HSB
Hochschule Bremen
City University of Applied Sciences

Stochastic Signals and Systems

Assignments 1 – 6

Prof. Dr.-Ing. Dieter Kraus

IWSS
Institute of Wateracoustics, Sonar Engineering and Signal Theory
Hochschule Bremen

Contents

1 Random variables and random Numbers 5
1.1 Preparation 5
1.1.1 Uniform distribution 5
1.1.2 Normal distribution 5
1.1.3 Approximation of the distribution function 5
1.1.4 Bivariate normal distribution 5
1.1.5 Monte-Carlo-method 5
1.2 Exercises with Matlab 6
1.2.1 Mean value, variance and standard deviation 6
1.2.2 Density function and histogram 6
1.2.3 Distribution function and frequency 6
1.2.4 Generation of a bivariate normal distribution 6
1.2.5 Bivariate normal distribution 7
1.2.6 Monte-Carlo-Method for approximation of π 7
2 Functions of random variables 8
2.1 Preparation 8
2.1.1 Normal distribution and uniform distribution 8
2.1.2 Exponential distribution 8
2.1.3 Sum of random variables 8
2.1.4 Product of random variables 8
2.1.5 χ^2-distribution 8
2.1.6 Normal distribution from uniform distribution 9
2.2 Exercises with MATLAB 10
2.2.1 Standard normal distribution 10
2.2.2 Exponential distribution 10
2.2.3 Sum of random numbers 10
2.2.4 Product of random numbers 10
2.2.5 χ^2-distribution 10
2.2.6 Normal distribution from uniform distribution 11

3 Least-squares estimation 12
3.1 Preparation 12
3.1.1 Polynomial model 12
3.1.2 Power model and exponential model 12
3.1.3 Sine model 12
3.1.4 Nonlinear LS-Approach to a circle 12
3.1.5 Linear LS-Approach to a circle 13
3.2 Exercises with MATLAB 14
3.2.1 Generating data 14
3.2.2 Model assignment 14
3.2.3 LS-Estimation of different models 14
3.2.4 Estimating the order of a polynomial 14
3.2.5 Comparison of observations and estimated model 14
3.2.6 LS-adjustment to a circle: estimating the centre 15
3.2.7 LS-Fit to a circle: iteration 15
3.2.8 LS-Fit to a circle: convergence 15
4 Parameter estimation, AR(p)-processes 16
4.1 Preparation 16
4.1.1 Discrete white noise 16
4.1.2 Generation of AR(p)-processes 16
4.1.3 LS-Estimation 16
4.1.4 Levinson-Durbin-Algorithm 16
4.2 Exercises with MATLAB 17
4.2.1 Sample covariance function 17
4.2.2 Generation of AR(p)−processes 17
4.2.3 LS-Estimation 17
4.2.4 Empiric Yule-Walker-Equation 17
4.2.5 Estimation of the model order 17
5 Discrete Fourier Transform 18
5.1 Preparation 18
5.1.1 FFT in MATLAB 18
5.1.2 Comparison of DFT and FFT 18
5.1.3 Nyquist-Criterion 18

5.1.4 Filling with zeroes 18
5.1.5 Windows 18
5.1.6 Ideal filter 18
5.1.7 Fast convolution 18
5.2 Exercises with MATLAB 19
5.2.1 Image frequencies 19
5.2.2 Aliasing 19
5.2.3 Filling with zeroes 19
5.2.4 Leakage effect 19
5.2.5 Effect of windows 19
5.2.6 Transversal (FIR) low pass filter 20
5.2.7 Fast convolution 21
6 Spectral Analysis 22
6.1 Preparation 22
6.1.1 Representation of the spectrum 22
6.1.2 Periodogram 22
6.1.3 Averaging of Periodograms 22
6.1.4 Trend removal 22
6.1.5 Smoothing of Periodograms 22
6.1.6 Prewhitening 22
6.2 Exercise with Matlab 23
6.2.1 Estimation of spectra 23
6.2.2 Estimation of transfer functions 23
6.2.3 Add Trend to an AR(p)-Process 23
6.2.4 Trend removal 23
6.2.5 Smoothing of Periodograms 23
6.2.6 Prewhitening 24
6.2.7 Spectrum estimation 24

Assignment 1

1 Random variables and random Numbers

1.1 Preparation

1.1.1 Uniform distribution

Specify the density function, distribution function and determine the expected value and variance of a on $[a, b]$ uniformly distributed random variable *X*.

1.1.2 Normal distribution

Specify the density function and the distribution function of a $\mathcal{N}(\mu,\sigma^2)$-distributed random variable *X*. How can the tabulated values of the standard normal distribution function be used to determine values for a $\mathcal{N}(\mu,\sigma^2)$-distributed random variable?

1.1.3 Approximation of the distribution function

What are the advantages in estimating the distribution function instead of estimating the corresponding density function?

1.1.4 Bivariate normal distribution

Specify the density function $f_{XY}(x, y)$ and the characteristic function $\Phi_{XY}(x, y)$ of two bivariate normal distributed random variables X and Y with the expected values μ_X, μ_Y, the variances σ_X^2, σ_Y^2 and the correlation coefficient ρ. Outline the level curves of the density function $\{(x, y): f_{XY}(x, y) = \text{const.}\}$ for $\rho = 0$ and $\rho = 1/2$. To what shape degenerate the level curves in case of $\rho = \pm 1$?

1.1.5 Monte-Carlo-method

Develop a method for estimating the constant π using realizations of a random variable uniformly distributed on the interval [0, 1].

(Note: Consider pairs of these random numbers like coordinates of random points in the plane. How large is approximately the relative frequency that a point is lying inside the unit circle?)

1.2 Exercises with Matlab

1.2.1 Mean value, variance and standard deviation

Generate 1000 on the interval [0, 1] uniformly distributed random numbers. Display the numbers by using the MATLAB-instruction `plot`. Before generating the random numbers, the initial seed has to be set to 0 via the MATLAB function `rng`. Save the vector of random numbers in `dat1_1`. Calculate the sample mean, sample variance and sample standard deviation using the random sequence in three different ways.

Using the MATLAB

1) loop `for` ... `end`,
2) functions `sum` and `length` and
3) functions `mean` and `cov`.

Compare the results. Take a look at the m-files `mean` and `cov` using `help` and/or `type`.

1.2.2 Density function and histogram

Write a function `density`, that estimates the density function of a random variable from a number of observations by applying the histogram function `hist`. Use the uniformly distributed random sequence from `dat1_1`. Show the estimates as bar graph and as line graph. Compare the theoretical density function with the result by drawing the theoretical density also into the line graph figure.

Repeat the experiment for 1000 standard normal distributed random numbers, where the initial value has to be set to 0. Save this random vector in `dat1_2`.

(Note: The function `density` shall be applicable to any random sequences. Therefore, take into account the interrelationship between the scaling and the length of the random sequence and the width of the interval.)

1.2.3 Distribution function and frequency

Estimate the distribution function of the random sequences given in `dat1_1` and `dat1_2`. For this, generate with the MATLAB instruction `linspace` a ramp within the interval [0, 1] and use it for drawing the estimate.

(Note: Sort the data.)

1.2.4 Generation of a bivariate normal distribution

Set the initial seed of `randn` to 0. Generate through `z1=randn(1000,2)` a matrix of normal distributed random numbers. Multiply `z1` from the right with the matrix `D = [1 .5; 0 .5]`, thus `z2 = z1 * D`. Separate the result through `x = z2( : ,1)` and `y = z2( : ,2)` into two vectors and add 1.5 to each element of `x` and 0.5 to each element of `y`. Store the vectors by using the MATLAB instruction `save dat1_3 x y`.

1.2.5 Bivariate normal distribution

Load `dat1_3` that includes the two vectors `x` and `y` of length 1000 as samples of the two bivariate normally distributed random variables X and Y. Estimate for this dataset $\mu_X, \mu_Y, \sigma_X^2, \sigma_Y^2$ and ρ by using the MATLAB function `cov`. Calculate the bivariate density function $f_{XY}(x, y)$ with the estimated parameters and display it as a 3dimensional and a level curve graphic.

(Note: Use the MATLAB-functions `mesh` and `contour`.)

1.2.6 Monte-Carlo-Method for approximation of π

Generate two on the interval [0, 1] uniformly distributed random sequences `x` and `y` of length 1000. The initial value has to be set to 0 again. How many points lie within the unit circle? Estimate thereby the constant π.

Assignment 2

2 Functions of random variables

2.1 Preparation

2.1.1 Normal distribution and uniform distribution

Transform the $N(\mu,\sigma^2)$−distributed random variable X into a standard normal distributed random variable. Transform the on $[a,b]$ uniformly distributed random variable X such that it is uniformly distributed on $[-1/2,1/2]$.

2.1.2 Exponential distribution

Determine the expected value, the variance and the density function of $Y=-\frac{1}{\alpha}\ln X$ for $\alpha>0$, when X is uniformly distributed on [0, 1]. Which other distribution has the same density function, as the exponential distribution for $\alpha=1/2$?

2.1.3 Sum of random variables

Determine the expected value, the variance and the density function of $Z=X+Y$, when X and Y are two stochastically independent and on [0, 1] identically uniformly distributed random variables.

2.1.4 Product of random variables

Calculate the expected value, the variance and the density function of $Z=XY$, when X and Y are two stochastically independent and on [0, 1] identically uniformly distributed random variables.

2.1.5 χ^2-distribution

Determine the expected value, the variance and the density of

$$Z=\sum_{i=1}^{4} X_i^2,$$

where $X_i (i=1,2,3,4)$ are four stochastically independent, standard normally distributed random variables. How could one create a random variable from two stochastically independent exponentially distributed random variables that possess the same distribution as Z? Calculate the density of Z through convolution of the density of an exponential distribution.

2.1.6 Normal distribution from uniform distribution

Calculate the bivariate density $f_{Y_1Y_2}(y_1, y_2)$ of

$$Y_1 = \sqrt{-2\ln X_1}\,\sin(2\pi X_2) \quad \text{and} \quad Y_2 = \sqrt{-2\ln X_1}\,\cos(2\pi X_2)$$

where X_1 and X_2 are stochastically independent and on [0, 1] identically uniformly distributed random variables.

2.2 Exercises with MATLAB

2.2.1 Standard normal distribution

Load `dat1_3` containing the random sequences **x** and **y** of two bivariate normal distributed random variables. Transform the random samples such that they obey a standard normal distribution. Make sure that your transformation is correct by showing the theoretical and the via `density` estimated density of the transformed random sequence **x** in a diagram.

2.2.2 Exponential distribution

Load the on [0, 1] uniformly distributed random sequence from `dat1_1` and transform it as explained in Section 2.1.2 with $\alpha = 1/2$. Calculate the sample mean and sample variance of the transformed sequence and compare the results with the theoretical values from Section 2.1.2. Show the in Section 2.1.2 calculated and via `density` estimated density function of the transformed random sequence in a figure.

2.2.3 Sum of random numbers

Generate two on [0, 1] uniformly distributed random sequences of the length 1000 after setting the initial seed to 0. Save both random sequences in `dat2_1`. Then add the two sequences element-by-element. Calculate the sample mean and sample variance of the sum and compare the results with the theoretical values from Section 2.1.3. Display the in Section 2.1.3 calculated and via `density` estimated density function in a diagram.

2.2.4 Product of random numbers

Multiply both on [0, 1] uniformly distributed random sequences from `dat2_1` element-by-element. Calculate the sample mean and sample variance of the product and compare the results with the theoretical values derived in Section 2.1.4. Depict the in Section 2.1.4 calculated and via `density` estimated density function in a figure.

2.2.5 χ^2-distribution

Generate four standard normal distributed random sequences of the length 1000, where before the initial value has to be set to 0. Save the data in `dat2_2`. Determine the sum of squares of the four random sequences element-by-element. Calculate the sample mean and sample variance of the sum and compare the results with the theoretical values obtained in Section 2.1.5. Present the in Section 2.1.5 calculated and via `density` estimated density function in a diagram.

2.2.6 Normal distribution from uniform distribution

Now use again `dat2_1` containing two on [0, 1] uniformly distributed random sequences. Build $\mathbf{y}_1$ as explained in Section 2.1.6. Calculate the sample mean and sample variance of $\mathbf{y}_1$ and compare the results with the theoretical values determined in Section 2.1.6. Show the in Section 2.1.6 calculated and via `density` estimated density function in a figure.

Assignment 3

3 Least-squares estimation

3.1 Preparation

3.1.1 Polynomial model

The signal model is given by

$$Y_i = a_0 + a_1 x_i + a_2 x_i^2 + \ldots + a_p x_i^p + Z_i \, .$$

Let the observations (x_i, y_i) $i = 1,2,\ldots,N$ be given. Derive the LS-Estimate for the unknown parameters a_i $i = 0,1,\ldots,p$ and the estimate for the variance of the noise. How large has the number N of observations to be at least? Express with regard to a MATLAB implementation the problem in vector notation.

3.1.2 Power model and exponential model

The power model is given by

$$Y_i = a \cdot x_i^b \cdot Z_i \, .$$

Linearize the model assuming $a > 0$, $x_i > 0$ and $Z_i > 0$. Why are the measuring errors assumed to be multiplicative? Estimate the parameters a and b as well as the variance of the measurement error after linearization. How large has to be the number N of observations at least? The exponential model is defined by

$$Y_i = a \cdot b^{x_i} \cdot Z_i \, .$$

Carry out all the tasks mentioned above also for this model.

3.1.3 Sine model

The amplitude a and phase b of the sine model

$$Y_i = a \sin(x_i + b) + Z_i$$

are unknown. Linearize the sine model. Estimate the parameters a and b as well as the variance of the measurement error after linearization. How large has to be the number N of observations at least?

3.1.4 Nonlinear LS-Approach to a circle

A number of data points (x_i, y_i) $i = 1,2,\ldots,N$ are given in the xy-plane. Fit to them a circle with unknown radius r and unknown centre (x_0, y_0). The model for the observation is given by

$$\sqrt{(x_i - x_0)^2 + (y_i - y_0)^2} = r_i = r + z_i,$$

where r_i indicates the distance between the i-th measurement point (x_i, y_i) and the centre (x_0, y_0) and z_i indicates the error of the radius. Express the sum of squares of measurement errors $q(x_0, y_0, r)$ and derive the necessary condition for their minimization. Is the resulting equation system solvable in an analytic way? Give reasons for your answer. Point out a method for solving the equation system numerically.

3.1.5 Linear LS-Approach to a circle

Supposed the centre $(\tilde{x}_0, \tilde{y}_0)$ of the circle from Section 3.1.4 is approximately known. Then it is possible to develop the function $r_i(x_0, y_0)$ into a Taylor series in place $(\tilde{x}_0, \tilde{y}_0)$. The approximation

$$\begin{aligned} r_i(x_0, y_0) &= \sqrt{(x_i - x_0)^2 + (y_i - y_0)^2} \\ &\approx r_i(\tilde{x}_0, \tilde{y}_0) + \left.\frac{\partial r_i(x_0, y_0)}{\partial x_0}\right|_{(\tilde{x}_0, \tilde{y}_0)} (x_0 - \tilde{x}_0) + \left.\frac{\partial r_i(x_0, y_0)}{\partial y_0}\right|_{(\tilde{x}_0, \tilde{y}_0)} (y_0 - \tilde{y}_0) \\ &\approx r_i(\tilde{x}_0, \tilde{y}_0) - \frac{x_i - \tilde{x}_0}{r_i(\tilde{x}_0, \tilde{y}_0)}(x_0 - \tilde{x}_0) - \frac{y_i - \tilde{y}_0}{r_i(\tilde{x}_0, \tilde{y}_0)}(y_0 - \tilde{y}_0) \end{aligned}$$

provides a new model for the error

$$z_i = r_i - r = r_i(\tilde{x}_0, \tilde{y}_0) - \frac{x_i - \tilde{x}_0}{r_i(\tilde{x}_0, \tilde{y}_0)}(x_0 - \tilde{x}_0) - \frac{y_i - \tilde{y}_0}{r_i(\tilde{x}_0, \tilde{y}_0)}(y_0 - \tilde{y}_0) - r$$

that is now linear in x_0 and y_0.

Formulate the minimization criterion for the linearized model. Determine the LS-Estimates $(\hat{x}_0, \hat{y}_0)$ and $\hat{r}$ from (x_i, y_i) $i = 1,2,\ldots,N$. The LS-Estimates obtained are only as good as the initial guess of the centre $(\tilde{x}_0, \tilde{y}_0)$. One can improve the estimates if the procedure mentioned above is repeated after replacing $(\tilde{x}_0, \tilde{y}_0)$ by $(\hat{x}_0, \hat{y}_0)$. Furthermore, state two different stopping criteria for this iterative procedure.

3.2 Exercises with MATLAB

3.2.1 Generating data

Now data for four different signal models have to be created. Generate therefore an on [0, 1] uniformly distributed random sequence `x1` and a standard normal distributed random sequence `z1`, each with a length of 100 values. Set in both cases first the initial seed to 0. Type now these commands in MATLAB in the following order:

```
x=x1*5+2;  z=z1*sqrt(0.004); xy1=[x  exp(1+x*0.6+z)];
x=x1*4*pi; z=z1*sqrt(0.05);  xy2=[x  2*sin(x+1)+z];
x=x1*5;    z=z1;             xy3=[x  -0.6*x.^3+0.9*x.^2+3*x+4.5+z];
x=x1*5;    z=z1*sqrt(0.004); xy4=[x  exp(0.3+log(x)*0.5+z)];
x=x1*2*pi; z=z1*0.5+6;       xy=[z.*cos(x)+4  z.*sin(x)+2];
save  dat3_1  xy1  xy2  xy3  xy4
save  dat3_2  xy
```

3.2.2 Model assignment

Load `dat3_1` that includes the four 100 × 2 matrices `xy1`, `xy2`, `xy3` and `xy4`. The two column vectors of each matrix correspond to the observations x_i and y_i of a particular signal model. Show each dataset x_i, y_i in a diagram and assign to each matrix a model.

(Note: For the model assignment you can exploit the fact that $\log g(x)$ behaves in case of the power model like a logarithmic function while $\log g(x)$ depends in case of the exponential model only linearly on x.)

3.2.3 LS-Estimation of different models

Estimate each of the parameters a, b and the variance of the measurement errors σ_Z^2 of the linearized models, i.e., of the exponential model, the power model and the sine model. Therefore, write a function `LSE` that is able to deal with these three models and with a polynomial model of arbitrary order.

3.2.4 Estimating the order of a polynomial

Estimate the order p of the polynomial model. Therefore, you have to depict the estimated variance versus the model order $p = 1,2,\ldots,10$. Consider the order that provides the smallest variance as the correct order. Estimate for that order the parameters a_i for $i = 1,2,\ldots,p$.

3.2.5 Comparison of observations and estimated model

For each of the models estimated above show the observations of the model (x_i, y_i) $i = 1,2,\ldots,N$ and the corresponding reconstructed model curve $(x_i, g(x_i))$ $i = 1,2,\ldots,N$ in one figure.

3.2.6 LS-adjustment to a circle: estimating the centre

Load `dat3_2` containing a 100×2 matrix that represent the coordinates of 100 measurement points in the *xy*-plane. Take a look at the measurement points and guess from them the coordinates of the centre location.

3.2.7 LS-Fit to a circle: iteration

Carry out a LS-Estimation as shown in Section 3.1.5. Use the estimated centre as an improved initial value and repeat the LS-Estimation as long as $\left\|(\tilde{x}_0(k), \tilde{y}_0(k)) - (\tilde{x}_0(k-1), \tilde{y}_0(k-1))\right\| < 10^{-10}$ is fulfilled, where $\left(\tilde{x}_0(k), \tilde{y}_0(k)\right)$ denotes the *k*-th LS-Estimate of the centre. Write a function `LSE_circle` that carries out the iteration. Depict the reconstructed circle and the measurement points in a diagram.

3.2.8 LS-Fit to a circle: convergence

Does this method converge, if the initial guess of the circle centre is very poor? Try it by giving your function `LSE circle` intentionally a very poor initial estimate of the centre location. Explain your observation.

Assignment 4

4 Parameter estimation, AR(p)-processes

4.1 Preparation

4.1.1 Discrete white noise

Specify the constant component, the covariance function and the spectral density of discrete white noise.

4.1.2 Generation of AR(p)-processes

Make oneself familiar with the MATLAB-function `filter`. The function `filter` is used for the generation of an AR(p)-process. Which values have to be entered for the filter coefficient vector **b**? Which value has to be assigned to the first element a_0 of the filter coefficient vector **a**?

4.1.3 LS-Estimation

How can one determine the least squares estimates of the parameters $a_1, a_2, ..., a_p; \sigma_Z^2$ for the given observations x_i $i = 1,2,\ldots,N$? How large has to be the number of observations N?

4.1.4 Levinson-Durbin-Algorithm

Which particular property of the coefficient matrix of the following Yule-Walker-Equation system

$$\begin{pmatrix} c_{XX}(0) & c_{XX}(1) & \ldots & c_{XX}(p-1) \\ c_{XX}(1) & c_{XX}(0) & \cdots & c_{XX}(p-2) \\ \vdots & \vdots & \ddots & \vdots \\ c_{XX}(p-1) & c_{XX}(p-2) & \cdots & c_{XX}(0) \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_p \end{pmatrix} = - \begin{pmatrix} c_{XX}(1) \\ c_{XX}(2) \\ \vdots \\ c_{XX}(p) \end{pmatrix}$$

permits the application of the Levinson-Durbin-Algorithm? Is the Levinson-Durbin-Algorithm be suited for solving the equation system of the LS-Estimation procedure? Give reasons for your answer. Which advantage offers the Levinson-Durbin-Algorithm with respect to the model order estimation? How can the model order be estimated?

4.2 Exercises with MATLAB

4.2.1 Sample covariance function

Write a function `covfct` for determining the sample and the modified sample covariance function. To calculate both sample covariance functions take the first 200 random numbers from `dat1_1`. Display the results and explain the differences between the functions. What indicates that a sequence of random numbers can be interpreted as a realisation of discrete white noise?

(Note: In MATLAB exists a similar function `xcov`. Ascertain the correctness of your function `covfct` by comparing the results of the experiments with `covfct` and `xcov`.)

4.2.2 Generation of AR(p)−processes

Load `dat1_2` that includes a realization of white noise. For the generation of an AR(p)-process you have to filter the white noise by a recursive filter with the filter coefficients $a_1 = 0.5$, $a_2 = 0.3$, $a_3 = 0.1$, $a_4 = 0.7$, $a_5 = 0.3$. Use the MATLAB-function `filter`. Save the white noise and the AR(p)-process for later use in `dat4_1`.

4.2.3 LS-Estimation

Carry out a LS-Estimation of the parameters $a_i \;\; i = 1,2,...,5$ and the variance σ_Z^2 of the AR(p)-process.

4.2.4 Empiric Yule-Walker-Equation

Determine the first 11 values of the sample covariance function, e.g. $\hat{c}_{XX}(0),\ldots,\hat{c}_{XX}(10)$, using function `covfct`. Estimate the parameters $a_i \;\; i = 1,2,...,5$ and σ_Z^2 by solving the empirical Yule-Walker-Equation via the

a) Gaussian Elimination Algorithm
b) Levison-Durbin Algorithm.

(Note: Use the MATLAB-function `toeplitz` to generate the coefficient matrix and read for a) the MATLAB-help for the operator `backslash` and for b) use the MATLAB-function `levinson`)

4.2.5 Estimation of the model order

Estimate the parameters $a_i \;\; i = 1,2,...,5$ and σ_Z^2 as in Section 4.2.4 b), but now for the wrong model order $p = 4$ and $p = 6$. Compare the estimated parameters with those obtained in Section 4.2.4 b). Which consequences have an underestimation or an overestimation of the model order? Estimate now for the model order $k = 1,2,...,10$ the variance $\sigma_{Z,k}^2$ with the Levinson-Durbin Algorithm. Present the result in a diagram. Draw also the values of the Akaike and Rissanen criterion and deduce from them the model order p of the AR(p)-process.

Assignment 5

5 Discrete Fourier Transform

5.1 Preparation

5.1.1 FFT in MATLAB

Make oneself familiar with the MATLAB function `fft`. Why do you get complex valued results in spite of real input values?

5.1.2 Comparison of DFT and FFT

What is the difference between DFT and FFT? Do you get different results by using both Fourier Transforms?

5.1.3 Nyquist-Criterion

Suppose we are sampling a sinusoidal signal of frequency f with the sampling frequencies $f_s = f$, $f_s = 2f$ and $f_s = 3f$. What signals do you get?

5.1.4 Filling with zeroes

Can you improve the resolving power of the Fourier Transform for a given data length by zero padding? Motivate your answer. Which effect does zero padding have? Is it important where you insert zeroes?

5.1.5 Windows

The MATLAB-functions `hanning`, `hamming`, `blackman` and `bartlett` should be used for the implementation of windows. Make oneself familiar with their functionality. Which properties of the Fourier Transforms of windows are particularly important for applications?

5.1.6 Ideal filter

Which properties does an ideal low pass filter possess? Which impact does the choice of windows have on the design of a transversal (FIR) low pass filter?

5.1.7 Fast convolution

The fast convolution is implemented in MATLAB by the function `fftfilt` which is based on the Overlap-Add-Method. Make oneself familiar with their functionality. The correct data length of Fourier Transforms is important in the use of the fast convolution. What happens, if one does not pay attention to it?

5.2 Exercises with MATLAB

5.2.1 Image frequencies

A sinusoidal signal $x(t) = \sin(2\pi f_0 t)$ with the frequency $f_0 = 2$ kHz is given. Sample 512 values of the signal with the sampling frequencies´ $f_s = 10$ kHz and 3 kHz. Represent the squared magnitude of the Fourier Transform. Explain the phenomenon of image frequencies.

(Note: The MATLAB-function `fft` can be used to calculate the DFT for this and all the following exercises.)

5.2.2 Aliasing

A rectangular signal of duration $T = 1$ ms is given, i.e.

$$x(t) = \begin{cases} 1 & 0 \le t \le T \\ 0 & \text{else} \end{cases}$$

Sample 512 values of the signal with the sampling frequencies $f_s = 4$ kHz and 16 kHz. Represent the squared magnitude of the Fourier Transform. Explain the impact of the sampling frequency on the aliasing effect by using the results obtained.

5.2.3 Filling with zeroes

Take the first 32 samples for $f_s = 10$ kHz from Section 5.2.1 and represent the squared magnitude of the Fourier Transform. Fill your data set with 480 zeroes and repeat the experiment now. Which effect does the zero padding of the data set have? Compare the result with that of Section 5.2.1. What can you say about the impact of the correct data length (without filled zeroes) on the resolving power of the Fourier Transform?

5.2.4 Leakage effect

Use the first 33 values for $f_s = 10$ kHz from Section 5.2.1 and represent the squared magnitude of the Fourier Transform. How does the leakage effect make itself noticeable? Execute the Fourier Transform with another number of samples once more. When does the leakage effect make itself noticeable?

5.2.5 Effect of windows

Represent the following windows

Rectangle-window:

$$w_n = 1, \qquad 0 \le n \le N-1$$

Bartlett-window:

$$w_n = \begin{cases} \frac{2n}{N-1}, & 0 \le n \le \frac{N-1}{2} \\ 2 - \frac{2n}{N-1}, & \frac{N-1}{2} < n \le N-1 \end{cases}$$

Hann-window

$$w_n = \frac{1}{2}\left[1 - \cos\left(\frac{2\pi(n-1)}{N+1}\right)\right], \qquad 0 \le n \le N-1$$

Hamming-window

$$w_n = 0,54 - 0,46\cos\left(\frac{2\pi n}{N-1}\right), \qquad 0 \le n \le N-1$$

Blackman-window

$$w_n = 0,42 - \frac{1}{2}\cos\left(\frac{2\pi n}{N-1}\right) + 0,08\cos\left(\frac{4\pi n}{N-1}\right), \qquad 0 \le n \le N-1$$

in the time domain for $n = 1,\ldots,51$. Use the relevant MATLAB function.

Calculate the logarithmic amplitude responses $20\log_{10}\left(|W(\Omega)|/\max|W(\Omega)|\right)$ using the FFT of length $L = 1024$ and depict them in diagrams. Determine the frequencies of the −3dB limits and the suppression of the highest sidelobe of the amplitude response. Discuss the results.

(Note: Depict always the time function and the corresponding amplitude response in one figure. For this reason, use the MATLAB function `subplot`.)

5.2.6 Transversal (FIR) low pass filter

The impulse response of an ideal low pass filter is given by

$$h_n = \frac{1}{2\pi}\int_{-\pi}^{\pi} H(\Omega)\, e^{j\Omega n} d\Omega = \frac{1}{2\pi}\int_{-\Omega_c}^{\Omega_c} e^{-j\Omega\alpha} e^{j\Omega n} d\Omega = \frac{\Omega_c}{\pi}\,\mathrm{si}\left(\Omega_c(n-\alpha)\right),$$

where Ω_c denotes the cutoff frequency. Because only a finite number of filter-coefficient can be processed in a computer, the impulse response is truncated to the finite length N. For $\alpha = (N-1)/2$ we get a causal low pass filter with linear phase and the property $\tilde{h}_n = \tilde{h}_{N-n-1}$.

Use the first 51 values of the impulse response with the cutoff frequency $\Omega_c = 1$ as filter coefficient. Represent every tapered impulse response $\tilde{h}_n = w_n \cdot h_n$ and its amplitude response (in dB) in a common figure. Determine the suppression of the highest side lobe and the bandwidth of the low pass filter at −3 dB und −20 dB level. Discuss the results.

(Note: A scaled version of function $\mathrm{si}(x) = (\sin x)/x$ is called `sinc` in MATLAB. You should see the MATLAB-help to `sinc` before you use it.)

5.2.7 Fast convolution

Write a function `firfilt` for a direct implementation of a transversal (FIR) filter. Create the rectangular signals

$$h_n = \begin{cases} 1 & \text{if } n = 0,1,\ldots,100 \\ 0 & \text{elsewhere} \end{cases}$$

and

$$x_n = \begin{cases} 1 & \text{if } n = 0,1,\ldots,900 \\ 0 & \text{elsewhere} \end{cases}.$$

Execute the filtering both directly with its function `firfilt` and with help of MATLAB function `fftfilt`, which implements the overlap-add-method. For that, an FFT-length of 256 for `fft-filt` should be used. Compare the results and the CPU-Processing time for the arithmetic operations of both approaches.

Assignment 6

6 Spectral Analysis

6.1 Preparation

6.1.1 Representation of the spectrum

Familiarise oneself with the MATLAB-functions `freqz` and `periodogram`.

6.1.2 Periodogram

Why does one use the Periodogram and not the Fourier-transform of a signal for spectrum estimation? Is the Periodogram itself suited to provide a reasonable estimate of the spectrum?

6.1.3 Averaging of Periodograms

Which are the advantages and disadvantages of the averaging of Periodograms? Why is one allowed to average Periodograms of successive data pieces of observations?

6.1.4 Trend removal

Why does a trend in the data disturb the spectrum analysis? How does a constant or linear trend affect a direct estimation of a spectrum? Discuss in that context the Periodogram of the sum of a trend and a noise signal without a constant component. Does a constant trend have an effect at discrete frequency locations?

6.1.5 Smoothing of Periodograms

For which spectra is the smoothing of Periodograms not suitable? What are the advantages and disadvantages of smoothing compared to averaging periodograms?

6.1.6 Prewhitening

Which advantages does prewhitening offer? Why does one not compute the spectrum directly from the estimated parameters of an AR(p) process?

6.2 Exercise with Matlab

6.2.1 Estimation of spectra

Write a function `spec`, which estimates the spectrum from any data record by averaging of Periodograms. The number of data pieces L and the window type employed should be freely selectable. Estimate the spectrum of the AR(p) process from Section 4.2.2 for $L = 1,2,5,10$ using the rectangular window in each case. Represent each time the spectrum and its estimate in a common figure. Repeat the investigations using the Hann-window and discuss the results.

6.2.2 Estimation of transfer functions

Write a function `Hw` which estimates the spectrum of a stochastic signal, the cross spectrum of two stochastic signals and from this the transfer function of the underlying system by means of averaging of Periodograms. Use as input and output signal the realizations of the white noise and the AR(p) process from Section 4.2.2, respectively. Estimate the transfer function of the recursive filter by means of the function `Hw` for $L = 5$ using the Hann-window. Represent the estimated and theoretical amplitude response in one figure and discuss the result.

(Note: The theoretical amplitude response can be easily determined employing the MATLAB function `freqz`.)

6.2.3 Add Trend to an AR(p)-Process

Load the data record `dat4_1`, which contains the realizations of the AR(p) process generated in Section 4.2.2. Add a trend to the data by means of `x = x + linspace(0,3,1000)'` and store the resulting vector in `dat6_1`.

6.2.4 Trend removal

Load the data record `dat6_1` containing a vector $\mathbf{x}$ of length 1000. Represent the vector as a graph in a figure. One recognizes that the data possess a linear trend. Determine this trend by means of least squares estimation. Remove the trend from the data and store the data in `dat6_2`.

6.2.5 Smoothing of Periodograms

Extend your function `spec` from Section 6.2.1 in such a way that now also a smoothing of Periodograms is possible over a freely selectable number of $2m + 1$ frequency bins. Make sure that the averaging at the edges of the frequency domain only utilizes the existing Periodogram values and that the Periodogram values at the frequencies $\Omega = 0$ and $\Omega = \pi$ are also not taken into account. Depict the smoothed Periodograms of the trend removed data vector given in `dat6_2` for all frequencies

$$\Omega_n = \frac{2\pi n}{N}, \quad n = 1, \ldots, \left\lfloor \frac{N-1}{2} \right\rfloor$$

and $m = 0,2,5,10$ in a diagram and discuss the results.

6.2.6 Prewhitening

In order to avoid a smoothing over spectral peaks, a prewhitening of the trend removed process stored in `dat6_2` has to be carried out first. Therefore, estimate the parameters $a_i :\; i = 1, 2, \ldots, p$ and the order p of the AR(p) process using the Levinson-Durbin algorithm and the MDL criterion respectively. With these estimates the prewhitened process $\hat{z}_n$ can be determined by

$$\hat{z}_n = x_n + \sum_{k=1}^{\hat{p}} \hat{a}_k x_{n-k} \,.$$

Represent the corresponding data vector $\hat{\mathbf{z}}$ in a figure and save it in `dat6_3`.

6.2.7 Spectrum estimation

Compute the smoothed Periodogram of $\hat{z}_n$ stored in `dat6_3` for $m = 10$ and depict the result in a diagram. Now, the spectrum of the AR(p) process shall be determined using the amplitude response of the underlying recursive filter and the smoothed Periodogram of $\hat{z}_n$. Finally, display the squared amplitude response of the recursive filter and the spectrum $\hat{C}_{XX}(\Omega)$ in a figure, and compare the latter with the results from Section 6.2.5.