

```
118 plt.savefig(subpasta + pastaOriginal + 'Função original')
119
120 #4.2
121 plt.figure()
122 plt.plot(x,y,color='black')
123
124 plt.xlabel('Tempo (s)')
125 plt.ylabel('Sinal de entrada (int)')
126 plt.savefig(subpasta + pastaOriginal + 'Função original Interpolada')
127
128 #4.3
129 #derivando por equação de diferenças
130 df = GeraEquacaoDeDiferencas(x,y)
131 df2 = GeraEquacaoDeDiferencas(x,df)
132 GerarGráficosDeEquacaoDeDiferencas(x,y,df,df2,N*T,paciente,subpasta + pastaOriginal
+ 'Equações de Diferenças')
133
134 #4.4
135 yf = fft(y)
136 xf = np.linspace(0.0, 1.0/(2.0*T), N//2)
137
138 plt.figure()
139 plt.plot(xf, 2.0/N * np.abs(yf[0:N//2]))
140 plt.savefig(subpasta + pastaOriginal + 'FFT')
141
142
143 #5
144 b, a = signal.butter(3, 0.5)
145 zi = signal.lfilter_zi(b,a)
146 y_filtrado, _ = signal.lfilter(b,a, y, zi=zi*y[0])
147 gerarCsv(x,y_filtrado,subpasta + pastaIIR )
148
149
150 #5.1
151 plt.figure()
152 plt.plot(x,y_filtrado,'ro')
153 plt.xlabel('Tempo (s)')
154 plt.ylabel('Sinal de entrada (int)')
155 plt.savefig(subpasta + pastaIIR + 'Função original')
156
157 #5.2
158 plt.figure()
159 plt.plot(x,y_filtrado,color='black')
160 plt.xlabel('Tempo (s)')
161 plt.ylabel('Sinal de entrada (int)')
162 plt.savefig(subpasta + pastaIIR + 'Função original Interpolada')
163
164 #5.3
165 df = GeraEquacaoDeDiferencas(x,y_filtrado)
166 df2 = GeraEquacaoDeDiferencas(x,df)
167 GerarGráficosDeEquacaoDeDiferencas(x,y_filtrado,df,df2,N*T,paciente,subpasta +
pastaIIR + 'Equações de Diferenças')
168
169 #5.4
170 yf2 = fft(y_filtrado)
171 plt.figure()
172 plt.plot(xf, 2.0/N * np.abs(yf2[0:N//2]), 'r')
173 plt.savefig(subpasta + pastaIIR + 'FFT')
174
175
176
```