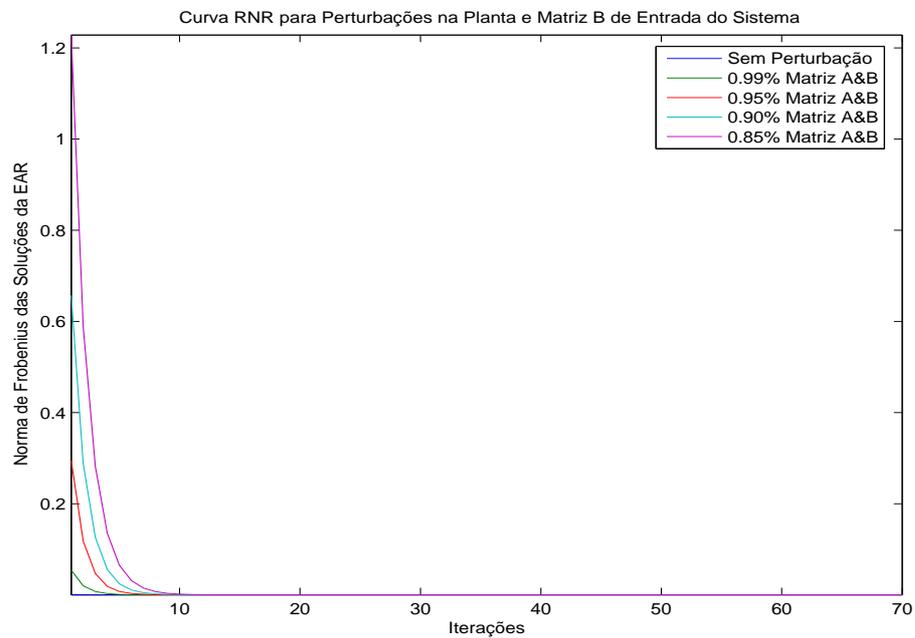


Figura 5.16: Norma de *Frobenius* da Solução da *RNR* para Perturbações na Planta e Matriz *B* de Entrada do Sistema.

As Figuras 5.17-5.19, mostram o comportamento da energia global da rede neuronal recorrente associada a *EAR* e ao fator de *Cholesky*, para este caso escolhemos os valores para η_v e η_z que retornavam um bom resultado e mostrar que a rede a partir de um ponto de operação anteriormente estável, pode convergir mais rapidamente.

As perturbações exercidas com maior influência são novamente das matrizes associadas a planta do sistema, ao passo que a matriz B de entrada no sistema para este caso não apresentou nenhuma modificação significativa com relação a convergência da *RNR* como pode ser analisada pela Figura 5.18.

Essa maior dependência da matriz A do sistema é visivelmente verificada observando-se a própria estrutura da *EAR*, Eq.(3.18), onde podemos verificar que dos cinco termos da equação, dois deles estão associado à matriz A e somente um deles está associado à matriz B .

Para a rede recorrente temos que analisar, não somente a estrutura da *EAR*, como também a estrutura das equações que representam a dinâmica do comportamento da *RNR*, Eqs.(4.25-4.28), sendo as camadas de entrada e saída Eq.(4.25) e Eq.(4.27), respectivamente as equações que sofrem maior influência da matriz A , por estarem associadas à *EAR* e as Eq.(4.26) e Eq.(4.28), associadas ao fator de *Cholesky*.

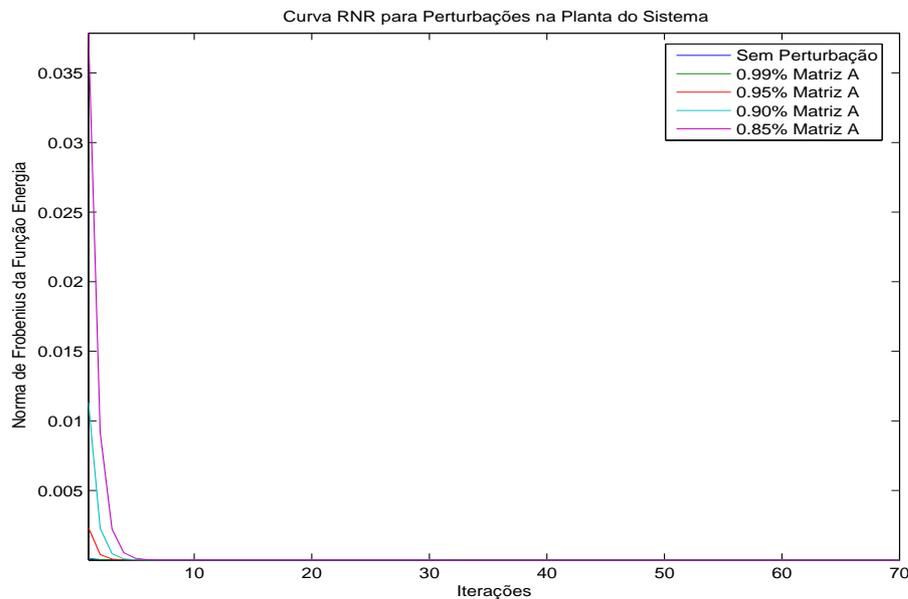
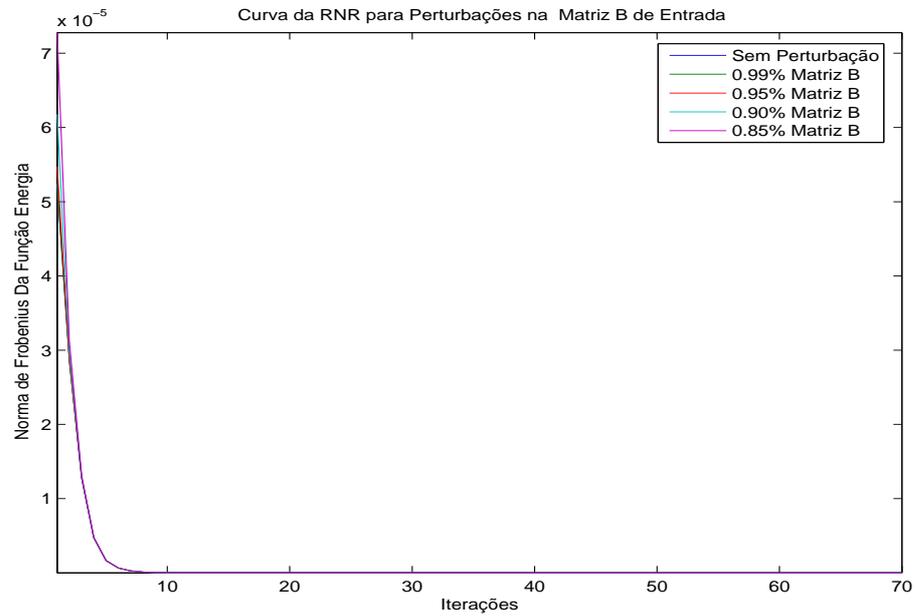
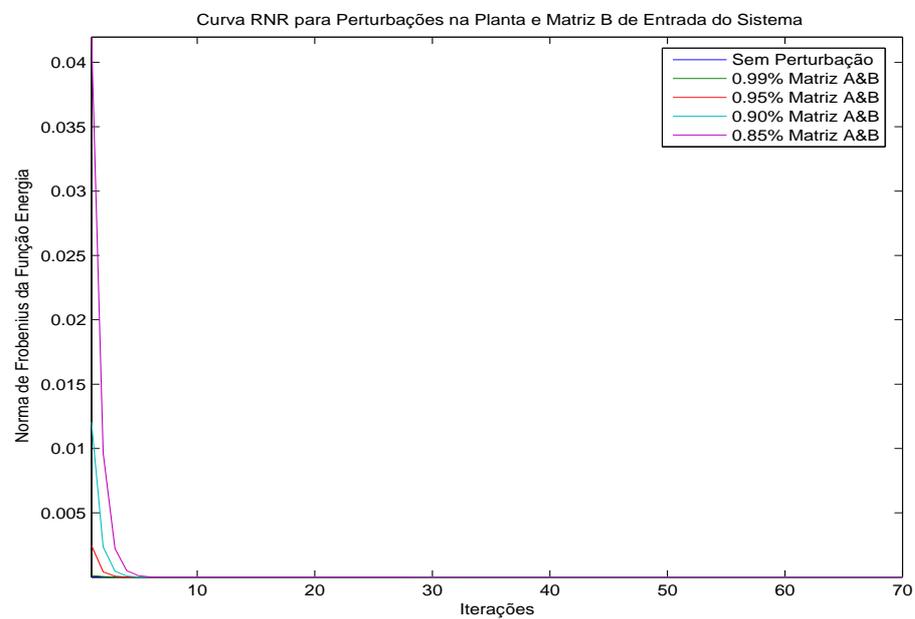


Figura 5.17: Energia para Perturbações na Planta do Sistema.

Figura 5.18: Energia para Perturbações na Matriz B de Entrada.Figura 5.19: Energia para Perturbações na Planta e Matriz B de Entrada.

Para a visualização da evolução do algoritmo da RNR tendo como valor de

referência a solução de *Schur*, para as perturbações nas matrizes *A* do sistema e *B* de entrada e ambas simultaneamente são mostradas pelas Figuras 5.20-5.22.

O comportamento da *RNR* tendo como referência a solução de *Schur*, retornou um boa solução convergindo rapidamente diferentemente da *RND* que apresentou-se mais sensível às perturbações, nestes experimentos.

Cada caso dever ser analisado, simulado e estudado separadamente, pois, as metodologias apresentadas, possuem suas vantagens e desvantagens, não sendo as mesmas solução para todas as dinâmicas, pois, cada sistema possui características intrínsecas próprias e não devem ser de modo algum descartadas do processo de modelagem, amostragem e linearização do sistema.

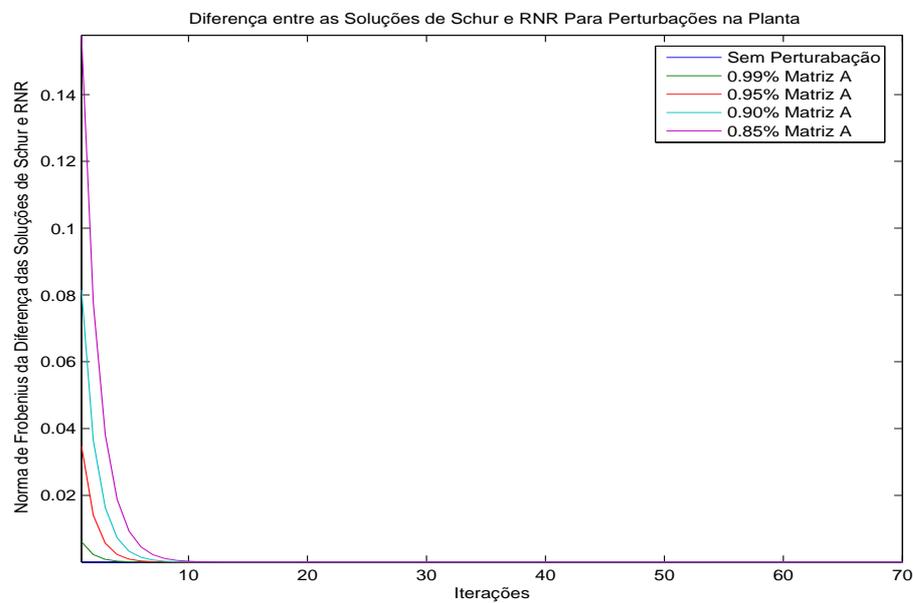


Figura 5.20: Norma de *Frobenius* da Diferença Entre as Soluções de *Schur* e *RNR* para Perturbações na Planta do Sistema.

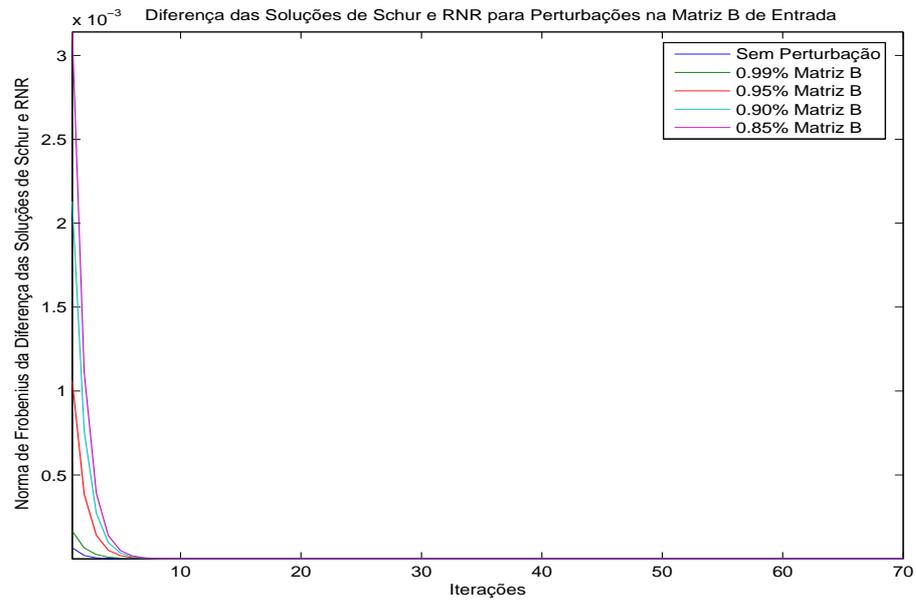


Figura 5.21: Norma de *Frobenius* da Diferença Entre as Soluções de *Schur* e *RNR* para Perturbações na Matriz *B* de Entrada.

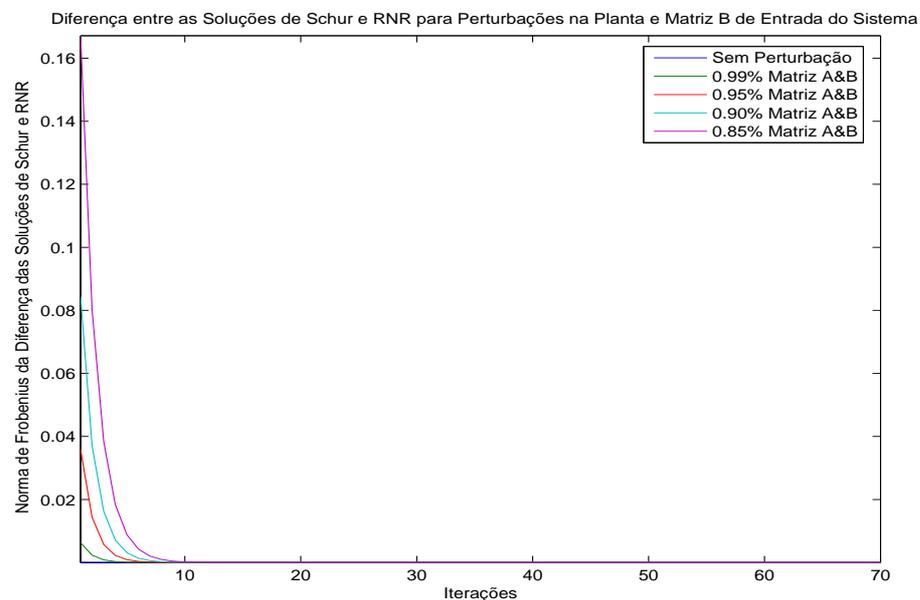


Figura 5.22: Norma de *Frobenius* da Diferença Entre as Soluções de *Schur* e *RNR* para Perturbações na Planta e Matriz *B* de Entrada.

5.3 Conclusão

Neste Capítulo foram discutidas questões sobre o desempenho, estabilidade, convergência e sintonia das metodologias apresentadas nas Seções 4.1 e 4.2.

No Apêndice *C* apresentam-se as matrizes das plantas dos sistemas utilizados nas simulações computacionais. A Seção 5.1, foi discutida uma metodologia para a determinação dos parâmetros η_v e η_z da *RNR*, que é de extrema importância estando relacionada diretamente à eficiência do treinamento da rede neuronal.

A Seção 5.1 subdivide-se na Seção 5.1.1, referente a discussão sobre as métricas que seriam aplicadas a solução e a *EAR*, assim como a função de ativação usada pela rede recorrente.

Na Seção 5.1.3 análises referentes aos experimentos do sistema *DFIG*. Um recurso usado para a determinação dos parâmetros foi a construção de uma superfície de modo a facilitar a visualização da influência dos parâmetros na computação do cálculo da *EAR* da *RNR* e ter uma visão mais global de quais os valores de η_v e η_z reproduziriam uma melhor solução para a *EAR*.

Uma das principais questões discutidas no Capítulo tinha como objetivo avaliar o comportamento das redes neuronais de modo a verificar o comportamento das mesmas em relação ao desempenho, convergência e qualidade da convergência das metodologias apresentadas no Capítulo 4, ou seja, a estabilidade, pois levando em consideração as restrições associadas a *EAR*, assim como questões puramente relacionada a configuração usadas para se computar as soluções, gerar as superfícies e critérios de parada do treinamento, estão fundamentados na estabilidade e solvabilidade da *RNR*.

A robustez frente a perturbações discutidas na Seção 5.2, apresentaram a dinâmica das redes neuronais em relação aos parâmetros da perturbados da *EAR*. Como resultado das perturbações podemos verificar que para o sistema *DFIG* usado, a *RND* consegue solucionar para pequenos índices de perturbações, mas quando as perturbações são acrescidas a convergência piora. Já a *RNR*, conseguiu contornar esse problema de modo que a mesma atingiu a mesma solução de *Schur* e com uma boa sintonia e escolha dos seus parâmetros, se apresenta como uma alternativa bastante viável.

CAPÍTULO 6

Conclusão

Este trabalho apresentou metodologias baseadas em computação neuronal para a resolução da equação algébrica de *Riccati*. Inicialmente apresentamos a estrutura algébrica da equação de *Riccati*, lemas, teoremas e corolários que nos apresentassem a forma do conjunto solução da *EAR* simétrica e real, estrutura esta de interesse, sendo a mesma a forma da *EAR* que foi solucionada pelas redes neurais direta e recorrente apresentadas.

Duas redes neuronais foram apresentadas para a resolução da *EAR*, uma rede neuronal direta e uma recorrente e as mesmas foram comparadas com o método de *Schur*. A estrutura da rede direta associa a *EAR* com o erro a ser minimizado pela *RND* a partir da aplicação do gradiente descendente.

Os resultados da *RND* segundo um ponto anterior de estabilidade, obteve resultados razoavelmente satisfatórios para pequenas variações, mas para maiores perturbações ou condições iniciais distantes de uma condição inicial boa a *RND* não converge para uma boa solução. Devemos também levar em consideração o sistema propriamente dito, pois, talvez nesta planta em específico a resposta não tenha sido tão satisfatória, sendo que testes com outras plantas nos apresentariam uma melhor visão da dinâmica do processo de treinamento da rede neuronal direta.

A rede neuronal recorrente usada para solucionar o problema de otimização restrita, por meio de outra estrutura de otimização convexa, em que é definido de uma função convexa que possui a função da *EAR* e do fator de *Cholesky* em sua formulação obteve um comportamento mais satisfatório frente à *RND* até mesmo em relação ao método de *Schur*.

A *RNR* apresentada obteve bons resultados inclusive para grandes perturbações nas plantas dos sistemas e para outras perturbações, e uma vez que a *RNR* esteja treinada, mesmo com variações, a rede pode convergir prontamente solucionando a *EAR* assim como o método de *Schur*.

O problema de ajuste dos parâmetros de aprendizagem e velocidade de convergência η_v e η_z , pôde ser em partes contornado, pela geração de superfícies que auxiliam no processo de escolha dos parâmetros η_v e η_z . Deste modo para cada um dos sistemas testados, uma vez escolhido os parâmetros η_v e η_z , temos então um intervalo conveniente em torno de uma região que retorna uma boa solução e a *RNR* converge para uma solução estável.

6.1 Propostas Futuras

Alguns comentários e perguntas interessantes surgiram ao longo do desenvolvimento deste trabalho, idéias provenientes dos estudos realizados e perguntas sobre que outras alternativas poderiam ser exploradas para a resolução da *EAR*, quais mudanças poderiam ser incluídas às metodologias apresentadas de maneira a otimizá-las, ou até mesmo diferentes técnicas fora da inteligência computacional.

Uma questão interessante seria sobre a possibilidade de criar formulações para solucionar o problema de atualização das matrizes de covariâncias do filtro de *Kalman*, que está associada à solução da equação algébrica de *Riccati* discreta. Uma linha de trabalho seria verificar a possibilidade de construir modelos baseados nessas metodologias para solucionar o problema de filtragem ótima.

Outra questão interessante seria acoplar outras técnicas inteligentes para fazer a escolha dos parâmetros da *RNR*, como por exemplo o desenvolvimento de um algoritmo genético para a escolha dos parâmetros de aprendizagem da *RNR*, ou mesmo um sistema neuro-fuzzy trabalhando em paralelo, para a escolha dos parâmetros de convergência e para o auxílio e supervisão das restrições associadas aos problemas.

A utilização de um modelo de otimização multiobjetivos tendo um algoritmo genético ou uma lógica fuzzy trabalhando em paralelo com a rede neuronal para solucionar a *EAR* e executar a sintonia automática da rede neuronal recorrente.

Referências Bibliográficas

- Ammar, Gregory e Volker Mehrmann (1991). On hamiltonian and symplectic hessenberg forms. *Linear Algebra and its Applications* **149**(15), 55–72.
- Athans, Michael e L.Peter Falb (1966). *OPTIMAL CONTROL- An Introduction to the Theory and Its Applications*. McGRAW-Hill Book Company . United States of America.
- Byers, Ralph (1985). Numerical condition of the algebraic equation. *Contemporary Mathematics* **47**, 35–49.
- Byers, Ralph (1987). Solving the algebraic riccati equation with the matrix sign function. *Linear Algebra and its Applications* **85**, 267–279.
- Caines, Peter E. (1988). *Linear Stochastic Systems*. Probability and Mathematical Statistics. Wiley Series.
- Carro, F. Incertis e J. M. Martinez Torres (1977). An extension on a reformulation of the algebraic equation problem. *IEEE Transaction on Automatic Control* **22**(1), 128–129.
- Charles Kenney, Alan J. Laub e Matt Wette (1989). A stability enhancing scaling procedure for schur-riccati solvers. *System and Control Letters* **12**(3), 241–250.
- Cloutier, James R. (1997). State-dependent riccati equation techniques: An overview. *Proceedings of the 1997 American Control Conference* **2**, 932–936.
- Cochocki, A. e Rolf Unbehauen (1993). *Neural Networks for Optimization and Signal Processing*. John Wiley and Sons, Inc.. New York, NY, USA.

- Faibusovich, L. E. (1986). Algebraic riccati and symplectic algebra. *International Journal of Control* **43**(3), 781–792.
- Fábio Nogueira, Ivanildo Abreu, João V. da Fonseca Allan Kardec B. e Orlando Fonseca (2006). Solução da equação algébrica de riccati via rede neural recorrente para o projeto do regulador linear quadrático. *XVI Congresso Brasileiro de Automática*.
- Ferreira, Carlos C. T. e J. V. Fonseca Neto (2003). Projeto do Filtro de Kalman via Alocação de Autoestrutura e Algoritmo Genético. *VI Simpósio Brasileiro de Automação Inteligente*.
- Ferreira, Carlos C. T., J. V. Fonseca Neto e F. A. Torrico (2003). Alocação de autoestrutura via controle lqg/ltr e computação evolutiva. *VI Simpósio brasileiro de Automação Inteligente*.
- Gardiner, Judith D. e Alan J. Laub (1986). A generalization of the matrix-sign-function solution for algebraic riccati equations. *International Journal of Control* **44**(3), 823–832.
- Gardiner, Judith D. e Alan Laub (1991). Parallel algorithm for algebraic riccati equations. *International Journal of Control* **54**(6), 1317–1333.
- Gülsu, Mustafa e Mehmet Sezer (2006). On the solution of the riccati equation by the taylor matrix method. *Applied Mathematics and Computation* **176**(2), 414–421.
- Golub, Gene H. e James H. Wilkinson (1976). Ill-conditioned eigensystems and the computation of the jordan canonical form. *SIAM Review* **18**(4), 578–619.
- Goodwin, Graham C. e Wai Sang Sin (1984). *Adaptive Filtering: Prediction and Control*. Information and System Science Series. Prentice Hall.
- Ham, Fredic M. e Emmanuel G. Collins (1996). A neurocomputing approach for solving the algebraic riccati equation. *IEEE International Conference on Neural Networks* **1**(3-6), 617–622.
- Haykin, Simon (1999). *Neural Networks: A Comprehensive Foundation*. Prentice Hall.

- J. J. Hench, C. He. Kučera e V. Mehrmann (1998). Dampening controllers via a riccati equation approach. *IEEE Transaction on Automatic Control* **43**(9), 1280–1284.
- Jain, L. C. e L. R. Medsker (1999). *Recurrent Neural Networks: Design and Applications*.
- Jonckheere, Edmond (1981). On the existence of a negative semidefinite antistabilizing solution to the discrete-time algebraic equation. *IEEE Transactions on Automatic Control* **AC-26**(3), 707–712.
- Kirk, Donald E. (1970). *Optimal Control Theory - An Introduction*. Prentice-Hall.
- Kleiman, David L. (1968). On an iterative technique for riccati equation computations. *IEEE Transactions on Automatic Control* pp. 114–115.
- Kosko, Bart (1992). *Neural Networks and Fuzzy Systems: A Dynamical Systems Approach to Machine Intelligence*. Prentice Hall.
- Kuriyama, Kazunobu e Tsutomu Mita (1997). Alternative derivation of the algebraic riccati equation in h_∞ control. *Automatica* **33**(1), 91–95.
- Kwon, W. H. e A. E Pearson (1977). A note on the algebraic riccati equation. *IEEE Transaction on Automatic Control* **22**(1), 143–144.
- Lancaster, Peter e Leiba Rodman (1995). *Algebraic Riccati Equations*. Clarendon Press Oxford.
- Laub, Alan J. e Pascal Gahinet (1997). Numerical improvements for solving riccati equation. *IEEE Transaction on Automatic Control* **42**(9), 1303 – 1308.
- Leden, B. (1976). Dead-beat control and the riccati equation. *IEEE Transaction on Automatic Control* **21**(5), 791–792.
- Levin, J. J. (1959). On the matrix riccati equation. *Proceedings of the American Mathematical Society* **10**(4), 519–524.
- Lewis, F. (1981). Riccati equation computation of supremal (f,g)-invariant and reachability subspaces. *IEEE Transactions on Automatic Control* **AC-26**(3), 725–728.

- LEWIS, Frank L. e Vassilis L. Syrmos (1995). *Optimal Control*. John Wiley and Sons, Inc.
- Loan, C. Van (1984). A symplectic method for approximating all the eigenvalues of a hamiltonian matrix. *Linear Algebra and its Applications* **61**, 233–251.
- Luenberger, D.G. (1984). *Linear and Nonlinear Programming*. 2nd ed ed.. Addison-Wesley Publ.Co.
- Martensson, K. (1972). Approaches to the numerical solution of the optimal control problems. Master's thesis. Division of Automatic Control, Lund Institute Technology. Report 7206. Sweden.
- Michel, Anthony N. e Derong Liu (2002). *Quantitative Analysis and Synthesis of Recurrent Neural Networks*. Marcel Dekker.
- Moore, John B. (1969). Application of the riccati equations in system engineering. *Electrical Engineering Transactions* p. 6.
- Nils R. Sandell, Jr. (1974). On newton's method for riccati equation solution. *IEEE Transactions on Automatic Control* **19**(3), 254–255.
- Oettli, Michael (1995). The Homotopy Method Applied to the Symmetric Eigenproblem. Phd thesis. Swiss Federal Institute of Technology Zurich.
- P. Lancaster, A.C.M. Ran e L. Rodman (1986). Hermitian solutions of the discrete algebraic riccati equation. *International Journal of Control* **44**(3), 777–802.
- Peter Benner, Ralph Byers (1998). Newtons method with exact line search for solving the algebraic riccati equation. *Scientific Parallel Computing*.
- Pinto, Vandilberto Pereira (2007). Modelagem e Simulação de uma Planta Eólica Controlada por um Regulador Linear Quadrático Conectada ao Sistema de Distribuição de Energia Elétrica. PhD thesis. Universidade Federal do Ceará.
- Potter, James E. (1966). Matrix quadratic solutions. *SIAM Journal on Applied Mathematics* **14**(3), 496–501.
- Reinhold, Van Nostrand (1992). Handbook of intelligent control: Neural, fuzzy, and adaptive approaches.

- Rodrigues-Canabal, J. (1973). The geometry riccati equation. *Stochastics* **1**, 129–149.
- Sergio Bittanti, Alan J. Laub e Willems, Jan C., Eds.) (1991). *The Riccati Equation*. Springer Communications And Control Engineering Series. Springer-Verlag New York, Inc.
- Staffans, Olof J. (1996). On the discrete and continuous time infinite dimensional algebraic riccati equations. *System and Control Letters* **29**(3), 131–138.
- Svensson, J. (1998). Grid Connected Voltage Source Converters Control Principles and Wind Energy Applications. Tese de doutorado. Chalmers University of Technology. Department of Electric Power Engineering.
- Trapp, George (1985). The riccati equation and the geometric mean. *Contemporary Mathematics* **47**, 437–445.
- V, Sima. (1981). An efficient schur method to solve the stabilizing problem. *IEEE Transactions on Automatic Control* **AC-26**(3), 724–725.
- Vaughan, David R. (1970). A nonrecursive algebraic solution for the discrete riccati equation. *IEEE Transaction on Automatic Control* **15**(5), 597–599.
- Wang, Jun e Guang Wu (1993). Recurrent neural networks for lu decomposition and cholesky factorization. *Mathematical and Computer Modeling* **18**(1993), 1–8.
- Wang, Jun e Guang Wu (1998). A multilayer recurrent neural network for solving continuous-time algebraic riccati equation. *Neural network* **11**(1998), 939–950.
- Zhang, Xiang-Sun (2000). *Neural Networks in Optimization*. Kluwer Academic.

APÊNDICE A

Método de Runge Kutta

Neste apêndice apresentaremos o método de integração numérica de *Runge Kutta* de 4ª ordem.

O método consiste no uso da expansão da série de *Taylor* para aproximar a solução dos valores anteriores e das derivadas das funções anteriores, escalonados pela constante h representando o tamanho do passo a ser dado no processo de integração.

O método de *Runge Kutta* pode ser usado para solucionar equações diferenciais ordinárias da seguinte forma,

$$\frac{dy}{dx} = f(x, y), y(0) = y_0 \quad (\text{A-1})$$

O modo de atualização usado pelo processo de integração é feito como se apresenta a seguir.

$$y_{i+1} = y_i + (a_1k_1 + a_2k_2 + a_3k_3 + a_4k_4) h \quad (\text{A-2})$$

sabendo que o valor de $y = y_i$ em x_i , podemos encontrar o valor de $y = y_{i+1}$.

A Eq.(A - 2) é balanceada pelos primeiros cinco termos da série de *Taylor*, deste modo temos,

$$\begin{aligned}
y_{i+1} = y_i + \frac{dy}{dx}|_{x_i, y_i}(x_{i+1} - x_i) + & \quad (A-3) \\
\frac{1}{2!} \frac{d^2y}{dx^2}|_{x_i, y_i}(x_{i+1} - x_i)^2 + & \\
\frac{1}{3!} \frac{d^3y}{dx^3}|_{x_i, y_i}(x_{i+1} - x_i)^3 & \\
+ \frac{1}{4!} \frac{d^4y}{dx^4}|_{x_i, y_i}(x_{i+1} - x_i)^4 &
\end{aligned}$$

Sabendo que

$$\frac{dy}{dx} = f(x, y) \quad (A-4)$$

e $x_{i+1} - x_i = h$, então

$$\begin{aligned}
y_{i+1} = y_i + f(x_i, y_i) h + & \quad (A-5) \\
\frac{1}{2!} f'(x_i, y_i) h^2 + & \\
\frac{1}{3!} f''(x_i, y_i) h^3 + & \\
\frac{1}{4!} f'''(x_i, y_i) h^4 &
\end{aligned}$$

Baseado na Eq.(A-3) e Eq.(A-5), uma das soluções usadas é:

$$y_{i+1} = y_i + \frac{1}{6} (k_1 + 2k_2 + 2k_3 + k_4) h \quad (A-6)$$

$$k_1 = f(x_i, y_i) \quad (A-7)$$

$$k_2 = f\left(x_i + \frac{1}{2}h, y_i + \frac{1}{2}k_1h\right) \quad (A-8)$$

$$k_3 = f\left(x_i + \frac{1}{2}h, y_i + \frac{1}{2}k_2h\right) \quad (A-9)$$

$$k_4 = f(x_i + h, y_i + k_3h) \quad (A-10)$$

Resultados Analíticos da Rede Neural

Este Apêndice complementa a abordagem neuronal recorrente apresentada no Capítulo 4 na Seção 4.2.

Apresentamos neste apêndice, resultados relacionados a análise de estabilidade a partir da aplicação da teoria de sistemas dinâmicos, mais especificamente, a aplicação do método direto de *Lyapunov* para a determinação da estabilidade de um sistema dinâmico da rede neuronal recorrente usada para solucionar a *EAR*. Estes resultados podem ser encontrados em (Wang e Wu 1998).

Teorema B.0.1. *Se a função de ativação for contínua e pelo menos diferenciável por partes, monótona e não-decrescente, isto é, $\frac{df_{ij}(\xi)}{d\xi} \geq 0$, para $-\infty < \xi < +\infty$, e $f_{ij}(\xi) = 0$ se $\xi = 0$, ($i, j = 1, \dots, n$), então a rede neural é assintoticamente estável ou seja, $\forall P(0), L(0), \exists \bar{L}, \bar{P}$ tal que $\lim_{t \rightarrow \infty} L(t) = \bar{L}$ e $\lim_{t \rightarrow \infty} P(t) = \bar{P}$.*

Prova: Seja $f_{ij}(\xi) \triangleq \frac{de_{ij}(\xi)}{d\xi}$. Desde que f_{ij} seja contínua e pelo menos diferenciável e $\frac{df_{ij}(\xi)}{d\xi} \geq 0$, $e_{ij}(\xi) > -\infty$, ou seja, e_{ij} é limitada inferiormente e $\frac{d^2e_{ij}(\xi)}{d\xi^2} \geq 0$, (conseqüentemente e_{ij} é convexa) para $-\infty < \xi < +\infty$ e $i, j = 1, \dots, n$. Uma vez que ξ é definida como a soma de e_{ij} e e_{ij} é convexa e limitada inferiormente, ξ também é convexa e limitada inferiormente. Além disso, ξ é convexa e limitada inferiormente, $\xi \rightarrow \infty$ quando $\|P\| \rightarrow \infty$ ou $\|L\| \rightarrow \infty$, ou seja, ξ é ilimitada radialmente.

De acordo com as definições da *RNR*, Eq.(4.25-4.27), $s_{ij} = s_{ji}$, $y_{ij}(t) = y_{ji}$,

simetria das matrizes R e F , para $i, j = 1, \dots, n$ e supondo-se que

$$g_{ij}(P) = \sum_{k=1}^n \sum_{l=1}^n p_{ik} s_{kl} p_{lj} - \sum_{k=1}^n (a_{ki} p_{kj} + p_{ik} a_{kj}) - q_{ij}$$

pode-se escrever,

$$\sum_{k=1}^n \sum_{l=1}^n u_{kl}(t) \frac{\partial g_{kl}}{\partial p_{ij}} = \sum_{k=1}^n \left\{ \sum_{l=1}^n [p_{ij}(t) s_{kl} u_{lj}(t) + u_{ik}(t) s_{kl} p_{lj}] - \right. \quad (\text{B-1}) \\ \left. - a_{ik} u_{kj} - u_{ik} a_{jk} \right\}$$

$$\sum_{k=1}^n \sum_{l=1}^n y_{kl}(t) \frac{\partial h_{kl}}{\partial p_{ij}} = -y_{ij} \quad (\text{B-2})$$

$$\sum_{k=1}^n \sum_{l=1}^n y_{kl}(t) \frac{\partial h_{kl}}{\partial l_{ij}} = 2 \sum_{p=j}^n y_{ip}(t) l_{pj}(t) \quad (\text{B-3})$$

Assume-se que

$$\eta_v > 0, \eta_z > 0, \quad h_{ij}(P, L) = \sum_{k=1}^{\min\{i,j\}} l_{ik} l_{jk} - p_{ij}, \quad \frac{de_{ij}(h_{ij})}{dh_{ij}} \triangleq f_{ij}(h_{ij}),$$

as Eqs.(4.20-4.24) e Eqs.(B-2)-(B-3) implicam que

$$\frac{d\xi(P, L)}{dt} = \sum_{k=1}^n \sum_{l=1}^n \left[\frac{de_{kl}(g_{kl})}{dt} + \frac{de_{kl}(h_{kl})}{dt} \right] \quad (\text{B-4}) \\ = \sum_{k=1}^n \sum_{l=1}^n \sum_{i=1}^n \left\{ \sum_{j=1}^n \left[\frac{\partial e_{kl}(g_{kl})}{\partial p_{ij}} \frac{dp_{ij}(t)}{dt} + \frac{\partial e_{kl}(h_{kl})}{\partial p_{ij}} \frac{dp_{ij}(t)}{dt} \right] + \sum_{j=1}^i \frac{\partial e_{kl}(h_{kl})}{\partial l_{ij}} \frac{dl_{ij}(t)}{dt} \right\}$$

$$\frac{d\xi(P, L)}{dt} = \sum_{k=1}^n \sum_{l=1}^n \sum_{i=1}^n \left\{ \sum_{j=1}^n \left[\frac{de_{kl}(g_{kl})}{dg_{kl}} \frac{\partial g_{kl}}{\partial p_{ij}} + \frac{de_{kl}(h_{kl})}{dh_{kl}} \frac{\partial h_{kl}}{\partial p_{ij}} \right] \frac{dp_{ij}(t)}{dt} + \right. \quad (\text{B-5}) \\ \left. + \sum_{j=1}^i \frac{de_{kl}(h_{kl})}{dh_{kl}} \frac{\partial h_{kl}}{\partial l_{ij}} \frac{dl_{ij}(t)}{dt} \right\}$$

$$\begin{aligned} \frac{d\xi(P, L)}{dt} &= \sum_{k=1}^n \sum_{l=1}^n \sum_{i=1}^n \left\{ \sum_{j=1}^n \left[f_{kl}(g_{kl}) \frac{\partial g_{kl}}{\partial p_{ij}} + f_{kl}(h_{kl}) \frac{\partial h_{kl}}{\partial p_{ij}} \right] \frac{dp_{ij}(t)}{dt} + \right. \\ &\quad \left. + \sum_{j=1}^n f_{kl}(h_{kl}) \frac{\partial h_{kl}}{\partial l_{ij}} \frac{dl_{ij}(t)}{dt} \right\} \end{aligned} \quad (\text{B-6})$$

$$\begin{aligned} \frac{d\xi(P, L)}{dt} &= \sum_{l=1}^n \left\{ \sum_{j=1}^n \sum_{k=1}^n \sum_{i=1}^n \left[u_{kl}(t) \frac{\partial g_{kl}}{\partial p_{ij}} + y_{kl}(t) \frac{\partial h_{kl}}{\partial p_{ij}} \right] \frac{dp_{ij}(t)}{dt} + \right. \\ &\quad \left. + \sum_{j=1}^i \sum_{k=1}^n \sum_{l=1}^n y_{kl}(t) \frac{\partial h_{kl}}{\partial l_{ij}} \frac{dl_{ij}(t)}{dt} \right\} \end{aligned} \quad (\text{B-7})$$

$$\begin{aligned} \frac{d\xi(P, L)}{dt} &= \sum_{i=1}^n \sum_{j=1}^n \left\{ \sum_{k=1}^n \sum_{l=1}^n [p_{ik}(t) s_{kl} p_{lj}(t) + u_{il}(t) s_{kl} p_{lj}(t)] - \sum_{k=1}^n [a_{ik} u_{kj}(t) \right. \\ &\quad \left. + u_{ik}(t) a_{jk}(t) - y_{ij}(t)] \right\} \frac{dp_{ij}(t)}{dt} + 2 \sum_{i=1}^n \sum_{j=1}^i \left[\sum_{k=j}^n y_{ik}(t) l_{kj}(t) \right] \frac{dl_{ij}(t)}{dt} \end{aligned} \quad (\text{B-8})$$

$$\frac{d\xi(P, L)}{dt} = - \sum_{i=1}^n \left\{ \frac{1}{\eta_v} \sum_{j=1}^n \left[\frac{dp_{ij}(t)}{dt} \right]^2 + \frac{2}{\eta_z} \sum_{j=1}^i \left[\frac{dl_{ij}(t)}{dt} \right]^2 \right\}$$

$$\begin{cases} < 0, & \text{se } \exists i, j, dp_{ij}(t)/dt \neq 0 \text{ ou } dl_{ij}/dt \neq 0; \\ = 0, & \text{se } \forall i, j, dp_{ij}(t)/dt = 0 \text{ e } dl_{ij}/dt = 0. \end{cases}$$

Portanto, ξ é uma função de *Lyapunov* e a rede neural recorrente é assintoticamente estável no sentido amplo.

Análise de Solvabilidade

A análise de estabilidade indica que a transição do estado de ativação forma um fluxo de gradiente que minimiza a função ξ energia. O teorema a seguir fornece

uma condição necessária e suficiente para que o fluxo de gradiente convirja para a solução simétrica e definida positiva da equação algébrica de *Riccati*, P , a partir de quase qualquer estado inicial.

Teorema B.0.2. *Suponha que toda função de ativação seja contínua, pelo menos diferenciável por partes, monótona não decrescente, isto é, $\frac{df_{ij}(\xi)}{d\xi} \geq 0$, para $-\infty < \xi < +\infty$ e $f_{ij}(\xi) = 0$, se e somente se, $\xi = 0$.*

A matriz de estado estacionário da rede neural recorrente é sempre a solução simétrica e definida positiva para a equação de *Riccati* para qualquer estado inicial simétrico $P(0)$ e quaisquer estados iniciais não-nulos $L(0)$, (ou seja, $\forall P(0) = P(0)^T$, $\forall L(0) \neq 0$, $\bar{P} = P$), se e somente se,

$$\text{rank}(\bar{P}BR^{-1}B^T - A) = n \quad (\text{B-9})$$

e todos os elementos diagonais em \bar{L} são não nulos, isto é, $\forall i, \bar{l}_{ii} \neq 0$ para $i = 1, 2, \dots, n$.

Prova Os requisitos relativos a simetridade da matriz de estado inicial $P(0)$ e matriz de estado inicial não-nula $L(0)$, são devidos à assimetria da matriz de estado inicial $P(0)$, o qual resulta em uma assimetria de $P(t)$, e uma matriz de estado inicial nula não permite qualquer estado de transição. Visto que f_{ij} é contínua, $f_{ij}(\xi) = 0$ se $\xi = 0$,

$$\begin{aligned} U(t) &\triangleq F[P(t)SP(t) - A^T P(t)P(t)A - Q] \\ Y(t) &\triangleq F[L(t)L(t)T - P(t)] \\ \bar{U} &= F[\bar{P}S\bar{P} - A^T \bar{P} - \bar{P}A - Q] = 0, \end{aligned}$$

se e somente se $\bar{P}S\bar{P} - A^T \bar{P} - \bar{P}A - Q = 0$ e

$$\bar{Y} = F[\bar{L}\bar{L}^T - \bar{P}] = 0, \quad \text{se e somente se} \quad \bar{L}\bar{L}^T = \bar{P}.$$

A Eq.(4.25) implica que $\bar{P}S\bar{U} + \bar{U}S\bar{P} - A\bar{U} - \bar{U}A^T - \bar{Y} = 0$ quando $P(t)$ converge, isto é, $\frac{dP(t)}{dt} = 0$. Similarmente, Eq.(4.26) implica que $\bar{Y}\bar{L} = 0$ quando $L(t)$ converge, ou seja, $\frac{dL(t)}{dt} = 0$. Uma vez que

$$\begin{aligned}
 & \bar{P}S\bar{U} + \bar{U}S\bar{P} - A\bar{U} - \bar{U}A^T - \bar{Y} = \\
 & = (\bar{P}S - A)\bar{U} + [(\bar{P}S - A)\bar{U}]^T - \bar{Y} \\
 & = (\bar{P}S - A)\bar{U} + [(\bar{P}S - A)\bar{U}]^T - \bar{Y},
 \end{aligned}$$

a condição necessária e suficiente para $\bar{P}S\bar{U} + \bar{U}S\bar{P} - A\bar{U} - \bar{U}A^T - \bar{Y} = 0$, se e somente se, $\bar{U} = 0 \wedge \bar{Y} = 0$ é que $(\bar{P}BR^{-1}B^T - A)$ possua *rank* completo. Similarmente, a condição necessária e suficiente para $\bar{Y}\bar{L} = 0$, se e somente se, $\bar{Y} = 0$ isto é, \bar{L} tenha *rank* completo. Desde que $L(t)$ seja uma matriz triangular, $\text{rank}(\bar{L}) = n$, se e somente se, $\forall i, \bar{l}_{ii} \neq 0$.

Portanto, a matriz de estado da rede neural recorrente converge para uma solução P simétrica e definida positiva da equação algébrica de *Riccati*, se e somente se, $\text{rank}(\bar{P}BR^{-1}B^T - A) = n, \forall i, \bar{l}_{ii} \neq 0$.

Observação O teorema B.0.2 estabelece uma condição necessária e suficiente para convergência global da solução definida positiva da equação algébrica de *Riccati* para quase qualquer estado inicial arbitrário, ou seja, a matriz de estado $P(t)$ pode ainda alcançar a solução desejada dados os estados iniciais sem satisfazer as condições do teorema. Um método de projeto é fixar $P(0)$ e $L(0)$ com valores relativamente grandes de modo que $P(t)$ e $L(t)$ sejam inicialmente definidas positivas.

APÊNDICE C

Plantas dos Sistemas

Apresentamos neste Apêndice as matrizes que construíram o sistema em espaço de estados Eq. (3.1) e Eq. (3.2), representando as plantas dos sistemas usados no Capítulo 5 e as matrizes Q , R da aeronave e do $DFIG$, assim como as matrizes dos sistemas de 3ª e 5ª ordem usadas.

As matrizes representando o comportamento dinâmico e as entradas do sistema da aeronave são:

$$A_{Aero} = \begin{bmatrix} -20.00 & 0 & 0 & 0 & 0 & 0 \\ 0 & -25.00 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1.00 & 0 \\ -0.744 & -0.032 & 0 & -1.540 & -0.00420 & 1.5400 \\ 0.3370 & -1.120 & 0 & 0.2490 & -1.00 & -5.200 \\ 0.0200 & 70.00 & 0.0386 & -0.996 & -0.00029 & -0.117 \end{bmatrix} \quad (C-1)$$

$$B_{Aero} = \begin{bmatrix} 20.00 & 0.00 \\ 0.00 & 25.0 \\ 0.00 & 0.00 \\ 0.00 & 0.00 \\ 0.00 & 0.00 \\ 0.00 & 0.00 \end{bmatrix} \quad (C-2)$$

As matrizes Q e R associadas ao problema LQR determinadas pelo algoritmo genético.

$$Q_{Aero} = \begin{bmatrix} 10.1 & 1.9 & 2.0 & 1.7 & 0.4 & 1.2 \\ 1.9 & 8.2 & 2.6 & 0.7 & 0.6 & 2.5 \\ 2.0 & 2.6 & 7.3 & 1.7 & 2.2 & 1.6 \\ 1.7 & 0.7 & 1.7 & 38.1 & 2.0 & 1.2 \\ 0.4 & 0.6 & 2.2 & 2.0 & 78.0 & 0.3 \\ 1.2 & 2.5 & 1.6 & 1.2 & 0.3 & 29.2 \end{bmatrix}, \quad (C-3)$$

$$R_{Aero} = \begin{bmatrix} 31.9 & 4.6 \\ 4.6 & 25.9 \end{bmatrix}, \quad (C-4)$$

Outra planta que aplicamos a metodologia foi a de um gerador eólico duplamente alimentado (*DFIG*). Detalhes sobre o modelo linearizado podem ser encontrados em (Svensson 1998) e (Pinto 2007). As matrizes são apresentadas a seguir:

Planta do sistema de malha aberta.

$$A_{DFIG} = \begin{bmatrix} -0.07 & 0.33 & 0 & 0 & 0 & 0 \\ -0.33 & -0.07 & 0 & 0 & 0 & 0 \\ 0 & 0 & -0.19 & 0 & 0 & 0 \\ 0 & 0 & 0 & -0.03 & 1.00 & 0 \\ 0 & 0 & 0 & -1.00 & -0.03 & 0 \\ 0 & 0 & 0 & -10.00 & -0.12 & 0.71 \end{bmatrix}, \quad (C-5)$$

Matriz do sistema realimentado

$$A_{DFIG} = \begin{bmatrix} -39.4 & 0.39 & 0 & 0 & 0 & 0 \\ -0.27 & -55.7 & 0 & 0 & 0 & 0 \\ 0 & 0 & -21.9 & 0 & 0 & 0 \\ 0 & 0 & 0 & -24.5 & 0.63 & 19.8 \\ 0 & 0 & 0 & -1.27 & -18.0 & 0.72 \\ 0 & 0 & 0 & -6.91 & -0.08 & -1.78 \end{bmatrix}, \quad (C-6)$$

A matriz de controle,

$$B_{DFIG} = \begin{bmatrix} -12.46 & 0 & 0 & 0 & 0 \\ 0 & -12.46 & 0 & 0 & 0 \\ 0 & 0 & -2.19 & 0 & 0 \\ 0 & 0 & 0 & 5.71 & 0 \\ 0 & 0 & 0 & 0 & 5.71 \\ 0 & 0 & 0 & -0.71 & 0 \end{bmatrix} \quad (C-7)$$

As matrizes Q e R , usadas para o $DFIG$, foram:

$$Q_{DFIG} = \begin{bmatrix} 2.59 & 0.34 & 1.13 & 0.55 & 0.35 & 1.83 \\ 0.34 & 4.39 & 1.42 & 0.29 & 1.59 & 4.57 \\ 1.13 & 1.42 & 3.49 & 5.00 & 0.39 & 0.36 \\ 0.55 & 0.29 & 5.00 & 11.15 & 0.33 & 0.36 \\ 0.35 & 1.59 & 0.39 & 0.33 & 12.34 & 0.29 \\ 1.83 & 4.57 & 0.36 & 0.36 & 0.29 & 14.93 \end{bmatrix} \quad (C-8)$$

$$R_{DFIG} = \begin{bmatrix} 6.53 & 0.13 & 0.11 & 0.16 & 0.12 \\ 0.13 & 6.93 & 0.15 & 0.0832 & 0.13 \\ 0.11 & 0.15 & 7.45 & 0.12 & 0.15 \\ 0.16 & 0.08 & 0.12 & 9.68 & 0.13 \\ 0.12 & 0.13 & 0.15 & 0.13 & 10.56 \end{bmatrix} \quad (C-9)$$

As matrizes dos sistemas de 3ª ordem e 5ª ordem usadas nas simulações apresentadas na Seção 5.1.1.

Matriz dinâmica do sistema de 3ª ordem,

$$A_3 = \begin{bmatrix} 1.00 & 0.50 & -1.30 \\ -0.50 & 1.50 & 0 \\ 0.90 & -1.20 & -1.50 \end{bmatrix} \quad (C-10)$$

Matriz referente às entradas do sistema

$$B_3 = \begin{bmatrix} 1.00 & 1.00 \\ -1.00 & 2.00 \\ 0.50 & 1.00 \end{bmatrix} \quad (C-11)$$

Matriz Dinâmica referente ao sistema de 5ª ordem

$$A_5 = \begin{bmatrix} 0 & 2.00 & 0 & 0 & 0 \\ -0.10 & -0.35 & 0.10 & 0.10 & 0.75 \\ 0 & 0 & 0 & 2.00 & 0 \\ 0.40 & 0.40 & -0.40 & -1.40 & 0 \\ 0 & -0.03 & 0 & 0 & -1.00 \end{bmatrix} \quad (\text{C-12})$$

Matriz referente às entradas do sistema de 5ª ordem

$$B_5 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \quad (\text{C-13})$$

APÊNDICE D

Método de Schur

Este Apêndice tem como finalidade fazer uma breve apresentação do método de *Schur*, que é considerado um dos métodos clássicos para a solução da *EAR*. No Capítulo 2, foram apresentadas tanto a estrutura algébrica da *EAR* e no Capítulo 3 aplicação da mesma por meio do problema *LQR*, sendo o método de *Schur* largamente usado para a solução da *EAR* no *LQR*.

As dificuldades numéricas relacionadas com a utilização dos autovetores de matrizes hamiltonianas quase-defectivas podem ser contornadas com a aplicação do método de *Schur*. Então, a equação contínua de *Riccati* pode ser resolvida através da redução de H para uma forma quase-triangular, a forma real de *Schur*, utilizando transformações ortogonais de similaridade.

O método de *Schur* possui uma série de vantagens em relação ao método dos autovetores. Inicialmente, a redução para a forma quase-triangular é um passo intermediário no cálculo dos autovetores pelo método *QR*, de forma tal que o método de *Schur* requer menos cálculo que o método dos autovetores.

Importante também é que o método de *Schur* não sofre com as dificuldades inerentes dos autovetores de matrizes Hamiltonianas defectivas ou aproximadamente defectivas. Este método funciona bem na presença de divisores elementares lineares ou não lineares de H .

O método de *Schur* é significativamente mais rápido que o método de *Newton* e é mais rápido do que o método da função *sin* matricial com refinamento iterativo.

Existe uma matriz ortogonal U , a qual transforma a matriz hamiltoniana H na forma real de *Schur*: Eq.(D-1)

$$T = U^T H U = \begin{bmatrix} T_{11} & T_{12} \\ 0 & T_{22} \end{bmatrix} \quad (\text{D-1})$$

sendo, T_{11} e T_{22} , matrizes superiores quase-triangulares. O bloco na diagonal de T_{11} e T_{22} são de no máximo (2×2)

A redução da equação (D-1) não é única e é sempre possível escolher a matriz U de forma tal que os autovalores de T_{11} tem parte real negativa, enquanto que os autovalores de T_{22} possuem parte real positiva.

Seja a matriz U particionada,

$$U = \begin{bmatrix} U_{11} & U_{12} \\ U_{21} & U_{22} \end{bmatrix} \quad (\text{D-2})$$

sendo, cada bloco tem dimensão $n \times n$. Então, pode-se provar que a matriz U_{11} é não singular e a solução semi-definida positiva da equação EAR é dada por:

$$P = U_{21} U_{11}^{-1} \quad (\text{D-3})$$

Além do mais, os autovalores de T_{11} são os autovalores da matriz $(A - BK)$, isto é, os pólos do sistema de malha fechada ótimo devido a relação $(A - BK) = U_{11} T_{11} U_{11}^{-1}$.

A redução ortogonal da matriz hamiltoniana na forma real de *Schur* é feita pelo método QR , visto que a matriz H é não simétrica, não é possível garantir um ordenamento especial dos autovalores na diagonal da forma quase-triangular.

É possível reordenar esta forma arbitrariamente pela troca sistemática dos pares adjacentes dos autovalores, implementando transformações ortogonais de similaridade.

Apresentamos a seguir um algoritmo para a resolução do problema de simetria e ordenação.

1. Seja $H := \begin{bmatrix} A & S \\ -Q & -A^T \end{bmatrix}$

2. Reduzir a matriz H para a forma real de *Schur* $T = U^T H U$ usando transformações ortogonal de similaridade

3. Reordenar a forma real de *Schur* usando transformações ortogonais de similaridade de modo que o bloco quasi triangular T_{11} formado no canto superior direito T possua autovalores com parte real negativa e acumular as transformações em U .
4. Solucionar $U_1^T P = U_{21}^T$ para P , sendo $\begin{bmatrix} U_{11}^T & U_{21}^T \end{bmatrix}^T$ os vetores de *Schur* correspondentes ao bloco T_{11} .

APÊNDICE E

Rede Neuronal Direta

Neste Apêndice apresentamos uma parte da demonstração do desenvolvimento da *RND* apresentada no Capítulo 4 na Seção 4.1.1, simulações e discussão sobre a convergência, desempenho e estabilidade da *RND* são discutidos na Seção E.1.

Para calcularmos o gradiente de $E(P)$ Eq.(4.6), usando as propriedades Eq.(4.7) e usando a regra da cadeia para cada um dos termos temos,

$$\frac{\delta \text{traco}(v^T R^T P^T P R v)}{\delta t} = \frac{\delta \text{traco}(v^T R^T P^T P R v)}{\delta t} + \frac{\delta \text{traco}(v^T R^T P^T P R v)}{\delta t}$$

$$P R v v^T R^T + P R v v^T R^T = 2 P R v v^T R^T$$

para o segundo termo $-v^T R^T P^T A^T P z$,

$$\frac{\delta \text{traco}(-v^T R^T P^T A^T P z)}{\delta t} = -A^T P z v^T R^T - A P R v z^T \quad (\text{E-1})$$

para o terceiro termo $-v^T R^T P^T P A z$,

$$\frac{\delta \text{traco}(-v^T R^T P^T P A z)}{\delta t} = -P A z v^T R^T - P R v z^T A^T \quad (\text{E-2})$$

para o quarto termo $-v^T R^T P^T Q z$,

$$\frac{\delta \text{traco}(-v^T R^T P^T Q z)}{\delta t} = -Q z v^T R^T \quad (\text{E-3})$$

para o quinto termo $-z^T P^T A P R v$,

$$\frac{\delta \text{traco}(-z^T P^T A P R v)}{\delta t} = -A P R v z^T - A^T P z v^T R^T \quad (\text{E-4})$$

para o sexto termo $z^T P^T A A^T P z$,

$$\frac{\delta \text{traco}(z^T P^T A A^T P z)}{\delta t} = A A^T P z z^T + A A^T P z z^T = 2 A A^T P z z^T \quad (\text{E-5})$$

para o sétimo termo $z^T P^T A P A z$,

$$\frac{\delta \text{traco}(z^T P^T A P A z)}{\delta t} = A P A z z^T + A^T P z z^T A^T \quad (\text{E-6})$$

para o oitavo termo $z^T P^T A Q z$,

$$\frac{\delta \text{traco}(z^T P^T A Q z)}{\delta t} = A Q z z^T \quad (\text{E-7})$$

para o nono termo $-z^T A^T P^T P R v$,

$$\frac{\delta \text{traco}(-z^T A^T P^T P R v)}{\delta t} = P R v z^T A^T - P A z v^T R^T \quad (\text{E-8})$$

para o décimo termo $z^T A^T P^T A^T P z$,

$$\frac{\delta \text{traco}(z^T A^T P^T A^T P z)}{\delta t} = A^T P z z^T A^T + A P A z z^T \quad (\text{E-9})$$

para o décimo primeiro termo $z^T A^T P^T P A z$,

$$\frac{\delta \text{traco}(z^T A^T P^T P A z)}{\delta t} = P A z z^T A^T + P A z z^T A^T = 2 P A z z^T A^T \quad (\text{E-10})$$

para o décimo segundo termo $z^T A^T P^T Q z$,

$$\frac{\delta \text{traco}(z^T A^T P^T Q z)}{\delta t} = Q z z^T A^T \quad (\text{E-11})$$

para o décimo terceiro termo $-z^T Q^T P R v$,

$$\frac{\delta \text{traco}(-z^T Q^T P R v)}{\delta t} = -Q z v^T R^T \quad (\text{E-12})$$

para o décimo quarto termo $z^T Q^T A^T P z$,

$$\frac{\delta \text{traco}(z^T Q^T A^T P z)}{\delta t} = A Q z z^T \quad (\text{E-13})$$

para o décimo quinto termo $z^T Q^T P A z$,

$$\frac{\delta \text{traco}(z^T Q^T P A z)}{\delta t} = Q z z^T A^T \quad (\text{E-14})$$

como o décimo sexto termo não é função de P , a derivada é nula.

E.1 Comportamento do Treinamento e Convergência da RND com Perturbações

Apresentamos nesta Seção considerações adicionais as simulações e análises feitas no Capítulo 5 Seção 5.2, que refletem o comportamento do treinamento da *RND* em relação às perturbações das matrizes *A* e *B* que compõem os parâmetros da *EAR* em relação da convergência, estabilidade e desempenho da *RND*.

As Figuras E.1 – E.3, representam a diferença entre as soluções de *Schur* e o erro da *RND*. É observado que existe uma dependência da rede direta em relação às condições iniciais, de modo que a diferença entre as soluções se mantém constante quase que ao longo de todo processo iterativo, o que significa que a *RND*, converge prematuramente, com algum grau de dependência da condição inicial. Como vantagem temos o seu desempenho, que é relativamente rápido sendo que não é necessário o cálculo de derivadas ou integrais em sua estrutura.

Alguns ajustes podem ser feitos para sintonizar a *RND* e desde modo melhorar o desempenho, convergência e qualidade da solução. Os experimentos realizados representam uma grande variação das matrizes associadas ao sistema, sendo que para uma série de processos não existe uma variação tão grande no processo, deste modo, para pequenas variações a *RND* torna-se uma alternativa a ser considerada em implementações práticas.

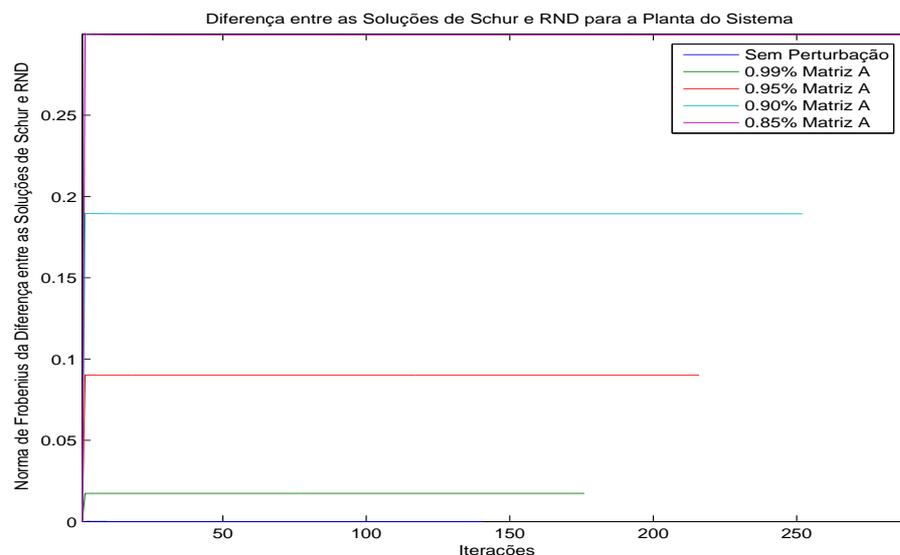


Figura E.1: Diferença da Norma de *Frobenius* para as Soluções da *EAR* usando o método de *Schur* e *RND* para Planta Perturbada

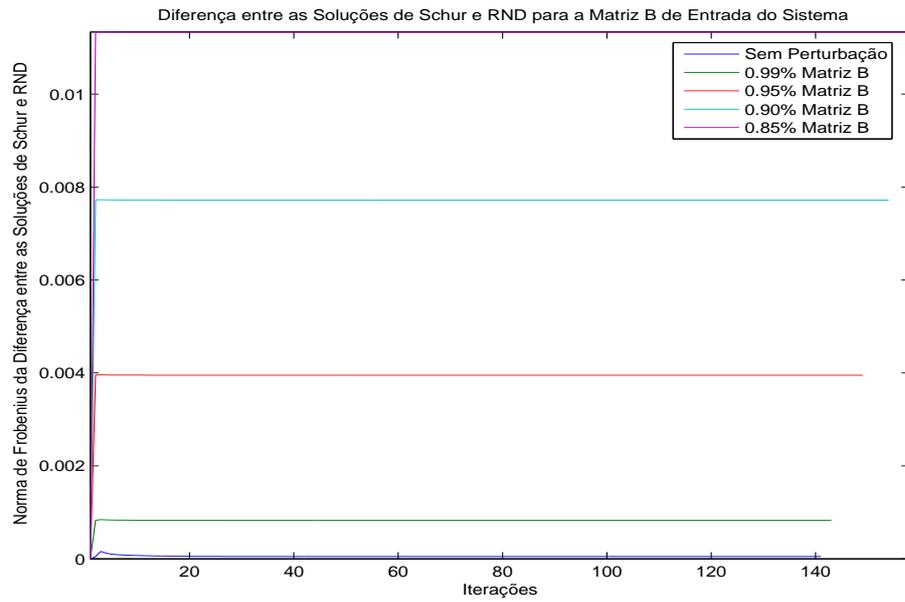


Figura E.2: Diferença da Norma de *Frobenius* das Soluções da *EAR* usando o método de *Schur* e *RND* para a Matriz *B* Perturbada

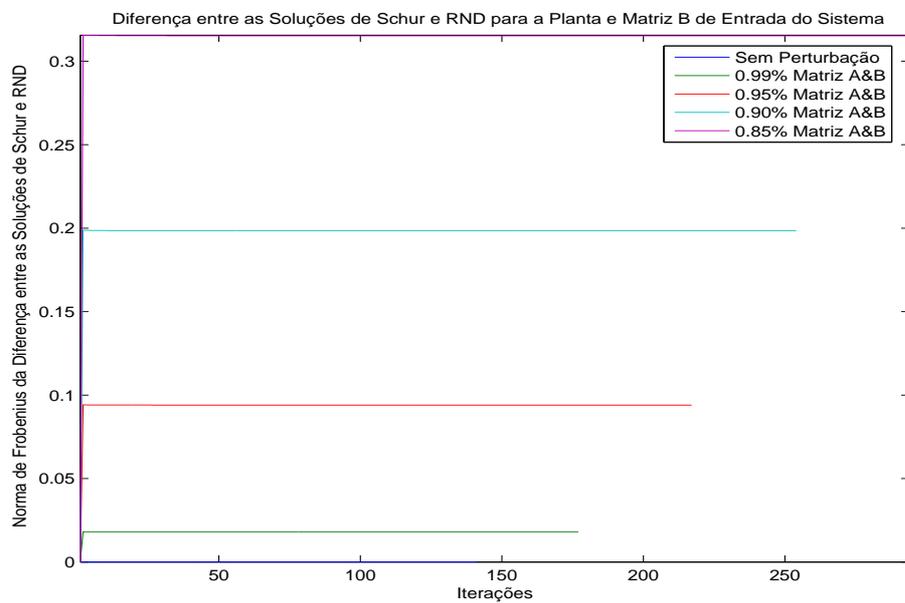


Figura E.3: Diferença da Norma de *Frobenius* das Soluções da *EAR* usando o método de *Schur* e *RND* para Planta e Matriz *B* Perturbadas

O retorno das soluções da *EAR* pela *RND* são mostradas pelas Figuras E.4-E.6, sendo que para representarmos as soluções usando a norma de *Frobenius* como métrica para a comparação das matrizes das soluções da *EAR* geradas pela *RND* durante o processo de treinamento da rede. O resultado desvia da solução a medida que as perturbações se acentuam e a solução da *EAR* não converge para zero.

Temos que levar em consideração que as perturbações em torno da condição inicial foram muito acentuadas e como foi citado na Seção 4.1, para este caso particular tivemos esse comportamento, não podendo garantir que o mesmo venha a ocorrer em outras plantas, além do mais a ordem do sistema deve ser levada em consideração que a que por ser bastante elevada implica em um maior número de variáveis envolvidas no processo e deste modo torna-se mais complexo a sintonia dos métodos usados para solucionar a *EAR*.

Novamente verificamos a maior influência da planta do sistema na solução da *EAR*, as Figuras, E.4 e E.6, que retornaram valores próximos de 3,6 para as maiores variações da planta e de 3,9 para perturbações da planta e entradas simultaneamente.

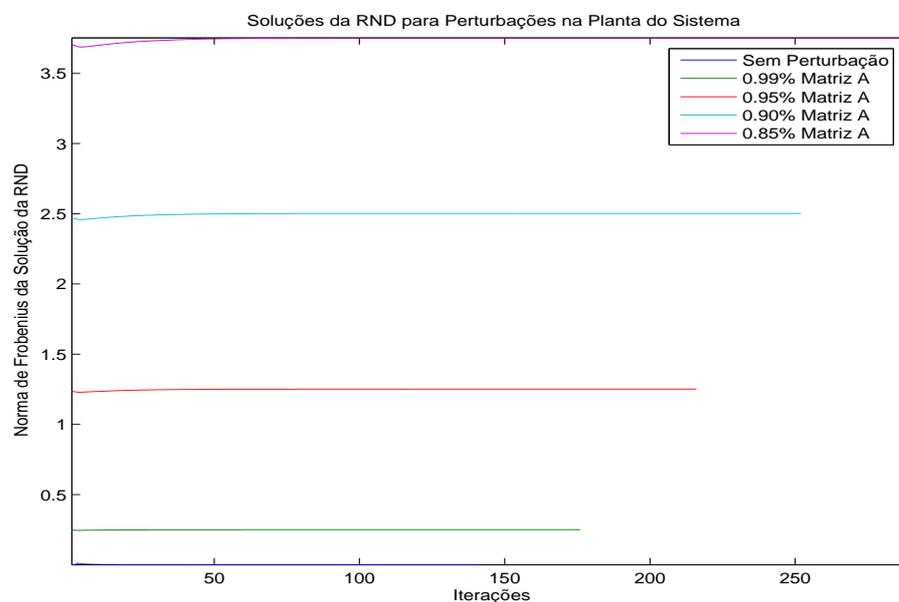


Figura E.4: Norma de *Frobenius* da Solução da *RND* para Perturbações na Planta

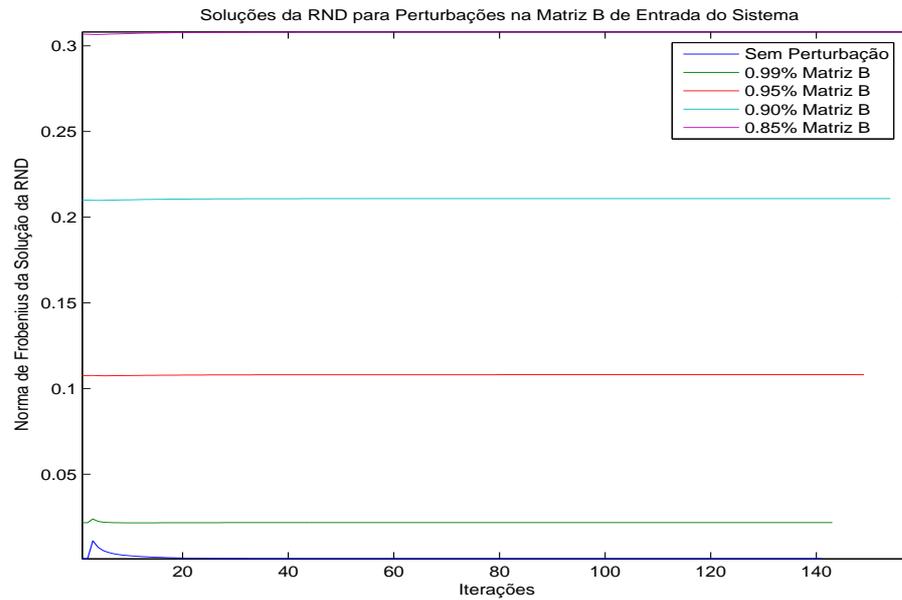


Figura E.5: Norma de *Frobenius* da Solução da *RND* para Perturbações na Matriz de Entrada *B*

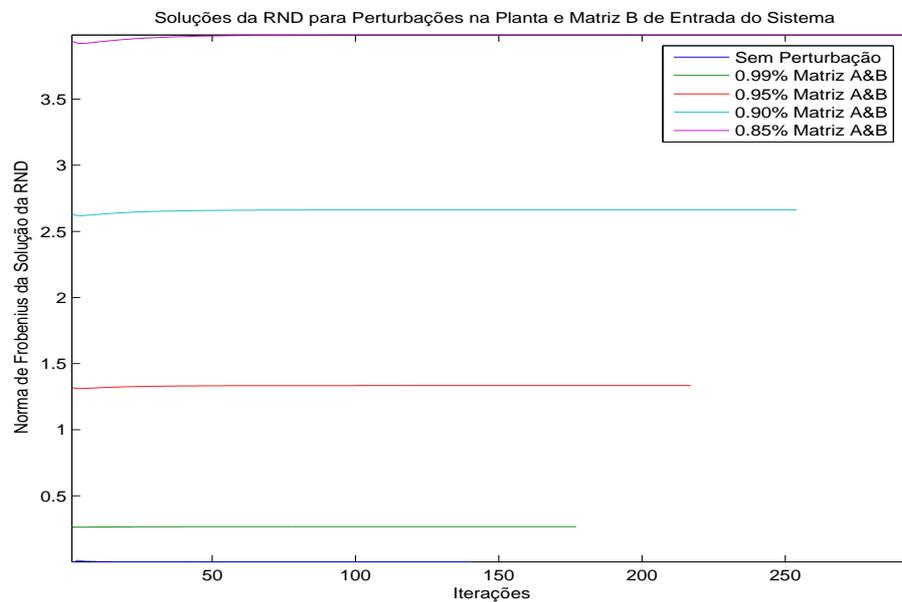


Figura E.6: Norma de *Frobenius* da Solução da *RND* para Perturbações na Planta e Matriz *B* de Entrada do Sistema

APÊNDICE F

Aeronave

Apresentamos neste Apêndice considerações e análises sobre o comportamento da rede neuronal recorrente, como complemento aos experimentos computacionais e questões relativas à escolha paramétrica responsável pelo sintonia da *RNR* responsáveis pelo desempenho, convergência e estabilidade discutidas nos Capítulo 5.

A seguir apresentamos algumas figuras referentes às variações dos parâmetros η_z e η_v com diferentes configurações.

Na Figura F.1, apresentamos as soluções obtidas para o sistema de 6^a ordem apresentado acima com uma maior faixa de variação dos parâmetros η_z e η_v Eq.(4.22) e Eq.(4.23) para que fosse possível observar de modo amplo a influência nas variações dos parâmetros.

Nas Figuras F.2 e F.3, foi selecionado um intervalo em que se obteve as menores normas das soluções apresentados na Figura F.1 o que visualmente é representado pelo primeiro quadrado próximo aos eixos dos parâmetros η_z e η_v . Estes resultados aqui foram apresentados pelo autor no *IX Congresso Brasileiro de Automática* (Fábio Nogueira 2006).

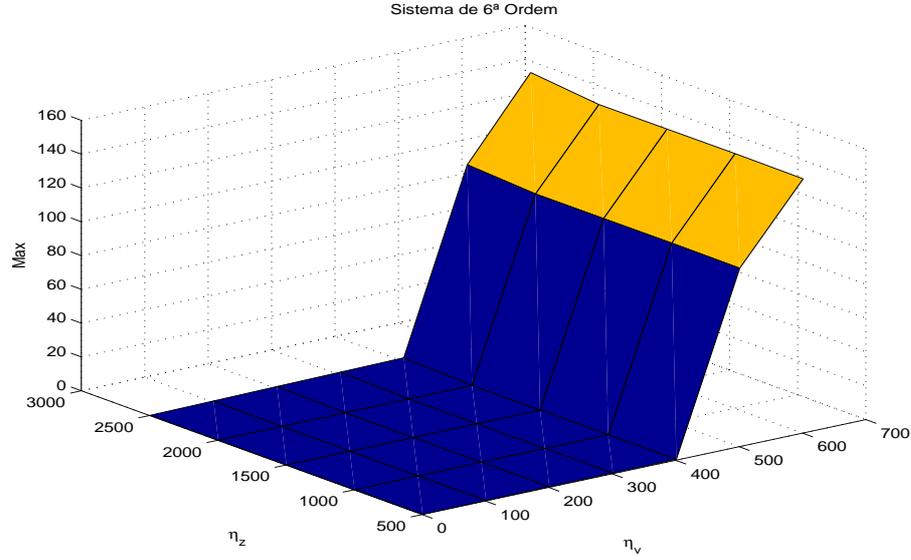


Figura F.1: n_{\max} da solução V para a variação η_v : 1-600 e η_z :500-2500 para um sistema de 6ª ordem da aeronave.

Diversos e exaustivos testes foram executados para a determinação tanto dos parâmetros η_z e η_v , como também do tempo de processamento T_p e número de amostragem N_{Sample} , pois, os mesmos também exercem grande influência no comportamento global do algoritmo, para a busca da solução da *EAR*. Os outros sistemas testes tiveram como condições iniciais as matrizes identidades. Neste experimento usamos uma condição homotópica (Oettli 1995) para verificar se a rede seria capaz de solucionar. A condição inicial usada foi a solução fornecida pelo método de *Schur*. A matriz usada como condição inicial $P(0)_{RNR}$, relação (F-1) apresentada abaixo,

$$P(0)_{RNR} = \begin{bmatrix} 0.3002 & 0.0551 & -0.1507 & -2.0658 & -0.2985 & 3.1657 \\ 0.0551 & 0.2027 & -0.4445 & -1.0346 & -1.2521 & 3.0215 \\ -0.1507 & -0.4445 & 27.8866 & 13.5111 & 13.9739 & -32.9266 \\ -2.0658 & -1.0346 & 13.5111 & 76.5739 & 24.7117 & -127.3341 \\ -0.2985 & -1.2521 & 13.9739 & 24.7117 & 34.0006 & -70.8805 \\ 3.1657 & 3.0215 & -32.9266 & -127.3341 & -70.8805 & 319.2099 \end{bmatrix} \quad (F-1)$$

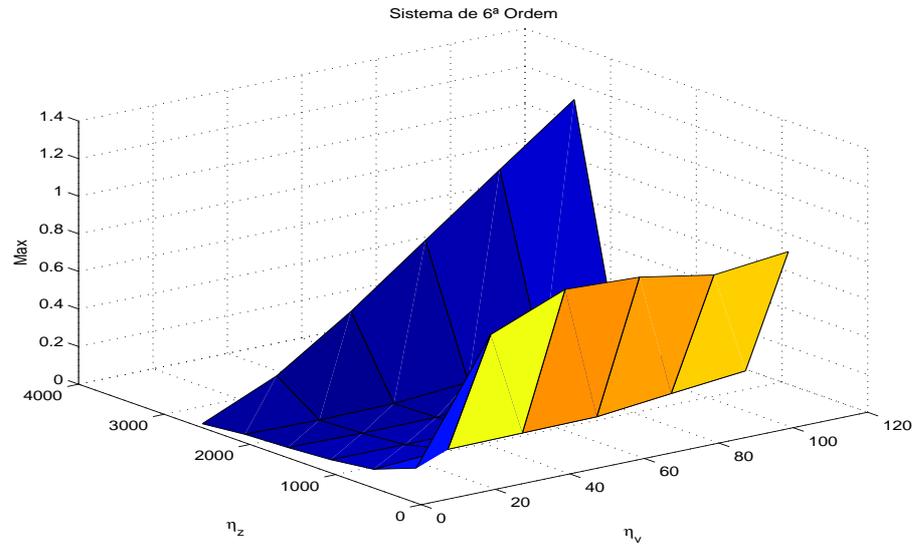


Figura F.2: n_{\max} da solução V para a variação η_v : 1-120 e η_z :500-2500 para um sistema de 6ª ordem da aeronave.

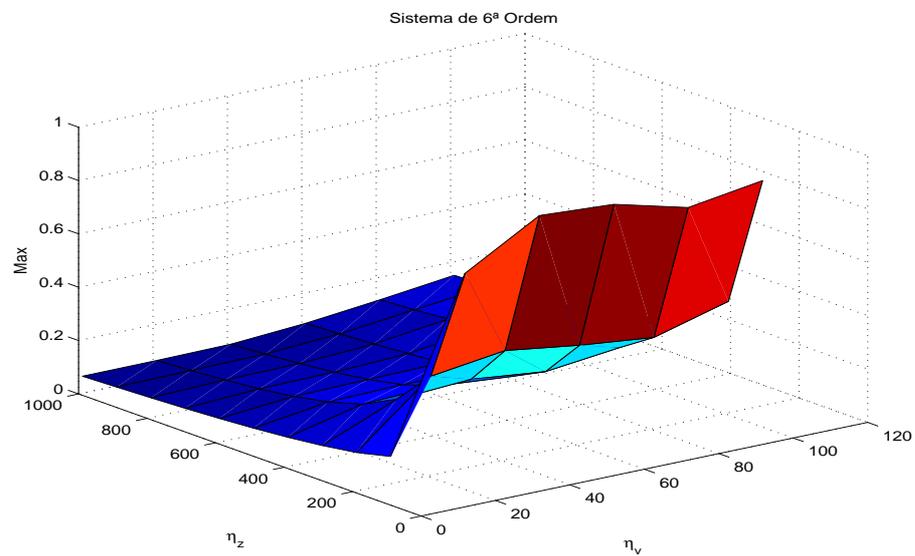


Figura F.3: n_{\max} da solução V para a variação η_v : 1-120 e η_z :100-1000 para um sistema de 6ª ordem da aeronave.

APÊNDICE G

Decomposição de Valores Singulares

Neste apêndice temos o interesse de mostrar um pouco do comportamento dos sistemas através da análise dos valores singulares de cada uma das plantas apresentadas sem controle com o sistema realimentado. Como estamos trabalhando com sistemas multivariáveis fizemos a escolha da decomposição de valores singulares.

Para a análise dos valores singulares de sistemas *MIMO*, que garantam a robustez é dado pela avaliação dos limites dos valores singulares mínimos do ganho de malha que devem ser grandes nas baixas frequências para garantir a robustez de desempenho e os valores singulares máximos de ganho de malha devem ser baixos nas altas frequências para garantir a robustez de estabilidade. Portanto, o controlador projetado satisfaz as condições de desempenho e estabilidade no domínio da frequência.

As figuras G.1 e G.2 apresentam os valores singulares dos dois sistemas testes principais testas com as metodologias neuronais. Na figura G.1 o sistema base para a planta do *DFIG*, sem controle apresenta um comportamento em baixas frequências um comportamento tem oscilatório e o desempenho prejudicados, já com a aplicação do regulador linear quadrático, percebe-se claramente que ocorreu um atenuação o que pode caracterizar um melhor comportamento do sistema, mesmo que com menores ganhos. Na análise de estabilidade em altas frequências não se verifica de modo visível um ganho ou perda de estabilidade, apenas no

início da transição das baixas para altas frequências, sendo que com o controlador o sistema entra em regime permanente mais rapidamente.

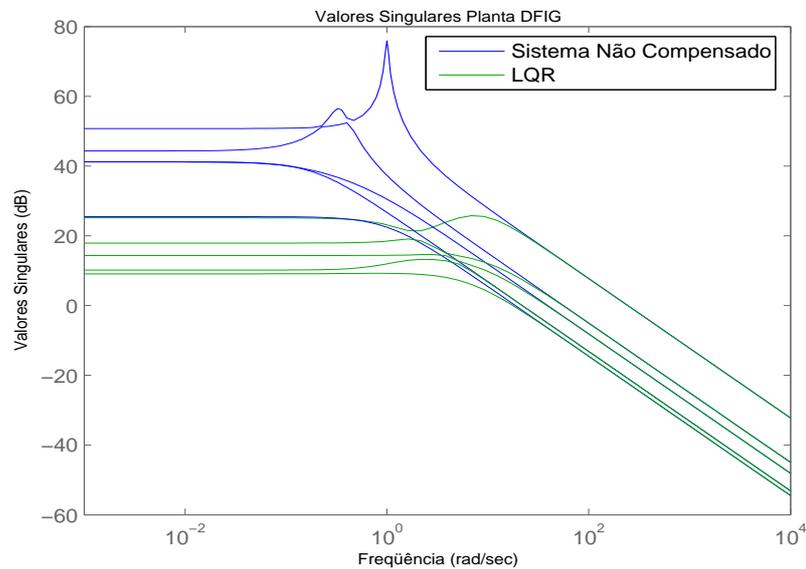


Figura G.1: Decomposição de Valores Singulares para o Planta do DFIG

Para a planta da aeronave, os valores singulares apresentados pela figura G.2, em baixas frequências é as condições de desempenho robusto são satisfeitas, e para esse caso os valores singulares mínimos tiveram um ligeiro ganho se comparados com o sistema base já os valores singulares máximos se percebeu ligeiro ganho se notando uma atenuação do oscilatório, em altas frequências as condições de estabilidades são satisfeitas.

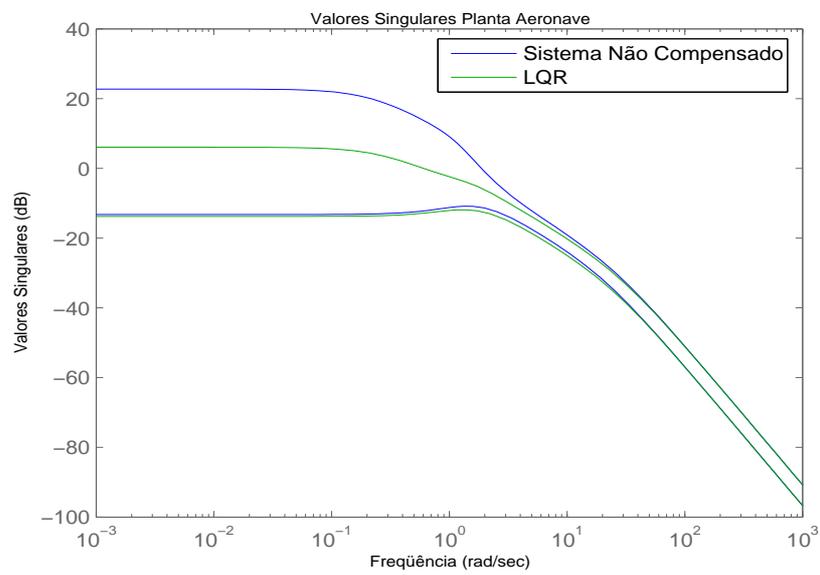


Figura G.2: Decomposição de Valores Singulares para o Planta da Aeronave