Exercises for Course on Modeling and System Identification (MSI)
Albert-Ludwigs-Universität Freiburg – Winter Term 2018-2019

# Exercise 8: Nonlinear Least Squares

**(to be returned on Jan 18, 2019, 10:00 in SR 00-010/014, or before in building 102, 1st floor, 'Anbau')**

Prof. Dr. Moritz Diehl, Tobias Schöls, Katrin Baumgärtner, Alexander Petrov

In this exercise you will learn how to practically solve nonlinear least squares problems, compute estimates of the covariance of your estimated parameters and make statements about the correctness of the model assumptions.

## Exercise Tasks

1. **Covariance approximation** (3 points)

   Consider a nonlinear function $f : \mathbb{R}^n \to \mathbb{R}$ that maps a random vector $X = (X_1, \dots, X_n)^\top$ to a scalar random variable $Y$, i.e.

   $$Y = f(X) = f(X_1, \dots, X_n).$$

   We have $\mathbb{E}\{X\} = \mu_x = (\mu_1, \dots, \mu_n)^\top$ and $\text{cov}(X) = \Sigma_x \in \mathbb{R}^{n \times n}$.

   (a) ON PAPER: Give an approximation of the expected value $\mathbb{E}\{Y\}$ and the covariance matrix $\text{cov}(Y)$ of $Y$ using a first order Taylor expansion of $f$ around $\mu_x$. (2 points)

   (b) ON PAPER: Suppose $X_1, \dots, X_n$ are independent. Simplify your covariance approximation from part (a). (1 point)

2. **Parameter estimation for output error minimization** (8 points)

   You operate a two-wheeled robot with unknown dimensions (left wheel radius $R_L$, right wheel radius $R_R$, and axle length $L$), as simulated in Exercise 7. After observing the movement of the robot, you would like to estimate these dimensions $\theta = (R_L, R_R, L)^\top$, with `lsqnonlin`[1] . Assuming that the robot system has only output errors, and that these errors are Gaussian with zero mean and variances $\sigma_x^2 = 1.6 \cdot 10^{-3}$ m$^2$ and $\sigma_y^2 = 4 \cdot 10^{-4}$ m$^2$, then the Maximum Likelihood Estimation problem to estimate $\theta$ is:

   $$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta} \in \mathbb{R}^3} \sum_{k=0}^{N} \|\boldsymbol{y}_k - M_k(\boldsymbol{u}, \boldsymbol{q}_0, \boldsymbol{\theta})\|^2_{\boldsymbol{\Sigma}_{\boldsymbol{y}}^{-1}},$$

   where $\boldsymbol{y}_k = (x, y)^\top \in \mathbb{R}^2$ with $x$ and $y$ being the coordinates of the robot and $N$ is the number of measurements; $\boldsymbol{\Sigma}_{\boldsymbol{y}}$ is the weighing matrix containing the variances on the $x$ and $y$ measurements defined as:

   $$\boldsymbol{\Sigma}_{\boldsymbol{y}} = \begin{bmatrix} \sigma_x^2 & 0 \\ 0 & \sigma_y^2 \end{bmatrix};$$

   $M_k(\boldsymbol{u}, \boldsymbol{q}_0, \boldsymbol{\theta})$ denotes the modeled position at timestep $k$ for given $\boldsymbol{u}, \boldsymbol{q}_0, \theta$ where $\boldsymbol{u} \in \mathbb{R}^{(N-1)\times 2}$ is a matrix that contains all applied control inputs, consisting of the angular velocity of the left and right wheel respectively ($\omega_L$ and $\omega_R$); $\boldsymbol{q}_0$ contains the robot's initial pose ($x$-$y$-position and orientation) $\boldsymbol{q}_0 = (x_0, y_0, \beta_0) = (0, 0, 0)$ which we assume to be perfectly known. The measurements are taken at a sampling time of $\Delta t = 0.01$ s.

[1] `lsqnonlin` takes as input a vector function $f(\theta) = [f_1(\theta), \dots, f_N(\theta)]$, and minimizes $\|f(\theta)\|_2^2$ with respect to $\boldsymbol{\theta}$. Thus, you have to stack the residuals obtained for different timesteps to obtain a *single* residual vector.

The kinematic model is given by

$$\dot{\mathbf{q}} = \begin{pmatrix} v \cdot \cos\beta \\ v \cdot \sin\beta \\ \frac{\omega_L R_L - \omega_R R_R}{L} \end{pmatrix}, \tag{1}$$

where the robot's velocity $v$ is given by $v = \frac{\omega_{\mathrm{L}} \cdot R_{\mathrm{L}} + \omega_{\mathrm{R}} \cdot R_{\mathrm{R}}}{2}$ and $\mathbf{q} = (x, y, \beta)^\top$ is the robot's pose.

(a) ON PAPER: First formulate a discrete time model for the robot's dynamics $F : \mathbb{R}^3 \to \mathbb{R}^3$ using a one-step Euler integrator and the kinematic model given in (1). Then formulate the output model

$$M_k : \mathbb{R}^{(N-1)\times 2} \times \mathbb{R}^3 \times \mathbb{R}^3 \to \mathbb{R}^2,\ (\boldsymbol{u}, \boldsymbol{q}_0, \boldsymbol{\theta}) \mapsto \hat{\boldsymbol{y}}_k$$

*Hint: You may use $F$ for the formulation of $M_k$.* (1 point)

(b) MATLAB: Implement a function:
`residual(theta, q0, u, k_deltaT, y, sigma_y)`, which computes the residual vector between the given measured location $\boldsymbol{y}_k$ and the modeled location $M_k(\boldsymbol{u}, \boldsymbol{q}_0, \boldsymbol{\theta})$ . Keep in mind to incorporate the measurement variances $\boldsymbol{\Sigma_y}$ correctly, i.e. weight the residual and to perform the right number of integration steps. (2 points)

(c) MATLAB: Use `lsqnonlin` to estimate $\theta^*$. (1 point)

(d) MATLAB: Compute the simulated trajectory using $\theta^*$ and plot it versus the measurements using the provided code and compare it to the 4th order polynomial fit shown in Figure 1 (this is what you did for the last exercise sheet).
ON PAPER: What do you observe? (1 point)

(e) ON PAPER: Check if the assumptions made on the noise were correct by plotting a histogram for the residual in $x$ and $y$ (using $\theta^*$). (1 point)

(f) MATLAB: Approximate the covariance matrix $\boldsymbol{\Sigma}_{\theta^*}$ of your estimate $\theta^*$ (check page 48 of the lecture notes). (2 points)

3. ON PAPER: Suppose you have identified the robot's kinematic model as done in task 2. You are now given another series of controls $\mathbf{U} \in \mathbb{R}^{N-1\times 2}$ and are asked to predict the robot's end pose and to give an estimate for the covariance matrix $\Sigma_{\boldsymbol{y}_N}$ of your prediction. Describe how this can be done. You may use any result or quantity introduced or computed in tasks 1 and 2. (2 points)

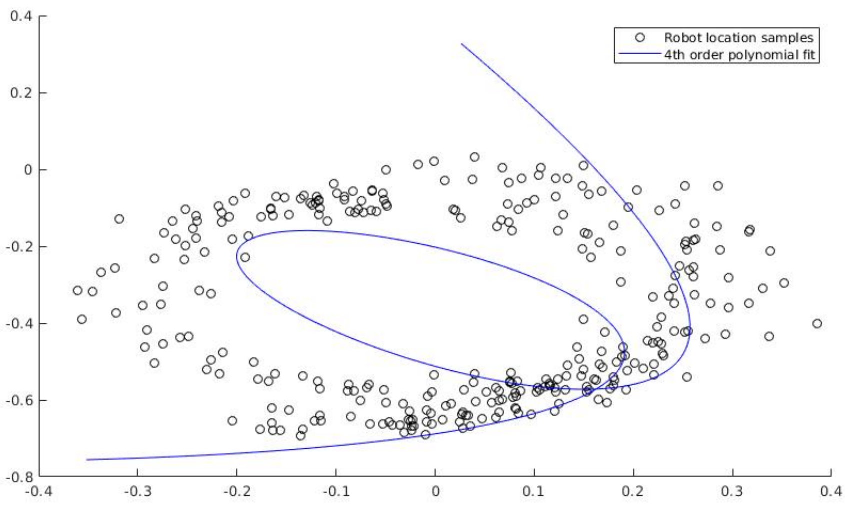

Figure 1: Fit from Exercise 7

*This sheet gives in total 13 points.*