Prof. Dr. Kathrin Flaßkamp
M.Sc. Matthias Hoffmann
Systems Modeling and Simulation
Saarland University

Optimization – Project 1 – WS 21/22

Constrained and unconstrained optimization

Project task 1.1 – Mass-spring-damper system

Consider the mass-spring-damper system

$$\ddot{p} = -\frac{1}{m}k \cdot p - \frac{1}{m}d \cdot \dot{p} \quad \Leftrightarrow \quad \dot{\mathbf{x}} = \begin{pmatrix} 0 & 1 \\ -\frac{1}{m}k & -\frac{1}{m}d \end{pmatrix}\mathbf{x}, \qquad \mathbf{x}_0 = \begin{pmatrix} p(0) \\ \dot{p}(0) \end{pmatrix} = \begin{pmatrix} 2 \\ 0 \end{pmatrix}$$

with state x consisting of the position p and the velocity $\dot{p}$. Assume that all three parameters m, k, d (i.e. mass, spring and damping coefficients) are unknown. In a real-world experiment, (noisy) measurements ($\mathbf{x}_{j,\text{meas}}$, $j = 1, \dots, N$) for an experiment starting with the initial condition $\mathbf{x}_0$ were collected. (Download the file `x_star_noise.mat`, in which measurements are taken for time steps $\Delta t = 0.01$.)

a) Does a unique solution for three estimation parameters exist? If no, explain why.

b) Write a MATLAB function which, for a fixed parameter set m, k, d, outputs the trajectory x on a suitable time interval as a solution to the linear mass-spring-damper system? *Hint:* You can either compute and implement the analytical solution, or make use of a numerical integration method (self-coded or, e.g. `ode45`).

c) Make sure your function from b) can output simulated data $\mathbf{x}_{j,\text{sim}}$, $j = 1, \dots, N$ on a time grid with step size $\Delta t = 0.01$ and of suitable length N, matching the resolution of the measurements. Now you can formulate the **Problem of Least-Squares** using noisy measurements $\mathbf{x}_{j,\text{meas}}$ and the simulated data $\mathbf{x}_{j,\text{sim}}$, $j = 1, \dots, N$: Start with a mathematical formulation on paper, then, write a MATLAB function

```
function res = get_residuum(param,sim_param)
```

that computes the objective value of the residuum function.

d) Implement the above sketched parameter estimation problem in MATLAB using `fminunc` and run your code. This might look like

```
sim_param = {x_0,delta,N,x_star_noise}; %delta is the time step size, N number of measurements
param_0 = [2;4;0.1]; %inital guess for parameters
opt_fun = @(param)get_residuum(param,sim_param);
[param_star,fval] = fminunc(opt_fun,param_0);
```

e) Assume your optimization of the three parameters yields the solution $\tilde{m}_{\text{est}}, \tilde{k}_{\text{est}}, \tilde{d}_{\text{est}}$. If now m is known, how can the best possible estimations for k and d be obtained? What is the alternative to estimating all three parameters?

Project task 1.2 – Denoising

Sensor signals typically show deterministic and stochastic errors. While deterministic errors can be corrected with an error model, stochastic errors need to be removed from the signal using filters. As these errors have a high frequency, low-pass filters can help to reduce them at the cost of a phase shift in the signal. Optimization techniques offer a different approach to this problem.

a) Load the data from `measurements.mat` and visualize them. `y_raw` is the measured data with noise and `t` the evenly spaced timings of the measurements.
Hint: Since this data has been generated synthetically, we can give you `y` as the baseline, so that you can check the performance of your denoising method. You should not use `y` within the optimization, though.

b) **Objective:** We aim to find a denoised signal `y_den` by minimizing the distance of `y_den` to `y_raw`, as in least-squares. Write an objective function in MATLAB and try to run with `fmincon`, although you do not have constraints, yet. What is the resulting `y_den`?

c) **Constraints:** To make it a meaningful denoising algorithm, constraints have to be added. Let us assume, the measured quantity's values cannot change by more than $1\,\mathrm{s}^{-1}$. Implement this idea as linear constraints (*Hint:* Forward difference/difference quotient, dt. Differenzenquotient) for a constrained optimization problem using `fmincon`. Run your code.

d) **Second objective:** In the previous solutions, you should have seen a massive improvement in the reconstruction of `y`, but some spikes remain. An additional factor in your cost function can help to further improve your denoising: The smoothness of the predicted signal can, for example, be described as

$$J_{\text{smooth}} = \sum_i ||y_{\text{den},i} - y_{\text{den},i-1}||_p^p, \;\; p \in \mathbb{N}^+$$

Implement this regularization function, or your alternative approach to it, and add it to the cost function. A weightening between the two objectives might be necessary for best reconstruction.

e) Compare your results with the provided code for a low-pass filter.

Your main script shall give a figure comparing the solutions of b), c), d) and e) in subfigures, for instance. It might look like the figure you find in opt_comparison.pdf

Please send your solutions to these tasks to matthias.hoffmann@uni-saarland.de by 11.12.2021.