

# SCD - Sitemas de Controle Dinâmicos

Igor Otoni, Egmon Pereira

<sup>1</sup>CEFET-MG Campus Timóteo

## 1. Exercício 1

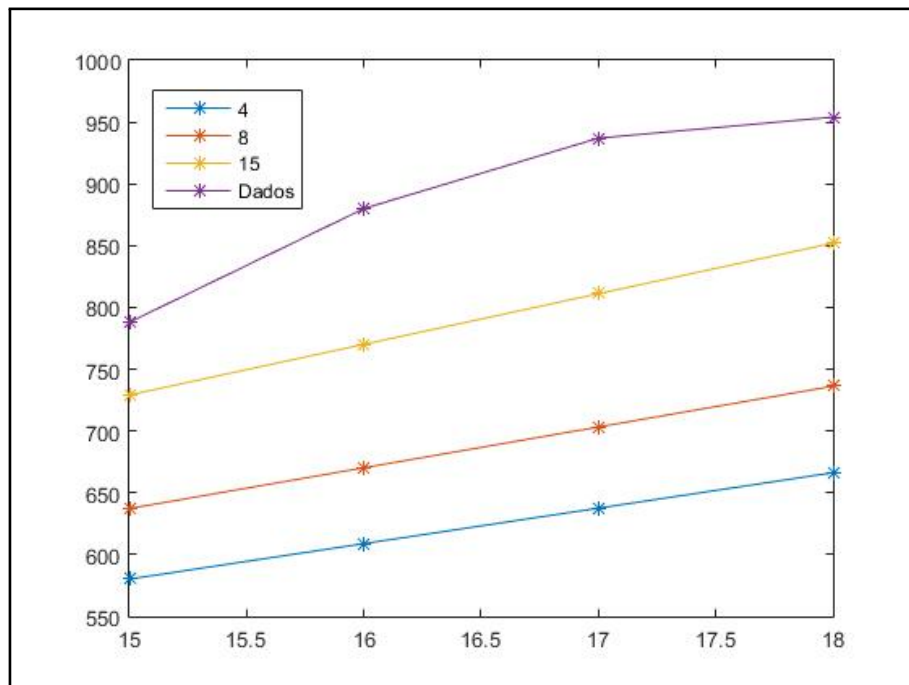
### 1.1. Código para resolução e gráfico para análise

```
1 % LIMPANDO DADOS DO MATLAB
2 clc; clear; close all;
3 % LETRA A)
4 x = [0 1 2 3];
5 y = [151 180 200 240];
6 [c_a] = polyfit(x, y, 1);
7 mat_a = ones(length(x),2);
8 mat_a(:,1) = x';
9 linear = mat_a*c_a';
10 x = [15 16 17 18];
11 y = [788 880 937 954];
12 mat_a = ones(length(x),2);
13 mat_a(:,1) = x';
14 previsao_4 = mat_a*c_a';
15 erro_4 = abs(y - previsao_4');
16 erro_quadratico_4 = ( (y(1) - previsao_4(1))^2 + (y(2) -
    previsao_4(2))^2 + (y(3) - previsao_4(3))^2 + (y(4) -
    previsao_4(4))^2 ) / 4
17 % LETRA B)
18 x = [0 1 2 3 4 5 6 7];
19 y = [151 180 200 240 260 300 350 380];
20 [c_b] = polyfit(x, y, 1);
21 mat_b = ones(length(x),2);
22 mat_b(:,1) = x';
23 x = [15 16 17 18];
24 y = [788 880 937 954];
25 mat_b = ones(length(x),2);
26 mat_b(:,1) = x';
27 previsao_8 = mat_b*c_b';
28 erro_8 = abs(y - previsao_8');
29 erro_quadratico_8 = ( (y(1) - previsao_8(1))^2 + (y(2) -
    previsao_8(2))^2 + (y(3) - previsao_8(3))^2 + (y(4) -
    previsao_8(4))^2 ) / 4
30 % LETRA C)
31 x = [0 1 2 3 4 5 6 7 8 9 10 11 12 13 14];
```

```

32 y = [151 180 200 240 260 300 350 380 415 465 510 545 622
      678 724];
33 [c_c] = polyfit(x, y, 1);
34 mat_c = ones(length(x),2);
35 mat_c(:,1) = x';
36 x = [15 16 17 18];
37 y = [788 880 937 954];
38 mat_c = ones(length(x),2);
39 mat_c(:,1) = x';
40 previsao_15 = mat_c*c_c';
41 erro_15 = abs(y - previsao_15');
42 erro_quadratico_15 = ( (y(1) - previsao_15(1))^2 + (y(2) -
      previsao_15(2))^2 + (y(3) - previsao_15(3))^2 + (y(4) -
      previsao_15(4))^2 ) / 4
43 % GRAFICO
44 plot_ = ones(length(x), 4);
45 plot_(:,1) = previsao_4;
46 plot_(:,2) = previsao_8;
47 plot_(:,3) = previsao_15;
48 plot_(:,4) = y;
49 plot(x, plot_ , '-*');
50 legend('4','8','15','Dados');

```



### 1.1.1. A

previsao\_4 =

```
580.2000
608.9000
637.6000
666.3000
```

```
erro_4 =
  207.8000  271.1000  299.4000  287.7000
```

```
erro_quadratico_4 =
  7.2272e+04
```

### **1.1.2. B**

```
previsao_8 =
  637.2619
  670.2738
  703.2857
  736.2976
```

```
erro_8 =
  150.7381  209.7262  233.7143  217.7024
```

```
erro_quadratico_8 =
  4.2181e+04
```

### **1.1.3. C**

```
previsao_15 =
  729.1619
  770.1405
  811.1190
  852.0976
```

```
erro_15 =
  58.8381  109.8595  125.8810  101.9024
```

```
erro_quadratico_15 =
  1.0440e+04
```

### **1.1.4. D**

Os valores dos parâmetros são diferentes para as letras *a*, *b* e *c*; [28.699999999999985 1.497000000000001e+02], [33.011904761904766 1.420833333333333e+02] e [40.978571428571420 1.144833333333333e+02], respectivamente.

### 1.1.5. E

Os erros são diferentes para cada uma das letras, conforme mais pontos foram sendo utilizados, mais o erro diminuiu. A letra *a* obteve pior erro e a letra *c* obteve menor erro. Isso aconteceu, pois, conforme mais pontos são incluídos, mais o erro é diluído entre as amostras: as variações são assimiladas de maneira melhor pelo modelo.

### 1.1.6. F

O melhor modelo foi o obtido pela letra *c*, pois foi o que melhor se aproximou e melhor predisse o comportamento do sistema, isso foi constatado tanto pela análise dos erros, quanto por inspeção gráfica. Porém foi um modelo mais difícil de ser obtido: precisou de mais informações e gastou mais processamento. Levando essas circunstâncias em consideração, a resposta para escolher de qual entre os modelos vai depender das necessidades e recursos de um cenário-problema real.

## 2. Exercício 2

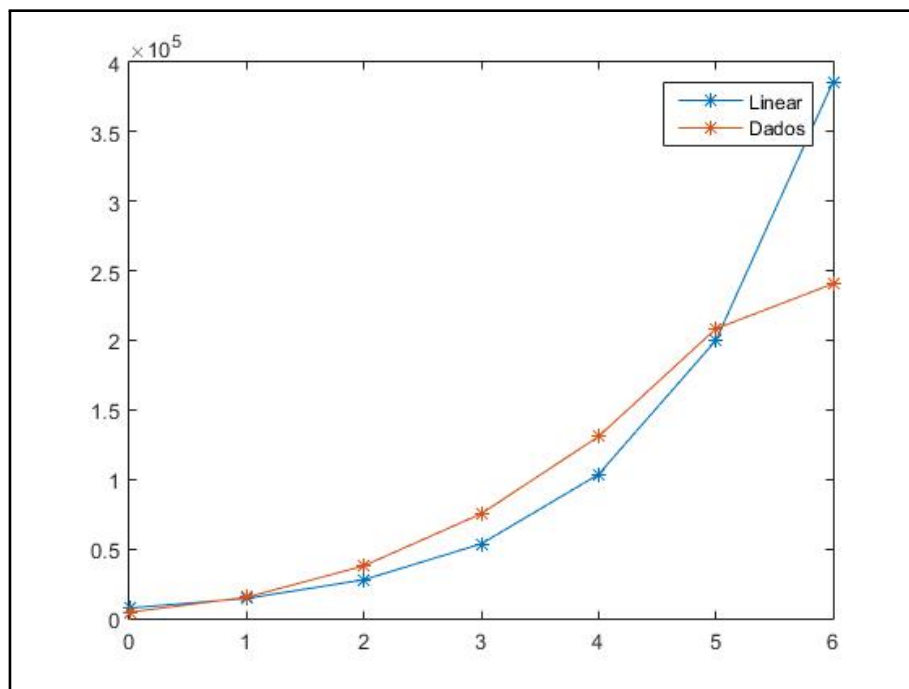
### 2.1. Código para resolução e gráfico para análise

```
1 % LIMPANDO DADOS DO MATLAB
2 clc; clear; close all;
3 % VETOR DE DADOS OU MEDICOES
4 x = [0 1 2 3 4 5 6];
5 y = [4499 15205 37808 75336 130799 208281 240749];
6 b_y = y;
7 % TRANSFORMACAO Y = P * E^(K * X) <=> LN(Y) = LN(P) + LN(K * X)
8 y = log(y);
9 % LETRA A)
10 % GERANDO MATRIZ LINEAR [x 1]
11 mat_a = ones(length(x), 2);
12 mat_a(:, 1) = x';
13 % FORMULA DO METODO DOS MINIMOS QUADRADOS
14 p = (inv(mat_a' * mat_a) * mat_a') * y';
15 p(2) = exp(p(2))
16 % OBTENDO DADOS/PREVISOES DO MODELO EXPONENCIAL
```

```

17 exponencial = p(2)*exp(p(1)*x);
18 % CALCULO DO ERRO
19 erro_7 = abs(b_y - exponencial)
20 erro_quadratico_7 = ( (b_y(1) - exponencial(1))^2 + (b_y(2)
    - exponencial(2))^2 + (b_y(3) - exponencial(3))^2 + (
    b_y(4) - exponencial(4))^2 + (b_y(5) - exponencial(5))^2
    + (b_y(6) - exponencial(6))^2 + (b_y(7) - exponencial
    (7))^2 ) / 4
21 % PREPARANDO MATRIZ DE DADOS E GRAFICO
22 plot_ = ones(length(x), 2);
23 plot_(:,1) = exponencial;
24 plot_(:,2) = b_y;
25 plot(x, plot_ , '-*');
26 legend('Linear','Dados');

```



### 2.1.1. A

```

p =
    0.6577
    8.9170

p =
    1.0e+03 *
    0.0007
    7.4577

```

```

erro_7 =
    1.0e+05 *
    0.0296    0.0081    0.1002    0.2169
    0.2725    0.0840    1.4508

erro_quadratico_7 =
    5.6106e+09

```

Percebe se que o método dos mínimos quadrados ajustou bem os parâmetros da função exponencial. O maior erro foi obtido na última amostra, que difere bastante da curva exponencial, ou seja, algo esperado.

### 3. Resolução em Python

#### 3.1. Código

```

1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3  """
4  Created on Wed Sep  4 15:14:06 2019
5
6  @author: egmon
7
8  Fontes:
9  http://www2.unirio.br/unirio/ccet/profmat/tcc/TCCJOAO.pdf
10 http://www.decom.ufop.br/marcone/Disciplinas/
    MetodosNumericoseEstatisticos/QuadradosMinimos.pdf
11 https://www.portugal-a-programar.pt/forums/topic/47019-
    gerar-gr%C3%A1ficos-com-python/
12 https://fenix.ciencias.ulisboa.pt/downloadFile
    /2251937252639182/LabNum_2018_v4.pdf
13
14 """
15
16 from numpy import matrix
17 import numpy as np
18 import matplotlib.pyplot as plt
19 import os
20 import math
21
22 def calculo(X,Y,a,b,n,m,t,p):
23     """
24     CALCULOS
25     O = [XT*X]^{-1} . XT . Y
26     """
27     C = (((X.T) * (X)).I) * ((X.T) * Y)
28     D = C.A1
29     if n == 1:

```

```

30         if len(D) == 2 and p == 1:
31             print('theta 1 = %0.4f' %D[0])
32             print('theta 2 = %0.4f' %D[1])
33         elif len(D) == 3 and p == 1:
34             print('\ntheta 1 = %0.4f' %D[0])
35             print('theta 2 = %0.4f' %D[1])
36             print('theta 3 = %0.4f' %D[2])
37     W = X.A1
38     Z = []
39     for i in range(len(W)):
40         if i % 2 == 0:
41             Z.append(W[i])
42
43     for i in range(len(W)):
44         if i % 2 == 0:
45             Z.append(W[i])
46
47     if n == 1:
48         graficoReta(Z,Y,D,a,b,m)
49
50     return proj(Y,D[0],D[1],t)
51
52 def proj(Y,a,b,t):
53     sal = a * (t - 1) + b
54     return sal
55
56
57 def graficoReta(X,Y,D,a,b,m):
58     i = 0
59     x = []
60     y = []
61     x = np.linspace(a,b,100)
62     while i < 100:
63         y.append(D[0]*(x[i]) + D[1])
64         i += 1
65     if m == 0:
66         plt.plot(x,y)
67         plt.plot(Y)
68         plt.xlabel("X")
69         plt.ylabel("Y")
70         plt.show()
71
72 def eqDiferencial(X,Y,t,a,b):
73     W = X.A1
74     Z = []
75     for i in range(len(W)):

```

```

76         if i % 2 == 0:
77             Z.append(W[i])
78     plt.plot(Z,Y)
79     plt.xlabel("X")
80     plt.ylabel("Y")
81     plt.show()
82     k = math.log(Y[5])/math.log(Y[0])
83     print('k =',k)
84     z = Y[0] * math.exp(k * t)
85     print('Y(%d) = %0.2f' %(t,z))
86
87
88 if __name__ == '__main__':
89     os.system('clear')
90     os.system('clear')
91     m = 0#1 nao imprime o grafico 0 Imprime o grafico
92     Salarios = matrix(
93         [[151],[180],[200],[240],[260],[300],[350],[380],[415],
94
95         [465],[510],[545],[622],[678],[724],[788],[880],[937],[954]]
96     )
97     ano = 15
98     t = ano
99     tam = len(Salarios)-ano
100     print("1)\na)")
101     Error = []
102     for i in range(tam):
103         X = matrix( [[-1,1],[0,1],[1,1],[2,1]] )
104         Y = matrix( [[151],[180],[200],[240]] )
105         XA = X
106         YA = Y
107         a = 0
108         b = len(Y)
109         c = 2000 + t
110         sal = calculo(X,Y,a,b,i + 1,m,t,1)
111         Error.append(sal)
112         print('Salario em %d = R$%0.2f' %(c,Salarios[t]))
113         print("Projecao para %d com %d dados = R$%0.2f\n"
114               %(c,len(Y),sal))
115         t += 1
116     erro = 0
117     for i in range(tam):
118         erro += abs(Error[i]-Salarios[ano])
119     print('\ne)')
120     print('Erro = %0.3f\n' %(erro/len(Y)))

```



```

118     t = ano
119     print('\nb)')
120     Error = []
121     for i in range(tam):
122         X = matrix(
123             [[-1,1],[0,1],[1,1],[2,1],[3,1],[4,1],[5,1],[6,1]]
124             )
125         Y = matrix(
126             [[151],[180],[200],[240],[260],[300],[350],[380]]
127             )
128         XB = X
129         YB = Y
130         a = 0
131         b = len(Y)
132         c = 2000 + t
133         sal = calculo(X,Y,a,b,i + 1,m,t,1)
134         Error.append(sal)
135         print('Salario em %d = R$%0.2f' %(c, Salarios[t]))
136         print("Projecao para %d com %d dados = R$%0.2f\n"
137               %(c, len(Y), sal))
138         t += 1
139     erro = 0
140     for i in range(tam):
141         erro += abs(Error[i]-Salarios[ano])
142     print('\ne)')
143     print('Erro = %0.3f\n' %(erro/len(Y)))
144
145     t = ano
146     print('\nc)')
147     Error = []
148     for i in range(tam):
149         X = matrix(
150             [[-1,1],[0,1],[1,1],[2,1],[3,1],[4,1],[5,1],[6,1],
151             [7,1],[8,1],[9,1],[10,1],[11,1],[12,1],[13,1]] )
152         Y = matrix(
153             [[151],[180],[200],[240],[260],[300],[350],[380],
154             [415],[465],[510],[545],[622],[678],[724]] )
155         XC = X
156         YC = Y
157         a = 0
158         b = len(Y)
159         c = 2000 + t
160         sal = calculo(X,Y,a,b,i + 1,m,t,1)
161         Error.append(sal)

```

```

155     print('Salario em %d = R$%0.2f' %(c, Salarios[t]))
156     print("Projecao para %d com %d dados = R$%0.2f\n"
           %(c, len(Y), sal))
157     t += 1
158     erro = 0
159     for i in range(tam):
160         erro += abs(Error[i] - Salarios[ano])
161     print('\ne')
162     print('Erro = %0.3f\n' %(erro/len(Y)))
163
164     print('\nd')
165     print('Sim, pois quantos mais dados, mais confiavel se
           torna a projecao.')
166
167     print('\n2')
168     X = [1,2,3,4,5,6,7]
169     Y = [4499,15205,37808,75336,130799,208281,240749]
170
171     ajuste = 0
172     if ajuste == 0:
173         Yln = []
174         for i in range(len(Y)):
175             Yln.append(math.log(Y[i]))
176         soma1 = 0
177         for i in range(len(X)):
178             soma1 += X[i]**2
179         soma2 = 0
180         for i in range(len(X)):
181             soma2 += X[i]*Yln[i]
182         print('%da + %0.2fb = %0.2f' %(len(X), sum(X), sum(
           Yln)))
183         print('%0.2fa + %0.2fb = %0.2f' %(sum(X), soma1,
           soma2))
184         a1 = -soma1 * len(X)
185         c1 = -soma1 * sum(Yln)
186         a2 = sum(X) * sum(X)
187         c2 = sum(X) * soma2
188         a = (c1 + c2)/(a1 + a2)
189         print('\na = %0.4f' %a)
190         b = (sum(Yln) - len(X) * a)/sum(X)
191         print('b = %0.4f' %b)
192         print('\ny = %0.0fe^%0.1fx' %(math.exp(a), b))
193         t = 6
194         y = math.exp(a) * math.exp(b * (t))
195         print('\nProjecao para 2000 e \t= %d' %(y))
196         print('Dado na Tabela \t\t= %d' %Y[t - 1])

```



1)

a)

theta 1 = 28.7000

theta 2 = 178.4000

Salario em 2015 = R\$788.00

Projeção para 2015 com 4 dados = R\$580.20

Salario em 2016 = R\$880.00

Projeção para 2016 com 4 dados = R\$608.90

Salario em 2017 = R\$937.00

Projeção para 2017 com 4 dados = R\$637.60

Salario em 2018 = R\$954.00

Projeção para 2018 com 4 dados = R\$666.30

e)

Erro = 164.750

b)

theta 1 = 33.0119

theta 2 = 175.0952

Salario em 2015 = R\$788.00

Projeção para 2015 com 8 dados = R\$637.26

Salario em 2016 = R\$880.00

Projeção para 2016 com 8 dados = R\$670.27

Salario em 2017 = R\$937.00

Projeção para 2017 com 8 dados = R\$703.29

Salario em 2018 = R\$954.00

Projeção para 2018 com 8 dados = R\$736.30

e)

Erro = 50.610

c)

theta 1 = 40.9786

theta 2 = 155.4619

Salario em 2015 = R\$788.00

Projeção para 2015 com 15 dados = R\$729.16

Salario em 2016 = R\$880.00

Projeção para 2016 com 15 dados = R\$770.14

Salario em 2017 = R\$937.00

Projeção para 2017 com 15 dados = R\$811.12

Salario em 2018 = R\$954.00

Projeção para 2018 com 15 dados = R\$852.10

e)

Erro = 10.928

d)

Sim, pois quantos mais dados, mais confiável se torna a projeção.

2)

$$7a + 28.00b = 76.23$$

$$28.00a + 140.00b = 323.34$$

$$a = 8.2593$$

$$b = 0.6577$$

$$y = 3863e^{0.7x}$$

Projeção para 2000 é = 199877

Dado na Tabela = 208281

-----  
Diferença = 8403

Erro = 1400.5151

Erro Quadrático = 3430.5474

Projeção para 2010 é = 385830

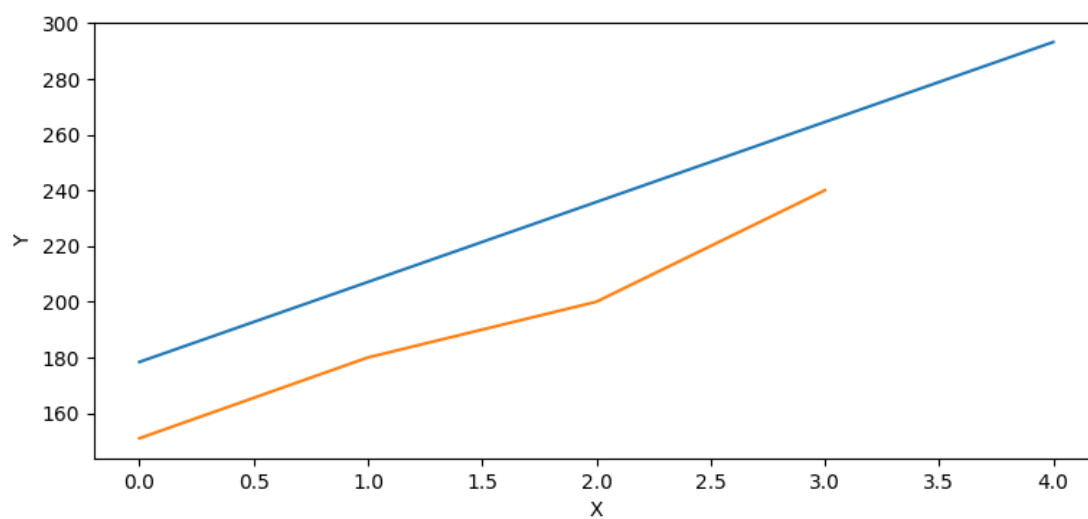
Dado na Tabela = 240749

-----  
Diferença = 145081

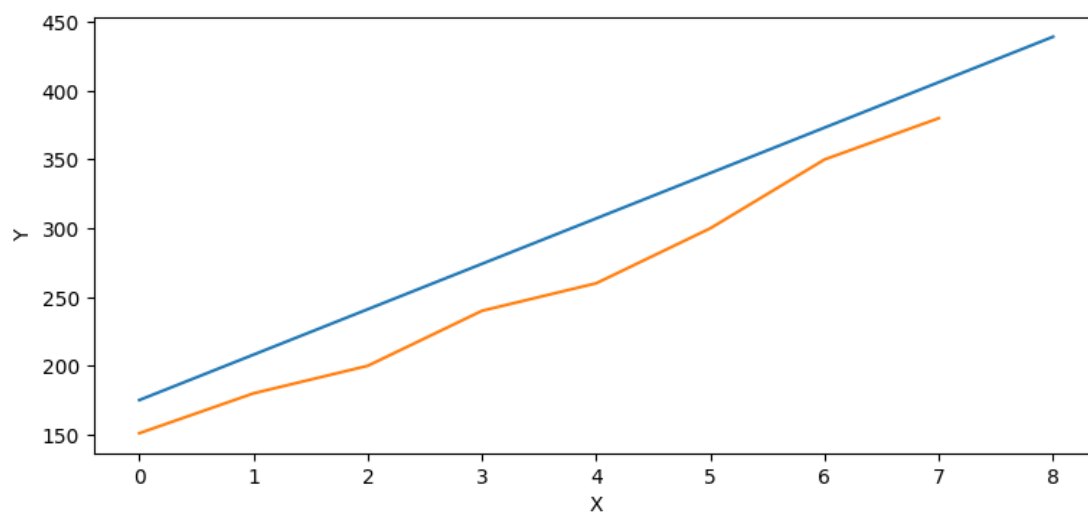
Erro = 20725.9443

Erro Quadrático = 54835.6942

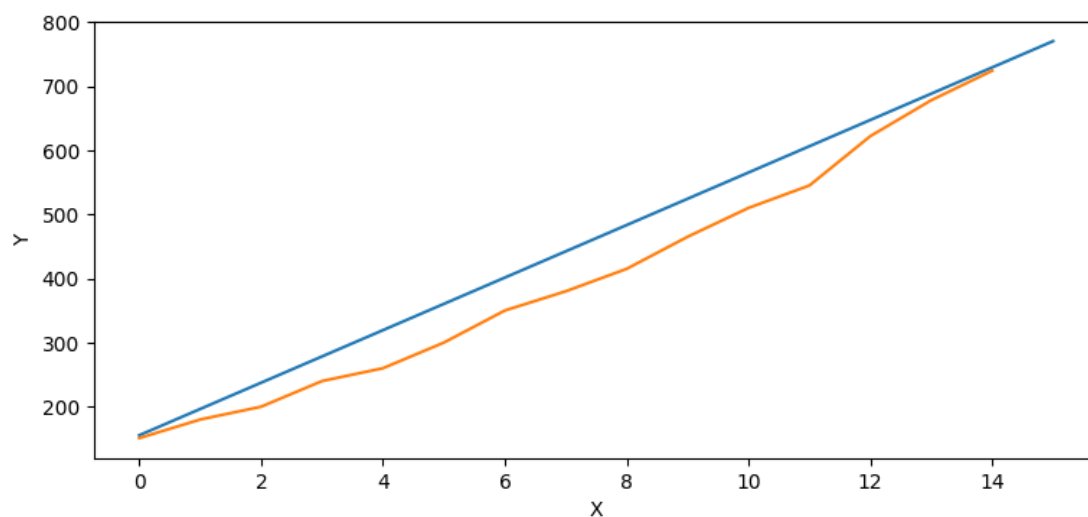
### 3.3. Gráficos



**Figura 1. Gráfico comparativo dos anos 2015 a 2018 Projetado vs Real com 4 dados**



**Figura 2. Gráfico comparativo dos anos 2015 a 2018 Projetado vs Real com 8 dados**



**Figura 3. Gráfico comparativo dos anos 2015 a 2018 Projetado vs Real com 15 dados**