Graz University of Technology
Signal Processing and Speech Communication Laboratory
Nonlinear Signal Processing (course 442.022) Problem set 1
Christian Knoll, `christian.knoll@tugraz.at`

A Flavor of Nonlinearity: Median Filtering

Problem 1.1

Perform the following tasks using Matlab:

1. Generate signals **(a)**–**(e)** from the list below.
2. Using `impulse_noise.m` corrupt these signals with impulse noise and apply mean (low-pass) and median filters using scripts `mfilt1.m` and `mean1.m`, respectively.
3. Plot the enhanced (filtered) signals overlaid with the noisy ones, and compute the corresponding SNR values; repeat the filtering with different filter lengths. What can be observed? Explain your observations.

(a) $x[n] = 0.25 \cdot n; \quad n = [0:511]$

(b) $x[n] = \sin(\pi n/25); \quad n = [0:511]$

(c) $x[n] = \mathrm{sinc}(0.1\pi n); \quad n = [-256:255]$

(d) A filtered white noise sequence (generated with `randn`). As a filter use an IIR filter with the following coefficients: $\mathbf{b} = [1]$ (numerator), and $\mathbf{a} = [1, -1.94, 0.96]$ (denominator). Filtering can be implemented with the function `filter`.

(e) A *.wav file with music or speech. To load the sound file into the Matlab workspace use `audioread`, to play the file use `sound`.

Problem 1.2

Perform the same analysis as in Problem 1.1, but now for an image. Use the provided `mfilt2.m` and `mean2.m` scripts. To load and display images in Matlab you will need `imread` and `imshow`. Impulse noise can again be generated using `impulse_noise.m`. Make sure the amplitude values of the noisy image do not exceed brightness values of 0 through 255 (8 bit unsigned integer).

Problem 1.3

Apply median filtering to a single harmonic signal $x[n] = \cos(\theta n)$. Try $\theta = 0.01\pi$, 0.1π, 0.15π, 0.25π, and 0.5π with filter orders 3, 11, 51, and 99. What can you infer from these simulations? Plot the corresponding input and output sequences to support you answer. What would you expect for an LTI system in this case?

Problem 1.4

In this problem we estimate the two-dimensional position $\mathbf{p} = [p_x,\ p_y]^T$ of a target using a sensor that measures the range (distance) r and the angle θ (see Fig. 1), collected in the vector $\mathbf{x} = [r,\ \theta]^T$. To obtain the position in Cartesian coordinates, we have to use the nonlinear transformation

$$\begin{aligned} \mathbf{p} &= \mathrm{h}(\mathbf{x}) \\ &= \begin{bmatrix} r \cdot \cos\theta \\ r \cdot \sin\theta \end{bmatrix}. \end{aligned}$$

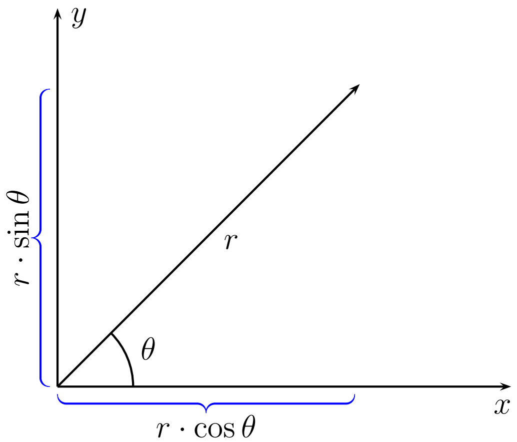

Figure 1: Illustration for problem 1.4

The sensor values are impaired by measurement noise. For the analysis, we know the true values of the sensor values $r_{\text{true}} = 10\,\text{m}$ and $\theta_{\text{true}} = 90\,°$. Consider the following cases:

1. $r \sim \mathcal{N}(r_{\text{true}}, \sigma_r^2)$ with $\sigma_r = 1\,\text{m}$, and θ is uniformly distributed with the true mean and a standard deviation of $10\,°$.

2. $r \sim \mathcal{N}(r_{\text{true}}, \sigma_r^2)$ with $\sigma_r = 0.1\,\text{m}$, and θ is uniformly distributed with the true mean and a standard deviation of $20\,°$.

3. $r \sim \mathcal{N}(r_{\text{true}}, \sigma_r^2)$ with $\sigma_r = 1\,\text{m}$, and θ is Gaussian distributed with the true mean and a standard deviation of $10\,°$.

4. $r \sim \mathcal{N}(r_{\text{true}}, \sigma_r^2)$ with $\sigma_r = 0.1\,\text{m}$, and θ is Gaussian distributed with the true mean and a standard deviation of $20\,°$.

For all cases, what do you *expect* the mean of $\mathbf{p}$, the estimated position, to be?

In Matlab, estimate the mean of $\mathbf{p}$ for these cases using $N = 10^4$ samples of the corresponding sensor values that you pass through the transformation. Compute the mean of the estimated positions and compare them to your expectations. Also, use estimated mean and covariance matrix of the position samples to plot a Gaussian approximation of their distribution using the provided function `plotGaussContour.m`. Describe your observations!