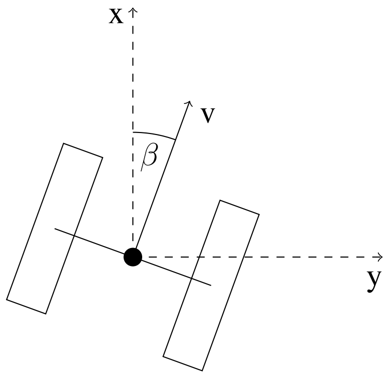
Exercises for Course on Modeling and System Identification (MSI)
Albert-Ludwigs-Universität Freiburg – Winter Term 2019-2020

Exercise 7: Recursive Least Squares

(to be returned on Dez 18th, 2019, 8:30 in HS 00 036 (Schick-Saal), or before in building 102, 1st floor, 'Anbau')

Prof. Dr. Moritz Diehl, Tobias Schöls, Naya Baslan, Jakob Harzer, Bryan Ramos

In this exercise you will implement a Recursive Least Squares (RLS) estimator and a forward simulation of a differential drive robot with unicycle dynamics. We will apply the RLS algorithm to position data of a 2-DOF movement in the X-Y plane, measured with a sampling time of $0.0159\,\mathrm{s}$. The movement of the robot depends on the angular velocities of the left and the right wheel ω_L and ω_R, as well as on their radii R_L and R_R. Differing radii influence the behaviour of the robot.



The system can be described by a state space model with three internal states. The state vector $\mathbf{x} = [x,\ y,\ \beta]^\top$ contains the position of the robot in the $X - Y$ plane and the deviation β from its initial orientation. The system can be controlled by the angular velocities of the wheels: $\mathbf{u} = [\omega_\mathrm{L},\ \omega_\mathrm{R}]^\top$. The output of the system is the position of the robot: $\mathbf{y} = [x,\ y]^\top$. The model follows as

$$\dot{\mathbf{x}} = \begin{pmatrix} v \cdot \cos\beta \\ v \cdot \sin\beta \\ \frac{\omega_\mathrm{L} R_\mathrm{L} - \omega_\mathrm{R} R_\mathrm{R}}{L} \end{pmatrix} \qquad \mathbf{y} = \begin{pmatrix} x \\ y \end{pmatrix} \tag{1}$$

with L being the length of the axis between the two wheels and the velocity v being

$$v = \frac{\omega_\mathrm{L} \cdot R_\mathrm{L} + \omega_\mathrm{R} \cdot R_\mathrm{R}}{2}.$$

1. **Recursive Least Squares applied to position data**

   In this task you will implement the Recursive Least Squares (RLS) algorithm in MATLAB and tune the *forgetting factors*. The robot's kinematic model introduced above is nonlinear. To obtain a linear-in-the-parameters (LIP) model, we approximate the position data it by a fourth order polynomial. You can assume that the noise on the X and Y measurements is independent. The experiment starts at $t = 0\,\mathrm{s}$.

   (a) MATLAB: Fit a 4-th order polynomial through the data using linear least-squares. Plot the data and the fit for the X- and Y-coordinate.
   Hint: You need one estimator for each coordinate.
   PAPER: Does the fit seem reasonable? Why do you think that is? (1 point)

(b) MATLAB: Implement the RLS algorithm as described in the script *(Check section 5.3.1)* to estimate 4-th order polynomials to fit the data. Do not use forgetting factors yet. Plot the result against the data.
PAPER: Compare the LS estimator from (a) with the RLS estimator you obtain after processing N measurements. Please give an explanation for your observation. (2 points)

(c) MATLAB: Add a forgetting factor α to your algorithm and try different values for α. Plot the results on the same plot as the previous question.
PAPER: How does α influence the fit? What is a reasonable value for α? (1 point)

(d) PAPER: How can you compute the covariance Σ_p of the position, if you know the covariance of the estimator $\Sigma_{\hat{\theta}}$?
Hint: For a random variable $\gamma = A\theta$, *where* A *is a matrix,* $\operatorname{cov}(\gamma) = A\operatorname{cov}(\theta)A^{\mathrm{T}}$. (1 point)

(e) MATLAB: Compute the *one-step-ahead* prediction at each point (i.e. extrapolate your polynomial fit to the next time step). We also provided code to plot the 1-σ confidence ellipsoid around this point, and the data.
PAPER: Do the confidence ellipsoids grow bigger or smaller as you take more measurements? Why do you think that is? (2 points)

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

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

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

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

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

This sheet gives in total 10 points