



**UNIVERSIDADE FEDERAL DO CEARÁ**  
**CENTRO DE TECNOLOGIA**  
**DEPARTAMENTO DE TELEINFORMÁTICA**  
**PROGRAMA DE PÓS-GRADUAÇÃO EM TELEINFORMÁTICA**  
**MESTRADO ACADÊMICO EM TELEINFORMÁTICA**

**HAROLDO CABRAL MAYA**

**REVISITANDO O PROBLEMA DE ESTIMAÇÃO DE CURVAS DE POTÊNCIA DE  
AEROGERADORES VIA MODELOS POLINOMIAIS ESPARSOS**

**FORTALEZA**

**2018**

HAROLDO CABRAL MAYA

REVISITANDO O PROBLEMA DE ESTIMAÇÃO DE CURVAS DE POTÊNCIA DE  
AEROGERADORES VIA MODELOS POLINOMIAIS ESPARSOS

Dissertação apresentada ao Curso de Mestrado Acadêmico em Teleinformática do Programa de Pós-Graduação em Teleinformática do Centro de Tecnologia da Universidade Federal do Ceará, como requisito parcial à obtenção do título de mestre em Engenharia Teleinformática. Área de Concentração: Sinais e Sistemas

Orientador: Prof. Dr. Guilherme de Alencar Barreto

FORTALEZA

2018

Dados Internacionais de Catalogação na Publicação  
Universidade Federal do Ceará  
Biblioteca Universitária  
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

---

- C118r Cabral Maya, Haroldo.  
Revisitando o Problema de Estimação de Curvas de Potência de Aeroogeradores via Modelos Polinômiais Esparsos / Haroldo Cabral Maya. – 2018.  
100 f. : il. color.
- Dissertação (mestrado) – Universidade Federal do Ceará, Centro de Tecnologia, Programa de Pós-Graduação em Engenharia de Teleinformática, Fortaleza, 2018.  
Orientação: Prof. Dr. Guilherme de Alencar Barreto.
1. Curva de Potência. 2. Turbinas Eólicas. 3. Computação Evolucionária. 4. Modelos Esparsos. 5. Regularização. I. Título.

CDD 621.38

---

HAROLDO CABRAL MAYA

REVISITANDO O PROBLEMA DE ESTIMAÇÃO DE CURVAS DE POTÊNCIA DE  
AEROGERADORES VIA MODELOS POLINOMIAIS ESPARSOS

Dissertação apresentada ao Curso de Mestrado Acadêmico em Teleinformática do Programa de Pós-Graduação em Teleinformática do Centro de Tecnologia da Universidade Federal do Ceará, como requisito parcial à obtenção do título de mestre em Engenharia Teleinformática. Área de Concentração: Sinais e Sistemas

Aprovada em: 30 de Julho de 2018

BANCA EXAMINADORA

---

Prof. Dr. Guilherme de Alencar Barreto (Orientador)  
Universidade Federal do Ceará (UFC)

---

Prof. Dr. Paulo Cesar Marques de Carvalho  
Universidade Federal do Ceará (UFC)

---

Prof. Dr. João Paulo Pordeus Gomes  
Universidade Federal do Ceará (UFC)

## AGRADECIMENTOS

Agradeço imensamente à minha mãe, Lúcia Cabral Costa de Oliveira, que além de me oferecer seu apoio incondicional em todas as minhas empreitadas pessoais e profissionais, é o maior exemplo próximo de bondade, sabedoria e integridade que tive o prazer de conviver e que venho tentando me igualar.

Agradeço imensamente ao meu orientador do mestrado, Prof. Dr. Guilherme de Alencar Barreto, que mesmo em meio as inúmeras atribuições profissionais que lhe são conferidas, foi capaz de gentilmente ceder parte de seu tempo em prol do meu desenvolvimento acadêmico. Sua postura profissional e dedicação ao trabalho foram os maiores ensinamentos que tive o prazer de receber e que honrarei perpetuando-as em minha eventual vida como pesquisador e/ou professor.

Agradeço à querida Georgia Effgen Santos, pelo amor superlativo e apoio irrestrito no caminho que trilho diariamente, e por, mesmo diante de todos os percalços, estar sempre segurando minha mão.

Agradeço aos colegas da pós-graduação que frequentam a sala de estudo número 1, vulgarmente conhecida como "Centaurinho", pelo ambiente descontraído, salutar e colaborativo em que todos se debruçam sobre os problemas dos demais.

Agradeço a todos os Professores que tive a honra de ter, pelas valiosas lições acadêmicas, morais e profissionais.

Agradeço ao apoio dos meus amigos mais próximos que acompanharam, torceram e vibraram com as minhas conquistas.

Agradeço a Universidade Federal do Ceará, pela possibilidade de engrandecimento pessoal e profissional proporcionado pela vida acadêmica.

Agradeço ao CNPq (processo no. 309451/2015-9) e à CAPES (processo no. 1707928/DS) pelo suporte financeiro a esta pesquisa. E também ao CENAPAD/UFC pela permissão de uso do seu sistema computacional, incluindo o software MATLAB®.

“Essencialmente, todos os modelos estão errados,  
mas alguns são úteis.”

(George E. P. Box)

## RESUMO

Neste trabalho, o clássico modelo polinomial para estimação da curva de potência de aerogeradores é revisitado com o intuito de tornar seu projeto automático e mais parcimonioso. Para este fim, propõe-se uma metodologia de projeto baseada em computação evolucionária que retorna a ordem ótima do modelo polinomial, bem como seleciona (por poda) os termos relevantes do polinômio. A metodologia proposta é comparada com o estado da arte em estimação da curva de potência de aerogeradores, tais como modelos logísticos (de 3, 4 e 5 termos), redes neurais artificiais e modelo polinomial ponderado. Comparação de desempenho com o método LASSO, amplamente utilizado na obtenção de modelos esparsos regularizados, é também realizada. Os resultados indicam um desempenho consistentemente superior da metodologia proposta em relação a todos os métodos alternativos avaliados.

**Palavras-chave:** Energias renováveis. Aerogeradores. Curva de potência. Modelo polinomial. Computação evolucionária.

## ABSTRACT

In this dissertation, the classical polynomial model for wind turbines power curve estimation is revisited aiming at an automatic and parsimonious design. For this purpose, with the help of genetic algorithms we introduce a methodology aiming at estimating the optimal order of the polynomial as well the relevant terms in this polynomial. The proposed methodology is compared with the state of the art in estimating the power curve of wind turbines, such as logistic models (with 3, 4 and 5 parameters), artificial neural networks and weighted polynomial regression. We also show that the proposed approach performs better than the standard LASSO approach for building regularized sparse models. The results indicate that the proposed methodology consistently outperforms all the evaluated alternative methods.

**Keywords:** Renewable energy. Wind turbines. Power curve. Polynomial model. Evolutionary computation.

## LISTA DE FIGURAS

Figura 1 – Fluxo de ar através da área circular de varredura das pás do aerogerador. . .	23
Figura 2 – Curva de potência típica de uma aerogerador com controle de potência por ângulo de passo. . . . .	25
Figura 3 – Gráfico de $w$ em relação a $u$ desenvolvido a partir da função bponderada de Tukey. . . . .	36
Figura 4 – Gráfico sigmóide simples da função $f(x) = \frac{1}{1+e^{-x}}$ . . . . .	37
Figura 5 – Representação em blocos de um neurônio artificial $\alpha$ . . . . .	39
Figura 6 – Rede MLP com duas camadas ocultas. . . . .	40
Figura 7 – Quantidade de termos utilizados em função de $\lambda$ para o modelo LASSO em TURB2. . . . .	53
Figura 8 – Curva da aptidão em função do valor de $\lambda$ para o modelo LASSO. . . . .	54
Figura 9 – Curva de potência do modelo PRISMA. Os pontos cinza são os dados do aerogerador TURB1 e a potência de saída foi normalizada pela potência nominal. As curvas tracejadas correspondem ao intervalo de $\pm\sigma_\epsilon$ . . . . .	58
Figura 10 – Curva de potência do modelo PRISMA. Os pontos cinza são os dados do aerogerador TURB2 e a potência de saída foi normalizada pela potência nominal. As curvas tracejadas correspondem ao intervalo de $\pm\sigma_\epsilon$ . . . . .	58
Figura 11 – Curva de potência do modelo PRISMA. Os pontos cinza são os dados do aerogerador TURB3 e a potência de saída foi normalizada pela potência nominal. As curvas tracejadas correspondem ao intervalo de $\pm\sigma_\epsilon$ . . . . .	59
Figura 12 – Boxplots dos resíduos para cada um dos modelos utilizados em TURB1. . .	63
Figura 13 – Boxplots dos resíduos para cada um dos modelos utilizados em TURB2. . .	63
Figura 14 – Boxplots dos resíduos para cada um dos modelos utilizados em TURB3. . .	63

## LISTA DE TABELAS

Tabela 1 – Especificação dos três aerogeradores. . . . .	50
Tabela 2 – Desempenho do modelo PRISMA em TURB1. . . . .	51
Tabela 3 – Desempenho do modelo PRISMA em TURB2. . . . .	51
Tabela 4 – Desempenho do modelo PRISMA em TURB3. . . . .	51
Tabela 5 – Desempenho dos modelos polinomial e polinomial robusto em TURB1. . .	52
Tabela 6 – Desempenho dos modelos polinomial e polinomial robusto em TURB2. . .	52
Tabela 7 – Desempenho dos modelos polinomial e polinomial robusto em TURB3. . .	52
Tabela 8 – Desempenho do modelo LASSO em TURB1. . . . .	54
Tabela 9 – Desempenho do modelo LASSO em TURB2. . . . .	55
Tabela 10 – Desempenho do modelo LASSO em TURB3. . . . .	55
Tabela 11 – Desempenho dos modelos logísticos em TURB1. . . . .	55
Tabela 12 – Desempenho dos modelos logísticos em TURB2. . . . .	56
Tabela 13 – Desempenho dos modelos logísticos em TURB3. . . . .	56
Tabela 14 – Desempenho do modelo MLP em TURB1. . . . .	56
Tabela 15 – Desempenho do modelo MLP em TURB2. . . . .	57
Tabela 16 – Desempenho do modelo MLP em TURB3. . . . .	57
Tabela 17 – Comparação de desempenho dos modelos quanto ao SQR em TURB1. . . .	59
Tabela 18 – Comparação de desempenho dos modelos quanto ao SQR em TURB2. . . .	60
Tabela 19 – Comparação de desempenho dos modelos quanto ao SQR em TURB3. . . .	60
Tabela 20 – Comparação de desempenho dos modelos quanto ao AIC em TURB1. . . .	60
Tabela 21 – Comparação de desempenho dos modelos quanto ao AIC em TURB2. . . .	61
Tabela 22 – Comparação de desempenho dos modelos quanto ao AIC em TURB3. . . .	61
Tabela 23 – Comparação de desempenho dos modelos quanto ao BIC em TURB1. . . .	61
Tabela 24 – Comparação de desempenho dos modelos quanto ao BIC em TURB2. . . .	62
Tabela 25 – Comparação de desempenho dos modelos quanto ao BIC em TURB3. . . .	62

## LISTA DE ABREVIATURAS E SIGLAS

4PL	<i>Four Parameter Logistic</i>
5PL	<i>Five Parameter Logistic</i>
AG	Algoritmo Genético
AIC	<i>Akaike's information criterion</i>
BIC	<i>Bayesian information criterion</i>
DE	<i>Evolução Diferencial</i>
FDP	Função de Densidade de Probabilidade
IRLS	<i>Iteratively Reweighted Least Squares</i>
LASSO	<i>Least Absolute Shrinkage and Selection Operator</i>
MAD	<i>Median Absolute Derivation</i>
MMV	Método da Máxima Verossimilhança
OLS	<i>Ordinary Least Squares</i>
PRISMA	<i>Polynomial Reduction Induced by Sparse Method Algorithm</i>
RNA	Rede Neural Artificial
SQR	Soma dos Quadrados dos Resíduos
WLS	<i>Weighted Least Squares</i>

## LISTA DE SÍMBOLOS

$a$	Base da progressão geométrica utilizada no modelo LASSO para seleccionar o conjunto de $\lambda$ s para teste
$A$	Área circular varrida pelas pás de um aerogerador de eixo horizontal
$b$	Limiar de ativação do neurônio
$c$	Limiar de identificação de um <i>outlier</i> utilizado na regressão robusta
$C$	Parcela constante da função de log-verossimilhança de um determinado conjunto de dados
$C_p$	Coefficiente de potência ( $P_a/P_w$ )
$d$	Caracterizador da norma
$D$	Dimensão dos dados de entrada de um neurônio artificial
$E_c$	Energia cinética
$f$	Uma função genérica qualquer
$g$	Função de densidade de probabilidade da potência elétrica
$J$	Função de custo da regressão cujo objetivo é reduzir seu valor de resposta
$k$	Cardinalidade de $\theta$
$k_\beta$	Cardinalidade do vetor $\beta$
$l_d$	Função de penalização L-norm
$L$	Função de log-verossimilhança
$\mathcal{L}$	Função de verossimilhança
$M$	Massa
$n$	Cardinalidade do conjunto amostral
$p$	Potência elétrica fornecida pelo rotor
$p$	Vetor de $p_i \quad i \in \{1 \dots n\}$
$P$	Potência mecânica genérica
$P_a$	Potência eólica absorvida
$P_w$	Potência eólica disponível
$t$	Tempo

$u$	Residual normalizado pelo seu desvio padrão estimado ( $\frac{\varepsilon_i}{\hat{\sigma}_\varepsilon}$ )
$v$	Velocidade do vento
$\mathbf{V}$	Matriz de Vandermonde de $v$
$x$	Variável aleatória qualquer
$y$	Variável dependente qualquer
$w_i$	Componente de ponderação de um determinado dado de índice $i$ em que $i \in \{1 \dots n\}$
$W$	Matriz diagonal $n \times n$ com $W_{ii} = w_i$
$X$	FDP de $x$
$z$	Dados de entrada de um neurônio artificial
$\alpha$	Identificação de um neurônio qualquer
$\beta$	Vetor dos coeficientes de uma regressão
$\gamma$	Constante utilizada para estimar o desvio padrão de uma distribuição através do MAD. Seu valor depende do tipo de distribuição em análise
$\varepsilon$	Residual da regressão
$\eta$	Eficiência de conversão de energia mecânica em elétrica do rotor
$\zeta$	Coefficiente de ativação do modelo PRISMA
$\theta$	Vetor de parâmetros de dependência da FDP $g$
$\theta_x$	Vetor de parâmetros de dependência da FDP $X$
$\lambda$	Constante que regulariza a intensidade da penalização $\ell_d$
$\lambda_{max}$	O menor valor de $\lambda$ cuja consequência é excluir todos os termos regressores do modelo LASSO
$\mu$	Valor esperado de potência elétrica
$\rho$	Densidade do ar
$\sigma^2$	Variância da potência elétrica
$\sigma_\varepsilon^2$	Variância de $\varepsilon$
$\hat{\sigma}$	Desvio padrão estimado
$\varphi(\cdot)$	Função de ativação do neurônio

$\psi$  Derivada da função  $\Psi$

$\Psi$  Função contínua, derivável em  $\mathbb{R}$  e com um mínimo global

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>17</b>
<b>1.1</b>	<b>Introdução</b>	<b>17</b>
<b>1.2</b>	<b>Motivação</b>	<b>18</b>
<b>1.3</b>	<b>Objetivos</b>	<b>18</b>
<i>1.3.1</i>	<i>Objetivos Gerais</i>	<i>18</i>
<i>1.3.2</i>	<i>Objetivos Específicos</i>	<i>19</i>
<b>1.4</b>	<b>Produção Científica</b>	<b>19</b>
<b>1.5</b>	<b>Organização Geral da Dissertação</b>	<b>19</b>
<b>2</b>	<b>FUNDAMENTOS DE GERAÇÃO EÓLICA</b>	<b>21</b>
<b>2.1</b>	<b>Necessidade da Modelagem da Curva de Potência</b>	<b>21</b>
<i>2.1.1</i>	<i>Avaliação e Previsão da Energia Produzida</i>	<i>21</i>
<i>2.1.2</i>	<i>Estimativa do Fator de Capacidade</i>	<i>22</i>
<i>2.1.3</i>	<i>Seleção de Aerogeradores</i>	<i>22</i>
<i>2.1.4</i>	<i>Monitoramento Online de Curvas de Potência</i>	<i>22</i>
<b>2.2</b>	<b>Fundamentos Teóricos da Modelagem de um Aerogerador de Eixo Horizontal</b>	<b>23</b>
<b>2.3</b>	<b>Problemas de Modelagem</b>	<b>26</b>
<i>2.3.1</i>	<i>Diferença nos Modelos de Aerogeradores</i>	<i>26</i>
<i>2.3.2</i>	<i>Comportamento nas Velocidades de Corte</i>	<i>26</i>
<i>2.3.3</i>	<i>Aerogerador Único versus Grupo de Aerogeradores</i>	<i>27</i>
<i>2.3.4</i>	<i>Fatores de Influência</i>	<i>27</i>
<b>2.4</b>	<b>Aquisição de Dados para Modelagem</b>	<b>28</b>
<i>2.4.1</i>	<i>Análise Estatística da Velocidade do Vento</i>	<i>29</i>
<i>2.4.2</i>	<i>Levantamento da Curva de Potência</i>	<i>29</i>
<b>2.5</b>	<b>Conclusões</b>	<b>30</b>
<b>3</b>	<b>MODELOS DE ESTIMAÇÃO DA CURVA DE POTÊNCIA</b>	<b>32</b>
<b>3.1</b>	<b>Modelo de regressão polinomial</b>	<b>32</b>
<b>3.2</b>	<b>Modelo de regressão polinomial robusto</b>	<b>33</b>
<b>3.3</b>	<b>Modelos logísticos de 3, 4 e 5 parâmetros</b>	<b>37</b>
<b>3.4</b>	<b>Rede Perceptron Multicamadas (MLP)</b>	<b>38</b>

3.5	<b>Critério de informação</b>	42
3.6	<b>Conclusões</b>	43
4	<b>METODOLOGIAS PROPOSTAS</b>	44
4.1	<b>LASSO</b>	44
4.2	<b>PRISMA</b>	46
4.3	<b>Conclusões</b>	48
5	<b>SIMULAÇÃO E RESULTADOS</b>	49
5.1	<b>Conjunto de dados</b>	49
5.2	<b>Modelo PRISMA</b>	50
5.3	<b>Modelos polinomial e polinomial robusto</b>	51
5.4	<b>Modelo LASSO</b>	53
5.5	<b>Modelos logísticos</b>	54
5.6	<b>Modelo MLP</b>	56
5.7	<b>Comparação entre os modelos apresentados</b>	56
5.7.1	<i>SQR</i>	57
5.7.2	<i>AIC/BIC</i>	59
6	<b>CONCLUSÕES E TRABALHOS FUTUROS</b>	64
6.1	<b>Conclusões</b>	64
6.2	<b>Trabalhos Futuros</b>	64
	<b>REFERÊNCIAS</b>	66
	<b>APÊNDICES</b>	70
	<b>APÊNDICE A – Desenvolvimento das principais equações</b>	70
A.1	<b>Função de Densidade de Probabilidade (FDP) de <math>p</math></b>	70
A.2	<b>Método da Máxima Verossimilhança</b>	71
A.3	<b>Parametrização da FDP de <math>p</math></b>	71
	<b>APÊNDICE B – Códigos-fontes utilizados para...</b>	76
B.1	<b>Função de aptidão</b>	76
B.2	<b>Função AIC</b>	78
B.3	<b>Função BIC</b>	78
B.4	<b>Teste de parâmetros dos modelos polinomial e robusto</b>	79
B.5	<b>Teste de parâmetros do modelo LASSO</b>	81
B.6	<b>Teste de parâmetros do modelo Logístico</b>	84

<b>B.7</b>	<b>Teste de parâmetros do modelo MLP . . . . .</b>	<b>97</b>
------------	--	-----------

## 1 INTRODUÇÃO

Este capítulo se inicia com uma breve introdução a respeito da importância de gerar energia elétrica a partir de fontes renováveis. Dentre os principais tipos de produção de energia renovável, é enfatizada a de origem eólica, pois trata-se da base do nosso objeto de estudo. Partindo disso, é salientada a importância da modelagem de aerogeradores, bem como as principais dificuldades em sua aplicação, para que se tenha uma previsão adequada do fornecimento elétrico. E então, é apresentada, de forma sucinta, a nossa proposta de modelagem esparsa, a qual iremos abordar detalhadamente em capítulo *a posteriori*.

Além do supracitado, este capítulo também aborda a motivação da realização deste trabalho, quais os objetivos almejados e é anunciada a publicação, por hora, oriunda do estudo desenvolvido.

Por fim, é especificada como esta dissertação será organizada nos capítulos posteriores, sendo feita uma breve contextualização a respeito de cada um deles.

### 1.1 Introdução

Uma matriz energética que proporcione o mínimo possível de agressão ambiental tem sido de fundamental interesse para a maioria dos países desenvolvidos e em desenvolvimento no mundo. Assim, há uma forte tendência mundial para que a produção energética se volte a fontes renováveis, como a solar e a eólica.

Com o crescimento da indústria eólica, aerogeradores estão sendo instalados em diversos tipos de ambientes que se diferem, por exemplo, em relação à umidade relativa do ar, densidade do ar, altitude, condições climáticas sazonais e complexidade do terreno. Estas características, além de outras não mencionadas e outras nem conhecidas, influenciam significativamente a produção energética local. Em face de tais dificuldades, faz-se necessário uma modelagem adequada da relação entre a velocidade do vento e a potência gerada (conhecida genericamente como *curva de potência*) pelos aerogeradores, com o intuito de prever corretamente sua produção energética, bem como o monitorar o seu desempenho.

Dada sua importância, a modelagem (ou estimação) de curvas de potência é um tema que sempre despertou (e ainda desperta) o interesse da comunidade de energias renováveis, haja visto o grande número de artigos disponíveis na literatura especializada, principalmente na forma de surveys (MARCIUKAITIS *et al.*, 2017; WANG *et al.*, 2018; SOHONI *et al.*, 2016; LEE *et*

*al.*, 2015; SHOKRZADEH *et al.*, 2014; LYDIA *et al.*, 2014; CLIFTON *et al.*, 2013; LYDIA *et al.*, 2013; LI *et al.*, 2001).

Dentre os métodos encontrados na literatura, destacam-se o modelo polinomial, o polinomial ponderado, o polinomial esparso, o modelo logístico de 3, 4 e 5 parâmetros e redes neurais artificiais. Em particular, o modelo polinomial tem seu uso disseminado entre os praticantes da área devido não somente à sua formulação mais simples quando comparado a outros modelos, tais como redes neurais, mas principalmente porque encontra-se disponível em planilhas de cálculo populares (e.g. Excel e LibreOffice Calc).

Mesmo sendo de uso mais simples, o modelo polinomial requer a especificação *a priori* da ordem do polinômio. A especificação correta da ordem do polinômio é essencial para um bom desempenho preditivo do modelo, bem como para evitar problemas numéricos. Uma vez escolhida a ordem, um método de estimação de parâmetros (geralmente, mínimos quadrados ordinários) é usado para estimar os coeficientes associados a cada termo do polinômio. No modelo polinomial clássico, o número de coeficientes a estimar é igual à ordem do polinômio mais 1.

## **1.2 Motivação**

Tendo em mente as dificuldades supracitadas, revisitaremos o modelo polinomial convencional a fim de propor uma metodologia de projeto baseada em computação evolucionária que retorna a ordem ótima do modelo polinomial, bem como seleciona os termos relevantes do polinômio e elimina os demais. A metodologia proposta é avaliada em um amplo estudo usando dados reais e comparada com métodos que compõem o estado da arte em estimação da curva de potência de aerogeradores. Os resultados indicam um desempenho consistentemente superior da metodologia proposta.

## **1.3 Objetivos**

### ***1.3.1 Objetivos Gerais***

Revisitar o projeto convencional do modelo polinomial amplamente usado na tarefa de estimação de curvas de potência de aerogeradores a fim de conferir-lhe automaticidade e esparsidade.

### 1.3.2 *Objetivos Específicos*

Para alcançar o objetivo geral, foram feitos:

1. Realizar um amplo levantamento bibliográfico do estado da arte em estimação da curva de potência.
2. Avaliação da aplicabilidade do método de regularização *Least Absolute Shrinkage and Selection Operator* (LASSO) no projeto automático de modelos polinomiais para estimação da curva de potência de aerogeradores.
3. Proposta de um método alternativo ao LASSO baseado em otimização metaheurística, particularmente algoritmos genéticos.
4. Disponibilização de uma toolbox ao estilo Matlab/Octave contendo as implementações das diversas técnicas de estimação de curva de potência estudadas.

## 1.4 **Produção Científica**

Tivemos as seguintes produções acadêmicas, sob a autoria de Maya, H. C. e Barreto, G. A., oriundas do estudo desenvolvido nesta dissertação:

- Submetido, e aceito, para ser apresentado no “2nd International Conference on Renewable Energy and Resources”, em Boston, sob o título “A survey and some new results on machine learning methods for the estimation of the power curve of wind turbines”.
- Submetido ao “XV Encontro Nacional de Inteligência Artificial e Computacional”, em São Paulo, sob o título "A GA-Based Approach for Building Regularized Sparse Polynomial Models for Wind Turbine Power Curves".

## 1.5 **Organização Geral da Dissertação**

O restante desta dissertação está organizada na seguinte formatação:

No capítulo 2 são feitas as fundamentações teóricas a respeito da modelagem da curva de potência de aerogeradores, tais como o estudo sobre as influências mecânicas do vento em aerogeradores de eixo horizontal, a transformação da energia mecânica em elétrica nas condições apresentadas e a análise das distribuições estatísticas pertinentes ao desenvolvimento de uma modelagem coerente.

No capítulo 3 são apresentados outros modelos de regressão utilizados para descrever

a geração da energia elétrica de aerogeradores a partir da velocidade do vento. A justificativa para o uso de cada um dos modelos no experimento de comparação também é desenvolvida em cada subseção pertinente, bem como o conceito e algoritmo de cada qual. Também apresentamos aqui critérios de avaliação do desempenho de regressões.

No capítulo 4 são apresentadas duas propostas alternativas para modelagem de aerogeradores.

No capítulo 5 são apresentados os resultados de testes comparativos entre todos os modelos elencados neste trabalho, bem como as conclusões oriundas de suas interpretações.

Por fim, no capítulo 6, são feitas as considerações finais e as perspectivas para trabalhos futuros.

## 2 FUNDAMENTOS DE GERAÇÃO EÓLICA

O objetivo principal proposto nesta dissertação é desenvolver modelos matemáticos esparsos, simples e eficazes que representem a transformação energética performada por um aerogerador. Para tal, este capítulo destina-se a fazer um levantamento teórico sobre os pontos de principal relevância para a compreensão da problemática que leve a conclusão do objetivo almejado.

### 2.1 Necessidade da Modelagem da Curva de Potência

A curva de potência reflete a resposta da potência elétrica gerada por um aerogerador em função da velocidade de vento local. Elas são fornecidas pelos fabricantes em forma tabular ou gráfica. No entanto, essas curvas são criadas sob condições padronizadas, de forma que podem não representar adequadamente as condições realistas do local em que o aerogerador está instalado. O desempenho do aerogerador nos parques eólicos também não é ideal devido ao seu desgaste e envelhecimento. Dessa forma, a derivação de uma função apropriada para descrever a forma real da curva de potência local é necessária em sistemas de energia eólica.

Modelos precisos dessas curvas são úteis para diversas aplicações em parques eólicos. Os principais são elencados a seguir (SOHONI *et al.*, 2016).

#### 2.1.1 Avaliação e Previsão da Energia Produzida

A avaliação dos recursos eólicos de uma região é feita para identificar áreas adequadas para a instalação de parques eólicos (MATHEW, 2006). Nesse processo, a estimativa da energia é feita usando os dados do vento disponíveis e curva de potência de aerogeradores. Prever a potência de saída do aerogerador em um local candidato também é necessário no dimensionamento e estudos de otimização de custos durante o estágio de projeto de um sistema energético eólico. A precisão na previsão de potência é importante, pois uma superestimação pode resultar em baixa confiabilidade e uma subestimação pode levar ao sobredimensionamento do sistema. Operadores de aerogeradores que comercializam energia diretamente para o mercado de eletricidade também especificam a potência de saída de seus aerogeradores com alta acurácia, de modo que eles sejam capazes de entregar a quantidade correta de energia comercializada (SCHLECHTINGEN *et al.*, 2013).

### **2.1.2 Estimativa do Fator de Capacidade**

O fator de capacidade de um aerogerador é um indicador de sua eficiência definido pela razão entre a potência média de saída e a potência nominal de saída do gerador (JOWDER, 2009). Essa informação é essencial para estudos de dimensionamento e otimização de custos, definir qual o aerogerador ideal para uma determinada região e classificar locais em potencial para produção de energia eólica (JOWDER, 2009; POWELL, 1981). Modelos de curva de potência do aerogerador e a estimação da velocidade de vento média local são usados para estimar o fator de capacidade de um aerogerador. Uma análise comparativa de quatro métodos de modelagem de curva de potência na estimativa do fator de capacidade de um aerogerador é apresentada em (CHANG *et al.*, 2014).

### **2.1.3 Seleção de Aerogeradores**

A curva de potência pode ser usada para comparar genericamente os modelos e pode ajudar na escolha do aerogerador a partir das opções disponíveis. A seleção das características do aerogerador que combinam com o regime de ventos do local ajuda a otimizar a eficiência do sistema de energia eólica (USTUNTAS; SAHIN, 2008).

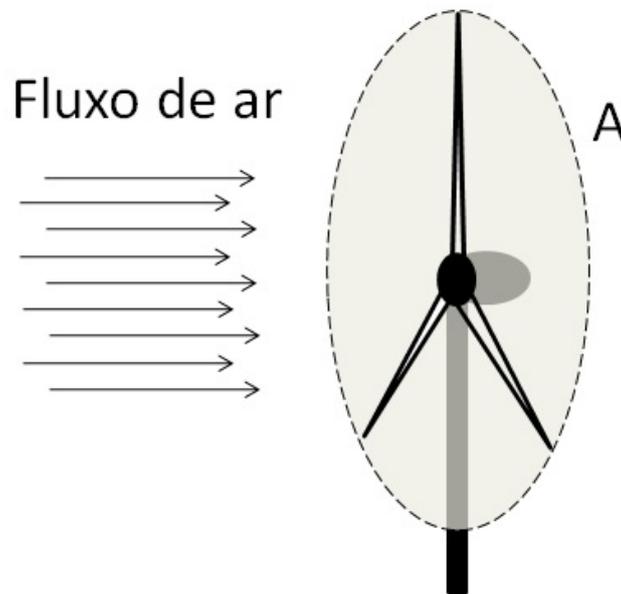
### **2.1.4 Monitoramento Online de Curvas de Potência**

Curvas de potência podem ser usadas para monitorar o desempenho de aerogeradores. Para tal, é necessária uma curva de referência que represente o desempenho esperado desse aerogerador de forma que se possa comparar com a curva real gerada *in loco*. Desvios dos valores reais da saída esperada podem indicar mau desempenho ou falhas (KUSIAK *et al.*, 2009). Diferentes tipos de falhas afetam o sistema do aerogerador diferentemente, e causarão o desvio da curva de potência em relação valor esperado de maneiras diferentes. Ferramentas que podem caracterizar e quantificar esses desvios podem auxiliar na identificação precoce de falhas e a análise estatística de *outlier* podem dar indicações da razão específica da anomalia. Monitoramento das condições do aerogerador pelo uso da modelagem de cópula da curva de potência é sugerida em (STEPHEN *et al.*, 2011; GILL; GALLOWAY, 2012). O reconhecimento precoce das falhas emergentes no equipamento, bem como seu reparo e manutenção em tempo oportuno, podem ajudar significativamente na melhoria do desempenho de aerogeradores.

## 2.2 Fundamentos Teóricos da Modelagem de um Aerogerador de Eixo Horizontal

Seja um aerogerador de eixo horizontal qualquer. Denotaremos por  $A$  a área circular (em  $m^2$ ) varrida por suas pás, tal como esboçado na Figura 1. Considere que o ar do local onde o aerogerador está instalado tenha densidade  $\rho$  constante (em  $kg/m^3$ ) e que a componente de velocidade do vento (em  $m/s$ ) que atravessar perpendicularmente a área  $A$  é constante e igual a  $v$ .

Figura 1 – Fluxo de ar através da área circular de varredura das pás do aerogerador.



Fonte: o autor.

Segundo a mecânica clássica, sabe-se que a potência  $P$  (em Watts) pode ser calculada como

$$P = \frac{\Delta E_c}{\Delta t}, \quad (2.1)$$

em que  $t$  é a medida do tempo (em segundos) e  $E_c$  é a energia cinética do movimento (em Joules). Ademais, sabe-se que a energia cinética é dada por

$$E_c = \frac{1}{2}Mv^2, \quad (2.2)$$

em que  $M$  é a massa (em  $kg$ ). Com  $v$  constante, pode-se escrever a variação da energia cinética como

$$\Delta E_c = \frac{1}{2}\Delta Mv^2. \quad (2.3)$$

Portanto, a potência eólica disponível associada a  $v$  em uma área  $A$ , denotada doravante por  $P_w$ , equivale a energia cinética por unidade de tempo e pode ser obtida por

$$\begin{aligned} P_w &= \frac{\Delta E_c}{\Delta t} \\ &= \frac{1}{2} \frac{\Delta M}{\Delta t} v^2 \\ &= \frac{1}{2} (\rho A v) v^2 \\ &= \frac{1}{2} \rho A v^3. \end{aligned} \quad (2.4)$$

Porém, apenas uma parte da potência eólica disponível será aproveitada para o movimento do rotor, sendo esta chamada de potência absorvida ( $P_a$ ).

O coeficiente de potência,  $C_p$ , definido por

$$C_p = \frac{P_a}{P_w}, \quad (2.5)$$

é uma grandeza adimensional que caracteriza a capacidade de um aerogerador absorver a potência eólica disponível (SHOKRZADEH *et al.*, 2014; CLIFTON *et al.*, 2013).

O valor do coeficiente de potência, que pode ser obtido a partir dos dados do fabricante, é influenciado por aspectos morfológicos e aerodinâmicos do aerogerador, como por exemplo o desenho de suas pás. O seu máximo teórico, conhecido como limite de Betz, é de  $16/27 \approx 0,5926$  (BETZ, 1920). Ademais, ainda deve-se contar com a eficiência do rotor,  $\eta$ , para se obter a potência elétrica gerada a partir do fluxo de vento, chegando assim à seguinte equação teórica:

$$p = \frac{1}{2} \eta C_p \rho A v^3. \quad (2.6)$$

Como a potência elétrica gerada será a principal potência analisada neste trabalho, a denotaremos simplesmente por  $p$ , sem nenhum sufixo que a identifique como tal e em caixa baixa, já que ao utilizá-la para desenvolver equações na forma matricial, o que ocorrerá em capítulos à frente, isso torna mais claro que se trata de um valor escalar.

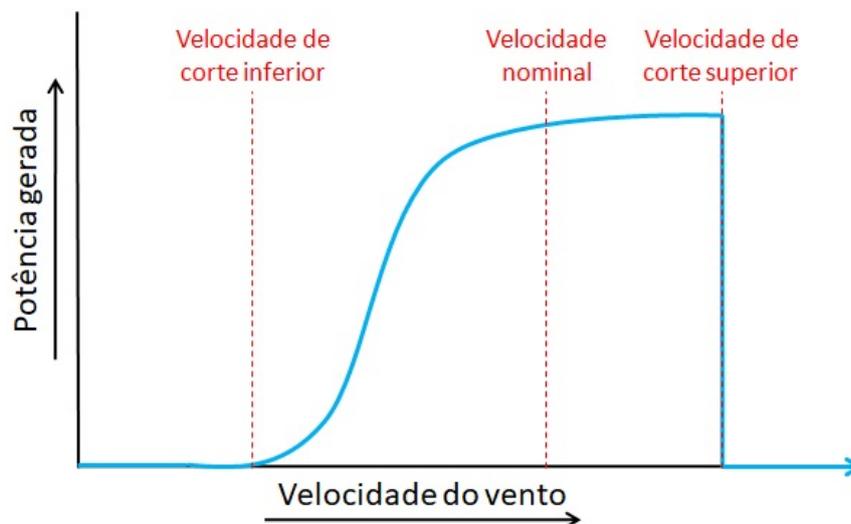
Aerogeradores reais são projetados para fornecer, no máximo, a sua potência nominal, podendo sofrer avarias caso a exceda. A potência elétrica fornecida tem comportamento não linear proporcional ao cubo da velocidade do vento para baixas velocidades, semelhante ao desenvolvimento teórico previamente apresentado. Porém, a medida que se aproxima da potência nominal, os aerogeradores comerciais dispõem de meios para limitar a energia eólica absorvida e, conseqüentemente, controlar a potência fornecida para que não exceda a potência nominal do rotor (MANWELL *et al.*, 2009).

Existem vários métodos que visam controlar a potência absorvida pelo rotor, dentre os quais podemos destacar dois como os principais utilizados atualmente:

- *controle do ângulo de passo*, que consiste no controle ativo do ângulo de ataque das pás do aerogerador. Dessa forma, pode-se alterar o valor do seu  $C_p$ , que influencia na potência eólica absorvida; e
- *controle por estol*, que é um tipo de controle passivo, que consiste em projetar as pás de forma que o efeito de estol<sup>1</sup> ocorra em velocidades superiores à velocidade eólica nominal.

Além disso, há uma velocidade de vento mínima para o funcionamento do aerogerador, conhecida como velocidade de corte inferior (*cut-in speed*), na qual a potência gerada é zero para velocidades iguais ou inferiores. Bem como há um limite de funcionamento superior, conhecido como velocidade de corte superior (*cut-out speed*), em que a potência gerada também é zero para velocidades iguais ou superiores, pois o funcionamento do aerogerador é interrompido com a finalidade de proteger a integridade de seus componentes por conta da alta força do vento (SOHONI *et al.*, 2016). Um gráfico típico da curva de potência de um aerogerador comercial com controle do ângulo de passo está mostrado na Figura 2, em que fica evidenciado o comportamento não linear da curva de potência.

Figura 2 – Curva de potência típica de uma aerogerador com controle de potência por ângulo de passo.



Fonte: o autor.

<sup>1</sup> Este efeito ocorre quando o fluxo de ar na superfície da pá deixa, abruptamente, de fixar-se à mesma, passando a girar de forma turbulenta, diminuindo assim a força de sustentação e aumentando a força de arrasto.

## 2.3 Problemas de Modelagem

Alguns aspectos requerem atenção durante a modelagem das curvas de potência dos aerogeradores. A seleção do modelo e a metodologia adotada depende do objetivo da modelagem, dos dados disponíveis, do impacto de vários parâmetros nessas curvas e de outras questões relacionadas. Os seguintes aspectos importantes devem ser considerados durante a modelagem da curva de potência (SOHONI *et al.*, 2016).

### 2.3.1 Diferença nos Modelos de Aerogeradores

As curvas de potência variam com diferentes fabricantes e modelos de aerogeradores. Portanto, o modelo de regressão utilizado para descrevê-los também deve se adequar as suas condições de funcionamento (DIAF *et al.*, 2008). Por exemplo, há diferença entre aerogeradores regulados por ângulo de passo e aerogeradores regulados por estol. Os controlados por ângulo de passo mantêm a produção constante da velocidade nominal até a velocidade de corte superior, enquanto o aerogerador regulados por estol possui uma redução da potência de saída acima da velocidade nominal do vento.

### 2.3.2 Comportamento nas Velocidades de Corte

O comportamento do aerogerador perto de velocidades de corte superior e inferior pode ser difícil de modelar (USTUNTAS; SAHIN, 2008). Estes limites são diferentes para diferentes modelos de aerogerador. Quando a curva de potência é derivada usando os dados medidos, alguns valores negativos e não-nulos de potências abaixo velocidade de corte inferior podem ser obtidos. Além disso, a histerese de velocidade de corte superior que ocorre durante o período entre o desligamento e a reinicialização do aerogerador afeta a produtividade da mesma (HORVATH *et al.*, 2007). Efeitos da histerese podem ser mais significativos com certos padrões de vento e terrenos, como, por exemplo, ventos instáveis e tempestuosos, que exigem frequentes partida e parada, resultando em perda considerável de produção de energia. A correção da curva de potência que leva em conta este comportamento do aerogerador na velocidade de corte superior pode reduzir erros de previsão de potência.

### 2.3.3 *Aerogerador Único versus Grupo de Aerogeradores*

As curvas fornecidas pelos fabricantes são adequadas para prever a potência de saída de um único aerogerador de um tipo específico. Em um grande parque eólico, alguns aerogeradores estão espalhados por uma área ampla. A produção de energia eólica envolve incertezas devido à natureza estocástica do vento e da variação da curva de potência (JIN; TIAN, 2010). A velocidade e direção do vento encontrados pelos aerogeradores em um parque eólico podem não ser os mesmos devido à variação do vento. Consequentemente, em um parque eólico, a energia produzida por aerogeradores com especificações idênticas também pode diferir, mesmo se a velocidade do vento for a mesma. O efeito de sombreamento dos aerogeradores causa essa diferença, como os aerogeradores que operam em posição sequencial com outros aerogeradores, que podem obter velocidades de vento reduzidas (MCLEAN, 2008). Essa diferença também pode acontecer devido a fatores como desgaste, envelhecimento e sujeira ou depósito de gelo nas lâminas. Com o crescimento dos projetos de energia eólica, tornou-se essencial o desenvolvimento de métodos para monitorar o desempenho não só de um único aerogerador, mas também o parque eólico como um todo. Portanto, desenvolver modelos para obter a relação entre velocidade do vento e potência de saída quando um grupo de aerogeradores é implantado em um parque eólico é necessário.

### 2.3.4 *Fatores de Influência*

Vários fatores podem causar o desvio da curva de potência do valor teórico (KUSIAK *et al.*, 2009; SCHLECHTINGEN *et al.*, 2013). Os fatores de influência mais importantes são mostrados a seguir e precisam atenção durante a modelagem.

1. **Condições de vento no local.** O vento é altamente estocástico por natureza. A velocidade e direção do vento mudam continuamente. O vento em um determinado local é afetado por fenômenos climáticos e topologia da região. A turbulência do vento em um dado local afeta a produção de energia (HEDEVANG, 2014). Obstáculos como árvores, edifícios e outras estruturas altas influenciam o vento.
2. **Densidade do Ar.** A pressão, temperatura e umidade local afetam a densidade do ar, afetando, portanto, a potência produzida. O efeito da variação da densidade do ar foi considerado para o desenvolvimento de curvas específicas em (OLAOFE;

FOLLY, 2013), por exemplo. É mostrado em (SCHLECHTINGEN *et al.*, 2013) que a temperatura tem a maior influência na densidade do ar e a consideração de seu efeito junto com a direção do vento resultou no melhor desempenho dos modelos.

3. **Extrapolação da Velocidade do Vento.** A velocidade do vento muda com a altura. Esse efeito de cisalhamento do vento é afetado pela rugosidade de terreno. A curva de potência usa a velocidade do vento medida à altura do cubo do aerogerador, mas essa altura varia com diferentes modelos e fabricantes, e nem sempre é possível medir a velocidade do vento a esta altura. Alguns métodos têm sido utilizados na literatura para expressar a variação de velocidade do vento com altura (DIAF *et al.*, 2008). Além disso, a velocidade do vento medida nos mastros é diferente da velocidade no local onde o aerogeradores se encontra e, às vezes quando os valores de velocidade do vento neste local específico não estão disponíveis, as medições de velocidade do vento em uma localização próxima são usadas para a determinação do perfil do vento. A precisão da conversão da velocidade do vento medido para a velocidade do vento na altura do cubo e na localização do aerogerador depende de fatores como o perfil vertical do vento no local, posição dos mastros em relação ao aerogerador, e o método utilizado para extrapolação.
4. **Condição do Aerogerador.** A curva de potência é afetada pela condição do aerogerador e equipamento associado. Envelhecimento e desgaste do aerogerador, anomalias e falhas, condições das pás, configurações do controlador e assim por diante, fazem com que a curva de potência se desvie dos valores reais (KUSIAK *et al.*, 2009; GILL; GALLOWAY, 2012).

## 2.4 Aquisição de Dados para Modelagem

Os dados necessários para modelar uma curva de potência são a velocidade do vento e a potência de saída. Esses dados devem ser medidos muitas vezes em intervalos periódicos de tempo. Uma vez que os dados necessários sejam coletados, a produção de energia do aerogerador pode ser analisada usando quatro abordagens distintas, ou seja, o uso direto da média dos dados de um curto intervalo de tempo, o método de caixas, o desenvolvimento da velocidade e curvas de potência a partir dos dados e a análise estatística usando medidas sumárias (MANWELL *et*

al., 2009).

#### 2.4.1 Análise Estatística da Velocidade do Vento

A distribuição de probabilidade da velocidade do vento descreve a possibilidade de ocorrência de alguns valores da velocidade do vento. As distribuições de probabilidade são geralmente caracterizadas pela função densidade de probabilidade  $f(v)$  ou pela função de densidade acumulada  $F(v)$  (MANWELL *et al.*, 2009). As duas distribuições de probabilidade comumente usadas na análise de dados de vento são a distribuição de Rayleigh e Weibull. A distribuição de Rayleigh requer apenas o conhecimento da velocidade média do vento  $\bar{v}$  e, portanto, é a distribuição de probabilidade de velocidade mais simples. O  $f(v)$  e  $F(v)$  da distribuição de Rayleigh é dada a seguir:

$$f(v) = \frac{\pi}{2} \left( \frac{v}{\bar{v}^2} \right) \exp \left[ -\frac{\pi}{4} \left( \frac{v}{\bar{v}} \right)^2 \right] \quad (2.7)$$

$$F(v) = 1 - \exp \left[ -\frac{\pi}{4} \left( \frac{v}{\bar{v}} \right)^2 \right] \quad (2.8)$$

E as funções  $f(v)$  e  $F(v)$  da distribuição de Weibull são apresentadas a seguir

$$f(v) = \left( \frac{k}{c} \right) \left( \frac{v}{c} \right)^{k-1} \exp \left[ -\left( \frac{v}{c} \right)^k \right] \quad (2.9)$$

$$F(v) = 1 - \exp \left[ -\left( \frac{v}{c} \right)^k \right] \quad (2.10)$$

onde  $k$  é o fator de forma e  $c$  é o fator de escala. Quanto maior o valor de  $k$ , menor é a variação da velocidade do vento observada.

#### 2.4.2 Levantamento da Curva de Potência

O IEC 61400-12-1 (IEC, 2005), é o padrão internacional comumente adotado para avaliar o desempenho de produção energética de aerogeradores. A norma descreve a metodologia de medição da curva de potência, que é determinada pela medição simultânea da velocidade do vento no local do teste e da potência eólica gerada. Uma calibração prévia do local é necessária

para determinadas condições do terreno. A produção anual de energia é calculada aplicando à curva de potência as distribuições de frequência da velocidade do vento complementadas por fontes de incerteza e seus efeitos. A norma prescreve a derivação da curva de potência usando a velocidade do vento na altura do cubo e sendo medida com um anemômetro de copo, mas se a velocidade do vento tiver uma grande variação sobre a área varrida pelo rotor, pode haver uma diferença significativa entre a velocidade do vento na altura do cubo e a velocidade do vento média em toda a área de varredura do rotor. Os métodos de medição e a precisão dos instrumentos de medição podem causar variações nas medições e podem levar a grandes erros de previsão.

O padrão IEC usa dados médios de dez minutos agrupados em intervalos de velocidade do vento de  $0,5 \text{ m/s}$  (método de caixas). O vento em um local específico pode ser afetado por vários fatores, como a topologia do local, os obstáculos e fenômenos climáticos. A curva de potência do IEC considera a condição de vento do local atual. Portanto, pode nem sempre ser apropriado aplicar-se às condições de vento de outros locais. Portanto, esforços de pesquisa são necessários para desenvolver curvas de energia específicas do local. Estas curvas podem incorporar as condições de vento do local específico, dando melhores resultados (OLAOFÉ; FOLLY, 2013; IEC, 2005).

A seleção apropriada do método de modelagem é um requisito importante para o planejamento e operação do sistema baseado em energia eólica. Os métodos que consideram apenas a velocidade do vento como entrada podem não levar em consideração a variação causada por vários parâmetros de influência. Métodos que consideram a influência desses parâmetros na curva de potência podem resultar em modelos mais precisos. O vento em um local específico pode ser afetado por vários fatores, como a topologia do local e os obstáculos e fenômenos climáticos. É mostrado em (OLAOFÉ; FOLLY, 2013) que o uso de curvas de potência de locais desenvolvidos, que usaram o conhecimento de parâmetros de aerogeradores para modelagem, resultou em uma avaliação de energia mais precisa do que a curva de potência do aerogerador. Os problemas discutidos acima, se abordados adequadamente, podem resultar em modelos eficientes de curvas de potência.

## 2.5 Conclusões

Neste capítulo foi apresentado inicialmente a importância da modelagem da curva de potência de um aerogerador. Isso é a base de estudo para prever e planejar a produção energética

de parques eólicos, monitorar o bom funcionamento do aerogerador, dentre outros.

Na sequência foi feito um estudo do comportamento físico teórico a respeito da transformação da energia cinética eólica em energia elétrica através de um aerogerador de eixo horizontal. Foi considerada a mecânica clássica e as limitações físicas mensuráveis conhecidas para que seja desenvolvido este esboço teórico. Com isso, chegamos a uma função que relaciona a velocidade do vento com a potência elétrica gerada. O gráfico dessa função é conhecido como curva de potência do aerogerador.

Em seguida foram apresentados os principais desafios para desenvolver um bom modelo para a curva de potência. Que podem ocasionar distorções significativas no formato da curva de potência a depender, por exemplo, de características ambientais ou estruturais do aerogerador.

Por fim, foram apresentadas as metodologias de aquisição dos dados que são utilizados para desenvolver a curva de potência de um aerogerador. Iniciou-se falando sobre a análise estatística da velocidade do vento e então findou na norma IEC 61400-12-1, comumente utilizada por produtores de aerogeradores para formar a curva de potência.

Ao passo desses conceitos preambulares sobre a natureza da problemática terem sido apresentados, o capítulo seguinte tratará de apresentar modelos comumente utilizados para regressão da curva de potência do aerogerador e também faremos uma abordagem sobre os critérios de otimização que serão utilizados para comparação entre os modelos apresentados.

### 3 MODELOS DE ESTIMAÇÃO DA CURVA DE POTÊNCIA

Existe uma infinidade de métodos regressivos para estimação da curva de potência, cada qual com suas características que os tornam mais ou menos apropriados para uma determinada aplicação. Nesta seção são apresentados modelos bastante utilizados na literatura científica atual, bem como seu princípio matemático e a justificativa para utilizá-los. O intuito disso é de avaliá-los e compará-los com o novo modelo de regressão polinomial esparso proposto nessa dissertação.

#### 3.1 Modelo de regressão polinomial

Regressão polinomial muito provavelmente é a técnica mais utilizada na prática para estimar a curva de potência de aerogeradores (LYDIA *et al.*, 2014; SHOKRZADEH *et al.*, 2014). Este modelo tenta ajustar um polinômio de ordem  $k - 1$  a um conjunto de  $n$  observações do par entrada-saída  $\{(p_i, v_i)\}_{i=1}^n$ , sendo  $p_i$  a  $i$ -ésima medição da potência elétrica gerada e  $v_i$  a velocidade do vento correspondente.

Do ponto de vista da modelagem estocástica, o ajuste do polinômio aos dados assume que o processo gerador dos dados pode ser descrito pela seguinte expressão:

$$p_i = \beta_1 + \beta_2 v_i + \beta_3 v_i^2 + \dots + \beta_k v_i^{k-1} + \varepsilon_i, \quad (3.1)$$

em que  $\beta_j, j = 1 \dots k$ , são os coeficientes do polinômio e  $\varepsilon_i$  é uma variável aleatória normal, com média nula e variância  $\sigma_\varepsilon^2$ .

Pode-se escrever a Eq. 3.1 na forma matricial como

$$p = \mathbf{V}\beta + \varepsilon, \quad (3.2)$$

em que  $p = [p_1 \ p_2 \ \dots \ p_n]^T$ ,  $\beta = [\beta_1 \ \beta_2 \ \dots \ \beta_k]^T$ ,  $\varepsilon = [\varepsilon_1 \ \varepsilon_2 \ \dots \ \varepsilon_n]^T$ , com a matriz  $V$ , de dimensão  $n \times k$ , sendo definida como

$$V = \begin{bmatrix} 1 & v_1 & \dots & v_1^{k-1} \\ 1 & v_2 & \dots & v_2^{k-1} \\ \vdots & \vdots & & \vdots \\ 1 & v_n & \dots & v_n^{k-1} \end{bmatrix}.$$

A estimativa de  $\beta$  é obtida via método *Ordinary Least Squares* (OLS), que consistem em minimizar o funcional  $J(\beta) = \|\varepsilon\|^2 = \varepsilon^T \varepsilon = \sum_{i=1}^n \varepsilon_i^2$ , cuja derivada em relação a  $\beta$  é dada

por

$$\begin{aligned}\frac{\partial J(\beta)}{\partial \beta} &= \frac{\partial \left[ (p - V\beta)^T (p - V\beta) \right]}{\partial \beta} \\ &= -2\mathbf{V}^T (p - V\beta).\end{aligned}\tag{3.3}$$

E a segunda derivada em relação a  $\beta$  é dada por

$$\begin{aligned}\frac{\partial^2 J(\beta)}{\partial \beta^2} &= \frac{\partial \left[ -2\mathbf{V}^T (p - V\beta) \right]}{\partial \beta} \\ &= 2\mathbf{V}^T V.\end{aligned}\tag{3.4}$$

Perceba que  $v_i \geq 0, \forall i \in \{1, \dots, n\}$ , pois a velocidade de vento mensurada sempre é positiva. Então podemos assumir que  $V^T V \succeq 0$ . Consequentemente, podemos afirmar que  $J(\beta)$  tem a concavidade voltada para cima, portanto há um valor de mínimo global para a função. Igualando a Eq. 3.3 ao vetor de zeros de dimensão  $k$  chega-se à equação do estimador OLS para os parâmetros do modelo polinomial:

$$\begin{aligned}-2\mathbf{V}^T (p - V\hat{\beta}) &= 0 \\ \therefore \mathbf{V}^T (p - V\hat{\beta}) &= 0 \\ \therefore \mathbf{V}^T V\hat{\beta} &= \mathbf{V}^T p \\ \therefore \hat{\beta} &= (V^T V)^{-1} V^T p.\end{aligned}\tag{3.5}$$

em que  $\hat{\beta}$  é uma estimaco de  $\beta$  e  $\mathbf{0}$  é um vetor de zeros com dimenso  $k$ .

### 3.2 Modelo de regresso polinomial robusto

Os dados para estimaco de curva de potncia costumam ser obtidos via sistemas SCADA<sup>1</sup>, estando assim susceptveis a erros de medio, falha em sensores e erros no sistema de comunicao (SOHONI *et al.*, 2016; SHOKRZADEH *et al.*, 2014). Os dados tambm so afetados por aerogeradores que no esto em produo pois foram desligadas pelo sistema de controle por outra razo que no a operao anmala. Assim, sistemas SCADA podem ter entradas nulas ou dados errneos, resultando em modelos inexatos da curva de potncia.

Pode-se optar por uma inspeco visual dos dados e a remoo manual de observaes supostamente anmalas antes de realizar a estimaco de parmetros do modelo polinomial. Contudo, a remoo manual pode levar a modelos inexatos uma vez que o volume de dados

<sup>1</sup> *Supervisory control and data acquisition.*

oriundos de sistemas SCADA é muito grande, tornando difícil a diferenciação entre observações corretas e anômalas.

Uma alternativa é usar alguma técnica de estimação de parâmetros que seja inerentemente robusta a *outliers*<sup>2</sup>, tal como *estimadores-M* (HUBER, 1964; BARRETO; BARROS, 2015). O princípio dessa técnica consiste em atribuir um componente multiplicativo de peso a cada dado amostral em função do seu residual. Esse peso tem como finalidade minimizar a influência de *outliers* tanto quanto possível (RUCKSTUHL, 2016). Portanto, considere a seguinte estimação, conhecida como Weighted Least Squares (WLS), que atribui um peso indexado para cada dado amostral

$$\begin{aligned} J(\beta) &= \sum_{i=1}^n w_i \left( p_i - \sum_{j=1}^k \beta_j v_i^{j-1} \right)^2 \\ &= \sum_{i=1}^n w_i (p_i - v_i^\top \beta)^2 \\ &= \sum_{i=1}^n w_i \varepsilon_i^2, \end{aligned} \quad (3.6)$$

em que  $v_i^\top = [1 \quad v_i \quad v_i^2 \quad \dots \quad v_i^{k-1}]$  e  $w_i$  é o componente de ponderação para o dado amostral de índice  $i$ . A Eq. 3.6 pode ser escrita na forma matricial como

$$J(\beta) = (p - V\beta)^\top W (p - V\beta), \quad (3.7)$$

em que  $W$  é uma matriz diagonal  $n \times n$  com  $W_{ii} = w_i$ . O mínimo de para o funcional da Eq. 3.7, caso exista, pode ser obtido através da sua primeira derivada em relação a  $\beta$

$$\begin{aligned} \frac{\partial J(\beta)}{\partial \beta} &= -2V^\top W (p - V\hat{\beta}) = 0 \\ \therefore V^\top W (p - V\hat{\beta}) &= 0 \\ \therefore V^\top W V \hat{\beta} &= V^\top W p \\ \therefore \hat{\beta} &= (V^\top W V)^{-1} V^\top W p, \end{aligned} \quad (3.8)$$

A segunda derivada em relação a  $\beta$ , dada por

$$\frac{\partial^2 J(\beta)}{\partial \beta^2} = 2V^\top W V, \quad (3.9)$$

pode nos informar a respeito da concavidade da função custo. Como visto na sessão anterior,  $v_i \geq 0 \forall i \in \{1, \dots, n\}$ , portanto a concavidade da Eq. 3.7 depende do valor de  $w_i$ .

<sup>2</sup> Uma observação que apresenta um grande afastamento das demais da série, ou que apresente alguma inconsistência, é conhecida como *outlier*.

Até então obtemos um método de otimização dos parâmetros  $\beta$ . Entretanto, além desses, faz-se necessário ajustar apropriadamente cada  $w_i$  de forma a mitigar a influência dos *outliers* como mencionado anteriormente. Para tal, é desenvolvida a problemática sob a ótica do estimador-M, que define-se como a solução que minimiza a função objetivo  $J(\beta)$  definida por (SUSANTI *et al.*, 2014; RUCKSTUHL, 2016)

$$J(\beta) = \sum_{i=1}^n \Psi \left( \frac{p_i - v_i^T \beta}{\sigma_\varepsilon} \right), \quad (3.10)$$

em que  $\Psi(\cdot) : \mathbb{R} \rightarrow \mathbb{R}$  é uma função contínua, derivável em  $\mathbb{R}$  e com um mínimo global.

A normalização do residual por seu desvio padrão,  $\sigma_\varepsilon$ , é feita com o intuito de assegurar a invariância em relação à escala utilizada. A estimação do desvio padrão da população a partir de uma amostra pode ser feita de forma robusta através da *Median Absolute Derivation* (MAD) (ROUSSEEUW; CROUX, 1993)

$$\hat{\sigma} = \gamma \cdot MAD \quad (3.11)$$

em que  $\gamma$  é uma constante dependente da distribuição. Para o caso de distribuições normais, ao qual se enquadram as aplicações desenvolvidas neste trabalho, temos  $\gamma = 1,4826$ .

A derivada da Eq. 3.10 em relação a  $\beta$  é

$$\begin{aligned} \frac{\partial J(\beta)}{\partial \beta} &= \sum_{i=1}^n \left[ -\frac{1}{\sigma_\varepsilon} \psi \left( \frac{p_i - v_i^T \beta}{\sigma_\varepsilon} \right) v_i \right] \\ &\therefore \sum_{i=1}^n \left[ \psi \left( \frac{p_i - v_i^T \beta}{\sigma_\varepsilon} \right) v_i \right] \\ &\therefore \sum_{i=1}^n \left[ \psi \left( \frac{\varepsilon_i}{\sigma_\varepsilon} \right) v_i \right], \end{aligned} \quad (3.12)$$

em que  $\psi$  é derivada de  $\Psi$ . A finalidade da função  $\psi$  é de atribuir um peso em relação ao residual, de sorte que há várias funções que podem caracterizá-la. Para o trabalho desenvolvido, foi utilizada a função bponderada de Tukey, definida por

$$\psi(u_i) = \begin{cases} \varepsilon_i \left[ 1 - \left( \frac{u_i}{c} \right)^2 \right]^2, & |u_i| \leq c \\ 0, & |u_i| > c \end{cases}, \quad (3.13)$$

em que  $u_i = \frac{\varepsilon_i}{\sigma_\varepsilon}$ , e  $c$  é um limiar, de tal forma que todo valor absoluto de  $u_i$  que for superior a  $c$  é considerado como *outlier*. Utilizamos a constante limitadora  $c = 4,685$ , o que proporciona 95% da eficiência do estimador de regressão.

Igualando as derivadas das funções custos WLS com a do estimador-M, obtemos

$$\sum_{i=1}^n [\psi(u_i) v_i] = \sum_{i=1}^n w_i e_i v_i \therefore \quad (3.14)$$

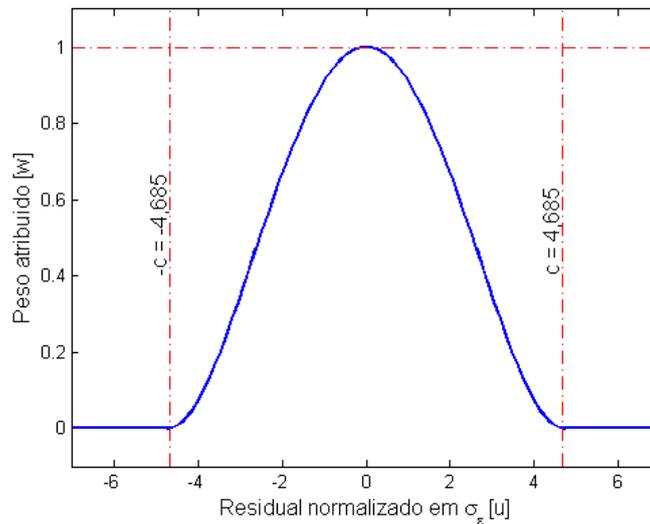
$$w_i = \frac{\psi(u_i)}{\varepsilon_i}$$

e com isso, podemos reescrever a equação Eq. 3.13 para  $w_i$  como

$$w_i = \begin{cases} \left[1 - \left(\frac{u_i}{c}\right)^2\right]^2, & |u_i| \leq c \\ 0, & |u_i| > c \end{cases}, \quad (3.15)$$

assim podendo atribuir o valor de  $w_i$  em função do residual  $e_i$ . O gráfico da Eq. 3.15 pode ser visto na Figura 3. Aqui podemos perceber que  $0 \leq w_i \leq 1, \forall i \in \{1, \dots, n\}$  então  $V^T W V \geq 0$ , o que significa que  $J(\beta)$  tem a concavidade voltada para cima e, por consequência, temos o mínimo da função na Eq. 3.8.

Figura 3 – Gráfico de  $w$  em relação a  $u$  desenvolvido a partir da função bponderada de Tukey.



Fonte: o autor.

Note, porém, que os pesos dependem dos erros (ou resíduos), os resíduos dependem dos coeficientes estimados, e os coeficientes estimados dependem dos pesos. Esta dependência faz com que um método iterativo seja necessário para estimar os parâmetros, sendo o *iteratively reweighted least-squares*<sup>3</sup> (IRLS) o mais usado para este fim.

O IRLS, aplicado ao problema em questão, consiste em executar o seguinte algoritmo:

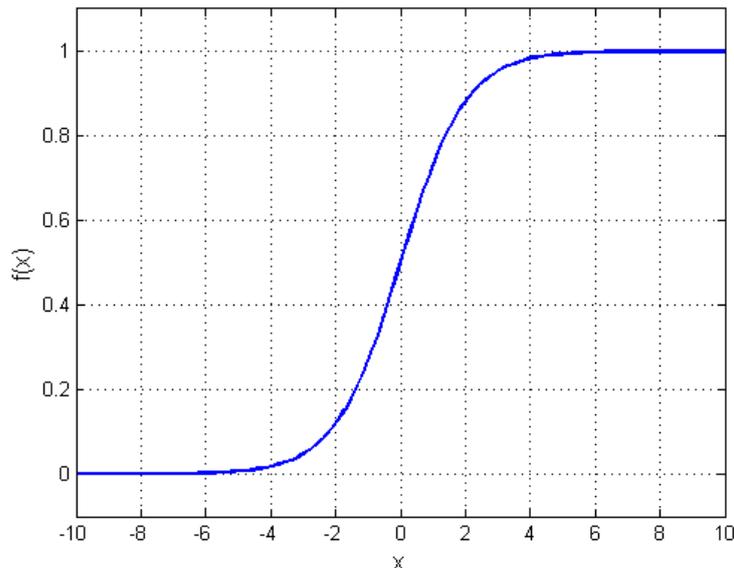
<sup>3</sup> Este método é usado na função `robustfit` implementado no Matlab®.

1. Calcular a estimação inicial de  $\hat{\beta}$  através do OLS (Eq. 3.5) e estabelecer os pesos iniciais como  $w_i = 1, \forall i \in \{1, \dots, n\}$ .
2. Calcular os residuais  $\varepsilon_i = p_i - v_i^T \beta$ .
3. Calcular  $\hat{\sigma}_\varepsilon$  através da Eq. 3.11 com  $\gamma = 1,4826$ .
4. Calcular  $u_i = \frac{\varepsilon_i}{\hat{\sigma}_\varepsilon}$ .
5. Calcular os pesos de  $w$  através da Eq. 3.15 com  $c = 4,685$ .
6. Atualizar os valores de  $\hat{\beta}$  através da Eq. 3.8.
7. Caso não tenha ocorrido a convergência dos parâmetros  $\hat{\beta}$ , retornar ao passo 2.

### 3.3 Modelos logísticos de 3, 4 e 5 parâmetros

Modelos logísticos são funções sigmoidais, ou seja, funções com o gráfico que se assemelha ao desenho da letra S (como exemplificada na Figura 4), que possuem parâmetros para ajustar o seu formato, sendo comumente usadas na aproximação da curva de potência de aerogeradores (SOHONI *et al.*, 2016; LYDIA *et al.*, 2013; LYDIA *et al.*, 2014). Neste trabalho iremos comparar o desempenho de dois modelos logísticos, cujos parâmetros serão ajustados por meio da técnica de otimização evolucionária conhecida como *evolução diferencial* (DE, sigla em inglês) (STORN; PRICE, 1997).

Figura 4 – Gráfico sigmóide simples da função  $f(x) = \frac{1}{1+e^{-x}}$ .



Fonte: o autor.

Podemos definir o modelo logístico de 3 parâmetros (3PL, sigla em inglês) que

relaciona a potência predita para um dado valor da velocidade como:

$$p(v) = \frac{\beta_1}{1 + e^{-\beta_2(v-\beta_3)}}, \quad (3.16)$$

em que definimos  $\beta = [\beta_1 \ \beta_2 \ \beta_3]^T$  como o seu vetor de parâmetros. Não há nenhuma restrição para os valores de  $\beta$  para este modelo.

Para o modelo logístico de 4 parâmetros (4PL, sigla em inglês), a potência predita para um dado valor da velocidade do vento pode ser definida pela seguinte expressão (KUSIAK *et al.*, 2009):

$$p(v) = \beta_1 \left( \frac{1 + \beta_2 e^{-\frac{v}{\beta_4}}}{1 + \beta_3 e^{-\frac{v}{\beta_4}}} \right), \quad (3.17)$$

em que definimos  $\beta = [\beta_1 \ \beta_2 \ \beta_3 \ \beta_4]^T$  como o seu vetor de parâmetros. Não há nenhuma restrição para os valores de  $\beta$  para este modelo.

Para o caso de um modelo logístico de 5 parâmetros (5PL, sigla em inglês), podemos utilizar a seguinte expressão (GOTTSCHALK; DUNN, 2005):

$$p(v) = \beta_1 + \frac{\beta_2 - \beta_1}{\left[ 1 + \left( \frac{v}{\beta_3} \right)^{\beta_4} \right]^{\beta_5}}, \quad (3.18)$$

em que  $\beta = [\beta_1 \ \beta_2 \ \beta_3 \ \beta_4 \ \beta_5]^T$  é o vetor de parâmetros correspondentes. Para esse modelo temos a restrição  $\beta_3, \beta_5 > 0$ .

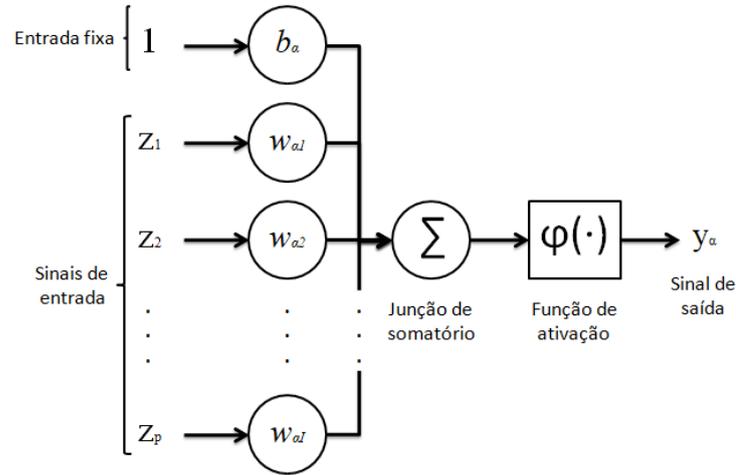
Ajustaremos os parâmetros de cada modelo logístico utilizando um algoritmo de DE (ENGELBRECHT, 2007) com as seguintes características:

- DE/best/1;
- critério de parada ocorre quando a aptidão for a mesma em 200 gerações seguidas;
- população de tamanho 150;
- fator de mutação 0,9; e
- fator de cruzamento 0,8.

### 3.4 Rede Perceptron Multicamadas (MLP)

O cérebro é um computador complexo, não linear e paralelo. Ele tem a capacidade de executar tarefas como reconhecimento de padrões, percepção e controle. Além dessas características, outras, como a capacidade de aprender, memorizar e ainda generalizar, estimularam

Figura 5 – Representação em blocos de um neurônio artificial  $\alpha$ .



Fonte: o autor.

pesquisas em modelagem algorítmica utilizando o paradigma de sistemas neurais biológicos (ENGELBRECHT, 2007).

Um neurônio artificial, cuja representação em blocos pode ser vista na Figura 5, é um modelo de um neurônio biológico. Podemos identificar três elementos básicos nesse modelo (HAYKIN, 2008):

1. Um conjunto de *sinapses* no qual cada um de seus elementos é caracterizado por um peso  $w$ . Por exemplo, um sinal  $z_j$  na entrada da sinapse  $j$  conectado ao neurônio  $\alpha$  é multiplicado pelo peso sináptico  $w_{\alpha j}$ . O peso sináptico de um neurônio artificial pode estar em um intervalo que inclui valores negativos e positivos.
2. Um somador que faz a combinação linear dos dados de entrada após a sua respectiva ponderação sináptica. Aqui soma-se também o limiar, denotado por  $b_\alpha$ , que atua diminuindo ou aumentando o resultado final da combinação linear.
3. Uma função de ativação, denotado por  $\varphi(\cdot)$ , que, em geral, é usada para limitar a amplitude da saída de um neurônio. Normalmente, a faixa de amplitude normalizada da saída de um neurônio é escrita como o intervalo unitário fechado  $[0, 1]$  ou, alternativamente,  $[-1, 1]$ .

Em termos matemáticos, sendo  $f : \mathbb{R}^D \rightarrow [0, 1]$  (ou  $f : \mathbb{R}^D \rightarrow [-1, 1]$ ) a função que representa um neurônio artificial e  $D$  a dimensão dos dados de entrada, temos que

$$f(z) = \varphi(w^\top z + b_\alpha) \quad (3.19)$$

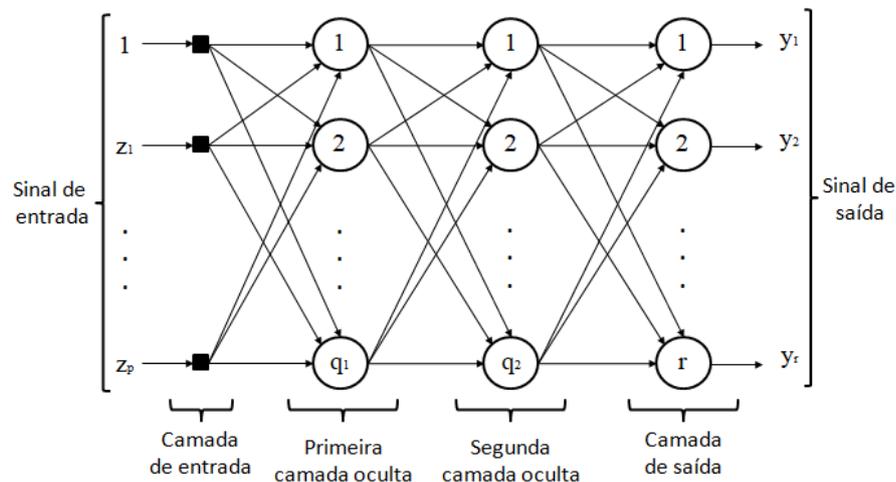
em que  $z = [z_1 \ z_2 \ \dots \ z_D]^T$  e  $w = [w_1 \ w_2 \ \dots \ w_D]^T$ . Alternativamente, podemos considerar  $b_\alpha = w_0$  e reescrever a 3.19 como

$$f(z) = \varphi(w'^T z') \quad (3.20)$$

em que  $z' = [1 \ z_1 \ z_2 \ \dots \ z_D]^T$  e  $w' = [b_\alpha \ w_1 \ w_2 \ \dots \ w_D]^T$ .

Uma Rede Neural Artificial (RNA), cujo paradigma é inspirado no sistema nervoso biológico, é uma cadeia de neurônios artificiais conectados entre si. As entradas de cada neurônio da RNA ou são dados oriundos de informações sobre variáveis externas (e.g. temperatura, pressão, umidade, velocidade do vento etc) ou são a saída de outros neurônios da rede. E a saída também apresenta duas condições de finalidade: ou é o valor de entrada para outro(s) neurônio(s) ou é o valor de saída da própria RNA. O treinamento da RNA é feito através do ajuste dos pesos a fim de que os valores de saída sejam tão próximos quanto possível de um valor esperado. Uma das principais formas de fazer isso é através do método *Back-propagation* (HAYKIN, 2008). Dessa forma, as RNAs podem aprender a relação não-linear entre os conjuntos de dados de entrada e saída.

Figura 6 – Rede MLP com duas camadas ocultas.



Fonte: o autor.

A rede *Multilayer Perceptron* (MLP) é uma RNA disposta em camadas de neurônios em que a saída de cada um deles se conecta à entrada de todos neurônios da camada subsequente, caso exista. A MLP é dividida em camadas, das quais podemos distinguir como:

- **Camada de entrada** - é a interface de entrada de dados da MLP. Essa camada é caracterizada pela quantidade de dados simultâneos recebidos pela MLP, com exceção, caso exista, do 1 fixo para o limiar.

- Camada oculta - São todas as camadas que não estão conectadas diretamente com a saída da MLP. Esta camada é caracterizada pela quantidade de neurônios contidos.
- Camada de saída - é a camada de neurônios diretamente ligadas a saída. Assim como a camada oculta, essa camada é caracterizada pela quantidade de neurônios contidos, ou, se preferir, pela quantidade de saídas da MLP, já que a saída de cada neurônio dessa camada representa a saída da própria MLP.

A Figura 6 ilustra um exemplo da rede MLP em que temos  $D$  variáveis de entrada,  $q_1$  na primeira camada oculta,  $q_2$  na segunda e uma camada de saída com  $r$  neurônios.

As funções de ativação dos neurônios ocultos devem ser não-lineares (e.g. sigmóides). Neurônios de saída podem usar tanto funções sigmóides quanto lineares, a depender da aplicação de interesse. Em classificação de padrões, geralmente funções sigmóides são usadas, enquanto funções lineares são usadas em problemas de regressão ou ajuste de curvas.

A rede MLP com uma camada oculta pode ser considerada como um aproximador universal de função. Assim, desde que possua uma quantidade suficiente de neurônios ocultos, ela pode aproximar qualquer função contínua com uma certa precisão (SANTOS, 2010).

Considere uma MLP com apenas uma camada oculta e um único neurônio linear de saída (i.e.  $r = 1$ ). Sua saída é dada por

$$y_n = \mathbf{w}^T \mathbf{h}_n + b_r, \quad (3.21)$$

onde  $\mathbf{w} = [w_1 \cdots w_{q_1}]^T$  é o vetor de pesos que conecta os neurônios ocultos ao neurônio de saída e  $b_r$  é o limiar deste neurônio. O vetor  $\mathbf{h}_n \in \mathbb{R}^{q_1}$ , que contém as saídas atuais dos neurônios ocultos, é calculado como

$$\mathbf{h}_n = [\varphi(\mathbf{m}_1^T \mathbf{x}_n + b_1), \dots, \varphi(\mathbf{m}_{q_1}^T \mathbf{x}_n + b_{q_1})]^T, \quad (3.22)$$

tal que  $\mathbf{x}_n = [x_1 \cdots x_D]^T$  é o vetor de entrada atual,  $\mathbf{m}_j = [m_{j1} \cdots m_{jD}]^T$  é o vetor de pesos do  $j$ -ésimo neurônio oculto,  $b_j$  denota seu limiar, e  $\varphi(\cdot)$  é a função de ativação não-linear associada.

Visto que estamos lidando com um problema de uma única entrada (velocidade do vento,  $v_n$ ) e uma única saída (potência gerada,  $p_n$ ), é preciso treinar apenas uma rede MLP(1,  $q$ , 1), i.e. uma rede com uma única entrada (além do limiar) e um neurônio de saída. Neste caso,  $y_n = \hat{p}_n$  e  $z_n = [-1 \ v_n]^T$ , com  $\hat{p}_n$  denotando a potência predita para a velocidade do vento atual  $v_n$ . Neste trabalho, os pesos e limiares da rede MLP aplicada ao problema de estimação da curva

de potência de aerogeradores são computados via algoritmo de Levenberg-Marquardt (HAGAN; MENHAJ, 1994).

### 3.5 Critério de informação

O dimensionamento de modelos é um problema recorrente na estatística aplicada. Podemos citar, como exemplos, a estimação da ordem de um modelo de séries temporais AR ou MA, a estimação do grau do polinômio que descreve um conjunto de dados, a seleção das variáveis a serem introduzidas em uma equação de regressão múltipla, assim por diante (DELEEUW, 1992). Segundo o princípio da parcimônia, entre modelos com iguais capacidade descritiva a respeito de um determinado fenômeno, opta-se em utilizar o mais simples (SEASHOLTZ; KOWALSKI, 1993). Podemos entender como modelo de regressão mais simples o que apresentar menor quantidade de parâmetros. Ao comparar funções de regressão através do MMV, negligenciamos o princípio da parcimônia, pois quanto maior a quantidade de parâmetros, maior a capacidade explicativa sobre um determinado fenômeno. Assim, é consenso entre os estatísticos que não há otimização neste tipo de procedimento comparativo (DELEEUW, 1992).

Há duas soluções muito utilizadas e difundidas pelos estatísticos para o problema de dimensão de modelos (DELEEUW, 1992; DZIAK *et al.*, 2012; BURNHAM; ANDERSON, 2004): uma é Critério de Informação de Akaike (AIC, sigla em inglês) e o outro é o Critério de Informação Bayesiano (BIC, sigla em inglês). Ambos os critérios são bem semelhantes e desenvolvidos a partir do MMV, porém fazem a consideração do número de parâmetros em sua avaliação. Em ambos, o modelo que apresentar o menor valor é considerado melhor avaliado.

Para um modelo de regressão que relaciona  $p$  à  $v$ , com distribuição dependente de  $m$  parâmetros e sendo  $n$  a quantidade de amostras coletadas, temos seu AIC dado por (AKAIKE, 1974).

$$AIC = -2 \ln \{ \mathcal{L}(\theta|p) \} + 2m \quad (3.23)$$

e seu BIC dado por (SCHWARZ, 1978)

$$BIC = -2 \ln \{ \mathcal{L}(\theta|p) \} + m \ln(n) \quad (3.24)$$

Podemos aplicar a Eq. A.13 nas Eqs. 3.23 e 3.24, chegando aos seguintes critérios de informação:

$$AIC = n \ln \{ SQR \} + 2k \quad (3.25)$$

e

$$BIC = n \ln \{SQR\} + k \ln (n) \quad (3.26)$$

Ao utilizar o AIC como critério de seleção, o problema mais comum que ocorre é o de chegar a um modelo demasiadamente grande. Em contra partida, utilizar o BIC faz com que o problema frequente seja o de obter um modelo com poucos parâmetros. Ademais, para um  $n$  muito pequeno, o erro mais comum é o subajuste, portanto geralmente AIC vem apresentado melhores resultados empíricos para este tipo de situação. Da mesma forma,  $n$  demasiadamente grande faz com que o sobreajuste seja o tipo de erro mais comum, assim utilizar um critério mais parcimonioso, como o BIC, geralmente proporciona resultados mais adequados. Infelizmente, o ponto em que o  $n$  se torna "grande" depende de vários aspectos da natureza do modelo de difícil identificação. Acaba sendo mais útil selecionar o critério que será utilizado em função do tipo de modelo que é desejado (DZIAK *et al.*, 2012).

### 3.6 Conclusões

A princípio, neste capítulo, foi feita uma breve revisão a respeito de alguns dos modelos comumente utilizados para regressão da curva de potência de aerogeradores. Iniciando pelo modelo polinomial clássico, em que seus parâmetros, são estimados pelo método dos mínimos quadrados ordinários (*ordinary least squares*, OLS), e por uma versão ponderada deste estimador. Em seguida, apresentamos modelos logísticos de 3, 4 e 5 parâmetros e finalizamos com o modelo baseado em redes neurais artificiais.

Por fim, fizemos uma breve reflexão a respeito do dimensionamento de modelos matemáticos. Sinalizamos o interesse antitético por um modelo que seja bem descritivo e ao mesmo tempo parcimonioso. Disto, salientamos a importância de se utilizar um critério de seleção de modelos adequado afim de avaliar e compará-los entre si.

O capítulo seguinte destina-se a apresentar duas novas metodologias de regressão de curva de potência. Uma baseada em uma metodologia estatística de regressão esparsa comumente utilizada na estatística e a outra num modelo original baseada em computação evolucionária.

## 4 METODOLOGIAS PROPOSTAS

Este capítulo destina-se a propor dois modelos de regressão esparsos<sup>1</sup> não utilizados para estimação de curva de potência de aerogeradores. O primeiro deles, que já é relativamente popular na estatística, é o *Least Absolute Shrinkage and Selection Operator*, ou LASSO. O segundo é uma nova proposta de regressão esparsa intitulada como *Polynomial Reduction Induced by Sparse Method Algorithm*, ou PRISMA.

### 4.1 LASSO

Caso a matriz  $V^T V$  seja singular a estimação de  $\beta$  por OLS seria inviável, o que impossibilitaria o desenvolvimento da Eq. 3.8. Mesmo não sendo singulares, matrizes com o determinante aproximadamente igual a zero já poderiam gerar instabilidade na otimização do algoritmo OLS. Para ilustrar essa instabilidade, tomemos como exemplo um caso em que duas das variáveis são altamente correlacionadas. Isso permite que o valor de  $\beta_i$  associado a uma variável cresça muito na direção positiva, enquanto a outra cresce proporcionalmente na direção negativa, cancelando o efeito da primeira variável. Isso produz um modelo de alta variância, tornando-o cada vez mais irrealista à medida que a correlação aumenta. Assim, o resultado final da produção de modelos de alta variância está diretamente relacionado ao potencial instabilidade numérica do procedimento de Mínimos Quadrados.

Um método bastante difundido (BOYD; VANDENBERGHE, 2004) para contornar a instabilidade numérica é o de somar um termo de penalização ao funcional  $J(\beta)$  em função do módulo de seus argumentos. Isso favorece a prevalência de valores menores em  $\beta$ , o que, por consequência, proporciona estabilidade e produz modelos de variância mais baixos.

A função de penalização  $d$ -norm, também conhecida por  $\ell_d$ , é uma função  $\mathbb{R}^n \rightarrow \mathbb{R}$ , definida por

$$\ell_d(\beta) = \|\beta\|_d = \left( \sum_{i=1}^n |\beta_i|^d \right)^{1/d}, \quad d \in \mathbb{N}^*. \quad (4.1)$$

é usada como função de penalização para valores elevados de  $\beta$  na forma  $[\ell_d(\beta)]^d$ . Assim, uma regressão por mínimos quadrados com penalidade  $\ell_d$  é a regressão que tem como objetivo

<sup>1</sup> Um modelo de regressão esparsa caracteriza-se pela tendência a excluir parâmetros com pouca ou nenhuma relevância para descrever o objeto de sua aplicação.

minimizar o seguinte funcional

$$J(\beta) = (p - V\beta)^\top (p - V\beta) + \lambda [\ell_d(\beta)]^d. \quad (4.2)$$

em que  $\lambda \geq 0$  é uma constante usada para regular a intensidade da penalização.

Dentre as possíveis penalizações *d-norm*, as mais utilizadas são as  $\ell_1$  e  $\ell_2$ . Em ambas, a estabilidade desejada é obtida. Porém, em  $\ell_2$ , cuja função de penalização é dada por

$$\begin{aligned} \lambda [\ell_2(\beta)]^2 &= \lambda \left[ \left( \sum_{i=1}^n |\beta_i|^2 \right)^{1/2} \right]^2 \\ &= \lambda \sum_{i=1}^n |\beta_i|^2 \\ &= \lambda \beta^\top \beta, \end{aligned} \quad (4.3)$$

temos a vantagem de se tratar de uma função convexa e diferenciável, possibilitando a obtenção de um valor ótimo de  $\beta$  pelos métodos tradicionais de derivação. Entretanto, a penalização é atenuada a medida que os valores de  $\beta$  se aproximam de zero. Isso se dá pela natureza de uma curva quadrática cujo o valor absoluto da derivada diminui a medida que se aproxima de um ponto crítico da função.

Em  $\ell_1$ , cuja função de penalização é dada por

$$\lambda [\ell_1(\beta)] = \lambda \sum_{i=1}^n |\beta_i|, \quad (4.4)$$

temos uma função não diferenciável, pois uma função definida pelo valor absoluto do seu argumento não tem derivada quando seu argumento é zero. Porém, como a derivada da função é constante em valor absoluto (com exceção, como supracitado, quando seu argumento for zero), temos uma tendência da penalização reduzir o valor dos coeficientes com a mesma intensidade até zerá-los. Isso resulta em uma característica interessante para regressões por mínimos quadrados com penalização  $\ell_1$ : podemos produzir uma solução esparsa. Ou seja, uma solução em que, a depender do valor de  $\lambda$  ser grande o suficiente, alguns coeficientes menos relevantes para a resposta da regressão serão zerados. Efetivamente, isso resulta na eliminação dos termos correspondente do sistema. Caso  $\lambda = 0$  nenhum termo será podado e a medida que cresce, os termos serão podados até que não reste mais nenhum.

Embora seja provável que tenha sido explorado anteriormente, a estimativa de parâmetros por mínimos quadrados com penalização  $\ell_1$  foi apresentada e popularizada sob o nome *Least Absolute Selection and Shrinkage Operator* (LASSO) por Tibshirani (TIBSHIRANI,

1996). Portanto, para o restante deste trabalho, usaremos o termo LASSO para denotar esse tipo de problema.

Os parâmetros da regressão LASSO não podem ser otimizados através do método tradicional da derivação. Essa desvantagem levou ao desenvolvimento de uma infinidade de técnicas para determinar seus parâmetros ótimos. Schmidt desenvolveu um estudo que elenca 8 métodos distintos de otimização (SCHMIDT, 2005). Vários desses algoritmos usam diretamente o problema de otimização sem restrições

$$\underset{\beta}{\text{minimize}} \quad (p - V\beta)^\top (p - V\beta) + \lambda \|\beta\|_1, \quad (4.5)$$

enquanto outras técnicas usam formulações restritas equivalentes, como por exemplo

$$\begin{aligned} \underset{\beta}{\text{minimize}} \quad & (p - V\beta)^\top (p - V\beta) \\ \text{subject to} \quad & \|\beta\|_1 \leq \frac{1}{\lambda}. \end{aligned} \quad (4.6)$$

## 4.2 PRISMA

Suponha uma regressão polinomial por mínimos quadrados conforme apresentada na Seção 3.1. Considere agora a introdução de coeficiente extra  $\zeta$  associado a cada termo da regressão, segundo a seguinte definição:

$$\hat{p}_i = \zeta_0 \beta_0 + \zeta_1 \beta_1 v_i + \dots + \zeta_{k_*} \beta_{k_*} v_i^{k_*} \quad (4.7)$$

$$= \sum_{j=0}^{k_*} \left\{ \zeta_j \beta_j v_i^j \right\} \quad (4.8)$$

$$= (\zeta \circ \beta)^T \mathbf{v}_i \quad (4.9)$$

em que  $k_*$  é o maior grau permitido ao polinômio,  $\beta$  é o vetor de coeficientes do polinômio,  $\zeta$  é um vetor de elementos binários (i.e.  $\zeta_j \in \{0, 1\}$ ,  $j = 0 \dots k_*$ ) chamados de *coeficientes de ativação*,  $\mathbf{v}_i$  é o vetor de velocidades associadas à  $i$ -ésima medida de velocidade  $v_i$ . O símbolo  $\circ$  denota um produto de Hadamard.

É importante enfatizar o papel dos coeficientes  $\zeta_j$  no modelo PRISMA. Um coeficiente  $\zeta_j$  com valor 0 elimina completamente o termo correspondente do polinômio. Segundo a formulação mostrada na Eq. (4.7), o modelo polinomial original torna-se um caso particular da metodologia proposta, em que  $\zeta_j = 1, \forall j$ .

Do exposto, buscamos uma solução que simultaneamente minimize a SQR do modelo e o número de termos do modelo polinomial. Isto é equivalente a procurar uma solução ótima

esparsa para o vetor  $\zeta$  e, conseqüentemente, para o vetor de coeficientes  $\beta$ . Um exemplo hipotético de uma solução esparsa para o problema de interesse em que  $k_* = 4$  é dada por  $\zeta = [1 \ 1 \ 0 \ 1 \ 0]^T$ . Neste caso, usamos apenas os coeficientes  $\beta_0$ ,  $\beta_1$  e  $\beta_3$ , o que leva à seguinte expressão para o preditor polinomial:

$$\hat{p}_i = \hat{\beta}_0 + \hat{\beta}_1 v_i + \hat{\beta}_3 v_i^3, \quad (4.10)$$

com os coeficientes  $\hat{\beta}_0$ ,  $\hat{\beta}_1$  e  $\hat{\beta}_3$  sendo estimados pela mesma expressão usada na Eq. (3.5), mas com as colunas para os quais  $\zeta_j = 0$  retiradas. No presente exemplo, as colunas  $j = 3$  e  $j = 5$  seriam eliminadas.

A matriz  $V$ , desenvolvida na Seção 3.1, é uma matriz de Vandermonde<sup>2</sup>. Isto faz com que uma escolha inadequada da ordem do polinômio (e.g.  $k > 9$ ) produza valores elevados para as algumas de suas colunas. Por exemplo, com valores típicos de velocidade do vento na faixa de 4 a 12 m/s, este fato conduz em geral a erros numéricos durante a inversão existente na Eq. (3.5). A metodologia proposta tem a propriedade inerente de evitar a ocorrência de tais problemas numéricos causados por valores elevados da ordem do polinômio.

Para encontrar o vetor-solução  $\zeta$  ótimo lançaremos mão de um método de otimização metaheurística, mais especificamente, algoritmos genéticos (AG) (ENGELBRECHT, 2007). Como os elementos de  $\zeta$  são binários, a codificação binária para os cromossomos será usada. Deste modo, o  $l$ -ésimo indivíduo de uma população de  $M$  cromossomos é definido como

$$\text{Indivíduo}_l = \zeta_l = [\zeta_{l,0} \ \zeta_{l,1} \ \cdots \ \zeta_{l,k_*}]^T, \quad (4.11)$$

para  $l = 1, \dots, M$ .

Para fins de uma justa comparação com as abordagens clássicas apresentadas na Seção 3, submetemos o modelo PRISMA a três baterias de testes para cada conjunto de dados, em que cada bateria utiliza uma função de aptidão do AG diferente, sendo elas o SQR, o AIC e o BIC.

Neste trabalho, limitamos o valor máximo da ordem do polinômio em  $k_* = 10$ , como o AG possuindo as seguintes especificações:

- $N_g = 10$  gerações;
- população de tamanho  $M = 500$ ;

<sup>2</sup> É uma matriz que apresenta em suas linhas a progressão geométrica, iniciada por 1, dos elementos de um determinado vetor (e.g.  $V = v_i^{j-1}$  em que  $i$  é o índice da linha de  $V$  bem como o índice do vetor  $v$ , e  $j$  é o índice da coluna de  $V$ ).

- cromossomo binário de tamanho  $k_* + 1 = 11$ ;
- seleção por torneio;
- cruzamento de um ponto;
- probabilidade de cruzamento  $p_c = 80\%$ ;
- probabilidade de mutação  $p_m = 0,1\%$ ; e
- sem elitismo.

### 4.3 Conclusões

Duas novas metodologias de regressão esparsas para curva de potência de aerogeradores foram apresentadas neste capítulo. No capítulo seguinte, essas duas e as metodologias apresentadas no Capítulo 3 serão avaliadas e comparadas entre si utilizando três conjuntos de dados reais.

## 5 SIMULAÇÃO E RESULTADOS

Para compararmos os desempenhos entre os modelos apresentados nos Capítulos 3 e 4, desenvolvemos uma bateria de testes para identificar o formato ótimo, segundo as avaliações SQR, AIC e BIC para cada um dos conjuntos de dados. Utilizamos o método de Monte Carlo<sup>1</sup>, com 100 repetições, nos casos em que os modelos utilizam variáveis aleatórias no desenvolvimento de sua regressão.

Para os modelos polinomiais (usual e ponderado), o número de parâmetros é igual a  $k_\beta = k + 1$ , em que  $k$  é a ordem do maior expoente. Para o modelo LASSO, o número de parâmetros é igual à cardinalidade do conjunto dos coeficientes não nulos e para o modelo PRISMA é a cardinalidade do conjunto dos coeficientes ativados. No caso do modelo neural, o número de parâmetros depende apenas do número de neurônios ocultos ( $q_1$ ), pois o número de entradas e o número de saídas é fixado em  $D = r = 1$ . Assim, o número total de parâmetros da rede MLP é dado por  $k_\beta = (D + 1) \cdot q_1 + (q_1 + 1) \cdot r = (1 + 1) \cdot q_1 + (q_1 + 1) \cdot 1 = 3q_1 + 1$ . Os modelos logísticos já têm a quantidade de termos fixos, bastando encontrar o valor ótimo para esses.

### 5.1 Conjunto de dados

Utilizamos três conjuntos de dados distintos para validação e comparação das metodologias aqui elencadas. O primeiro conjunto de dados, denotado por TURB1, foi coletado do Parque eólico da Prainha, no Ceará. Os dois outros conjuntos foram utilizados primeiramente em (LEE *et al.*, 2015) e são compostos por dados de velocidade e potência de 6 aerogeradores ao todo (quatro costeiras e duas em alto mar), denotadas de WT1 até WT6, respectivamente, e estão disponíveis para uso público<sup>2</sup>. No presente trabalho, usamos os dados referentes aos aerogeradores WT1 e WT5, aqui denotados como TURB2 e TURB3, respectivamente. A Tabela 1 reúne as informações a respeito dos três aerogeradores.

Os dados ambientais referentes aos aerogeradores, dentre eles a velocidade do vento, são coletados por sensores em uma torre meteorológica, enquanto os dados de potência são medidos no próprio aerogerador. Para os aerogeradores TURB2 e TURB3, cada torre meteorológica está associada a 2 aerogeradores, o que significa que cada medida de velocidade

<sup>1</sup> Método que se baseia em repetir sucessivas simulações um elevado numero de vezes para calcular suas características numéricas.

<sup>2</sup> <https://aml.engr.tamu.edu/2001/09/01/publications/>

Tabela 1 – Especificação dos três aerogeradores.

	TURB1	TURB2	TURB3
Altura do cubo (m)	40	80	70
Diâmetro do rotor (m)	40	80	90
Vel. de corte inferior (m/s)	3	3,5	3,5
Vel. de corte superior (m/s)	13	20	25
Vel. nominal aprox. (m/s)	12	13	15
Potência nominal (MW)	0,5	2	3
Pontos coletados	2250	47542	16443
Localização	Costeira	Costeira	Alto mar

Fonte: (LEE *et al.*, 2015).

feita naquela torre é pareada com as de potência das 2 torres associadas.

É prática corrente arranjar os dados coletados em fazendas eólicas em blocos de 10 min, período em que velocidades do vento são consideradas estacionárias (i.e. aproximadamente constantes). consequentemente, tomam-se as médias das leituras da potência de saída e da velocidade do vento a cada intervalo de 10 min.

Todos os modelos implementados e experimentos foram realizados no ambiente de programação do software Matlab<sup>®</sup>, disponibilizado pelo Centro de Computação de Alto Desempenho<sup>3</sup> (CENAPAD) da UFC.

## 5.2 Modelo PRISMA

No modelo PRISMA, temos a liberdade de estabelecer qual função objetivo será utilizada como critério de avaliação do AG. Neste trabalho, como mencionado anteriormente, nos limitamos a trabalhar apenas com SQR, AIC e BIC. Porém, qualquer função que avalie a regressão pode ser facilmente adaptada para o algoritmo desenvolvido.

A função objetivo utilizada tem influência direta na parcimônia do modelo. Essa função será utilizada como lastro para estabelecer quais os termos com menos influência na regressão, que serão podados visando uma representação parcimoniosa entre a variável independente e a explicativa. Além disso, devemos estabelecer o grau máximo admitido para o polinômio de regressão. Os scripts desenvolvidos em Matlab<sup>®</sup> podem ser vistos nos Anexos B.1, B.2 e B.3.

As duas maiores vantagens que o modelo PRISMA apresenta em relação aos demais é que não é preciso testar qual o grau do polinômio que apresenta a melhor aptidão e nem a quantidade de termos que serão utilizados. Essas avaliações já são intrínsecas ao algoritmo

<sup>3</sup> <http://cenapad.ufc.br>

evolutivo desenvolvido.

O grau máximo admitido para os polinômios foi o décimo, conseqüentemente, no máximo onze termos podem ser utilizados. Dessa forma, modificamos a função objetivo para cada um dos critérios de avaliação estabelecidos e obtemos os resultados dos testes para cada um dos conjuntos de dados que podem ser vistos nas Tabelas 2, 3 e 4.

Tabela 2 – Desempenho do modelo PRISMA em TURB1.

<b>Aptidão utilizada</b>	<b>Termos</b>	<b>SQR</b>	<b>AIC</b>	<b>BIC</b>
SQR	11	652019,5	30146,6	30215,2
AIC	6	652744,2	30139,1	30179,1
BIC	2	658657,3	30151,4	30168,6

Fonte: o autor.

Tabela 3 – Desempenho do modelo PRISMA em TURB2.

<b>Aptidão utilizada</b>	<b>Termos</b>	<b>SQR</b>	<b>AIC</b>	<b>BIC</b>
SQR	11	8061526,3	756066,0	756171,3
AIC	10	8061835,3	756065,9	756162,3
BIC	9	8062567,5	756068,2	756155,9

Fonte: o autor.

Tabela 4 – Desempenho do modelo PRISMA em TURB3.

<b>Aptidão utilizada</b>	<b>Termos</b>	<b>SQR</b>	<b>AIC</b>	<b>BIC</b>
SQR	11	1563180,6	234537,9	234630,4
AIC	5	1563844,5	234532,9	234579,1
BIC	4	1563412,9	234526,3	234564,9

Fonte: o autor.

### 5.3 Modelos polinomial e polinomial robusto

A quantidade de termos que são utilizados nos modelos polinomial e robusto é igual ao grau do polinômio de regressão mais um. Foi utilizado um algoritmo desenvolvido no Matlab<sup>®</sup>, cujo *script* pode ser visto no Anexo B.4, que coleta o SQR, AIC e BIC destes modelos a partir do polinômio de grau zero (função constante) até o de décimo grau. Os resultados do teste para os três conjuntos de dados podem ser vistos nas Tabelas 5, 6 e 7.

No modelo polinomial, como era de se esperar, o SQR é inversamente proporcional ao grau do polinômio. Porém, como a otimização do modelo robusto é feita de forma iterativa

Tabela 5 – Desempenho dos modelos polinomial e polinomial robusto em TURB1.

Grau	Termos	Modelo Polinomial			Modelo Robusto		
		SQR	AIC	BIC	SQR	AIC	BIC
0	1	25020356,8	38333,2	38344,6	25031326,4	38334,2	38345,6
1	2	1772430,3	32378,7	32395,8	2402965,2	33063,5	33080,6
2	3	1415554,6	31874,8	31897,7	4897965,9	34667,7	34690,6
3	4	775057,5	30521,6	30550,2	821880,8	30653,5	30682,1
4	5	657428,8	30153,2	30187,5	661150,1	30165,9	30200,2
5	6	657395,0	30155,1	30195,1	662119,7	30171,2	30211,2
6	7	656508,4	30154,1	30199,8	661623,7	30171,5	30217,3
7	8	654097,2	30147,8	30199,2	659348,2	30165,8	30217,2
8	9	652598,0	30144,6	30201,8	657598,3	30161,8	30219,0
9	10	652595,5	30146,6	30209,5	657655,9	30164,0	30226,9
10	11	652019,5	30146,6	30215,2	657652,8	30166,0	30234,6

Fonte: o autor.

Tabela 6 – Desempenho dos modelos polinomial e polinomial robusto em TURB2.

Grau	Termos	Modelo Polinomial			Modelo Robusto		
		SQR	AIC	BIC	SQR	AIC	BIC
0	1	54797884,6	847162,6	847180,1	54807471,8	847170,9	847188,5
1	2	10471227,1	768481,6	768507,9	10889986,6	770345,9	770372,2
2	3	9510838,1	763910,1	763945,2	9586288,2	764285,8	764320,9
3	4	8396532,7	757987,8	758031,6	9144563,8	762045,0	762088,9
4	5	8118261,8	756387,5	756440,1	8304130,9	757463,7	757516,3
5	6	8095235,7	756254,4	756315,8	8316917,1	757538,8	757600,2
6	7	8081444,8	756175,4	756245,5	8309463,7	757498,2	757568,4
7	8	8071144,9	756116,7	756195,7	8303393,3	757465,5	757544,4
8	9	8062948,0	756070,4	756158,1	8310869,2	757510,2	757597,9
9	10	8062567,1	756070,2	756166,6	8302835,6	757466,3	757562,7
10	11	8061526,3	756066,0	756171,3	8303970,6	757474,8	757580,0

Fonte: o autor.

Tabela 7 – Desempenho dos modelos polinomial e polinomial robusto em TURB3.

Grau	Termos	Modelo Polinomial			Modelo Robusto		
		SQR	AIC	BIC	SQR	AIC	BIC
0	1	16907567,6	273669,3	273684,7	16910132,2	273671,8	273687,2
1	2	2135872,8	239652,6	239675,8	2158604,9	239826,7	239849,8
2	3	1982377,6	238428,3	238459,2	1984509,7	238446,0	238476,8
3	4	1610660,9	235015,9	235054,4	1648933,1	235402,1	235440,6
4	5	1580943,5	234711,7	234757,9	1596930,7	234877,1	234923,4
5	6	1566196,3	234559,6	234613,5	1587649,0	234783,3	234837,2
6	7	1565555,1	234554,9	234616,5	1589836,1	234807,9	234869,6
7	8	1563314,8	234533,3	234602,7	1587216,2	234782,8	234852,2
8	9	1563209,0	234534,2	234611,3	1588370,9	234796,8	234873,8
9	10	1563180,8	234535,9	234620,7	1588115,2	234796,1	234880,9
10	11	1563180,6	234537,9	234630,4	1588546,7	234802,6	234895,1

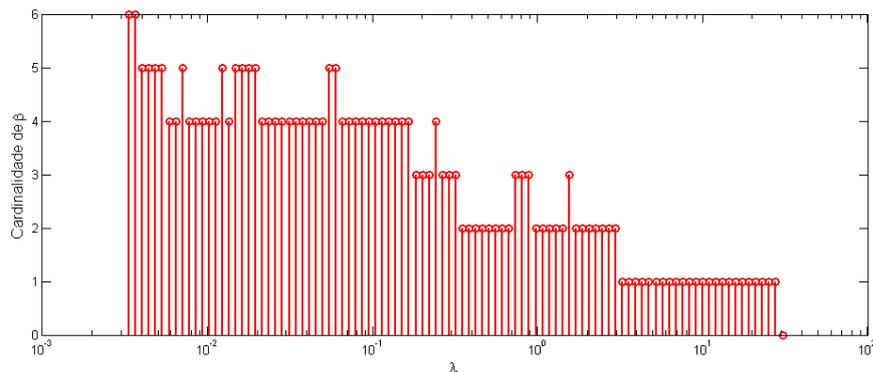
Fonte: o autor.

até que se tenha a convergência dos parâmetros  $\beta$ , o comportamento do SQR em relação ao grau do polinômio depende muito do critério de parada do algoritmo e isso faz com que este modelo não apresente o mesmo comportamento decrescente do modelo polinomial regular.

#### 5.4 Modelo LASSO

O parâmetro de controle de penalização  $\lambda$  tem influência direta na quantidade de termos de regressão no modelo LASSO e, conseqüentemente, nas avaliações SQR, AIC e BIC. Valores pequenos de  $\lambda$  tendem a utilizar mais termos e à medida que seu valor cresce, a quantidade tende a diminuir até que todos os coeficientes de regressão sejam eliminados. O menor valor de  $\lambda$  que exclui todos os termos do polinômio é denotado como  $\lambda_{max}$ . À título de ilustração, podemos perceber na Figura 7 essa relação através do gráfico de barras que relaciona a quantidade de coeficientes utilizado em função de  $\lambda$  para o conjunto de dados TURB2.

Figura 7 – Quantidade de termos utilizados em função de  $\lambda$  para o modelo LASSO em TURB2.



Fonte: o autor.

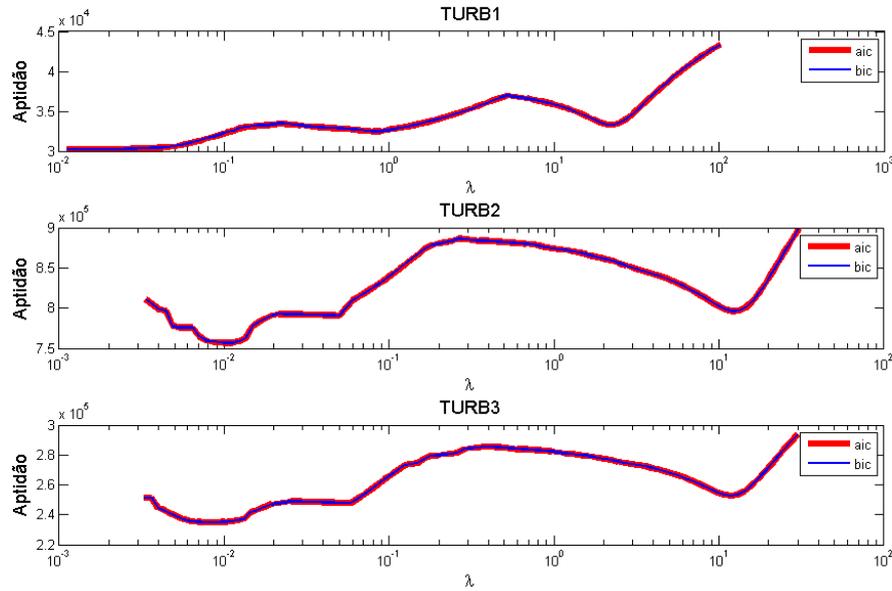
Para estimar o  $\lambda$  ótimo primeiramente identificamos numericamente o  $\lambda_{max}$  e então testamos o resultado da regressão para os cento e um valores da seguinte progressão geométrica:

$$\{0, a^1, a^2, \dots, a^{100}\} \quad (5.1)$$

em que  $a^{100} = \lambda_{max}$ . O *script* desenvolvido em Matlab<sup>®</sup> para este teste pode ser visto em B.5.

Na Figura 8 temos a curva do desempenho em função de  $\lambda$  para os conjuntos de dados. Com exceção do AIC em TURB1, todos critérios de seleção apontaram para um mesmo  $\lambda$  ótimo. As Tabelas 8, 9 e 10 nos apresentam os valores em torno desse ótimo para cada conjunto de dados.

Figura 8 – Curva da aptidão em função do valor de  $\lambda$  para o modelo LASSO.



Fonte: o autor.

Tabela 8 – Desempenho do modelo LASSO em TURB1.

$\lambda$	Termos	SQR	AIC	BIC
0,00000	10	656259,4	30159,2	30222,1
0,01118	5	664858,8	30178,5	30212,8
0,01227	5	663352,7	30173,4	30207,7
0,01346	5	662043,9	30168,9	30203,3
0,01477	5	661058,6	30165,6	30199,9
0,01621	5	660573,8	30163,9	30198,3
0,01780	5	660840,6	30164,9	30199,2
0,01953	5	662224,1	30169,6	30203,9
0,02143	5	665280,8	30179,9	30214,2
0,02352	5	670976,3	30199,1	30233,4
0,02582	5	681549,4	30234,3	30268,6
0,02834	4	695620,3	30278,3	30306,9

Fonte: o autor.

## 5.5 Modelos logísticos

Otimizar a quantidade de termos de forma parcimoniosa nos modelos logísticos é um contrassenso pois, diferente dos modelos até então desenvolvidos nesse capítulo, eles são fixos por definição para cada um dos modelos logísticos. Como o valor do AIC/BIC é calculado a partir do SQR e do número de termos utilizados na regressão (que neste caso é invariante), o melhor desempenho de SQR corrobora com o AIC/BIC.

Apesar de não ter aqui um hiperparâmetro a ser ajustado, necessita-se que seus

Tabela 9 – Desempenho do modelo LASSO em TURB2.

$\lambda$	Termos	SQR	AIC	BIC
0,00705	5	9684054,7	764772,2	764824,8
0,00774	4	8612929,6	759197,5	759241,4
0,00850	4	8441543,6	758241,9	758285,8
0,00932	4	8304027,3	757461,1	757504,9
0,01023	4	8197018,7	756844,5	756888,3
0,01123	4	8196949,1	756844,1	756887,9
0,01233	5	8490363,4	758518,1	758570,7
0,01353	4	9251249,0	762596,5	762640,3
0,01485	5	12588160,8	777241,3	777293,9
0,01630	5	14117189,3	782691,4	782744,0
0,01788	5	15531216,1	787229,7	787282,3

Fonte: o autor.

Tabela 10 – Desempenho do modelo LASSO em TURB3.

$\lambda$	Termos	SQR	AIC	BIC
0,00477	6	2264938,2	240625,4	240679,3
0,00524	6	1945615,0	238126,5	238180,5
0,00575	5	1761552,8	236490,4	236536,6
0,00631	5	1677260,1	235684,1	235730,4
0,00692	5	1615330,3	235065,5	235111,8
0,00760	4	1582976,5	234730,8	234769,4
0,00834	4	1584371,0	234745,3	234783,8
0,00915	4	1597054,1	234876,4	234914,9
0,01005	4	1624374,0	235155,3	235193,8
0,01103	4	1670521,7	235615,9	235654,5
0,01210	4	1740678,6	236292,4	236330,9

Fonte: o autor.

parâmetros sejam otimizados para que apresentem o menor valor SQR. Para esta finalidade, desenvolvemos uma função no Matlab<sup>®</sup> para estimar os valores ótimos, que pode ser vista no Anexo B.6.

Nas Tabelas 11, 12 e 13 podemos ver que independente do método de avaliação ou conjunto de dados utilizados, o 5PL é melhor avaliado para as condições de testes estabelecidas neste trabalho.

Tabela 11 – Desempenho dos modelos logísticos em TURB1.

Modelo	Termos	SQR	AIC	BIC
3PL	3	688504,8	30276,0	30276,0
4PL	4	688338,4	30254,6	30283,2
5PL	5	662715,8	30171,2	30205,5

Fonte: o autor.

Tabela 12 – Desempenho dos modelos logísticos em TURB2.

<b>Modelo</b>	<b>Termos</b>	<b>SQR</b>	<b>AIC</b>	<b>BIC</b>
3PL	3	8119980,6	756393,5	756428,6
4PL	4	8113822,0	756359,5	756403,3
5PL	5	8079488,4	756159,9	756212,5

Fonte: o autor.

Tabela 13 – Desempenho dos modelos logísticos em TURB3.

<b>Modelo</b>	<b>Termos</b>	<b>SQR</b>	<b>AIC</b>	<b>BIC</b>
3PL	3	1574249,2	234637,9	234668,8
4PL	4	1574249,4	234639,9	234670,1
5PL	5	1565754,3	234553,0	234599,2

Fonte: o autor.

## 5.6 Modelo MLP

Para otimizar a regressão via MLP foi utilizada a *toolbox* nativa do Matlab® que serve para desenvolver e treinar redes MLP. Variamos a quantidade de neurônios ocultos de 1 até 10 e avaliamos os resultados para identificar qual a quantidade de neurônios que é melhor avaliada pelos critérios de avaliação utilizados. O *script* desenvolvido pode ser visto no Anexo B.7 e os resultados dos testes com a rede MLP podem ser vistos nas Tabelas 14, 15 e 16.

Tabela 14 – Desempenho do modelo MLP em TURB1.

<b>Neurônios</b>	<b>Termos</b>	<b>SQR</b>	<b>AIC</b>	<b>BIC</b>
1	4	688865,2	30256,3	30284,9
2	7	669510,0	30197,6	30243,3
3	10	659447,8	30169,8	30232,7
4	13	654101,6	30157,8	30237,8
5	16	653049,4	30160,2	30257,4
6	19	653876,3	30168,9	30283,3
7	22	652980,4	30171,9	30303,5
8	25	652441,6	30176,1	30324,8
9	28	652502,0	30182,3	30348,1
10	31	652249,9	30187,4	30370,4

Fonte: o autor.

## 5.7 Comparação entre os modelos apresentados

Nas Figuras 9, 10 e 11 são mostradas as curvas de potência obtidas pelo uso da metodologia PRISMA em cada conjunto de dados. Estão mostradas também as curvas cor-

Tabela 15 – Desempenho do modelo MLP em TURB2.

Neurônios	Termos	SQR	AIC	BIC
1	4	9047712,8	758176,9	758220,7
2	7	9020514,4	758023,1	758093,3
3	10	8069836,1	756113,0	756209,5
4	13	8066036,8	756096,6	756219,4
5	16	8069356,6	756121,4	756270,5
6	19	8062222,2	756086,2	756261,5
7	22	8061133,1	756085,7	756287,4
8	25	8060717,5	756089,3	756317,3
9	28	8060641,3	756094,8	756349,1
10	31	8060634,4	756102,0	756382,6

Fonte: o autor.

Tabela 16 – Desempenho do modelo MLP em TURB3.

Neurônios	Termos	SQR	AIC	BIC
1	4	1573872,4	234636,0	234674,5
2	7	1567915,9	234579,6	234641,2
3	10	1563947,1	234544,0	234628,7
4	13	1564084,6	234551,4	234659,3
5	16	1563566,1	234551,9	234683,0
6	19	1562815,9	234550,1	234704,2
7	22	1562684,6	234554,7	234732,0
8	25	1562112,6	234554,7	234755,1
9	28	1562497,3	234564,7	234788,2
10	31	1562032,4	234565,8	234812,5

Fonte: o autor.

respondentes ao intervalo de 1 desvio-padrão acima e abaixo da curva estimada. No caso, o desvio-padrão é o dos resíduos,  $\sigma_e$ , calculado após o modelo ter seus parâmetros finais estimados.

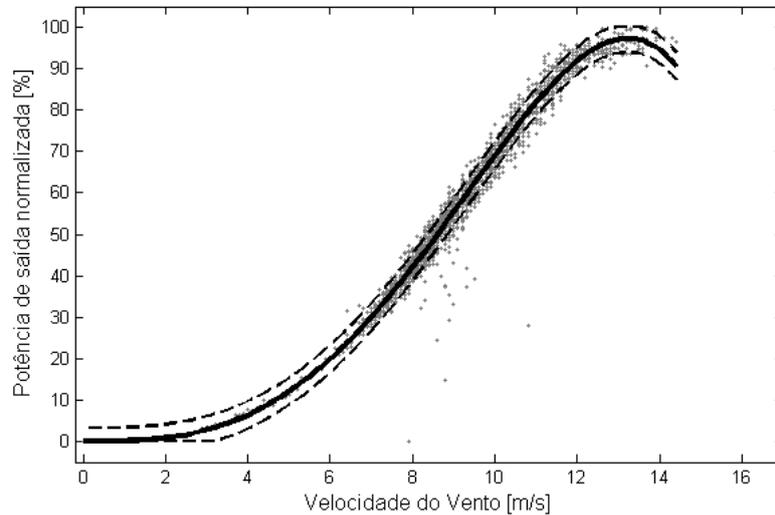
Vale ressaltar, que todos os outros modelos geram curvas de potência semelhantes às aquelas mostradas nas Figuras 9, 10 e 11, como era de se esperar, pois foram projetados para isso. Assim, evitamos colocar todas as curvas de potência juntas para não sobrecarregar a figura. Contudo, mesmo parecidos, os modelos tiveram desempenho geral muito diferentes a depender do critério de avaliação utilizado. Nos subtópicos a seguir iremos tratar de analisar cada caso.

### 5.7.1 SQR

O melhor desempenho de cada metodologia utilizada segundo o SQR é reunida nas Tabelas 17, 18 e 19. Cada uma trata de um dos conjuntos de dados utilizados neste trabalho.

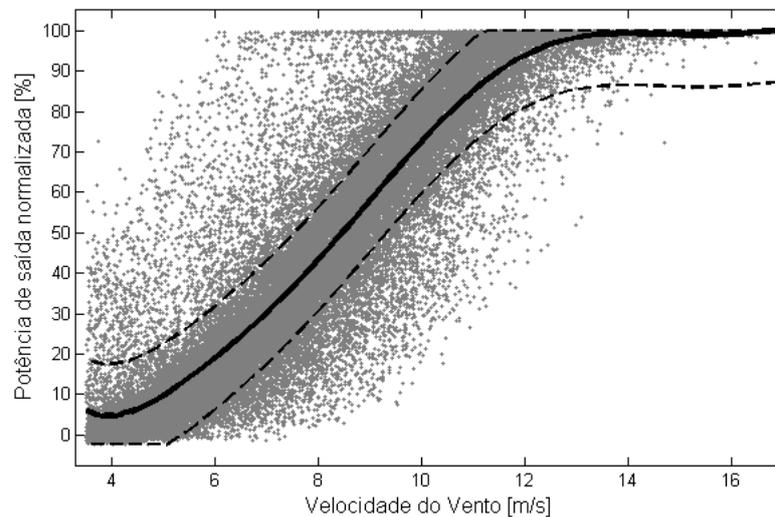
Nos três conjuntos de dados, os modelos que têm o melhor desempenho foram os que apresentam mais termos de regressão. Isso torna evidente a tendência de favorecer regressões

Figura 9 – Curva de potência do modelo PRISMA. Os pontos cinza são os dados do aerogerador TURB1 e a potência de saída foi normalizada pela potência nominal. As curvas tracejadas correspondem ao intervalo de  $\pm\sigma_\epsilon$ .



Fonte: o autor.

Figura 10 – Curva de potência do modelo PRISMA. Os pontos cinza são os dados do aerogerador TURB2 e a potência de saída foi normalizada pela potência nominal. As curvas tracejadas correspondem ao intervalo de  $\pm\sigma_\epsilon$ .

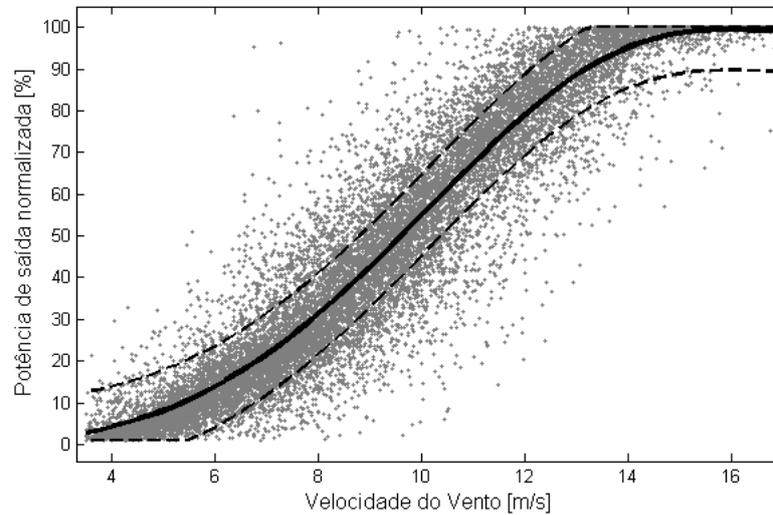


Fonte: o autor.

com maior quantidade de termos que modelos que utilizam o critério SQR como lastro de otimização possuem. O modelo ponderado não seguiu essa tendência pois sua otimização iterativa se divide em dois objetivos: além de procurar minimizar o SQR, tem como objetivo eliminar *outliers*.

Para obter o melhor resultado, como esperado, o modelo MLP utilizou o máximo de

Figura 11 – Curva de potência do modelo PRISMA. Os pontos cinza são os dados do aerogerador TURB3 e a potência de saída foi normalizada pela potência nominal. As curvas tracejadas correspondem ao intervalo de  $\pm\sigma_e$ .



Fonte: o autor.

Tabela 17 – Comparação de desempenho dos modelos quanto ao SQR em TURB1.

Modelo	Termos	SQR	Diferença acumulada
PRISMA	11	652019,5	-
Polinomial	11	652019,5	+0
MLP	31	652249,9	+230,4
Ponderado	9	657598,3	+5578,8
LASSO	5	660573,3	+8553,8
Logístico 5	5	662715,8	+10696,3
Logístico 4	4	688338,4	+36318,9
Logístico 3	3	688504,8	+36485,3

Fonte: o autor.

neurônios na camada oculta, resultando em 31 termos utilizados. O modelo polinomial obteve sua otimização em um polinômio de décimo grau, e o modelo PRISMA ativou todos os termos disponíveis em todos os conjuntos de dados, ficando igual ao polinomial.

### 5.7.2 AIC/BIC

O melhor desempenho de cada metodologia utilizada segundo o AIC é reunida nas Tabelas 20, 21 e 22, e segundo o BIC é reunida nas Tabelas 23, 24 e 25.

Ao analisar essas tabelas, percebe-se que o modelo PRISMA foi o que obteve melhor desempenho em todos os casos, pois foi o modelo que, simultaneamente, produziu menor

Tabela 18 – Comparação de desempenho dos modelos quanto ao SQR em TURB2.

<b>Modelo</b>	<b>Termos</b>	<b>SQR</b>	<b>Diferença acumulada</b>
MLP	31	8060634,4	-
PRISMA	11	8061526,3	+891,9
Polinomial	11	8061526,3	+891,9
Logístico 5	5	8079488,4	+18854
Logístico 4	4	8113822,0	+53187,6
Logístico 3	3	8119980,6	+59346,2
LASSO	4	8196949,1	+136314,7
Ponderado	10	8302835,6	+242201,2

Fonte: o autor.

Tabela 19 – Comparação de desempenho dos modelos quanto ao SQR em TURB3.

<b>Modelo</b>	<b>Termos</b>	<b>SQR</b>	<b>Diferença acumulada</b>
MLP	31	1562032,4	-
PRISMA	11	1563180,6	+1148,2
Polinomial	11	1563180,6	+1148,2
Logístico 5	5	1565754,3	+3721,9
Logístico 4	4	1574249,2	+12216,8
Logístico 3	3	1574249,4	+12217
LASSO	4	1582976,5	+20944,1
Ponderado	8	1587216,2	+25183,8

Fonte: o autor.

Tabela 20 – Comparação de desempenho dos modelos quanto ao AIC em TURB1.

<b>Modelo</b>	<b>Termos</b>	<b>AIC</b>	<b>Diferença acumulada</b>
PRISMA	6	30139,1	-
Polinomial	9	30144,6	+5,5
MLP	13	30157,8	+18,7
LASSO	10	30159,2	+20,1
Ponderado	9	30161,8	+22,7
Logístico 5	5	30171,2	+32,1
Logístico 4	4	30254,6	+115,5
Logístico 3	3	30276,0	+136,9

Fonte: o autor.

AIC/BIC com uma quantidade relativamente pequena de parâmetros.

Podemos confirmar também, como mencionado na seção 3.5, a tendência do AIC apresentar melhores resultados para modelos com mais termos se comparado ao BIC. Tomemos como exemplo a regressão pelo modelo PRISMA em TURB1 que resulta em 6 termos quando

Tabela 21 – Comparação de desempenho dos modelos quanto ao AIC em TURB2.

Modelo	Termos	AIC	Diferença acumulada
PRISMA	10	756065,9	-
Polinomial	11	756066,0	+0,1
MLP	22	756085,7	+19,8
Logístico 5	5	756159,9	+94
Logístico 4	4	756359,5	+293,6
Logístico 3	3	756393,5	+327,6
LASSO	4	756844,1	+778,2
Ponderado	5	757463,7	+1397,8

Fonte: o autor.

Tabela 22 – Comparação de desempenho dos modelos quanto ao AIC em TURB3.

Modelo	Termos	AIC	Diferença acumulada
PRISMA	5	234532,9	-
Polinomial	8	234533,3	+0,4
MLP	10	234544,0	+11,1
Logístico 5	5	234553,0	+20,1
Logístico 3	3	234637,9	+105
Logístico 4	4	234639,9	+107
LASSO	4	234730,8	+197,9
Ponderado	8	234782,8	+249,9

Fonte: o autor.

Tabela 23 – Comparação de desempenho dos modelos quanto ao BIC em TURB1.

Modelo	Termos	BIC	Diferença acumulada
PRISMA	2	30168,6	-
Polinomial	5	30187,5	+18,9
LASSO	5	30198,3	+29,7
Ponderado	5	30200,2	+31,6
Logístico 5	5	30205,5	+36,9
MLP	10	30232,7	+64,1
Logístico 3	3	30253,1	+84,5
Logístico 4	4	30283,2	+114,6

Fonte: o autor.

utiliza AIC como função objetivo e 2 termos quando utiliza BIC.

A título de ilustração, podemos apresentar a regressão resultante do modelo PRISMA em TURB1 utilizando BIC. Com dois termos de ativação, sendo eles  $\{\zeta_3, \zeta_5\}$ , de modo que os coeficientes estimados foram  $\{\hat{\beta}_3 = 0,1049859009; \hat{\beta}_5 = -0,0003598923\}$ , resultando na

Tabela 24 – Comparação de desempenho dos modelos quanto ao BIC em TURB2.

Modelo	Termos	BIC	Diferença acumulada
PRISMA	9	756155,9	-
Polinomial	9	756158,1	+2,2
MLP	10	756209,5	+53,6
Logístico 5	5	756212,5	+56,6
Logístico 4	4	756403,3	+247,4
Logístico 3	3	756428,6	+272,7
LASSO	4	756887,9	+732
Ponderado	5	757516,3	+1360,4

Fonte: o autor.

Tabela 25 – Comparação de desempenho dos modelos quanto ao BIC em TURB3.

Modelo	Termos	BIC	Diferença acumulada
PRISMA	4	234526,3	-
Logístico 5	5	234599,2	+72,9
Polinomial	8	234602,7	+76,4
MLP	10	234628,7	+102,4
Logístico 3	3	234668,8	+142,5
Logístico 4	4	234670,1	+152,2
LASSO	4	234769,4	+243,1
Ponderado	6	234837,2	+310,9

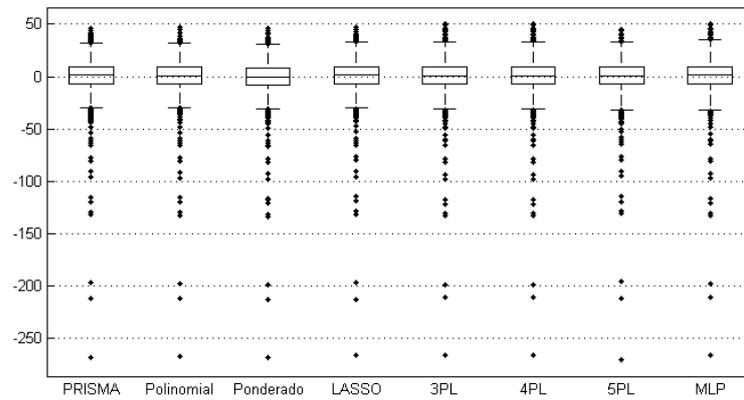
Fonte: o autor.

Figura 9, preteritamente apresentada, e na equação

$$p(v) = 0,1049859009v^3 - 0,0003598923v^5 \quad (5.2)$$

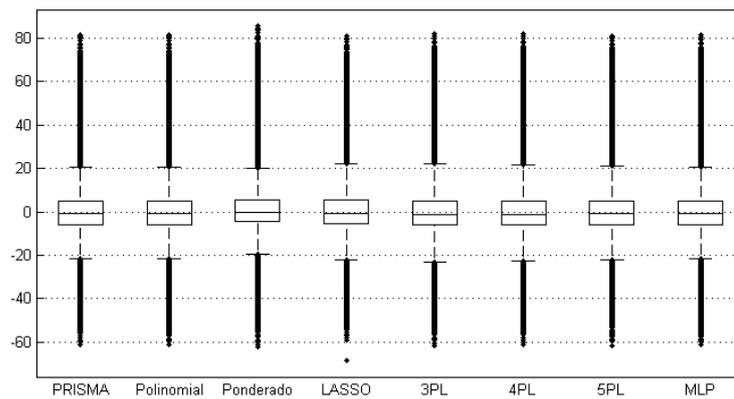
A título de completude, mostramos nas Figuras 12, 13 e 14 os boxplots dos resíduos para todos os modelos avaliados. Como era de se esperar, para o tipo de aplicação que estamos abordando, há muitos *outliers* presentes nos dados (SOHONI *et al.*, 2016; SHOKRZADEH *et al.*, 2014). A presença de *outliers* acaba interferindo negativamente no processo de estimação de parâmetros de todos os modelos e gerando assim uma distribuição de resíduos que também apresenta *outliers*, conforme pode ser verificado nas referidas figuras. O mais importante aqui é que o modelo PRISMA obteve o melhor desempenho, mesmo em um cenário com vários *outliers*.

Figura 12 – Boxplots dos resíduos para cada um dos modelos utilizados em TURB1.



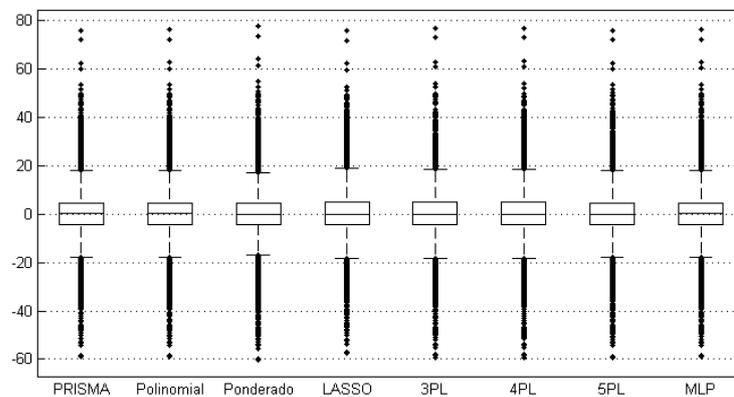
Fonte: o autor.

Figura 13 – Boxplots dos resíduos para cada um dos modelos utilizados em TURB2.



Fonte: o autor.

Figura 14 – Boxplots dos resíduos para cada um dos modelos utilizados em TURB3.



Fonte: o autor.

## 6 CONCLUSÕES E TRABALHOS FUTUROS

### 6.1 Conclusões

O trabalho desenvolvido nessa dissertação tratou essencialmente de apresentar modelos de regressão aplicados à estimação da curva de potência de aerogeradores que visassem estabelecer uma solução mais parcimoniosa que modelos comumente utilizados. Além disso, é importante que os modelos propostos apresentados ainda sejam competitivos quanto a capacidade de minimizar o residual quadrático em relação aos demais modelos. Para tal, fez-se um amplo levantamento bibliográfico e, tomada confirmação da vacância de uma metodologia esparsa de seleção automática de parâmetros, aplicamos o método de regularização LASSO para estimação da curva de potência de aerogeradores. Ao passo que foi verificado o bom funcionamento do LASSO, desenvolvemos uma nova metodologia de regressão esparsa baseada em algoritmos genéticos, que chamamos de PRISMA.

Utilizamos 3 conjuntos de dados distintos para observar as resultantes da regressão de cada modelo elencado nesta dissertação. Avaliamos e comparamos todos os resultados através do SQR, AIC e BIC, resultando em 9 tipos de teste. Constatamos que dos 7 casos, o PRISMA foi o modelo melhor avaliado, e nos outros dois restantes ficou em segundo lugar, sendo superado pelo MLP.

Enquanto não organizamos uma página na internet para tal, todos os códigos que foram desenvolvidos nesta dissertação serão disponibilizados a quem interessar através de uma solicitação por e-mail, mediante as devidas referências no caso de se tornar útil.

### 6.2 Trabalhos Futuros

Como trabalho futuro pensamos em desenvolver um novo método de regressão que use modelos polinomiais fracionários para estimação da curva de potência. Isto é, um modelo em que não só a quantidade de termos do polinômio, mas também os expoentes fracionários, sejam determinados por otimização metaheurística (ROYSTON, 2017; BAGNARDI *et al.*, 2004).

Ainda que, tanto na indústria de aerogeradores quanto na academia, a prática no desenvolvimento de modelos da curva de potência leve em consideração apenas a velocidade do vento (e, às vezes, a direção do vento), estamos atualmente refinando o modelo proposto a fim de introduzir outras variáveis de entrada, além da velocidade do vento, que influenciam no valor da

potência gerada pelo aerogerador, tais como direção, densidade do ar, umidade, intensidade da turbulência, etc.

Além dos supracitados, ainda podemos avaliar o uso de modelos de aprendizado de máquinas baseados em kernel, como *Support Vector Regression* (SANTOS; BARRETO, 2017) e Processos Gaussianos (MATTOS *et al.*, 2017).

## REFERÊNCIAS

- AKAIKE, H. A new look at the statistical model identification. **IEEE Transactions on Automatic Control**, v. 19, n. 6, p. 716–723, 1974.
- BAGNARDI, V.; ZAMBON, A.; QUATTO, P.; CORRAO, G. Flexible meta-regression functions for modeling aggregate dose-response data, with an application to alcohol and mortality. **American Journal of Epidemiology**, v. 159, n. 11, p. 1077–1086, 2004.
- BARRETO, G. A.; BARROS, A. L. B. P. On the design of robust linear pattern classifiers based on m-estimators. **Neural Processing Letters**, v. 42, n. 1, p. 119–137, 2015.
- BETZ, A. Das maximum der theoretisch möglichen ausnützung des windes durch windmotoren. **Zeitschrift für das gesamte Turbinenwesen**, v. 26, p. 307–309, 1920.
- BOYD, S.; VANDENBERGHE, L. **Convex Optimization**. 6th ed.. ed. [S.l.]: Cambridge University Press, 2004.
- BURNHAM, K. P.; ANDERSON, D. R. Multimodel inference understanding AIC and BIC in model selection. **Sociological Methods and Research**, v. 33, n. 2, p. 261–304, 2004.
- CHANG, T. P.; LIU, F. J.; KO, H. H.; CHENG, S. P.; SUN, L. C.; KUO, S. C. Comparative analysis on power curve models of wind turbine generator in estimating capacity factor. **Energy**, v. 73, p. 88–95, 2014.
- CLIFTON, A.; KILCHER, L.; LUNDQUIST, J. K.; FLEMING, P. Using machine learning to predict wind turbine power output. **Environmental Research Letters**, n. 8, p. 1–8, 2013.
- DELEEUW, J. Introduction to akaike (1973) information theory and an extension of the maximum likelihood principle. In: KOTZ, S.; JOHNSON, N. L. (Ed.). **Breakthroughs in Statistics: Foundations and Basic Theory**. New York: Springer-Verlag, 1992. (Springer Series in Statistics, Perspectives in Statistics, v. 1), p. 599–609.
- DIAF, S.; BELHAMEL, M.; HADDADI, M.; LOUCHE, A. Technical and economic assessment of hybrid photovoltaic/wind system with battery storage in corsica island. **Energy Policy**, v. 36, n. 2, p. 743–754, 2008.
- DZIAK, J. J.; COFFMAN, D. L.; LANZA, S. T.; LI, R. **Sensitivity and specificity of information criteria**. University Park, PA: The Methodology Center, 2012.
- ENGELBRECHT, A. P. **Computational Intelligence: An Introduction**. 2nd. ed. [S.l.]: Wiley, 2007.
- GILL, B. S. S.; GALLOWAY, S. Wind turbine condition assessment through power curve copula modeling. **IEEE Transactions on Sustainable Energy**, v. 3, n. 1, p. 94–101, 2012.
- GOTTSCHALK, P. G.; DUNN, J. R. The five-parameter logistic: A characterization and comparison with the four-parameter logistic. **Analytical Biochemistry**, v. 343, n. 1, p. 54–65, 2005.
- HAGAN, M. T.; MENHAJ, M. Training feed-forward networks with the Marquardt algorithm. **IEEE Transactions on Neural Networks**, v. 5, n. 6, p. 989–993, 1994.

- HAYKIN, S. **Neural Networks and Learning Machines**. 3rd. ed. [S.l.]: Pearson Education, Inc., 2008.
- HEDEVANG, E. Wind turbine power curves incorporating turbulence intensity. **Wind Energy**, v. 17, n. 2, p. 173–195, 2014.
- HORVATH, L.; PANZA, T.; KARADZA, N. The influence of high wind hysteresis effect on wind turbine power production at bura-dominated site. **European Wind Energy Conference and Exhibition**, p. 1017–1022, 2007.
- HUBER, P. J. Robust estimation of a location parameter. **Annals of Mathematical Statistics**, v. 35, n. 1, p. 73–101, 1964.
- IEC. **Wind turbines - Power performance measurement of electricity producing wind turbines**. Geneva, Suíça, 2005.
- JIN, T.; TIAN, Z. Uncertainty analysis for wind energy production with dynamic power curves. **IEEE 11th International Conference on Probabilistic Methods Applied to Power Systems**, p. 745–750, 2010.
- JOWDER, F. A. L. Wind power analysis and site matching of wind turbine generators in kingdom of bahrain. **Applied Energy**, v. 86, n. 4, p. 538–545, 2009.
- KUSIAK, A.; ZHENG, H.; SONG, Z. On-line monitoring of power curves. **Renewable Energy**, v. 34, n. 6, p. 1487–1493, 2009.
- LEE, G.; DING, Y.; GENTON, M. G.; XIE, L. Power curve estimation with multivariate environmental factors for inland and offshore wind farms. **Journal of the American Statistical Association**, v. 110, p. 56–67, 2015.
- LI, S.; WUNSCH, D. C.; O’Hair, E.; GIESSELMANN, M. G. Comparative analysis of regression and artificial neural network models for wind turbine power curve estimation. **Journal of Solar Energy Engineering**, v. 123, n. 4, p. 327–332, 2001.
- LYDIA, M.; KUMAR, S. S.; SELVAKUMAR, A. I.; KUMAR, G. E. P. A comprehensive review on wind turbine power curve modeling techniques. **Renewable and Sustainable Energy Reviews**, v. 30, p. 452–460, 2014.
- LYDIA, M.; Suresh Kumar, S.; Immanuel Selvakumar, A.; Edwin Prem Kumar, G. Advanced algorithms for wind turbine power curve modeling. **IEEE Transactions on Sustainable Energy**, v. 4, n. 3, p. 827, 2013.
- MANWELL, J.; MCGOWAN, J.; ROGERS, A. **Wind energy explained : theory, design, and application**. 2nd. ed. [S.l.]: John Wiley & Sons, Ltd., 2009.
- MARCIUKAITIS, M.; ZUTAUTAITE, I.; MARTISAUSKAS, L.; JOKSAS, B.; GECEVICIUS, G.; SFETSOS, A. Non-linear regression model for wind turbine power curve. **Renewable Energy**, v. 113, p. 732–741, 2017.
- MATHEW, S. **Wind Energy-Fundamentals, Resource Analysis and Economics**. [S.l.]: Springer, 2006.
- MATTOS, C. L. C.; DAI, Z.; DAMIANOU, A.; BARRETO, G. A. Deep recurrent gaussian processes for outlier-robust system identification. **Journal of Process Control**, p. 82–94, 2017.

- MCLEAN, J. R. **WP2.6-Equivalent wind power curves**. [S.l.], 2008. Tech. Rep. EIE/06/022/SI2.442659.
- OLAOFE, Z. O.; FOLLY, K. A. Wind energy analysis based on turbine and developed site power curves: a case-study of darling city. **Renewable Energy**, v. 53, p. 306–318, 2013.
- POWELL, W. R. An analytical expression for the average output power of a wind machine. **Solar Energy**, v. 26, n. 1, p. 77–80, 1981.
- ROUSSEEUW, P. J.; CROUX, C. Alternatives to the median absolute deviation. **Journal of the American Statistical Association**, v. 88, n. 424, p. 1273–1283, 1993.
- ROYSTON, P. Model selection for univariable fractional polynomials. **The Stata journal**, v. 17, n. 3, p. 619–629, 2017.
- RUCKSTUHL, A. **Robust Fitting of Parametric Models Based on M-Estimation**. [S.l.], 2016.
- SANTOS, J. D. A.; BARRETO, G. A. A regularized estimation framework for online sparse lssvr models. **Neurocomputing**, v. 238, p. 114–125, 2017.
- SANTOS, J. D. de A. **Sobre a estimação do número de neurônios ocultos da rede MLP: uma nova técnica baseada em PCA e SVD**. Tese (Doutorado) — Universidade Federal do Ceará, 2010.
- SCHLECHTINGEN, M.; SANTOS, I. F.; ACHICHE, S. Using datamining approaches for wind turbine power curve monitoring: a comparative study. **IEEE Transactions on Sustainable Energy**, v. 4, n. 3, p. 671–679, 2013.
- SCHMIDT, M. **Least Squares Optimization with L1-Norm Regularization**. [S.l.], 2005. CS542B Project Report. Disponível em: <[https://www.cs.ubc.ca/~schmidtm/Documents/2005\\_Notes\\_Lasso.pdf](https://www.cs.ubc.ca/~schmidtm/Documents/2005_Notes_Lasso.pdf)>.
- SCHWARZ, G. E. Estimating the dimension of a model. **Annals of Statistics**, v. 6, n. 2, p. 461–464, 1978.
- SEASHOLTZ, M. B.; KOWALSKI, B. The parsimony principle applied to multivariate calibration. **Analytica Chimica Acta**, v. 277, n. 2, p. 165–177, 1993.
- SHOKRZADEH, S.; JOZANI, M. J.; BIBEAU, E. Wind turbine power curve modeling using advanced parametric and nonparametric methods. **IEEE Transactions on Sustainable Energy**, v. 5, n. 4, p. 1262–1269, 2014.
- SOHONI, V.; GUPTA, S. C.; NEMA, R. K. A critical review on wind turbine power curve modelling techniques and their applications in wind based energy systems. **Journal of Energy**, v. 2016, p. 1–18, 2016.
- STEPHEN, B.; GALLOWAY, S. J.; MCMILLAN, D.; HILL, D. C.; INFELD, D. G. A copula model of wind turbine performance. **IEEE Transactions on Power Systems**, v. 26, n. 2, p. 965–966, 2011.
- STORN, R.; PRICE, K. Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces. **Journal of Global Optimization**, v. 11, n. 4, p. 341–359, 1997.

SUSANTI, Y.; PRATIWI, H.; H., S. S.; LIANA, T. M estimation, s estimation and mm estimation in robust regression. **International Journal of Pure and Applied Mathematics**, v. 91, n. 3, p. 349–360, 2014.

TIBSHIRANI, R. Regression shrinkage and selection via the lasso. **Journal of the Royal Statistical Society - Series B**, v. 58, n. 1, p. 267–288, 1996.

USTUNTAS, T.; SAHIN, A. D. Wind turbine power curve estimation based on cluster center fuzzy logic modeling. **Journal of Wind Engineering and Industrial Aerodynamics**, v. 96, n. 5, p. 611–620, 2008.

WANG, J.; ZHAO, X.; GUO, X.; LI, B. Analyzing the research subjects and hot topics of power system reliability through the Web of Science from 1991 to 2015. **Renewable and Sustainable Energy Reviews**, v. 82, n. 1, p. 700–713, 2018.

## APÊNDICE A – DESENVOLVIMENTO DAS PRINCIPAIS EQUAÇÕES

### A.1 Função de Densidade de Probabilidade (FDP) de $p$

Seja  $f$  uma função de regressão<sup>1</sup> qualquer que relaciona a variável dependente  $p$  à varável independente  $v$  de acordo com os  $k_\beta$  coeficientes de regressão do vetor-coluna<sup>2</sup>  $\beta = [\beta_1 \ \beta_2 \ \dots \ \beta_{k_\beta}]^\top$ , por tanto

$$p = f(v|\beta) + \varepsilon \quad (\text{A.1})$$

onde  $\varepsilon$  é o residual da regressão que assumimos ser uma variável aleatória independente, de distribuição normal<sup>3</sup> com média nula e variância fixa igual a  $\sigma_\varepsilon^2$ .

Como  $f$  é uma função determinística e  $\varepsilon$  tem distribuição normal,  $p$  também terá distribuição normal e independente. Assim, o valor esperado de  $p$  é

$$\begin{aligned} E[p] &= E[f(v|\beta) + \varepsilon] \\ &= E[f(v|\beta)] + E[\varepsilon] \\ &= E[f(v|\beta)] + 0 \\ &= \mu \end{aligned} \quad (\text{A.2})$$

em que  $\mu$  é o valor esperado de potência elétrica, que se dá em função da velocidade esperada do vento. E a sua variância é

$$\begin{aligned} \text{Var}[p|v] &= \text{Var}[f(v|\beta) + \varepsilon] \\ &= \text{Var}[f(v|\beta)] + \text{Var}[\varepsilon] \\ &= 0 + \sigma_\varepsilon^2 \\ &= \sigma_\varepsilon^2 \\ &= \sigma^2 \end{aligned} \quad (\text{A.3})$$

Como variância de uma equação determinística é zero, a variância de  $p$ , denotada por  $\sigma^2$ , é igual a do residual.

<sup>1</sup> Trata-se de uma função que tem como saída o valor esperado de uma variável, conhecida como variável dependente, e como argumento uma ou mais variáveis, conhecidas como variáveis independentes.

<sup>2</sup> Assumimos que todos os vetores são vetores-coluna, salvo indicação contrária.

<sup>3</sup> É uma distribuição de probabilidade contínua, parametrizada pela sua esperança matemática  $\mu$  e seu desvio padrão  $\sigma$ . Sua PDF é definida por  $f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{(x-\mu)^2}{2\sigma^2}\right]$ .

A partir do que foi obtido preteritamente nessa sessão, podemos definir a FDP de  $p$ , que denotaremos por  $g$ , como

$$g(p|\theta) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp \left[ -\frac{(p-\mu)^2}{2\sigma^2} \right] \quad (\text{A.4})$$

onde  $\theta = (\beta, \sigma^2)$  é seu vetor de parâmetros de dependência com cardinalidade  $k = k_\beta + 1$ .

Perceba que não temos como parâmetro de  $g$  o valor esperado da potência  $\mu$ . Isso ocorre pois  $\mu = E[f(v|\beta)]$ , assim é dependente de  $\beta$ . Portanto, além dos coeficientes de regressão,  $g$  também depende da variância do residual para ser bem definido.

## A.2 Método da Máxima Verossimilhança

Seja  $x$  uma variável aleatória com uma FDP  $X(x|\theta_x)$  qualquer dependente dos parâmetros contidos no vetor  $\theta_x$ . Considere que se tenha um conjunto amostral independente de tamanho  $n$  da variável aleatória  $x$ . Podemos avaliar quantitativamente um determinado conjunto de parâmetros  $\theta_x$  através da função de verossimilhança, definida por

$$\begin{aligned} \mathcal{L}(\theta_x|x_1, x_2, \dots, x_n) &= X(x_1|\theta_x) \times X(x_2|\theta_x) \times \dots \times X(x_n|\theta_x) \\ &= \prod_{i=1}^n X(x_i|\theta_x) \end{aligned}, \quad (\text{A.5})$$

de forma que quanto maior o valor resultante da função de verossimilhança para um determinado conjunto de parâmetros  $\theta_x$ , mais este conjunto proporciona uma FDP próxima à realidade.

Porém, ao invés de estabelecer um conjunto de parâmetros arbitrariamente, podemos obter o ponto de máximo da função de verossimilhança em função de  $\theta_x$ , caso exista, obtendo o valor ótimo para os parâmetros. Este método de estimação de parâmetros é conhecido como Método da Máxima Verossimilhança (MMV).

## A.3 Parametrização da FDP de $p$

Suponha que tenhamos coletado  $n$  amostras distintas  $(p_i, v_i)$ ,  $i = 1 \dots n$ , referentes às velocidades de vento e sua respectiva potência elétrica gerada. Podemos utilizá-las para fazer uma estimativa de  $\mu$  e  $\sigma^2$  de  $g$  através do MMV. Sendo a função de verossimilhança para  $g$  dado

por

$$\begin{aligned}
 \mathcal{L}(\theta|p) &= \prod_{i=1}^n g(p_i|\theta) \\
 &= \prod_{i=1}^n \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{(p_i - \mu)^2}{2\sigma^2}\right] \\
 &= \left(\frac{1}{\sqrt{2\pi\sigma^2}}\right)^n \exp\left[-\frac{1}{2\sigma^2} \sum_{i=1}^n (p_i - \mu)^2\right] \\
 &= (2\pi\sigma^2)^{-\frac{n}{2}} \exp\left[-\frac{1}{2\sigma^2} \sum_{i=1}^n (p_{ei} - \mu)^2\right]
 \end{aligned} \tag{A.6}$$

em que  $p$  é o vetor definido por

$$p = \begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_n \end{bmatrix} \tag{A.7}$$

O objetivo é maximizar o seu resultado em função do valor de seus parâmetros. Visto o termo exponencial da Eq. A.6, é mais conveniente trabalharmos com uma equação equivalente em termos do logaritmo natural, conhecida como função log-verossimilhança, desenvolvida a seguir

$$\begin{aligned}
 L(\theta|p) &= \ln[\mathcal{L}(\theta|p)] \\
 &= \ln\left\{ (2\pi\sigma^2)^{-\frac{n}{2}} \exp\left[-\frac{1}{2\sigma^2} \sum_{i=1}^n (p_i - \mu)^2\right] \right\} \\
 &= \ln\left[(2\pi\sigma^2)^{-\frac{n}{2}}\right] + \ln\left\{ \exp\left[-\frac{1}{2\sigma^2} \sum_{i=1}^n (p_i - \mu)^2\right] \right\} \\
 &= -\frac{n}{2} \ln(2\pi\sigma^2) - \frac{1}{2\sigma^2} \sum_{i=1}^n (p_i - \mu)^2 \\
 &= -\frac{n}{2} \ln(2\pi) - \frac{n}{2} \ln(\sigma^2) - \frac{1}{2\sigma^2} \sum_{i=1}^n (p_i - \mu)^2
 \end{aligned} \tag{A.8}$$

Podemos obter o máximo da função de log-verossimilhança através das técnicas das derivadas em relação aos parâmetros  $\mu$  e  $\sigma^2$ . Assim, faremos a derivada parcial de  $L(\theta|p)$  para cada um dos parâmetros e a igualaremos a zero para achar seu(s) ponto(s) crítico(s). Em seguida, faz-se necessário fazer a segunda derivada parcial para avaliarmos o comportamento da função em relação ao parâmetro selecionado e confirmar se o ponto crítico encontrado é, como desejado, o de máximo da função.

A derivada de  $L(\theta|p)$  em relação a  $\mu$  é

$$\begin{aligned}
 \frac{\partial L(\theta|p)}{\partial \mu} &= \frac{1}{\sigma^2} \sum_{i=1}^n (p_i - \mu) = 0 \\
 \therefore \sum_{i=1}^n (p_i - \mu) &= 0 \\
 \therefore \sum_{i=1}^n (p_i) - \sum_{i=1}^n (\mu) &= 0 \\
 \therefore \sum_{i=1}^n (p_i) - n\mu &= 0 \\
 \therefore \mu &= \frac{1}{n} \sum_{i=1}^n p_i = \bar{p}
 \end{aligned} \tag{A.9}$$

Temos apenas um ponto crítico em  $L(\theta|p)$  em relação a  $\mu$ . Através da segunda derivada podemos classificar o comportamento da função.

$$\frac{\partial^2 L(\theta|p)}{\partial \mu^2} = -\frac{n}{\sigma^2} < 0 \tag{A.10}$$

Como  $n$  é um número inteiro não nulo, por tanto positivo, e  $\sigma^2$  é um número real positivo, a segunda derivada de  $L(\theta|p)$  em relação a  $\mu$  sempre será negativa. Ou seja,  $L(\theta|p)$  é uma função côncava em relação a  $\mu$ , por tanto a primeira derivada apresenta o único ponto de máximo da função.

A derivada de  $L(\theta|p)$  em relação a  $\sigma^2$  é

$$\begin{aligned}
 \frac{\partial L(\theta|p)}{\partial \sigma^2} &= -\frac{n}{2\sigma^2} + \frac{1}{2\sigma^4} \sum_{i=1}^n (p_i - \mu)^2 = 0 \\
 \therefore \frac{1}{2\sigma^4} \left[ -n\sigma^2 + \sum_{i=1}^n (p_i - \mu)^2 \right] &= 0 \\
 \therefore -n\sigma^2 + \sum_{i=1}^n (p_i - \mu)^2 &= 0 \\
 \therefore \sigma^2 &= \frac{1}{n} \sum_{i=1}^n (p_i - \mu)^2
 \end{aligned} \tag{A.11}$$

Onde também temos apenas um ponto crítico em  $L(\theta|p)$  em relação a  $\sigma^2$ . Vejamos o comporta-

mento da função através da segunda derivada.

$$\begin{aligned}
\frac{\partial^2 L(\theta|p)}{\partial(\sigma^2)^2} &= \frac{n}{2\sigma^4} - \frac{1}{\sigma^6} \sum_{i=1}^n (p_i - \mu)^2 \\
&= \frac{1}{\sigma^4} \left[ \frac{n}{2} - \frac{1}{\sigma^2} \sum_{i=1}^n (p_i - \mu)^2 \right] \\
&= \frac{1}{\sigma^4} \left\{ \frac{n}{2} - \frac{1}{\left[ \frac{1}{n} \sum_{i=1}^n (p_i - \mu)^2 \right]} \sum_{i=1}^n (p_i - \mu)^2 \right\} \\
&= \frac{1}{\sigma^4} \left[ \frac{n}{2} - n \right] \\
&= -\frac{n}{2\sigma^4} < 0
\end{aligned} \tag{A.12}$$

Considerando que  $\sigma^2 = \frac{1}{n} \sum_{i=1}^n (p_i - \mu)^2$ , podemos afirmar que a segunda derivada parcial de  $L(\theta|p)$  sempre será negativa. Ou seja,  $L(\theta|p)$  é uma função côncava em relação a  $\sigma^2$ , por tanto a primeira derivada apresenta o único ponto de máximo da função.

Com os pontos críticos desenvolvidos preteritamente, podemos desenvolver A.8 para achar o valor de máxima verossimilhança dado uma amostra:

$$\begin{aligned}
L(\theta|p) &= -\frac{n}{2} \ln(2\pi) - \frac{n}{2} \ln(\sigma^2) - \frac{1}{2\sigma^2} \sum_{i=1}^n (p_i - \mu)^2 \\
&= -\frac{n}{2} \ln(2\pi) - \frac{n}{2} \ln \left[ \frac{1}{n} \sum_{i=1}^n (p_i - \mu)^2 \right] - \frac{1}{2 \left[ \frac{1}{n} \sum_{i=1}^n (p_i - \mu)^2 \right]} \sum_{i=1}^n (p_i - \mu)^2 \\
&= -\frac{n}{2} \ln(2\pi) + \frac{n}{2} \ln(n) - \frac{n}{2} \ln \left[ \sum_{i=1}^n (p_i - \mu)^2 \right] - \frac{n}{2} \\
&= -\frac{n}{2} \ln \left[ \sum_{i=1}^n (p_i - \mu)^2 \right] - \frac{n}{2} [\ln(2\pi) - \ln(n) + 1] \\
&= -\frac{n}{2} \ln \left[ \sum_{i=1}^n (p_i - \mu)^2 \right] - \frac{n}{2} \left[ \ln \left( \frac{2\pi e}{n} \right) \right] \\
&= -\frac{n}{2} \ln \left[ \sum_{i=1}^n (p_i - \bar{p})^2 \right] - \frac{n}{2} \ln \left( \frac{2\pi e}{n} \right) \\
&= -\frac{n}{2} \ln(SQR) + C
\end{aligned} \tag{A.13}$$

onde a soma dos quadrados dos resíduos (SQR) é definido por  $SQR = \sum_{i=1}^n (p_i - \bar{p})^2 = \sum_{i=1}^n \varepsilon_i^2$  e  $C = -\frac{n}{2} \ln \left( \frac{2\pi e}{n} \right)$  é um valor constante para um determinado conjunto de dados.

Podemos concluir que a máxima verossimilhança de uma amostra é inversamente diretamente proporcional ao valor do SQR que ela apresenta. Portanto, quanto menor o valor do SQR de uma amostra, melhor avaliada esta amostra será segundo a sua verossimilhança.

Neste trabalho iremos utilizar o mesmo conjunto de dados para comparar alguns métodos de regressão. Portanto, iremos suprimir o termo constante  $C$  da Eq. A.13.

## APÊNDICE B – CÓDIGOS-FONTES UTILIZADOS PARA...

### B.1 Função de aptidão

```

1 function [res, iBest, bBest, aptBest] = Aptidao(alphas, p,
    V, flagRobusta)
2 %% Função de aptidão
3 % Variáveis de entrada
4 % - alphas: Matriz com os alphas. Cada linha representa
    um arranjo de
5 % coeficientes que serão ativados a depender dos valores
    das colunas, que
6 % podem assumir os valores 0 ou 1. Se a coluna de índice
    i estiver em 1,
7 % o coeficiente i será utilizado na regressão. Caso seja
    0, não será.
8 % - p: Valor da potência mensurada.
9 % - V: Matriz de Vandermonde da velocidade mensurada.
10 % - flagRobusta: Indicativo se a regressão será robusta (
    IWLS)
11
12 %% Inicialização das variáveis de saída.
13 res = zeros(length(alphas), 1); % Aptidão da população.
14 iBest = 0; % Índice da melhor aptidão.
15 bBest = []; % Coeficientes da melhor aptidão.
16 aptBest = inf; % Melhor (menor) aptidão da população.
17
18 %% Início da análise dos conjuntos de alphas.
19 for i = 1:size(alphas,1) % Fazer para cada linha de
    alpha.
20     indices = find(alphas(i, :)); % Índice dos coef.
        utilizados.

```

```

21     nParam = length(indices); % Quantidade de coef.
        utilizados.
22     if nParam < 1 % Caso não tenha coeficientes,
        penalize.
23         apt = inf;
24     else % Caso tenha, seguir com o procedimento normal
        .
25         Vt = V(:, m); % Redefina a matriz V em função
            dos parâmetros utilizados.
26     %           Método alternativo: Zerar os parâmetros não
        utilizados.
27     %           Vt = zeros(size(V));
28     %           Vt(:, indices) = V(:, indices);
29
30     %% Resolver B.
31     if flagRobusta % Caso robusta, resolver B por
        IWLS.
32         B = robustfit(Vt,p,'bisquare', [], 'off');
33     else % Caso contrário, resolver B pelo método
        das derivadas utilizando decomposição QR.
34         if exist('B', 'var') == 1
35             clear B;
36         end
37         [Q, R, perm] = qr(Vt,0);
38         tol = abs(R(1)) * max(size(Vt)) * eps(class
            (R));
39         xrank = sum(abs(diag(R)) > tol);
40         if xrank == size(Vt,2)
41             B(perm, :) = R \ (Q'*p);
42         else
43             B(perm, :) = [R(1:xrank,1:xrank) \ (Q
                (:,1:xrank)')*p]; zeros(size(Vt,2)-

```

```

                                xrank,1)];
44         end
45     end
46
47     %% Cálculo de aptidão.
48     p_pred = Vt*B; % Potência predita.
49     erro = p - p_pred;
50     SSE = erro'*erro; % Soma dos erros quadráticos.
51     %         apt = SSE;
52     %         apt = AIC(nParam, length(p), SSE);
53     apt = BIC(nParam, length(p), SSE);
54     end
55     res(i) = apt; % Inserir valor de aptidão do
                    conjunto alpha i.
56     if apt < aptBest % Se for o melhor conjunto,
                    atualizar valores de saída.
57         iBest = i;
58         bBest = B;
59         aptBest = apt;
60     end
61 end
62 end

```

## B.2 Função AIC

```

1 function res = AIC(k, N, SSE)
2     res = N*log(SSE) + 2*(k+1);
3 end

```

## B.3 Função BIC

```

1 function res = BIC(k, N, SSE)
2     res = N*log(SSE) + log(N)*(k+1);
3 end

```

#### B.4 Teste de parâmetros dos modelos polinomial e robusto

```

1 %% Preparações preambulares.
2 clear all; close all;
3 %
4
5 %% Aquisição de dados.
6 dados = load('dados\Normalized_WT1.dat'); % Carregar dados
7
8 v = dados(:, 1); % Vetor de velocidade do vento.
9 p = dados(:, 2); % Vetor de potência gerada.
10 n = numel(v); % Tamanho dos vetores.
11 %
12 %% Parâmetros.
13 k = 10; % Maior grau testado.
14 termos = k + 1; % Cardinalidade do conjunto regressor.
15 %
16
17 %% Declarar variáveis.
18 sqrPol = zeros(termos,1);
19 aicPol = zeros(termos,1);
20 bicPol = zeros(termos,1);
21 sqrRob = zeros(termos,1);
22 aicRob = zeros(termos,1);
23 bicRob = zeros(termos,1);
24 %
25

```

```
26 %% Início dos testes.
27 for i = 1 : termos
28     %% Preparar dados.
29     V = ones(n, i); % Matriz de Vandermonde de v.
30     for j = 2:i
31         V(:, j) = v.*V(:, j-1);
32     end
33     %
34
35     %% Resolver B polinomial.
36     [Q, R] = qr(V,0); % Decomposição matricial QR.
37     BPol = R \ (Q'*p);
38
39
40     %% Resolver B robusto.
41     BRob = robustfit(V,p, 'bisquare', [], 'off');
42     %
43
44     %% Analisar resultados.
45     p_hat = V*BPol;
46     erro = p - p_hat;
47     SQR = erro'*erro;
48
49     sqrPol(i) = SQR;
50     aicPol(i) = AIC(i, length(v), SQR);
51     bicPol(i) = BIC(i, length(v), SQR);
52
53     p_hat = V*BRob;
54     erro = p - p_hat;
55     SQR = erro'*erro;
56
57     sqrRob(i) = SQR;
```

```

58     aicRob(i) = AIC(i, length(v), SQR);
59     bicRob(i) = BIC(i, length(v), SQR);
60 end
61 %
62
63 %% Apresentar dados
64 for i = 0:k;
65     fprintf('%d %.1f %.1f %.1f %.1f %.1f %.1f\n', i, sqrPol
        (i+1), aicPol(i+1), bicPol(i+1), sqrRob(i+1), aicRob
        (i+1), bicRob(i+1))
66 end

```

## B.5 Teste de parâmetros do modelo LASSO

```

1 %% Preparações preambulares.
2 clear all; close all;
3 %
4
5 %% Aquisição de dados.
6 dados = load('dados\Normalized_WT1.dat'); % Carregar dados
7
8 v = dados(:, 1); % Vetor de velocidade do vento.
9 p = dados(:, 2); % Vetor de potência gerada.
10 n = numel(v); % Tamanho dos vetores.
11 k = 10; % Maior grau do polinômio.
12 %
13 %% Preparar dados.
14 V = ones(n, k + 1); % Matriz de Vandermonde de v.
15 for i= 2:(k + 1)
16     V(:, i) = v.*V(:, i-1);
17 end

```

```
18 %
19
20 %% Resolver LASSO.
21 [B, S] = lasso(V, p, 'LambdaRatio', 0);
22 %
23
24 %% Analisar resultados.
25 for i=1:size(B,2)
26     p_hat = V*B(:,i);
27     erro = p - p_hat;
28     nParam = S.DF(i);
29     SQR = erro'*erro;
30
31     sqr(i) = SQR;
32     aic(i) = AIC(nParam, length(v), sqr(i));
33     bic(i) = BIC(nParam, length(v), sqr(i));
34 end
35 %
36
37 %% Apresentar resultado gráficos
38
39 figure; % Curva da aptidão em função do valor de  $\lambda$ 
        para o modelo LASSO.
40 semilogx(S.Lambda, bic, 'r-', 'linewidth', 4)
41 hold on;
42 semilogx(S.Lambda, bic, 'b-', 'linewidth', 2)
43 hold off;
44 xlabel('\lambda', 'fontsize', 12);
45 ylabel('Aptidão', 'fontsize', 12);
46
47 figure; % Quantidade de termos utilizados em função de  $\lambda$ 
        para o modelo LASSO.
```

```
48 stem(S.Lambda, S.DF, 'r-', 'linewidth', 2)
49 set(gca, 'xscal', 'log')
50 xlabel('\lambda', 'fontsize', 12);
51 ylabel('Cardinalidade de \beta', 'fontsize', 12);
52 %
53
54 %% Apresentar dados em tabela (apenas a parte significativa
55   ).
56 [~, ind(1)] = min(sqr);      % Índice da melhor avaliação
57   SQR.
58 [~, ind(2)] = min(aic);     % Índice da melhor avaliação
59   AIC.
60 [~, ind(3)] = min(bic);     % Índice da melhor avaliação
61   BIC.
62 ind = sort(unique(ind));    % Juntar os índices das três
63   avaliações em ordem crescente e sem repetições.
64 a = min(ind);              % Menor índice.
65 b = max(ind);              % Maior índice.
66 c = b - a;                 % Intervalo entre os índices.
67
68 k = 10 % Tabela com no mínimo 10 linhas de dados.
69 if (c < k)
70     extras = ceil((k - c)/2); % Quantidade de índices
71     extras para cada extremo.
72     a = a - extras;
73     if a < 1 % Tratar caso o índice seja inválido.
74         b = b + 1 - a;
75         a = 1;
76     end
77     b = b + extras;
78     if b > numel(sqr)
79         a = a - (b - numel(sqr))
```

```

74         b = numel(sqr);
75     end
76 end
77 for i = a:b
78     fprintf('%0.5f %d %0.1f %0.1f %0.1f \n', S.Lambda(i), S.DF(
79         i), sqr(i), aic(i), bic(i))

```

## B.6 Teste de parâmetros do modelo Logístico

```

1 function [vBest y_hat err SSE FIT]=logisticfit(x,y,k,
2     noNegativePot,kindFit)
3 %
4 % Logistic curve fitting for power curve estimation with
5 % coefficients
6 % computed via ordinary differential evolution (DE) method.
7 %
8 % For k = 3 (default) we will have 3 variables (L, p, m)
9 % and the function
10 % will be defined by:
11 %  $y(x) = L / \{1 + \exp[-p*(x - m)]\}$ 
12 %
13 % For k = 4 we will have 4 variables (a, m, n, t) and the
14 % function will be
15 % defined by:
16 %  $y(x) = a * \{ [1 + m * \exp(-x/t)] / [1 + n * \exp(-x/t)] \}$ 
17 %
18 % And for k = 5 we will have 5 variables (a, b, c, d, g)
19 % and the function
20 % will be defined by:
21 %  $y(x) = d + (a - d) / \{ [1 + (x/c)^b]^g \}$  with c, g > 0

```

```
18 % INPUTS
19 % =====
20 %
21 % x: vector with input observations (regressors).
22 % y: vector with output observations (same dimension as x)
    .
23 % k: Kind of logistic expression (3, 4 or 5 parameters). 3
    as default.
24 % noNegativePot: Penalty for negative power. True as
    default.
25 %
26 % OUTPUTS
27 % =====
28 %
29 % w: estimated coefficients (via ordinary least-squares)
30 % yhat: predicted output values
31 % errors: prediction errors (residuals)
32 % SSE: sum-of-squared errors
33 % R2: coefficient of determination (R2=1 - SSE/var(y))
34 %
35 % Author: Haroldo C. Maya
36 % Date: January 8th, 2017
37 %
38 if nargin < 2
39     error('LOGISTICFIT requires at least three input
        arguments. ');
40 end
41 if nargin < 3
42     k = 3;
43 end
44 if (k < 3) || (k > 5)
45     error('k most be between 3 and 5.');
```

```
46 end
47 if nargin < 4
48     noNegativePot = true;
49 end
50 if nargin < 5
51     kindFit = 'bic';
52 end
53
54 x=x(:); y=y(:); % input,output data always as column
    vectors
55
56 N=length(x); % number of input-output pairs (xi,yi)
57
58 %% ATTENTION!! Values for change are here.
59 gen = 100000; % Number of generations
60 B1 = 0.9; % Initial mutation factor [0 ~~ oo]
61 B2 = B1; % Final mutation factor [0 ~~ oo]
62 FacCross = 0.8; % Crossing Factor
63 stopCrit = 200; % If the sse stays stable for "stopCrit
    " generations, stop it.
64 %
65
66
67 %% Initialization of variables
68 dim = k; % Vector dimension.
69 size_pop = 30*dim; % Population size (It is recommended a
    minimum of ten times the number of dimensions)
70
71 % Initialize population vectors
72 V_pop = rand(size_pop, dim)-0.5; % Initialization of the
    initial population as a random matrix.
73 if k == 5
```

```

74     V_pop(:,3) = abs(V_pop(:,3)) + eps;
75     V_pop(:,5) = abs(V_pop(:,5)) + eps;
76 end
77
78 V_off = zeros(size_pop, dim);
79 % Initialize DE vectors
80 mutVec = zeros(size_pop, 2); % Mutation vector
81 croVec = zeros(size_pop, dim); % Crossing vector
82 ssePop = zeros(size_pop, 1); % SSE of the population
83 sseOff = zeros(size_pop, 1); % SSE of offspring
84 fitPop = zeros(size_pop, 1); % Fitness of the population
85 fitOff = zeros(size_pop, 1); % Fitness of offspring
86
87 sseBest = zeros(gen + 1, 1); % Best SSE of generation.
88 fitBest = zeros(gen + 1, 1); % Best fitness of
    generation.
89 %%%%%%%%%%%
90
91 %% Initial analysis
92 for i = 1 : size_pop
93     if k == 3
94         L = V_pop(i, 1);
95         p = V_pop(i, 2);
96         m = V_pop(i, 3);
97         y_hat = L./(1 + exp(-p*(x - m)));
98     elseif k == 4
99         a = V_pop(i, 1);
100        m = V_pop(i, 2);
101        n = V_pop(i, 3);
102        tau = V_pop(i, 4);
103        y_hat = a*((1 + m*exp(-x/tau))./(1 + n*exp(-x/tau))
        );

```

```

104 elseif k == 5
105     a = V_pop(i, 1);
106     b = V_pop(i, 2);
107     c = V_pop(i, 3);
108     d = V_pop(i, 4);
109     g = V_pop(i, 5);
110     if ((c <= 0) || (g <= 0))
111         y_hat = inf;
112     else
113         y_hat = d + (a - d) ./ ((1 + (x./c).^b).^g);
114     end
115 end
116 %% Attribute calculation
117 err = y - y_hat;           % Errors
118 ssePop(i) = err'*err;     % Sum-of-squared errors
119
120 switch(kindFit)
121     case 'sse'
122         fitPop(i) = ssePop(i); % Sum-of-squared errors.
123     case 'aic'
124         fitPop(i) = N*log(ssePop(i)) + 2*(k+1); %
125             Akaike information criterion (AIC).
126     case 'bic'
127         fitPop(i) = N*log(ssePop(i)) + (k+1)*log(N); %
128             Bayesian information criterion (BIC).
129     case 'fpe'
130         fitPop(i) = N*log(ssePop(i)) + N*log((N+k)/(N-k
131             )); % Final prediction error criterion (FPE)
132         .
133     case 'r2'
134         fitPop(i) = 1 - ssePop(i)/(sum((mean(y)-y).^2))
135             ; % R-Squared (R2).

```

```

131     case 'r2a'
132         fitPop(i) = 1 - ((N-1)/(N - (k+1)))*ssePop(i)/(
            sum((mean(y)-y).^2)); % R-Squared adjusted (
            R2A).
133     otherwise
134         fitPop(i) = N*log(ssePop(i)/N) + (k+1)*log(N);
            % Bayesian information criterion (BIC).
135     end
136
137     %% This part is generated to save time in the first
            iteration of Differential Evolution
138     mutVec(i, :) = randperm(size_pop, 2); % Selection of
            the indices of two distinct and random elements used
            to generate the donor vectors.
139     croVec(i, ceil(dim*rand)) = 1; % Selection of
            one of the variables of each element of the trial
            vector to be obligatorily part of the crossover.
140 end
141
142 if strcmp(kindFit, 'r2') || strcmp(kindFit, 'r2a')
143     [fitBest(1), iBest] = max(fitPop); % Best fitness of
            the initial population
144 else
145     [fitBest(1), iBest] = min(fitPop); % Best fitness of
            the initial population
146 end
147 sseBest(1) = ssePop(iBest); % SSE of the best
            fitness of the initial population
148 vBest = V_pop(iBest, :); % Best vector of the
            initial population
149
150 %% Differential evolution

```

```

151 for t = 2 : gen
152
153     B = B2 + (B1 - B2)*(gen - t)/gen; % Dynamic mutation
        factor.
154
155     %% Generating offspring
156     % Mutation: generate donor vectors ui = V_best + B*(Va
        - Vb)
157     u = repmat(vBest, size_pop, 1) + B*(V_pop(mutVec(1:
        size_pop, 1), :) - V_pop(mutVec(1:size_pop, 2), :));
158
159     % Generate the set of progeny vectors
160     m_V_don = ((rand(size_pop, dim) <= FacCross) + croVec)
        ~= 0; % Mask for elements of the trial vector
        that will participate in the crossing.
161     m_V_pop = 1 - m_V_don;
                                     % Mask for
        elements of the current population vector that will
        participate in the crossing.
162     V_off = V_pop.*m_V_pop + u.*m_V_don;
                                     % Offspring
163     croVec = zeros(size_pop, dim);
                                     % Clear crossing
        vector
164
165     %% To analyze offspring
166     for i = 1 : size_pop
167         if k == 3
168             L = V_off(i, 1);
169             p = V_off(i, 2);
170             m = V_off(i, 3);
171             y_hat = L./(1 + exp(-p*(x - m)));

```

```

172     elseif k == 4
173         a = V_off(i, 1);
174         m = V_off(i, 2);
175         n = V_off(i, 3);
176         tau = V_off(i, 4);
177         y_hat = a*((1 + m*exp(-x/tau))./(1 + n*exp(-x/
            tau)));
178     elseif k == 5
179         a = V_off(i, 1);
180         b = V_off(i, 2);
181         c = V_off(i, 3);
182         d = V_off(i, 4);
183         g = V_off(i, 5);
184         if ((c <= 0) || (g <= 0))
185             y_hat = inf;
186         else
187             y_hat = d + (a - d)./((1+(x./c).^b).^g);
188         end
189     end
190     %% Calculate Attributes
191     err = y - y_hat;           % Erros.
192     sseOff(i) = sum(err.^2);   % Sum-of-squared errors
193     .
194     switch(kindFit)
195     case 'sse'
196         fitOff(i) = sseOff(i); % Sum-of-squared
            errors.
197     case 'aic'
198         fitOff(i) = N*log(sseOff(i)) + 2*(k+1); %
            Akaike information criterion (AIC).
199     case 'bic'

```

```

199         fitOff(i) = N*log(sseOff(i)) + (k+1)*log(N)
           ; % Bayesian information criterion (BIC)
           .
200     case 'fpe'
201         fitOff(i) = N*log(sseOff(i)) + N*log((N+k)
           /(N-k)); % Final prediction error
           criterion (FPE).
202     case 'r2'
203         fitOff(i) = 1 - sseOff(i)/(sum((mean(y)-y)
           .^2)); % R-Squared (R2).
204     case 'r2a'
205         fitOff(i) = 1 - ((N-1)/(N - (k+1)))*sseOff(
           i)/(sum((mean(y)-y).^2)); % R-Squared
           adjusted (R2A).
206     otherwise
207         fitOff(i) = N*log(sseOff(i)) + (k+1)*log(N)
           ; % Bayesian information criterion (BIC)
           .
208     end
209
210     if ((noNegativePot)&&(sum(y_hat < 0) > 0))
211         fitOff(i) = 10*fitOff(i); % If there is power
           below 0, penalize it.
212     end
213
214     %% This part is generated to save time in the first
           iteration of Differential Evolution
215     mutVec(i, :) = randperm(size_pop, 2); % Selection
           of the indices of two distinct and random
           elements used to generate the donor vectors.
216     croVec(i, ceil(dim*rand)) = 1; % Selection
           of one of the variables of each element of the

```

```

        trial vector to be obligatorily part of the
        crossover.
217     end
218
219     %% Greedy selection
220     idx = fitOff < fitPop;           % Mask of offspring
        elements better than current generation
221     V_pop(idx, :) = V_off(idx, :); % Update population to
        the next generation.
222     fitPop(idx) = fitOff(idx);      % Update fitness to the
        next generation.
223     ssePop(idx) = sseOff(idx);     % Update SSE to the
        next generation.
224
225
226     %% Extraction and presentation of data
227     if strcmp(kindFit, 'r2') || strcmp(kindFit, 'r2a')
228         [fitBest(t), iBest] = max(fitPop); % Best fitness.
229     else
230         [fitBest(t), iBest] = min(fitPop); % Best fitness.
231     end
232     sseBest(t) = ssePop(iBest);      % SSE of the best
        fitness.
233     vBest = V_pop(iBest, :);        % Best vector.
234
235     if k == 3
236         L = vBest(1);
237         p = vBest(2);
238         m = vBest(3);
239         if p >= 0
240             sinal1 = '-';
241         else

```

```
242         sinal1 = '';
243     end
244     if m >= 0
245         sinal2 = '-';
246     else
247         sinal2 = '+';
248     end
249     y_hat = L./(1 + exp(-p*(x - m)));
250     st = sprintf('%.4f/{1 + exp[%c%.4f*(x %c %.4f)]}',
251         L, sinal1, abs(k), sinal2, abs(m));
252 elseif k == 4
253     a = vBest(1);
254     m = vBest(2);
255     n = vBest(3);
256     tau = vBest(4);
257     if m >= 0
258         sinal1 = '+';
259     else
260         sinal1 = '-';
261     end
262     if tau >= 0
263         sinal2 = '-';
264     else
265         sinal2 = '';
266     end
267     if n >= 0
268         sinal3 = '+';
269     else
270         sinal3 = '-';
271     end
272     y_hat = a*((1 + m*exp(-x/tau))./(1 + n*exp(-x/tau)))
273         );
```

```

272     st = sprintf('%.4f{[1 %c %.4fexp(%cx/%.4f)]/[1 %c
           %.4f*exp(%cx/%.4f)]}' , a, sinal1, abs(m), sinal2
           , abs(tau), sinal3, abs(n), sinal2, abs(tau));
273 elseif k == 5
274     a = vBest(1);
275     b = vBest(2);
276     c = vBest(3);
277     d = vBest(4);
278     g = vBest(5);
279     if d >= 0
280         sinal1 = '';
281         sinal3 = '-';
282     else
283         sinal1 = '-';
284         sinal3 = '+';
285     end
286     if a >= 0
287         sinal2 = '';
288     else
289         sinal2 = '-';
290     end
291     if b >= 0
292         sinal4 = '';
293     else
294         sinal4 = '-';
295     end
296     y_hat = d + (a - d)./((1+(x./c).^b).^g);
297     st = sprintf('%c%.4f + (%c%.4f %c %.4f)/[(1 + (x
           /%.4f)^{(%.4f)} )^{%.4f}]' , sinal1, abs(d),
           sinal2, abs(a), sinal3, abs(d), c, sinal4, abs(b
           ), g);
298 end

```

```

299     clc
300     fprintf('\n\n ### Generation %d ###\n\n', t);
301     fprintf(' Best fitness(%s): %1.4f\n', kindFit, fitBest(
        t));
302     fprintf(' Best SSE: %1.4f\n', sseBest(t));
303     fprintf(' %s\n',st)
304
305
306     %% stopping criterion
307     if ((t > stopCrit)&&(round(fitBest(t)*10000) == round(
        fitBest(t - stopCrit)*10000)))
308         break
309     end
310 end
311
312 err = y - y_hat; % errors
313 SSE = ssePop(end);
314 FIT = fitPop(end);
315
316 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
317 %
        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
318 %%% STEP 3: Plot power curve, histogram and boxplot of the
        errors %%
319 %
        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
320 % Power curve plotting
321 xmin=min(x); xmax=max(x);
322 incr=0.1; % increments for curve plotting purposes
323 xgrid=xmin:incr:xmax; % x interval for curve plotting

```

```

324 xgrid=xgrid(:);
325
326 if k == 3
327     ygrid = L./(1 + exp(-p*(xgrid - m)));
328 elseif k == 4
329     ygrid = a*((1 + m*exp(-xgrid/tau))./(1 + n*exp(-xgrid/
        tau)));
330 elseif k == 5
331     ygrid = d + (a - d)./((1+(xgrid./c).^b).^g);
332 end
333
334 figure;
335 plot(x,y,'ro',xgrid,ygrid,'b-'); grid
336 xlabel('wind speed [m/s]')
337 ylabel('generated power [KWatts]')
338
339 % Error histogram
340 figure;
341 %histfit(errors);
342 hist(err, 20)
343 xlabel('prediction errors')
344 title('Histogram - prediction errors (residuals)')
345
346 % Boxplot of prediction errors
347 figure;
348 boxplot(err);
349 title('Boxplot - prediction errors (residuals)')

```

## B.7 Teste de parâmetros do modelo MLP

```

1 clear; clc; close all
2

```

```
3 X=load('dados\Normalized_WT1.dat');
4 x=X(:,1); % speed samples
5 y=X(:,2); % power samples
6
7 [x i] = sort(x);
8 y = y(i);
9 k = 10;
10 sqr = zeros(k,1);
11 aic = zeros(k,1);
12 bic = zeros(k,1);
13 MC = 50;
14
15 for g=1:MC
16     for i=1:k;
17         %% Parâmetros da rede %%
18         NNeuroniosCamadaOculta = i;
19         PorcTreino = 0.7;
20         PorcValidacao = 0.30;
21         PorcTeste = 0.0;
22
23         %% Verificação de erros
24         if (PorcTreino + PorcValidacao + PorcTeste ~= 1)
25             error('ERRO: A divisão dos grupos de treino,
26                 validação e teste não foi adequada.');
```

```
26         end
27
28         net = fitnet(NNeuroniosCamadaOculta);
29         net.name = 'Haroldo's neural network';
30         net.userdata = 'Apenas testes e mais testes.';
31
32         net.divideParam.trainRatio = PorcTreino;
33         net.divideParam.valRatio = PorcValidacao;
```

```
34     net.divideParam.testRatio = PorcTeste;
35
36     %% Função de treino
37
38     [net tr] = train(net,x',y');
39     y2 = net(x');
40
41     %% Analisar resultados.
42     erro = y-y2';
43     SQR = erro'*erro;
44     sqr(i) = sqr(i) + SQR/MC;
45     aic(i) = aic(i) + AIC(NNeuroniosCamadaOculta*3+1,
46         length(y), SQR)/MC;
47     bic(i) = bic(i) + BIC(NNeuroniosCamadaOculta*3+1,
48         length(y), SQR)/MC;
49     fprintf('Neurônios = %d\nSQR = %.1f\nAIC = %.1f\n
50         nBIC = %.1f\n\n', i, sqr(i), aic(i), bic(i))
51 end
52 end
53
54 %% Imprimir tabela.
55 for i=1:k;
56     fprintf('%d %.1f %.1f %.1f\n', i, sqr(i), aic(i), bic(i)
57         ))
58 end
59
60 aux = [errors];
61 save('erroMLP.dat', 'aux', '-ascii');
62 beep
```