

Exercise 04: 2d-Visualization with matplotlib

Objective

In the technical field, the visualization of data is a very helpful tool. Often in calculations intermediate results are displayed graphically, to check for plausibility or to find errors in the calculation. Furthermore, visualizations facilitate the documentation of scientific work (experiments, calculations, ...).

In this course knowledge about 2D visualization of numerical data with the package matplotlib is presented. As a result, the participants are able to plot and save the results of a simulation calculation in a quality suitable for publication (e.g. for further use in \LaTeX or Word).

Exercise 04.1

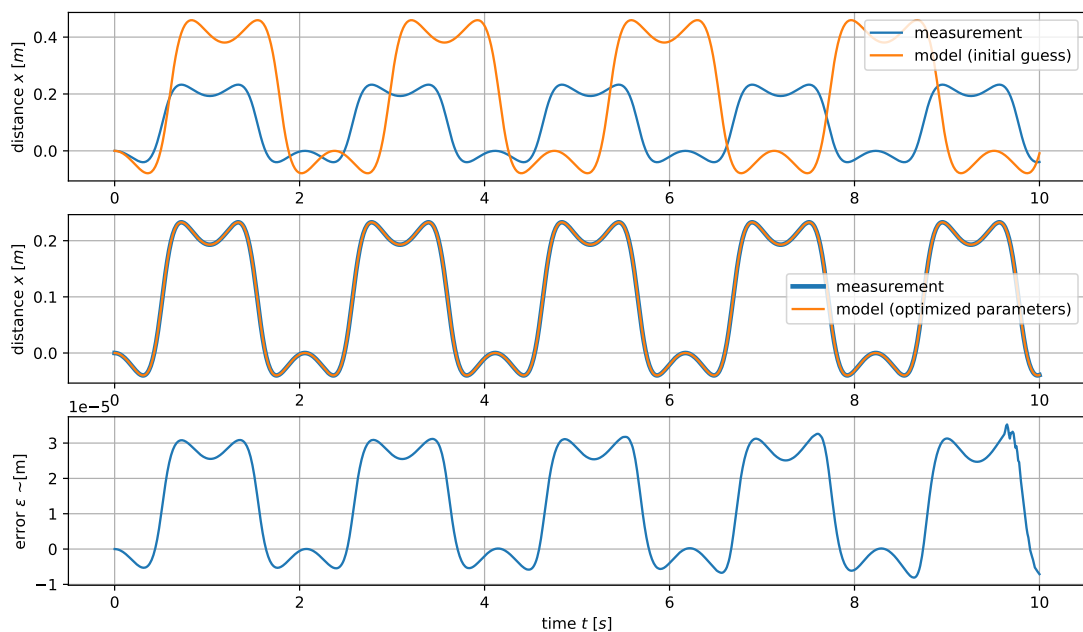
Result visualization of the identification task from Exercise 03 (see given file:

`skeleton-code/01_plot.py`)

Note: Only the curves of the result variable x (distance traveled by the trolley; $\hat{=}$ first component of the state vector) are used.

1. Create a list to store the intermediate results of the optimization (in the top level of the script, outside of any function) and modify the `min_target` function to append `res` to this list at each optimization step.
2. Create a new `figure` with a 3x1 subplot layout (See [reference](#), [example](#)).
3. Plot the data from the results file and the result of the first optimization step in subplot 1.
4. Plot the data from the result file and the result of the last optimization step in subplot 2.
5. Plot the error (i.e. the difference) between result file and optimized simulation model in subplot 3.
6. Complete the plot (axis labels, legend, auxiliary lines) and save it in pdf and png format.

Result:



7. What could be improved about the depiction? Try to implement your own ideas.

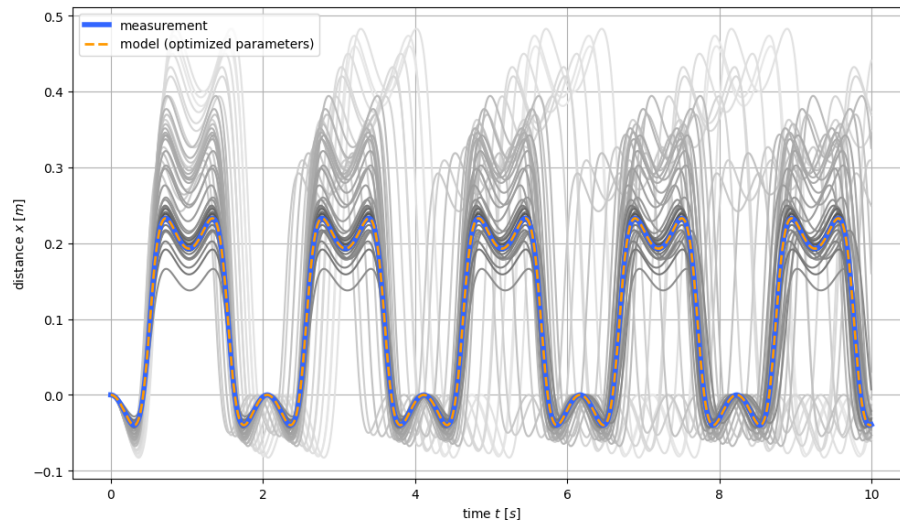
Exercise 04.2

2d visualization of the progress of optimization (also see given file: `skeleton-code/01_plot.py`)

Now all intermediate results of the optimization are to be displayed in one diagram. To maintain its readability, a color gradient corresponding to the optimization step should be provided.

1. Create a new `figure` object including subplot (1x1 layout).
2. Create a plot for each intermediate result except for the final result and assign a gray tone between 0 and 1 (linearly scaled to the length of the list)
3. Plot the measurement in color
4. Plot the result of the optimization in color
5. Complete the diagram (see task 04.1)

Result:



Exercise 04.3 (optional)

3d plot of the same data (also see given file: `skeleton-code/01_plot.py`)

1. Generate a meshgrid for the all intermediate plots (see [reference](#)).
2. Generate a 3d plot with `plot_surface(...)` (see [reference](#), [example](#)).
3. What is the influence of the parameters `rstride` and `cstride` ?
4. Complete the plot, try different colormaps (`matplotlib.cm`).

Result:

