Inteligência Computacional – Trabalho 1

Vanessa Carvalho do Nascimento (Mat.: 471584)

17 de novembro de 2021

Problema 1

O programa a ser descrito está baseado em lógica fuzzy (inferência de Mamdani). Foi criada uma função chamada **fuzzy** que pedirá três informações de forma sequencial: pressão no pedal do freio, velocidade da roda e velocidade do carro, que serão armazenados nas variáveis ppf, vr e vc, respectivamente, e a partir desses parâmetros o programa efetuará os cálculos necessários e retornará a pressão no freio (pf).

```
function fuzzy()


//-------Recebendo os valores necessários para o cálculo-------

ppf = input("Pressão no pedal do freio: ")
vr = input("Velocidade da roda: ")
vc = input("Velocidade do carro: ")

```

Figura 1: Recebendo os valores de ppf, vr e vc.

Foi definida uma função **f(x, r, s, t)**, responsável por retornar o valor de pertencimento de x a um conjunto fuzzy indicado por uma função triangular definida pelos pontos: (r,0), (s,1) e (t,0). Tal função possui a forma:

$$f(x,r,s,t) = max(min(\frac{x-r}{s-r}, \frac{t-x}{t-s}), 0)$$

Segundo é mostrado na figura a seguir, um valor x possuirá um pertencimento ao conjunto definido pela função triangular **f** igual a $\frac{x-r}{s-r}$, $\frac{t-x}{t-s}$ ou 0.

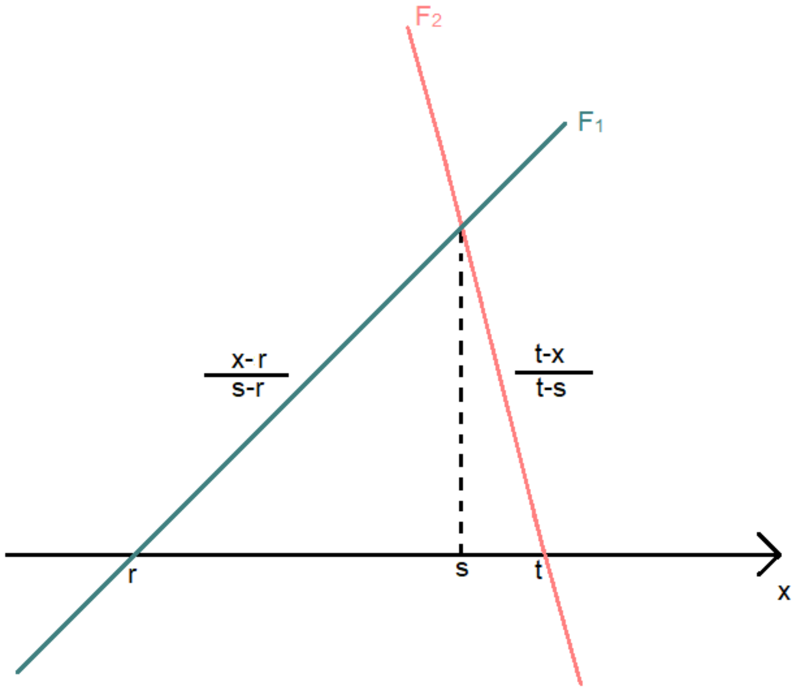

Figura 2: Esquema para explicar a função **f**.

Fazendo $F_1(x) = \frac{x-r}{s-r}$ e $F_2(x) = \frac{t-x}{t-s}$,

$$f(x, r, s, t) = max(min(F_1(x), F_2(x)), 0)$$

Tem-se que para $x < r$ ou $x > t$, $F_1(x) \cdot F_2(x) < 0$, ou seja, exclusivamente uma delas retornará um valor negativo e tal valor será o menor entre $F_1(x)$ e $F_2(x)$, mas o maior entre ele e 0 será 0, portanto para esse caso $f(x) = 0$.

Para $r \geq x \leq s$, o menor valor entre $F_1(x)$ e $F_2(x)$ será $F_1(x)$ e o maior entre ele e 0 será ele mesmo, o que é desejado.

Por fim, para $s < x \leq t$, o menor valor entre $F_1(x)$ e $F_2(x)$ será $F_2(x)$ e o maior entre ele e 0 será ele mesmo, o que é desejado.

A figura abaixo apresenta a implementação de **f**.

```
//-------Função para calcular o pertencimento de um valor x dada uma função triangular estabelecida com os valores (a, b, c) -------
deff('y = f(x, r, s, t)', 'y=max(min((x-r)/(s-r),(t-x)/(t-s)), 0)')

```

Figura 3: Implementação de **f(x, r, s, t)**.

São fornecidas quatro regras e elas são implementadas da seguinte forma, em que a_f é o valor para "aplicar o freio" e l_f é o valor para "liberar o freio":

```
//------- Aplicação das regras --------

//Regra 1:

ppf_med = f(ppf,30, 50, 70)  //Se a pressão no pedal do freio for média
af = ppf_med                //Então, aplicar o freio

//Regra 2:

ppf_alta = f(ppf, 50, 100, 100)               //Se a pressão no pedal de freio for alta
vc_alta = f(vc, 40, 100, 100)                 //E a velocidade do carro for alta
vr_alta = f(vr, 40, 100, 100)                 //E a velocidade das rodas for alta
af = af + min(ppf_alta, vc_alta, vr_alta)  //Então, aplicar o freio (AND entre as sentenças->min()) ---- É desejada a soma dos af's

//Regra 3:

vr_baixa = f(vr, 0, 0, 60)                  //Se a velocidade das rodas for baixa
//os valores gerados pelas outras sentenças da Regra 3 já foram calculados na regra anterior
lf = min(ppf_alta, vc_alta, vr_baixa)       //Então, liberar o freio (AND entre as sentenças->min())

//Regra 4:
ppf_baixa = f(ppf, 0, 0, 50)              //Se a pressão no pedal de freio for baixa
lf = lf+ppf_baixa                         //Então, liberar o freio ---- É desejada a soma dos lf's
```

Figura 4: Aplicação das regras.

Para o cálculo do centroide é preciso analisar dois casos: $l_f > a_f$ ou $l_f \leq a_f$. Dessa forma, obtém-se as seguintes figuras, que foram divididas em quatro partes para facilitar o cálculo. No Caso 2 da figura abaixo, está explícito que $l_f < a_f$, no entanto, quando $l_f = a_f$, aplicando o método correspondente ao Caso 2, obtém-se a resposta correta, como será visto mais a frente. Por isso, na implementação do código será considerado o Caso 2 como sendo $l_f \leq a_f$.

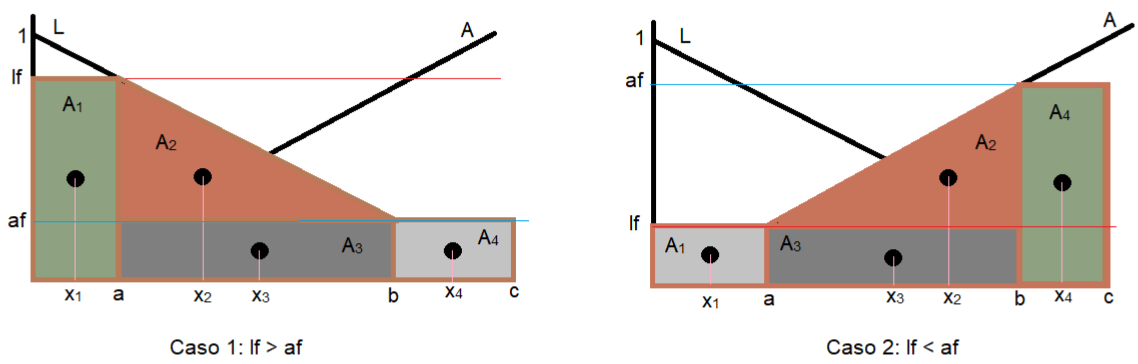

Figura 5: Esquema de divisão da área formada para o cálculo do centroide.

Nesses casos, deve-se localizar três valores importantes: a, b e c. Além disso, x_1, x_2, x_3 e x_4 são os centroides das figuras e A_i é a área da figura i. Para o cálculo do centroide $\overline{x}$ a partir das quatro partes, deve-se fazer:

$$\overline{x} = \frac{\sum_{i=1}^{n} x_i A_i}{\sum_{i=1}^{n} A_i}$$

Neste caso, $n = 4$. Nos gráficos apresentados, **A** e **L** são funções de pertinência: **A** equivale a aplicar o freio e **L** equivale a liberar o freio.

A possui a forma:

$$A(x) = \frac{x}{100}$$

E **L** possui a forma:

$$L(x) = 1 - \frac{x}{100}$$

Portanto, para o Caso 1, tem-se que:

$$1 - \frac{a}{100} = l_f \Rightarrow a = 100 - 100 \cdot l_f$$

Analogamente,

$$b = 100 - 100 \cdot a_f$$

Já para o Caso 2:

$$\frac{a}{100} = l_f \Rightarrow a = 100 \cdot l_f$$

Analogamente,

$$b = 100 \cdot a_f$$

Para ambos os casos, $c = 100$.
O código abaixo mostra tal implementação.

```
//-----------------Cálculo do centroide-----------------------


//Cálculo de a, b e c

if lf>af then
a=100-100*lf
b=100-100*af
else
a=100*lf
b=100*af
end

c=100
```

Figura 6: Cálculo de a, b e c.

Será útil calcular o maior e o menor valor entre a_f e l_f.

```
//Cálculo do maior e do menor valor entre af e lf

maior = max(af, lf)
menor = min(af, lf)

```

Figura 7: Cálculo do maior e do menor valor.

Assim, para calcular as áreas, percebe-se que em ambos os casos $A_1 = a \cdot l_f$ e $A_4 = a_f \cdot (c - b)$. Para evitar o sinal negativo em A_2 faz-se $A_2 = \frac{(b-a)\cdot(\mathbf{maior}-\mathbf{menor})}{2}$. Por fim, A_3 dependerá de **menor** e será dado por $A_3 = \mathbf{menor} \cdot (b - a)$. A Figura seguinte apresenta a codificação do cálculo das áreas.

```
//Cálculo das áreas

A1 = a*lf
A2 = (b-a)*(maior-menor)/2
A3 = (b-a)*menor
A4 = af*(c-b)
```

Figura 8: Cálculo das áreas.

Para o cálculo dos centroides, tem-que $x_1 = \frac{a}{2}$, $x_3 = \frac{a+b}{2}$ e $x_4 = \frac{b+c}{2}$. O valor de x_2 dependerá do caso. Se for o Caso 1, então $x_2 = \frac{2 \cdot a+b}{3}$. Se for o Caso 2, então $x_2 = \frac{a+2 \cdot b}{3}$.

```
//Cálculo do centroide de cada parte

x1 = a/2

if lf>af then
x2 = (2*a+b)/3
else
x2 = (a+2*b)/3
end

x3 = (a+b)/2
x4 = (b+c)/2
```

Figura 9: Cálculo do centroide de cada pedaço.

Em seguida, faz-se o somatório das multiplicações entre o valor de cada área pelo valor do centroide correspondente e o resultado é armazendado em **somaxarea**. A variável **somarea** armazena a soma das áreas. Por fim, efetua-se a divisão de **somaxarea** por **somarea** e obtém-se a pressão no freio **pf**.

```
//Cáculo do somatório das multiplicações entre o valor de cada área pelo valor do centroide correspondente
somaxarea = A1*x1+A2*x2+A3*x3+A4*x4


//Soma das áreas
somarea = A1+A2+A3+A4


//Calculo da pressão no freio

pf = somaxarea/somarea
disp('pressao no freio', pf)
```

Figura 10: Cálculo da pressão no freio.

Como dito anteriormente, quando $l_f = a_f$, o método do Caso 2 retorna o valor correto de **pf**. Há três modos possíveis para essa igualdade: quando a_f e l_f estão acima, abaixo ou exatamente no ponto de intersecção de **L** e **A**.

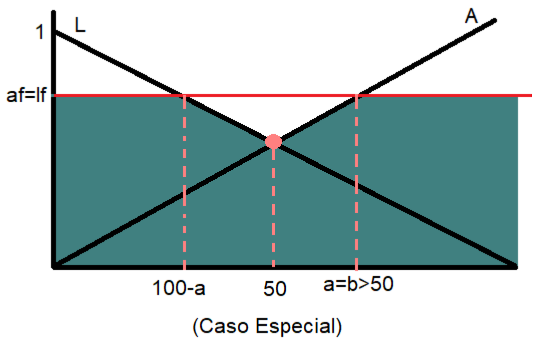

Figura 11: Acima do ponto de intersecção (Caso especial).

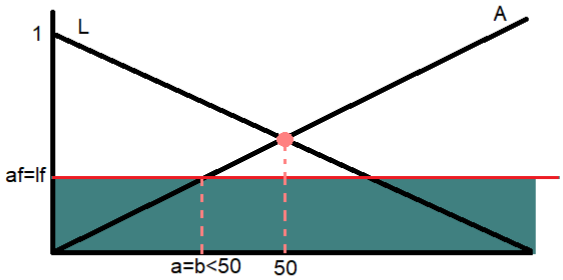

Figura 12: Abaixo do ponto de intersecção.

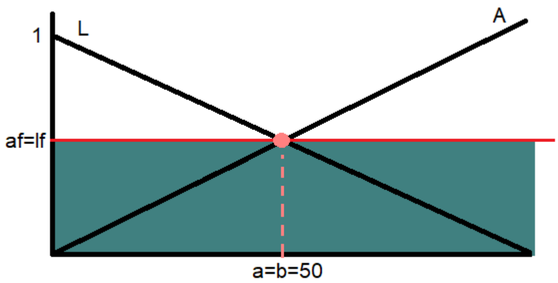

Figura 13: Exatamente no ponto de intersecção.

Em todos os casos, por simetria, percebe-se que o centroide deve estar localizado em $pf = \frac{100}{2} = 50$. Utilizando o método do Caso 2, tem-se que:

$$a_f = l_f \Rightarrow a = 100 \cdot l_f = 100 \cdot a_f = b \Rightarrow a = b$$

Dessa forma, criando uma variável **p**, tal que:

$$p = a_f = l_f = \textbf{maior} = \textbf{menor}$$

Tem-se, $A_1 = a \cdot p$, $A_2 = A_3 = 0$ e $a_4 = p(c - b)$.

Além disso, $x_1 = \frac{a}{2}$ e $x_4 = \frac{b+c}{2}$.
Assim,

$$\textbf{somaxarea} = A_1 \cdot x_1 + A_4 \cdot x_4 = \frac{a^2 \cdot p}{2} + \frac{p(c^2 - b^2)}{2} = \frac{p \cdot c^2}{2}$$

E a soma dos A_i's:

$$\textbf{somarea} = a \cdot p + p(c - b) = p \cdot c$$

Logo,

$$pf = \frac{\textbf{somaxarea}}{\textbf{somarea}} = \frac{c}{2} = \frac{100}{2} = 50$$

O que permite acrescentar a igualdade dentro do Caso 2 e, portanto, ele se torna $l_f \leq a_f$.

Para a visualização da área formada para o cálculo do centroide, definiu-se uma função chamada **piece** para fazer a modelagem. Dadas as fórmulas de A(x) e L(x), tem-se que o ponto de intersecção ocorre em $x = 50$. Analisando os gráficos apresentados, se $a \neq b$ então $a < 50 < b$ e a figura formada possui 3 estágios. O primeiro e o terceiro estágio são constantes, com o primeiro no nível l_f e o segundo no nivel a_f. O segundo estágio vai depender do caso. Ele será igual a L(x) no primeiro caso e será igual a A(x) no segundo caso, ou seja, se for o Caso 1, em [a, b], $piece(x) = 1 - \frac{x}{100}$ e se for o Caso 2, em [a, b], $piece(x) = \frac{x}{100}$.

Já quando $a = b$ há três possibilidades como dito anteriormente. Os casos em que $a = b \leq 50$ geram um área retangular bem delimitada, que será corretamente representada usando o método descrito para $a \neq b$. Portanto, o primeiro **if** do código abaixo para a plotagem aborda os casos em que: $a \leq 50 \leq b$ **OR** $(a == b)$ **AND** $a \leq 50$.

Por fim, tratando do caso especial que é abordado pelo **else**, em que $a = b > 50$ será preciso dividir a função em quatro partes. A última (chamada de terceira parte no código) será a mesma para qualquer caso, mas serão definidas três novas partes para o caso especial, de modo que para $x \leq 100 - a$, tem-se **piece(x)** $= af = lf$. Para $100 - a < x \leq 50$, tem-se **piece(x)** $= L(x)$. Por fim, para $50 < x < b$, tem-se **piece(x)** $= A(x)$.

```
//-----------------Plot-----------------
function  y = piece(x)

if a<=50 & 50<=b | a==b & a<50 then

// primeira parte
y(x <= a) = lf

// segunda parte
x2 = a < x & x <= b

if lf>af then
    y(x2) = 1 - x(x2)/100
    
 else
     y(x2) = x(x2)/100
end

else   // Caso especial
// primeira parte (caso especial)
y(x <= 100-a) = lf

// segunda parte (caso especial)
x2 = 100-a < x & x <=50
y(x2) = 1 - x(x2)/100

// terceira parte (caso especial)
x3 = 50 < x & x <a
y(x3) = x(x3)/100
end

// terceira parte
y(b <= x & x <= c) = af
endfunction
```

Figura 14: Implementação da função **piece(x)**.

Por fim, plota-se o gráfico de **piece(x)** para x em [0, c].

```
x = [0:100]'
y = piece(x)
plot(x,y)
xlabel("Pressão no freio", "fontsize", 2)
a = gca()
a.data_bounds = [0 0; c 1]

endfunction
```

Figura 15: Código para plotar o gráfico.

Para exemplificar, chamando-se a função **fuzzy** e fazendo $ppf = 60$, $vr = 55$ e $vc = 80$, obtém-se $pf = 63.892591$.

```
Scilab 6.1.0 Console

--> fuzzy
Pressão no pedal do freio: 60

Velocidade da roda: 55

Velocidade do carro: 80


  "pressao no freio"

   63.892591

--> |
```

Figura 16: Exemplo para $ppf = 60$, $vr = 55$ e $vc = 80$ retornando $pf = 63.892591$.

A figura seguinte mostra a área usada para o cálculo do centroide.

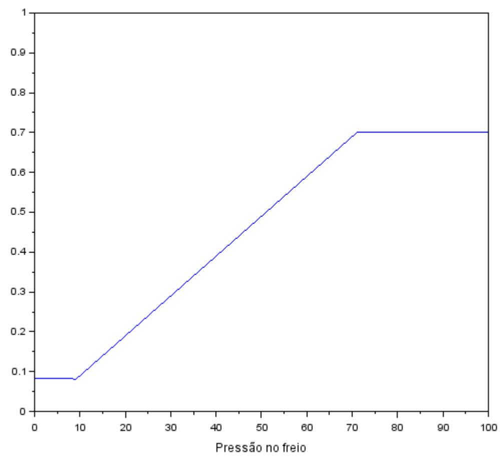

Figura 17: Gráfico do exemplo mostrado.

Para tal caso, $a = 8.3$ e $b = 70$.

Problema 2

Utilizando o conjunto de dados do aerogerador, deve-se determinar os modelos de regressão polinomial (graus 2 a 7) com parâmetros estimados pelo método dos mínimos quadrados. Em cada caso o modelo é analisado pelas métricas: R^2 e R^2_{aj}.

O código abaixo plota o gráfico que representa os dados do aerogerador, e que x (variável de entrada) é a velocidade do vento em m/s e y (variável de saída) é a potência gerada em kWatts.

```
base = fscanfMat("aerogerador.dat")

x = base(:,1) //velocidade do vento em m/s
y = base(:,2) //potência gerada em kWatts

plot(x, y, "*")

```

Figura 18: Código para plotar o gráfico que representa os dados do aerogerador.

A partir dos dados fornecidos é possível montar o gráfico a seguir.

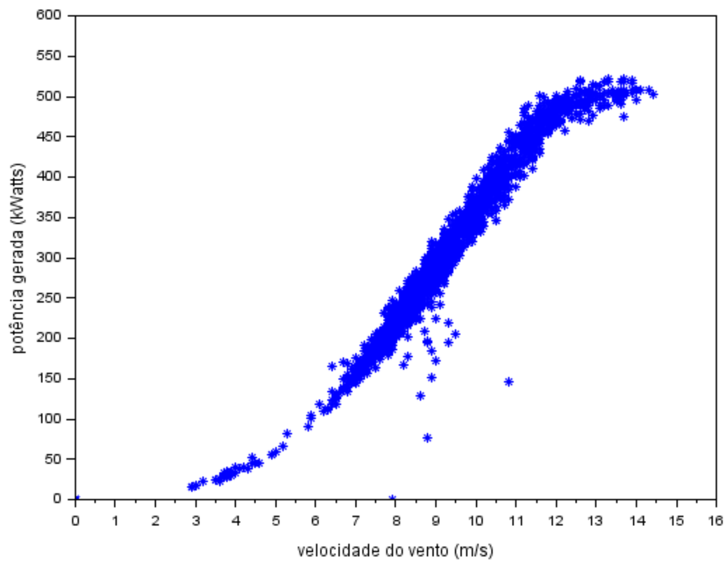

Figura 19: Gráfico plotado.

- Para Grau 2:

$$y_{prev} = -92.980030 + 26.723141x + 1.6931193x^2$$

Métricas: $R^2 = 0.9434239$ e $R^2_{aj} = 0.9433735$.

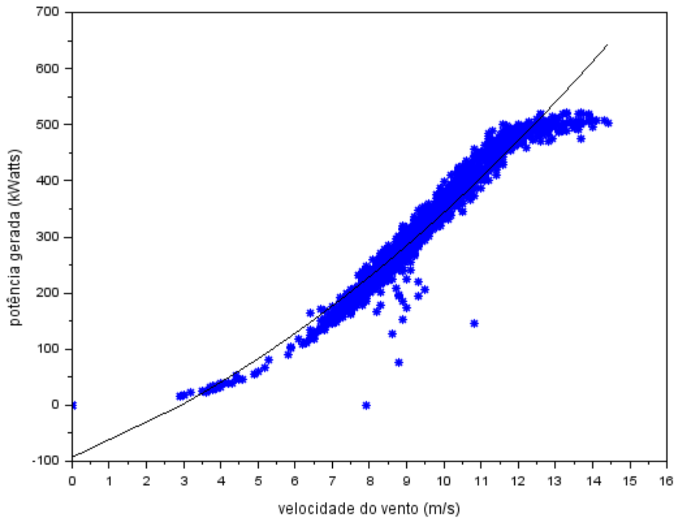

Figura 20: Gráfico de y_{prev} para Grau 2.

- Para Grau 3:

$$y_{prev} = 32.623510 - 58.760424x + 15.051913x^2 - 0.5924080x^3$$

Métricas: $R^2 = 0.9690229$ e $R^2_{aj} = 0.9689815$

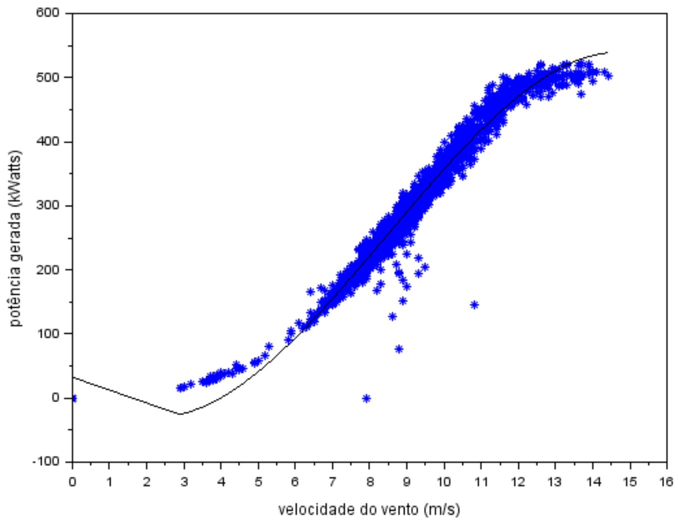

Figura 21: Gráfico de y_{prev} para Grau 3.

- Para Grau 4:

$$y_{prev} = -0.3913261 + 10.372887x - 5.0035997x^2 + 1.4338950x^3 - 0.0676697x^4$$

Métricas: $R^2 = 0.9737242$ e $R^2_{aj} = 0.9736774$.

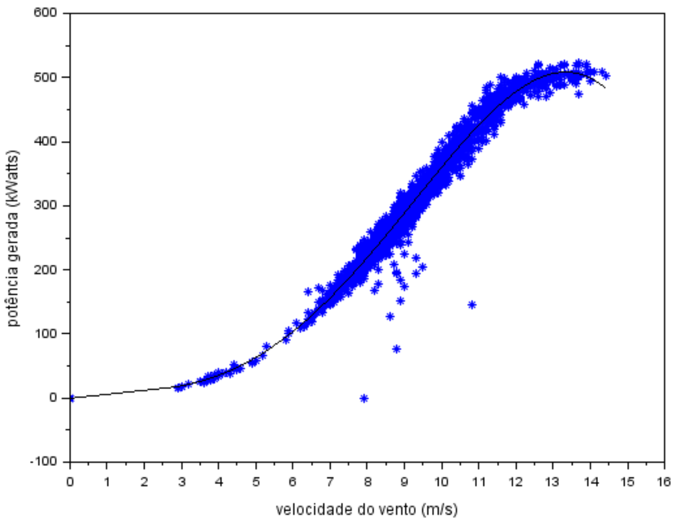

Figura 22: Gráfico de y_{prev} para Grau 4.

- Para Grau 5:

$$y_{prev} = -0.1798263 + 8.1638761x - 3.9304553x^2 + 1.2462259x^3 - 0.0537025x^4 - 0.0003753x^5$$

Métricas: $R^2 = 0.9737256$ e $R^2_{aj} = 0.9736671$.

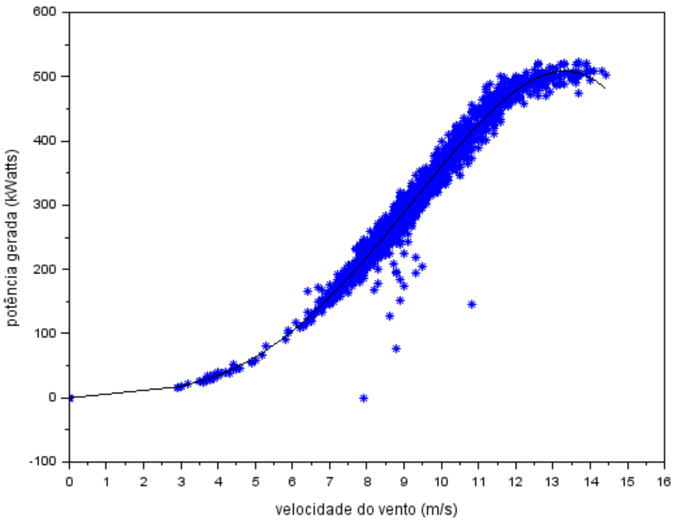

Figura 23: Gráfico de y_{prev} para Grau 5.

- Para Grau 6:

$y_{prev} = 0.2054094 - 24.641287x + 17.576585x^2 - 4.0297484x^3 + 0.5613640x^4 - 0.0347935x^5 + 0.0007445x^6$

Métricas: $R^2 = 0.9737610$ e $R^2_{aj} = 0.9736908$.

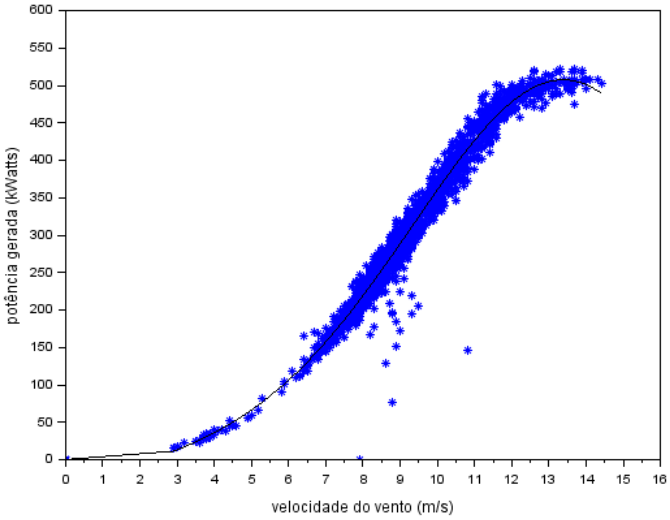

Figura 24: Gráfico de y_{prev} para Grau 6.

- Para Grau 7:

$$y_{prev} = -0.0879035 + 109.57219x - 90.005763x^2 + 29.913857x^3 - 4.8965685x^4 + 0.4404919x^5 - 0.0206212x^6 + 0.0003889x^7$$

Métricas: $R^2 = 0.9738574$ e $R^2_{aj} = 0.9737758$.

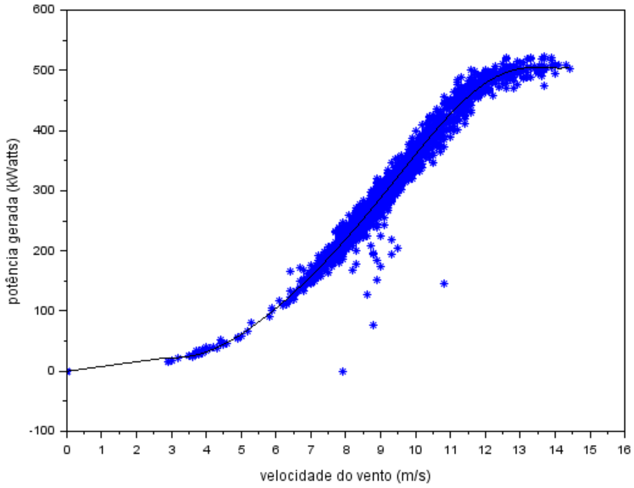

Figura 25: Gráfico de y_{prev} para Grau 7.

A figura abaixo apresenta o código utilizado para a modelagem. Trata-se do caso de Grau 7.

```
base = fscanfMat("aerogerador.dat")

x = base(:,1) //velocidade do vento em m/s
y = base(:,2) //potência gerada em kWatts

plot(x, y, "*")
xlabel("velocidade do vento (m/s)", "fontsize", 2)
ylabel("potência gerada (kWatts)", "fontsize", 2)

n = length(x)

X = [ones(n,1) x x.^2 x.^3 x.^4 x.^5 x.^6 x.^7]

beta = (X'*X)^(-1)*X'*y

y_prev = beta(1) + beta(2)*x + beta(3)*x.^2 + beta(4)*x.^3 + beta(5)*x.^4 + beta(6)*x.^5 + beta(7)*x.^6 + beta(8)*x.^7

SQE=sum((y-y_prev).^2)
Syy=sum((y-mean(y)).^2)

R2 = 1 - SQE/Syy

p=length(beta)

Raj2 = 1-SQE*(n-1)/(Syy*(n-p))
plot(x, y_prev, "k-")
```

Figura 26: Código para Grau 7.