

```
In[27]:= SetDirectory[NotebookDirectory[] <> ".."];
```

```
data = {{0, 1.899}, {0.2, 1.887}, {0.4, 1.875}, {0.6, 1.863}, {0.8, 1.852},
  {1, 1.841}, {1.2, 1.823}, {1.4, 1.818}, {1.6, 1.806}, {1.8, 1.795},
  {2, 1.784}, {2.2, 1.766}, {2.4, 1.76}, {2.6, 1.748}, {2.8, 1.737}, {3, 1.724},
  {3.2, 1.708}, {3.4, 1.702}, {3.6, 1.691}, {3.8, 1.679}, {4, 1.668},
  {4.2, 1.652}, {4.4, 1.644}, {4.6, 1.631}, {4.8, 1.621}, {5, 1.609}, {0, 1.896},
  {0.2, 1.885}, {0.4, 1.867}, {0.6, 1.858}, {0.8, 1.846}, {1, 1.834},
  {1.2, 1.827}, {1.4, 1.811}, {1.6, 1.799}, {1.8, 1.788}, {2, 1.781},
  {2.2, 1.769}, {2.4, 1.757}, {2.6, 1.743}, {2.8, 1.734}, {3, 1.718},
  {3.2, 1.712}, {3.4, 1.695}, {3.6, 1.684}, {3.8, 1.671}, {4, 1.665},
  {4.2, 1.654}, {4.4, 1.642}, {4.6, 1.631}, {4.8, 1.619}, {5, 1.608}};
comercialModelFit = LinearModelFit[data, {1, F}, F];
comercialModel = InverseFunction[comercialModelFit];
```

```
comercialPlot = Show[
  Plot[comercialModelFit[F], {F, Min[data[[All, 1]]], Max[data[[All, 1]]]},
  ListPlot[data, PlotStyle -> Red],
  AxesLabel -> {"Massa (kg)", "Tensão elétrica (V)"}
];
Export["Images/Comercial-Plot.pdf", comercialPlot];
```

```
data = {{0, 1.273}, {0.2, 1.262}, {0.4, 1.249},
  {0.6, 1.242}, {0.8, 1.233}, {1, 1.226}, {0, 1.272}, {0.2, 1.261},
  {0.4, 1.253}, {0.6, 1.244}, {0.8, 1.233}, {1, 1.225}};
nonComercialModelFit = LinearModelFit[data, {1, F}, F];
nonComercialModel = InverseFunction[nonComercialModelFit];
```

```
nonComercialPlot = Show[
  Plot[nonComercialModelFit[F],
  {F, Min[data[[All, 1]]], Max[data[[All, 1]]]},
  ListPlot[data, PlotStyle -> Red],
  AxesLabel -> {"Massa (kg)", "Tensão elétrica (V)"}
];
Export["Images/NaoComercial-Plot.pdf", nonComercialPlot];
```

```
data =
  {{0, -101.1}, {20, -73.4}, {30, -65.3}, {40, -64.3}, {50, -53.6}, {60, -44.7}};
torquimeterModel = LinearModelFit[data, {1, F}, F];
```

```
DoProcessing[file_, export_, title_, tf_, offset_, filter_, reference_] := (
  data = Import[file, "TSV", NumberPoint -> ","];
  frequency = 1 / (data[[2, 1]] - data[[1, 1]]);
  data[[All, 2]] = tf / @data[[All, 2]];
  data[[All, 2]] = LowpassFilter[data[[All, 2]], filter / frequency];
  data[[All, 2]] += offset;
  plot = Show[
    ListPlot[data, Joined -> True, PlotLabel -> title,
    PlotRange -> Full, AxesLabel -> {"Tempo (s)", "Massa (kg)"}],
    Plot[reference, {x, Min[data[[All, 1]]], Max[data[[All, 1]]]},
    PlotStyle -> {Dashed, Red, Thin}, PlotRange -> Full]
  ];
  Export[export, plot];
  Return[plot];
);
```

```
(* Comercial *)
DoProcessing["Data/Comercial_0-1kg.lvm", "Images/Comercial_0-1kg.pdf",
  "Comercial 0-1kg", comercialModel, 0.45, 10, 1];
```

```

DoProcessing["Data/Comercial_0-4kg_etapa.lvm",
  "Images/Comercial_0-4kg_etapa.pdf",
  "Comercial 0-4kg (em 2 etapas)", comercialModel, 0.45, 10, 4];
DoProcessing["Data/Comercial_0-4kg.lvm", "Images/Comercial_0-4kg.pdf",
  "Comercial 0-4kg", comercialModel, 0.45, 10, 4];
DoProcessing["Data/Comercial_1-0kg.lvm", "Images/Comercial_1-0kg.pdf",
  "Comercial 1-0kg", comercialModel, 0.45, 10, 1];
DoProcessing["Data/Comercial_4-0kg.lvm", "Images/Comercial_4-0kg.pdf",
  "Comercial 4-0kg", comercialModel, 0.45, 10, 4];

(* Não comercial *)
DoProcessing["Data/NaoComercial_0-1000g_direto.lvm",
  "Images/NaoComercial_0-1000g.pdf",
  "Não-Comercial 0-1000g", nonComercialModel, 0.2, 5, 1];
DoProcessing["Data/NaoComercial_0-400g_direto.lvm",
  "Images/NaoComercial_0-400g.pdf",
  "Não-Comercial 0-400g", nonComercialModel, 0.2, 5, 0.4];
DoProcessing["Data/NaoComercial_0-400g_etapa.lvm",
  "Images/NaoComercial_0-400g_etapa.pdf",
  "Não-Comercial 0-400g (em 2 etapas)", nonComercialModel, 0.2, 5, 0.4];
DoProcessing["Data/NaoComercial_1000g-0_direto.lvm",
  "Images/NaoComercial_1000g-0.pdf",
  "Não-Comercial 1000-0g", nonComercialModel, 0.2, 5, 1];
DoProcessing["Data/NaoComercial_400g-0_direto.lvm",
  "Images/NaoComercial_400g-0.pdf",
  "Não-Comercial 400-0g", nonComercialModel, 0.2, 5, 0.4];
DoProcessing["Data/NaoComercial_400g-0_etapa.lvm",
  "Images/NaoComercial_400g-0_etapa.pdf",
  "Não-Comercial 400-0g (em 2 etapas)", nonComercialModel, 0.2, 5, 0.4];

heatingData1 =
  Import["Data/NaoComercial_aquecimento.lvm", "TSV", NumberPoint → ","];
heatingData2 = Import["Data/NaoComercial_aquecimento_1.lvm",
  "TSV", NumberPoint → ","];
heatingData2[[All, 1]] += Max[heatingData1[[All, 1]]];
heatingData = Join[heatingData1, heatingData2];

frequency = 1 / (heatingData[[2, 1]] - heatingData[[1, 1]]);
heatingData[[All, 2]] = LowpassFilter[heatingData[[All, 2]], 0.1 / frequency];
heatingData[[All, 2]] = nonComercialModel /@ heatingData[[All, 2]] + 2.8;

plot = ListPlot[heatingData, Joined → True,
  PlotLabel → "Não-Comercial -- Drift de aquecimento",
  PlotRange → Full, AxesLabel → {"Tempo (s)", "Força (N)"}];
Export["Images/NaoComercial-Aquecimento.pdf", plot];

```

```

frequencyData =
  Import["Data/NaoComercial_batida_5.lvm", "TSV", NumberPoint → ","];
frequency = 1 / (frequencyData[[2, 1]] - frequencyData[[1, 1]]);

frequencyDataFourier = frequencyData;
frequencyDataFourier[[All, 2]] =
  frequencyDataFourier[[All, 2]] - Mean[frequencyDataFourier[[All, 2]]];
frequencyDataFourier[[All, 2]] = Abs[Fourier[frequencyDataFourier[[All, 2]]]];
frequencyDataFourier[[All, 1]] =
  Table[i, {i, 0, frequency, (frequency) / (Length[frequencyData] - 1)}];

interp = Interpolation[frequencyDataFourier,
  InterpolationOrder → 3, Method → "Spline"];

maximum = FindMaximum[interp[ $\omega$ ], { $\omega$ , 20}];

spectrum = Plot[interp[ $\omega$ ], { $\omega$ , 0, 100}, PlotRange → Full, Filling → Axis,
  AxesLabel → {"Frequência (Hz)", "Magnitude do Espectro"}, ImageSize → Full];
Export["Images/NaoComercial-Spectrum.pdf", spectrum];

spectrumHighlight = Show[
  Plot[interp[ $\omega$ ], { $\omega$ , 18, 25}, PlotRange → {{18, 25}, Full}, Filling → Axis,
    AxesLabel → {"Frequência (Hz)", "Magnitude do Espectro"}],
  ListPlot[{{ $\omega$  /. maximum[[2]], maximum[[1]]}}, Filling → Axis,
    PlotRange → {{18, 25}, Full}, PlotStyle → Red]
];
Export["Images/NaoComercial-SpectrumHighlight.pdf", spectrumHighlight];

In[52]:= Export[NotebookFileName[] <> ".pdf", EvaluationNotebook[]];

```