STOCHASTIC SIGNALS AND SYSTEMS

WS 2018

ASSIGNMENT 3

LEAST - SQUARES ESTIMATION

Date: 08/01/2019

by

Shinde Mrinal Vinayak (Matriculation No.: 5021349)
Yama Sanath Kumar(Matriculation No.: 5021387)

guided by

Prof. Dr. -Ing. Dieter Kraus

Contents

List of Figures

Chapter 1

Functions of random variables

1.1 Normal distribution and uniform distribution

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

Solution:

Description

The standard normal distribution is a special case of the normal distribution. It is the distribution that occurs when a normal random variable has a mean, (μ) of zero and a standard deviation, (σ)of one. The normal random variable of a standard normal distribution is called a standard score. Every normal random variable X can be transformed into a standard normal distribution via the following equation:

$$Y = \frac{X - \mu}{\sigma}$$

Where X is a normal random variable, μ is the mean of X, and σ is the standard deviation of X.

To transform a normal distribution of $X \sim N(\mu, \sigma)$ into into a standard normal distribution, $Y \sim N(0, 1)$ must be considered. The probability density function of normal distribution can be given as,

$$f_x(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{(x-\mu)^2}{2\sigma^2}\right\}, \quad x \in R$$

The working of the transformation is possible when, $y = g(x) = \frac{x-\mu}{\sigma}$. When, the mean ($\mu$) is 0, $y = g(x = \frac{x}{\sigma})$. Using the inverse property, it can be written as,

$$x = g^{-1}(y) = \sigma x + \mu$$

The probability density function of Y can be given as,

$$f_y(y) = \frac{f_x(x)}{|\frac{dF_Y}{dy}|} \;\; at \;\; x = g^{-1}(y)$$

The probability density function of X after inserting $x = g^{-1}(y)$ can be given as,

$$f_x(x) = \frac{\frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{(\sigma y + \mu - \mu)^2}{2\sigma^2}\right\}}{\frac{1}{\sigma}}$$

$$f_x(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{y^2}{2}\right\}, \quad y \in R$$

Taking the derivative of the function *g(x)*,

$$\frac{\partial g(x)}{\partial x} = \frac{\partial}{\partial x}(\frac{x}{\sigma} - \frac{\mu}{\sigma}) = \frac{1}{\sigma}$$

Substituting this in the probability density function of Y, we get

$$f_y(y) = \frac{1}{\sqrt{2\pi}}\exp\left\{-\frac{y^2}{2}\right\}, \quad y \in R$$

This is the transformed form of normal distribution variable to standard normal distribution variable.

Transformation of uniformly distributed random variable

Consider the random variable $X \sim R[a, b]$ which is a rectangular distribution between interval a and b. The maximum value of this function is $1/(b-a)$. Our aim is to obtain the desired random variable, $Y \sim R[-1/2, 1/2]$. In order to achieve this, we create another random variable $\tilde{X} = X - a$.

The corresponding density function has changed from rectangular function with limits $[a, b]$ to limits $[0, b-a]$. The height of both X and $\tilde{X}$ will be the same, i.e, $1/(b-a)$. Extending this further and taking another random variable, $\hat{X} = (X-a)/(b-a) = \tilde{X}/(b-a)$. Thus for this random variable, the graph envelops everything inside the rectangular region 0 and 1, with the maximum value of $\tilde{X}/(b-a)$.

Now, this random variable will help us in generating our desired random variable, $Y \sim R[-1/2, 1/2]$ For our desired variable, we consider

$$Y = \hat{X} - (1/2)$$

$$Y = ((X-a)/(b-a)) - 1/2$$

Hence we obtain the desired uniformly distributed random variable, i.e., if X is a random variable

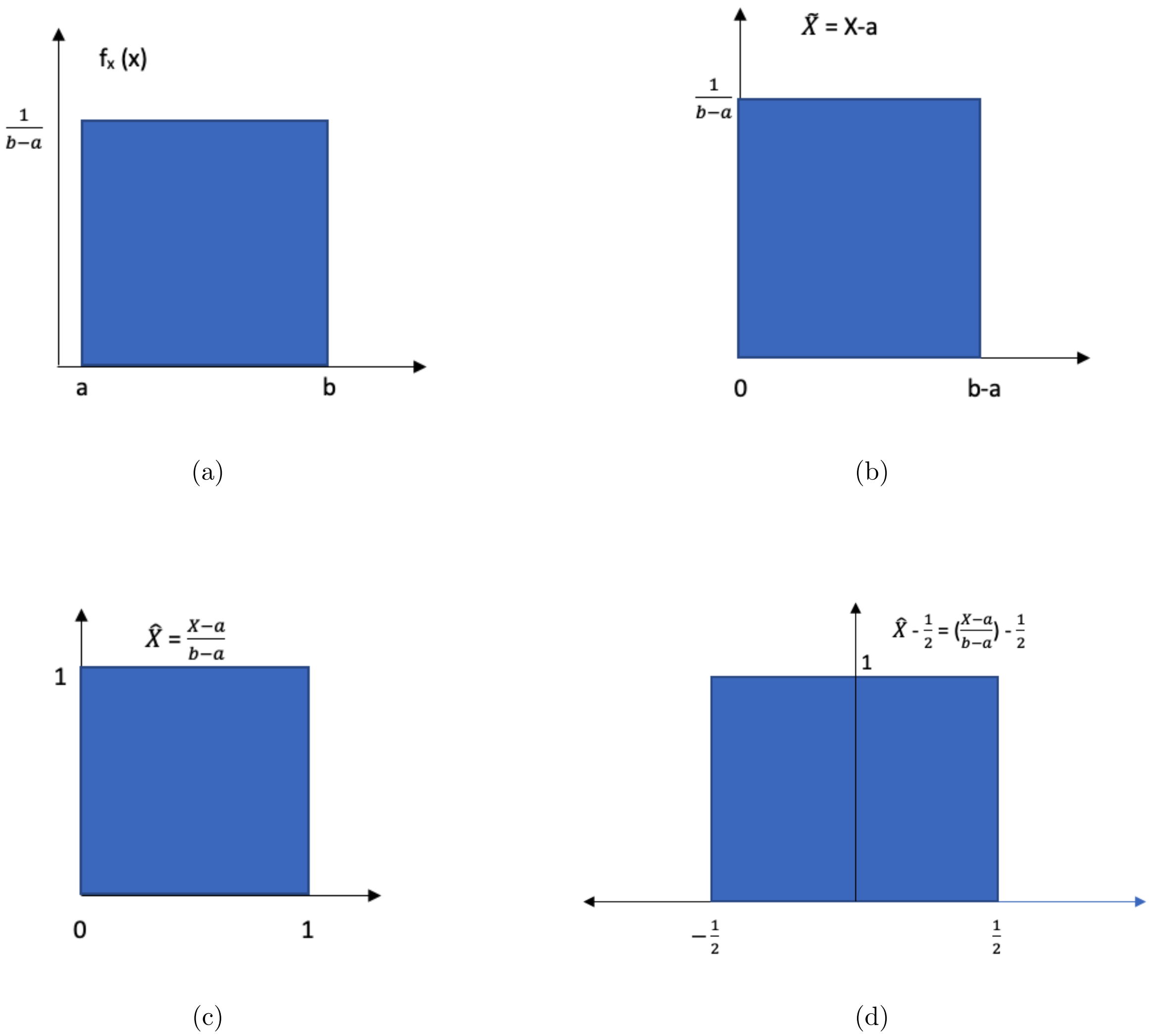


Figure 1.1: Transformation of uniformly distributed random variable

in [a, b], the Y is also a rectangular distribution in $[-1/2, 1/2]$. In practical this type of variable helps us in modelizing the quantization error, which jumps from one point to another over a region.

1.2 Exponential distribution

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

Solution:

The distribution function $F_Y(y)$ defined by the probability of the random variable, $Y = -\frac{1}{\alpha}\ln(X)$ for $\alpha > 0$ and X is uniformly distributed on [0, 1] can be given as,

$$F_Y(y) = P(Y \leq y)$$

$$F_Y(y) = P(\frac{-1}{\alpha}\ln X \leq y) = P(e^{-\alpha y} \leq X)$$

$$F_Y(y) = 1 - e^{-\alpha y}$$

The relationship between the density and the distributed function in the used, i.e.,

$$f_Y = \frac{\partial F_Y}{\partial y}$$

Where, f_y is the probability density function and F_y is the distribution function of Y. Therefore,

$$F_Y = [1 - e^{-\alpha y}].1_{[0,1]}(y)$$

The derivative will be,

$$\frac{\partial F_Y}{\partial y} = \alpha e^{-\alpha y}.1_{[0,1]}(y)$$

Now, the probability density function is given as,

$$f_y(y) = \alpha e^{-\alpha y}$$

The expected value (mean):

$$E(Y) = \int_{-\infty}^{\infty} y f_Y(y) dy$$

Now substituting for $f_Y y$,

$$E(Y) = \int_{0}^{\infty} y\alpha e^{-\alpha y} dy$$

$$E(Y) = \alpha \left[-\frac{1}{\alpha} y e^{-\alpha y} \Big|_0^\infty - \int_0^\infty -\frac{1}{\alpha} e^{-\alpha y} dy \right]$$

$$E(Y) = -\frac{1}{\alpha} e^{-\alpha y} \Big|_0^\infty = \frac{1}{\alpha}$$

The variance :

$${\sigma_Y}^2 = E(Y - E(Y))^2 = E(Y^2) - E(Y)^2$$

$E(Y^2)$ can be calculated as,

$$E(Y^2) = \int_{-\infty}^{\infty} y^2 f_y(Y) dy = \int_0^\infty y^2 \alpha e^{-\alpha y} dy$$

$$E(Y^2) = \alpha \left[-\frac{1}{\alpha} y e^{-\alpha y} \Big|_0^\infty - \int_0^\infty -\frac{2}{\alpha} y e^{-\alpha y} dy \right]$$

$$E(Y^2) = 2 \int_0^\infty y e^{-\alpha y} dy = \frac{2}{\alpha^2}$$

$${\sigma_Y}^2 = \frac{2}{\alpha^2} - \frac{1}{\alpha^2} = \frac{1}{\alpha^2}$$

The other distribution function:

If we Suppose $X \sim X_n^2$ then the density function is given as

$$f_X(x) = \frac{1}{2^{\frac{n}{2}}\Gamma(\frac{n}{2})}.x^{\frac{n}{2}-1}e^{-\frac{x}{2}}$$

When $n = 2$,

$$\Gamma(1) = 1$$

$$f_X(x) = \frac{1}{2}e^{\frac{-x}{2}}$$

The value of density function $f_Y(y)$ when $\alpha = 1/2$ is given as,

$$f_y(y) = \frac{1}{2}e^{\frac{-y}{2}}$$

Comparing $f_X(x)$ and $f_Y(y)$, we conclude that the other distribution function having the same density function as the exponential function is X_2^2 distribution.

1.3 Sum of Random Variables

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

Solution:

Description:

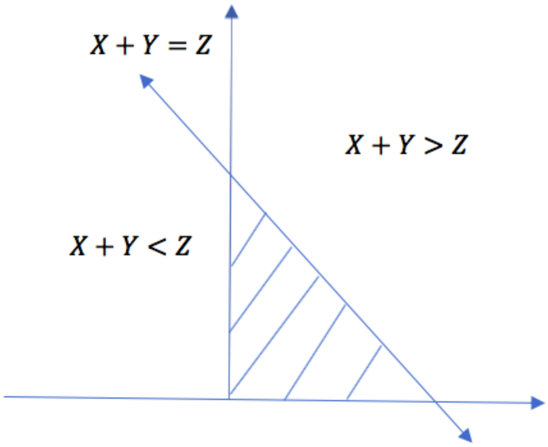


Figure 1.2: function $X + Y = Z$

Let X and Y be two independent random variables distributed on the interval [0, 1] that is, $X, Y \sim R(0, 1)$ and $Z = g(X, Y) = X + Y$. The distribution function for the random variable Z can be expressed as,

$$F_Z(z) = P(Z \leq z) = P(X + Y \leq z) = \int_{-\infty}^{\infty} \int_{\infty}^{z-x} f_{XY}(\mathrm{xy})\ \mathrm{dx}\ \mathrm{dy}$$

The density function of the random variable Z is the derivative pf the distribution function,

$$f_Z(z) = \frac{\partial F_Z(z)}{\partial z}$$

Substituting $F_Z(z)$ in $f_Z(z)$,

$$f_Z(z) = \frac{\partial}{\partial z} \int_{-\infty}^{\infty} \int_{-\infty}^{z-x} f_{XY}(\mathrm{x,y})\ \mathrm{dy}\ \mathrm{dx}$$

$$f_Z(z) = \int_{-\infty}^{\infty} \frac{\partial}{\partial z} \left(f_{XY}(\text{x, z-x}) - f_{XY}(\text{x}, -\infty)\right) \text{dx}$$

$$f_Z(z) = \int_{-\infty}^{\infty} f_{XY}(\text{x, z-x}) \text{ dx}$$

If we assume X and Y as two independent random variables, then we get the density function for sum of two random variables as,

$$f_Z(z) = \int_{-\infty}^{\infty} f_X(\text{x}) f_Y(\text{z - x})\text{dx}$$

This is a convolution integral and it represents the sum of two random variables.

$$f_Z(z) = \int_{-\infty}^{\infty} 1_{[0,1]}(\text{z-x}) \text{ dx} = \int_{0}^{1} 1_{[0,1]}(\text{z-x}) \text{ dx}$$

Consider $u = z - x$, $\Rightarrow dx = -du$

$$f_Z(z) = -\int_{z}^{z-1} 1_{[0,1]}(\text{u}) \text{ du} = \int_{z-1}^{z} 1_{[0,1]}(\text{u}) \text{ du}$$

The function $1_{[0,1]}(\text{u})$ is always zero expect when $0 \leq \text{u} \leq 1$ And, we can discuss the following cases:

Case 1: z<0 and z>2:

$$f_Z(z) = \int_{z-1}^{z} 1_{[0,1]}(\text{u}) \text{ du}$$

$$f_Z(z) = 0$$

Case 2: $\mathbf{0 \leq z \leq 1}$:

$$f_Z(z) = \int_{z-1}^{z} 1_{[0,1]}(\mathrm{u})\ \mathrm{du} = \int_{z-1}^{0} 1_{[0,1]}(\mathrm{u})\ \mathrm{du} + \int_{0}^{z} 1_{[0,1]}(\mathrm{u})\ \mathrm{du} = u|_0^z$$

$$f_Z(z) = z$$

Case 3: $\mathbf{1 < z \leq 2}$:

$$f_Z(z) = \int_{z-1}^{1} 1_{[0,1]}(\mathrm{u})\ \mathrm{du} = u|_{z-1}^{1} = 1 - z + 1$$

$$f_Z(z) = 2 - z$$

The resulting density function can be expressed as:

$$f_z(z) = \begin{cases} 0, & \text{otherwise} \\ \text{z}, & 0 \leq z \leq 1 \\ \text{2 - z}, & 1 \leq z \leq 2 \end{cases}$$

The expected value (mean):

$$E(Z) = \int_{-\infty}^{\infty} z f_Z(z) dz = \int_{0}^{1} z^2 dz + \int_{1}^{2} z(2-z) dz$$

$$E(Z) = \left.\frac{z^3}{3}\right|_0^1 + \int_1^2 2z dz - \int_1^2 z^2 dz$$

$$E(Z) = \frac{1}{3} + z^2|_1^2 - \left.\frac{z^3}{3}\right|_1^2 = \frac{1}{3} + 3 - \frac{7}{3}$$

$$E(Z) = 1$$

For two independent random variables, E(Z) = E(X+Y) = E(X) + E(Y) and for uniformly distributed random variable on the interval [0,1], E(X) = E(Y) = $\frac{1}{2}$, hence E(Z) = $\frac{1}{2} + \frac{1}{2} = 1$.

The variance :

$$\sigma_Z{}^2 = E(Z-1)^2 = \int_{-\infty}^{\infty} (Z-1) f_z(z) dz$$

We have Z=X+Y,

$$\sigma_Z{}^2 = E(Z-1)^2 = E(X+Y-1)^2 = E\left(\left(X-\frac{1}{2}\right)+\left(Y-\frac{1}{2}\right)\right)^2$$

$$\sigma_Z{}^2 = E(X+Y-1)^2 = E\left(\left(X-\frac{1}{2}\right)^2 + 2\left(X-\frac{1}{2}\right)\left(Y-\frac{1}{2}\right)+\left(Y-\frac{1}{2}\right)^2\right)$$

We know that, Cov(X,Y) = 0. Because X and Y are independent.

$$\sigma_Z{}^2 = \frac{1}{12} + \frac{1}{12}$$

$$\sigma_Z{}^2 = \frac{1}{6}$$

1.4 Product of random variables

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

Solution: Let $Z = XY$ be the product of two random variables X and Y, where X and Y are identical, independent and uniformly distributed random variables, i.e., $X, Y \sim R[0,1]$

In order to get the density function of the product of the two random variables, we introduce an auxiliary variable, $W = Y$. We also consider that the density of the bivariate function $f(x,y)$ is known. Therefore, Y is a function of $h(x,y)$,

$$x = \frac{z}{y} = \frac{z}{w} = g^{-1}(z,w)$$

$$y = w = h^{-1}(z,w)$$

To determine the distribution of the transformed random variables, the Jacobian matrix is emplyed,

$$J(x,y) = \begin{bmatrix} \frac{\partial g}{\partial x} & \frac{\partial g}{\partial y} \\ \frac{\partial h}{\partial x} & \frac{\partial h}{\partial y} \end{bmatrix} = \begin{bmatrix} y & x \\ 0 & 1 \end{bmatrix}$$

$$|\det(J(x,y))| = |y| = |w|$$

The bivariate density function is given by,

$$f_{Z,W}(z,w) = \frac{f_{XY}(g^{-1}(z,w), h^{-1}(z,w))}{|\det(J)|}$$

$$f_{Z,W}(z,w) = \frac{f_{XY}(z/w, w)}{|w|}$$

As the variable X and Y are independent, the probability function can be written as,

$$f_{XY} = f_X\left(\frac{z}{w}\right) f_Y(w)$$

The probability density function of Z is given by,

$$f_Z(z) = \int_{-\infty}^{\infty} f_{ZW}(z, w)dw$$

$$f_Z(z) = \int_{-\infty}^{\infty} \frac{f_{X,Y}(\frac{z}{w}, w)}{|w|}dw$$

$$f_Z(z) = \int_{-\infty}^{\infty} \frac{f_X(\frac{z}{w})f_y(w)}{|w|}dw$$

$$f_Z(z) = \int_{0}^{1} \frac{f_X(\frac{z}{w})}{|w|}dw$$

Case 1: z<0 i.e., $\mathbf{\frac{z}{w}}$<0:

$$f_Z(z) = 0$$

Case 2: $\mathbf{0 < z \leq 1}$, i.e, $\mathbf{z \leq w < 1}$: This means, $0 < \frac{z}{w} \leq 1$

$$f_Z(z) = \int_{z}^{1} \frac{1}{w}dw = \ln(w)|_z^1$$

$$f_Z(z) = -\ln(z)$$

Case 3: z>1 i.e., $\mathbf{\frac{z}{w}}$>1:

$$f_Z(z) = 0$$

Then, the density function can be expressed by,

$$f_z(z) = \begin{cases} 0 & z < 0 \\ \text{-ln(z)} & 0 \leq z \leq 1 \\ 0 & z > 1 \end{cases}$$

The expected value (mean):

$$E[Z] = E[XY] = E[X].E[Y]$$

$$E(Z) = \int_{-\infty}^{\infty} z f_Z(z) dz = -\int_0^1 z\ln(z) dz = -\left(\frac{z^2}{2} ln(z) \Big|_0^1 - \int_0^1 \frac{z^2}{2} \frac{1}{z} dz \right) = \frac{z^2}{4} \Big|_0^1$$

$$E(Z) = \frac{1}{4}$$

The variance :

$${\sigma_Z}^2 = E[Z^2] - (E[Z])^2$$

So,

$$E[Z^2] = \int_{-\infty}^{\infty} z^2 f_Z(z) dz = -\int_0^1 z^2 \ln(z) dz = \frac{z^3}{9} \Big|_0^1 = \frac{1}{9}$$

$$(E[Z])^2 = \frac{1}{16}$$

Hence, the variance is,

$${\sigma_Z}^2 = \frac{1}{9} - \frac{1}{16} = \frac{7}{144}$$

1.5 χ^2 distribution

Task: 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.

Solution: χ^2 distribution defines a property such that its mean coincides with the number of degrees of freedom, n. It is mathematically defined as,

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

Where, X_i is a standard normally distributed random variable. Here, $n = 4$, therefore, the mean is 4 for χ_4^2 distribution. The variance of the distribution is nothing but, twice the degree of freedom. Hence, the variance is 8 for χ_4^2 distribution.

The expected value (mean): Conventionally, it can also be calculated by,

$$E[Z] = \int_{-\infty}^{\infty} z f_Z(z) dz = \int_0^1 z \left(\frac{1}{4}.z.e^{-\frac{z}{2}} \right) dz$$

$$E[Z] = 4$$

The variance: Conventionally, it can also be calculated by,

$$\sigma_Z{}^2 = E[Z^2] - (E[Z])^2$$

$$\sigma_Z{}^2 = \int_{-\infty}^{\infty} z^2 f_Z(z)dz - 16$$

$$\sigma_Z{}^2 = 8$$

Generating a random variable from two exponential distractions: Let us suppose we have two exponentially distributed random variables X_1 and X_2. Hence we can calculate the density of the sum of X_1 and X_2 by generating a third random variable Z. If X_i, $(i = 1, 2, 3, ..., n)$n are stochastically independent exponentially distributed random variables with rate parameter, α and mean, $1/\alpha$, then the probability density function is,

$$f_Z(x) = f_{X_1+X_2+...+X_n}(x) = \alpha e^{-\alpha x}\frac{(\alpha x)^{n-1}}{(n-1)!}$$

Thus for $n = 2$,

$$f_Z(x) = f_{X_1+X_2}(x) = \alpha e^{-\alpha x}\frac{(\alpha x)^{2-1}}{(2-1)!} = \alpha^2 x e^{-\alpha x}$$

If $\alpha = 1/2$,

$$f_Z(x) = \frac{1}{4}xe^{-\frac{1}{2}x} \tag{1}$$

Let us consider a random variable Z, which represents a function of a standard normally distributed random variables, such that

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

Its probability density function is given by,

$f_Z(x) = \frac{1}{2^n/2\Gamma(n/2)} Z^{\frac{n}{2}-1} e^{-z/2}, \quad$ z > 0

$$f_Z(z) = \frac{1}{4} z e^{-\frac{1}{2}z} \tag{2}$$

From equation (1) and equation (2), we conclude that the density of random variable Z as a sum of two exponentially distributed random variables is the same as that ofχ^2 distributed random variable Z with four degrees of freedom, if $\alpha = 1/2$.

Probability Density as convolution of density of exponential random variables: We have four stochastically independent, standard normally distributed random variables, X_i where $(i = 1, 2, 3, 4)$. When these random variables are squared and grouped such that we have two exponentially distributed random variables X and Y where,

$X = X_1^2 + X_2^2$ and $Y = Y_3^2 + Y_4^2$

Therefore, the density function for the random variables can be expressed as,

$$f_X(x) = \frac{1}{2} e^{-\frac{x}{2}}$$

and

$$f_Y(y) = \frac{1}{2} e^{-\frac{y}{2}}$$

Now, if we assume that X and Y generate a new random variable Z, such that $Z = X + Y$. Then the bivariate density function for X and Y gives the probability distribution function of Z.

$$f_Z(z) = \int_{-\infty}^{\infty} f_{(X,Y)}(x, y) dxdy = \int_{-\infty}^{\infty} f_X(x) f_Y(y) dxdy$$

Substituting $Y = Z - X$,

$$f_Z(z) = \int_0^z f_X(x) f_Y(z-x) dx$$

$$f_Z(z) = \frac{1}{4}.z.e^{-\frac{z}{2}}$$

1.6 Normal distribution from uniform distribution

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

$Y_1 = \sqrt{-2\ln X_1}\sin(2\pi X_2)$ and each on [0,1] uniformly distributed random variables.

Solution: Given two random variables X_1 and X_2 which are independent, identical and uniformly distributed on the interval [0,1] that is $X_1 \sim R(0,1), Y_1 \sim R(0,1)$. The task is to generate bivariate density function of two random variables Y_1 and Y_2 where,

$$Y_1 = \sqrt{-2\ln(X_1)}sin(2\pi X_2) = g_1(X1, X2)$$

$$Y_2 = \sqrt{-2\ln(X_1)}cos(2\pi X_2) = g_2(X1, X2)$$

First step is relating Y_1 and Y_2 with X_1 and X_2 as follows:

$$Y_1^2 + Y_2^2 = -2\ln(X_1)$$

$$x_1 = g_1^{-1}(y_1, y_2) = \exp\left(-\frac{y_1^2 + y_2^2}{2}\right)$$

$$x_2 = g_2^{-1}(y_1, y_2) = \frac{1}{2\pi}\arctan\frac{y_1}{y_2}$$

To calculate the bivariate density function,

$$f_{Y_1Y_2}(y_1y_2) = \frac{f_{x_1x_2}(x_1,x_2)}{|\det(J)|}\Bigg|_{x_1=g_1^{-1}(y_1,y_2),\ x_2=g_2^{-1}(y_1,y_2)}$$

$$f_{Y_1Y_2}(y_1y_2) = f_{x_1x_2}(x_1,x_2)/\frac{\partial(x_1,x_2)}{\partial(y_1,y_2)} \tag{3}$$

where,

$$\frac{\partial(x_1,x_2)}{\partial(y_1,y_2)} = |\det J|$$

Also, $f_{X_1X_2}(x_1x_2) = f_{X_1}(x_1)f_{Y_1}(y_1)$

To determine the determinant of the Jacobian Matrix, J,

$$J = \begin{bmatrix} \frac{\partial g_1(x_1,x_2)}{\partial x_1} & \frac{\partial g_1(x_1,x_2)}{\partial x_2} \\ \frac{\partial g_2(x_1,x_2)}{\partial x_1} & \frac{\partial g_2(x_1,x_2)}{\partial x_2} \end{bmatrix} = \begin{bmatrix} \frac{sin(2\pi x_2)}{x_1\sqrt{-2\ln x_1}} & 2\pi\sqrt{-2\ln x_1}cos(2\pi x_2) \\ \frac{cos(2\pi x_2)}{x_1\sqrt{-2\ln x_1}} & -2\pi\sqrt{-2\ln x_1}sin(2\pi x_2) \end{bmatrix}$$

$$|\det(J)| = |\frac{2\pi}{x_1}|$$

$$|\det(J)| = \frac{2\pi}{\exp\Bigg(-\frac{y_1^2+y_2^2}{2}\Bigg)} \tag{4}$$

The joint probability density function of X_1 and X_2 will be:

$$f_{X_1X_2}(x_1x_2) = \begin{cases} 1 & 0 \leq x_1, x_2 \leq 1 \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

Now, substituting equation (4) and equation (5) in equation (3), we get,

$$f_{x_1x_2}(x_1x_2) = \frac{1}{2\pi}exp(-\frac{y_1^2+y_2^2}{2})$$

Bivariate density function can be written as a product of two functions as shown below:

$$f_{y_1y_2}(y_1y_2) = \frac{1}{2\pi}\exp(-\frac{y_1^2}{2}).\frac{1}{2\pi}exp(-\frac{y_2^2}{2}) = f_{y_1}(y_1).f_{y_2}(y_2)$$

$$f_{y_1}(y1) = \frac{1}{2\pi}\exp(-\frac{y_1^2}{2})$$

$$f_{y_2}(y2) = \frac{1}{2\pi}\exp(-\frac{y_2^2}{2})$$

Each of these two functions represents a density function of a standard uniformly distributed random variable that is $Y_1 \sim N(0,1)$ and $Y_2 \sim N(0,1)$. Thus Y_1 and Y_2 can be termed now as independent and Standard Normally Distributed Random Variables.

Chapter 2

Exercises with Matlab

2.1 Standard normal distribution

Task: 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 value 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 =`$[\mathrm{x} - 0.6 * \mathrm{x}.^3 + 0.9 * \mathrm{x}.^2 + 3 * \mathrm{x} + 4.5 + \mathrm{z}]$;

```
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_1  xy
```

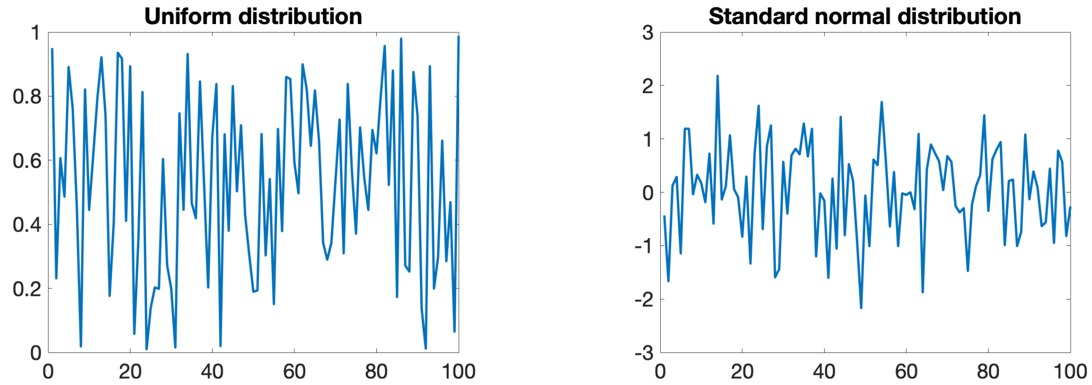
Solution: At first, a uniformly distributed random sequence, `x1` and a standard normal distributed random sequence, `z1` with length as 100 are created. Then, four different signal models are created using the equations mentioned in the task. The four models are: exponential model, sine model, polynomial model and power model. These models are finally saved in `dat3_1`.

MATLAB code:

```
clear all;
close all;
clc;

N = 100;
rand ('state',0);
x1 = rand(N,1);
figure()
plot(x1,'LineWidth',2.5)
title('Uniform distribution')
set(gca,'FontSize',22)

randn ('state',0);
z1 = randn(N,1);
figure()
plot(z1,'LineWidth',2.5)
title('Standard normal distribution')
set(gca,'FontSize',22)
```

```
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
```

Output: The plot of uniform distribution and standard normal distribution is shown in Figure 2.1.



(a) Uniform distribution

(b) Standard normal distribution

Figure 2.1: Distributions

Inference: Four different models which represent exponential, sine, polynomial and power model are created. Also, the uniform distribution with mean as 0 and variance as 1 is generated.

2.2 Model assignment

Task: Load `dat3_1` that includes the four 100 x 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.)

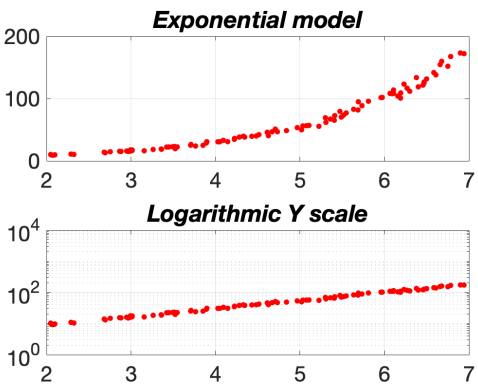
Solution: The four models are first loaded and then are plotted. From the first set of data, i.e., `xy1` exponential graph is plotted. And similarly rest of the models are graphed from rest of the data sets. The exponential and power model is also represented using logarithmic Y scale.

MATLAB code:

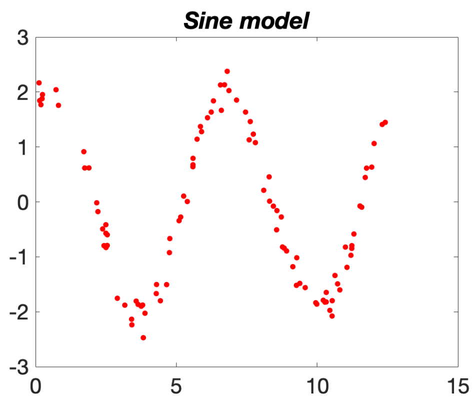
```
clear all;
close all;
clc

load dat3_1;
figure()
subplot(2,1,1)
plot(xy1(:,1),xy1(:,2),'.r','MarkerSize',15)
grid on
set(gca,'FontSize',22)
title('Exponential model','FontAngle', 'italic', 'FontWeight', 'bold')
subplot(2,1,2)
semilogy(xy1(:,1),xy1(:,2),'.r','MarkerSize',15)
grid on
set(gca,'FontSize',22)
title('Logarithmic Y scale','FontAngle', 'italic', 'FontWeight', 'bold')

figure()
plot(xy2(:,1),xy2(:,2),'.r','MarkerSize',15)
set(gca,'FontSize',22)
title('Sine model','FontAngle', 'italic', 'FontWeight', 'bold')

figure()
plot(xy3(:,1),xy3(:,2),'.r','MarkerSize',15)
grid on
set(gca,'FontSize',22)
title('Polynomial model','FontAngle', 'italic', 'FontWeight', 'bold')

figure()
subplot(2,1,1)
plot(xy4(:,1),xy4(:,2),'.r','MarkerSize',15)
grid on
set(gca,'FontSize',22)
```

```
title('Power Model','FontAngle', 'italic', 'FontWeight', 'bold')
subplot(2,1,2)
semilogy(xy4(:,1),xy4(:,2),'.r','MarkerSize',15)
grid on
ytick = 10.^(0:3);
yticklab = cellstr(num2str(round(-log10(ytick(:))), '10^-^%d'));
set(gca,'YTick',ytick,'YTickLabel',yticklab,'TickLabelInterpreter', ...
    'tex','FontSize',22)
title('Logarithmic Y scale','FontAngle', 'italic', 'FontWeight', 'bold')
```

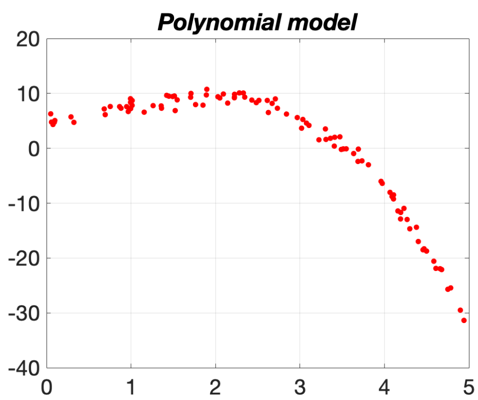
Output: The output after execution of the code is shown in Figure 2.2



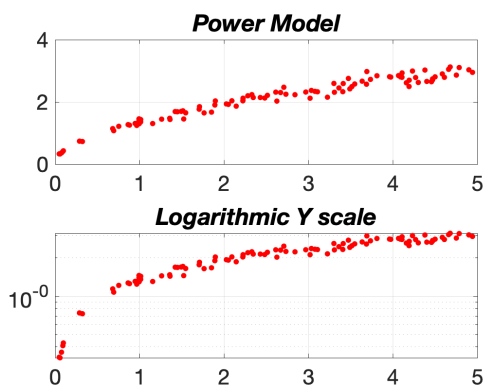
(a) Exponential model



(b) Sine model



(c) Polynomial model



(d) Power model

Figure 2.2: Four different models

Inference: We plotted the generated data sets and assigned them a proper model. The logarithm graph of the exponential model is observed to be a straight line because for the exponential model $g(x)$, $\log(g(x))$ will depend linearly on x. And similarly, for $g(x)$ as a power model, $\log(g(x))$ depends logarithmically on x.

2.3 LS-Estimation of different models

Task: Estimate each of the parameter a, b and the variance of the measurement errors ${\sigma_z}^2$ of the linearized models, i.e. of the exponential model, the power model and the sine model. Write therefore a function `LSE` that is able to deal with these three models and with a polynomial model of arbitrary order.

Solution: We estimate the parameters and the variance of the measurement errors ${\sigma_z}^2$ of the exponential model, the power model, the sine model and the polynomial model. We wrote a function `LSE` to obtain the above mentioned parameters. The inputs of this function are the data sets from `dat3_1` i.e. (x_i, y_i), the model whose parameters we want and the integer p. In the case of polynomial model, p represents the order of the polynomial, while for the rest of the model $p = 1$. We ask the user to input the value of p for the polynomial model.

MATLAB function `LSE`:

```
function [coefficient, sigma] = LSE(x, y, model, p)
if strcmp(model,'exponential') == 1
    H = [ones(length(x),1) x];
   [av, sigma] = getVariance(length(x), H, log(y), p);
    coefficient = exp(av);
    fprintf('\nParameters for Exponential model for degree = %d are:',p)
    fprintf('\na = %f, b = %f, variance = %f', ...
```

```
        coefficient(1,:),coefficient(2,:),sigma)
elseif strcmp(model,'power') == 1
    H = [ones(length(x),1) log(x)];
    [av, sigma] = getVariance(length(x), H, log(y), p);
    coefficient(1,:) = exp(av(1));
    coefficient(2,:) = av(2);
    fprintf('\n\nParameters for Power model for degree = %d are:',p)
     fprintf('\na = %f, b = %f, variance = %f', ...
        coefficient(1,:),coefficient(2,:),sigma)
elseif strcmp(model,'sine') == 1
    H=[sin(x) cos(x)];
   [av, sigma] = getVariance(length(x), H, y, p);
    coefficient(2,:) = atan(av(2)/av(1));
    coefficient(1,:) = av(1)/cos(coefficient(2,:));
    fprintf('\n\nParameters for Sine model for degree = %d are:',p)
    fprintf('\na = %f, b = %f, variance = %f', ...
        coefficient(1,:),coefficient(2,:),sigma)
elseif strcmp(model,'polynomial') == 1
    for i = 1:length(x)
        for j = p+1:-1:1
            H(i,j) = (x(i).^(j-1));
        end
    end
    [coefficient, sigma] = getVariance(length(x), H, y, p);
    fprintf('\n\nParameters for Polynomial model for degree = %d are:',p)
    for i = 1:length(coefficient)
         fprintf('\n%f',coefficient(i,:))
    end
    fprintf('\nvariance = %f',sigma)
end


    function [av, sigma] = getVariance(ln, H, y1, order)
       P = H*(inv(H'*H))* H';
       PO = eye(ln) - P;
       av = (inv(H'*H))*H'*y1;
       sigma = y1'*PO'*y1/(ln-(order+1));
```

```
    end

end
```

MATLAB code:

```
clear all
close all
clc

load dat3_1;
p = input('Input the order of the polynomial:');
[coefficient_exp , sigma_exp] = LSE(xy1(:,1),xy1(:,2),'exponential',1);
[coefficient_sin , sigma_sin] = LSE(xy2(:,1),xy2(:,2),'sine',1) ;
[coefficient_pow , sigma_pow] = LSE(xy4(:,1),xy4(:,2),'power',1);
[coefficient_ply , sigma_ply] = LSE(xy3(:,1),xy3(:,2),'polynomial',p);
```

Output: The output after execution of the code is shown below. We gave input as $p = 1$ and $p = 3$. The code prints the parameters for each of the model.

```
Input the order of the polynomial:1

Parameters for Exponential model for degree = 1 are:
a = 2.714761, b = 1.823818, variance = 0.003046

Parameters for Sine model for degree = 1 are:
a = 2.015541, b = 1.008430, variance = 0.037934

Parameters for Power model for degree = 1 are:
a = 1.354748, b = 0.499158, variance = 0.003047

Parameters for Polynomial model for degree = 1 are:
16.827522
-6.241403
variance = 43.876759>>
```

```
Input the order of the polynomial:3

Parameters for Exponential model for degree = 1 are:
a = 2.714761, b = 1.823818, variance = 0.003046

Parameters for Sine model for degree = 1 are:
a = 2.015541, b = 1.008430, variance = 0.037934

Parameters for Power model for degree = 1 are:
a = 1.354748, b = 0.499158, variance = 0.003047

Parameters for Polynomial model for degree = 3 are:
4.709133
2.477967
1.187414
-0.640214
variance = 0.767853>>
```

Inference: Estimation of the wanted parameters for our generated data is done successfully.

2.4 Estimating the order of a polynomial

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

Solution: The same function `LSE` is used to obtain the variance of the polynomial model. The polynomial order p, is taken from 1 to 10 and the parameters at each order are shown in the output.

MATLAB code:

```
clear all
close all
clc

load dat3_1;
for p = 10:-1:1
[~, sigma_ply(p)] = LSE(xy3(:,1),xy3(:,2),'polynomial',p);
end
plotGraph(sigma_ply)

function plotGraph(sigma)
p = 1:10;
plot(p,sigma,'-ro',...
    'LineWidth',3,...
    'MarkerEdgeColor','k',...
    'MarkerFaceColor','r',...
    'MarkerSize',10)
grid on
set(gca,'FontSize',28)
str = 'Variance as a function of polynomial order';
set(gca,'Title',text('String',str,'FontAngle', 'italic', ...
    'FontWeight', 'bold'), ...
        'xlabel',text('String', 'polynomial order (p)',...
```

```
    'FontAngle','italic'),...
    'ylabel',text('String', 'variance','FontAngle','italic'), ...
    'FontSize',28)

end
```

Output: The written code plots variance versus polynomial order. It can be seen that the variance decreases drastically and after a point it remains constant. The plot can be seen in Figure 2.3.

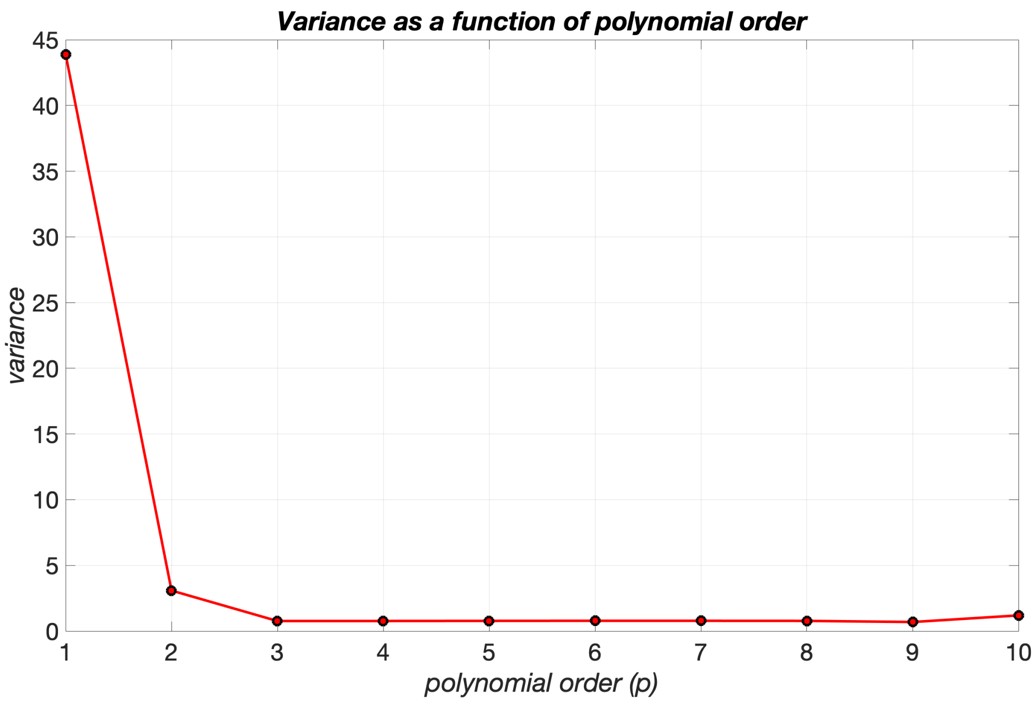


Figure 2.3: Variance as a function of the polynomial order

The output of the execution is shown below. The parameters for polynomial order 1 to 10 are printed and shown.

```
Parameters for Polynomial model for degree = 10 are:
5.299264
-2.723607
-6.276594
76.732053
-153.649104
145.967431
-78.227471
24.897769
-4.667275
0.476042
-0.020380
variance = 1.195394

Parameters for Polynomial model for degree = 9 are:
6.774025
-32.555802
146.734740
-261.125133
245.242899
-133.100953
43.098831
-8.217127
0.850924
-0.036910
variance = 0.691175
```

```
Parameters for Polynomial model for degree = 8 are:
5.111471
-2.399149
18.477101
-28.173307
22.562530
-10.156769
2.535997
-0.329173
0.017318
variance = 0.777727

Parameters for Polynomial model for degree = 7 are:
4.528678
6.766773
-11.812788
14.262726
-8.260890
2.382645
-0.344544
0.019745
variance = 0.783534

Parameters for Polynomial model for degree = 6 are:
4.950627
0.985690
3.220401
-1.829663
0.360478
-0.056690
0.003666
variance = 0.785786
```

```
Parameters for Polynomial model for degree = 5 are:
4.902030
1.537868
2.113314
-0.960441
0.040257
-0.001117
variance = 0.777667

Parameters for Polynomial model for degree = 4 are:
4.892581
1.614992
2.000118
-0.899167
0.026369
variance = 0.769498

Parameters for Polynomial model for degree = 3 are:
4.709133
2.477967
1.187414
-0.640214
variance = 0.767853

Parameters for Polynomial model for degree = 2 are:
0.830208
11.804629
-3.541588
variance = 3.083713

Parameters for Polynomial model for degree = 1 are:
16.827522
-6.241403
variance = 43.876759>>
```

Inference: The variance reduces from 43.87 at $p = 1$ to 0.767 at $p = 3$. Hence the polynomial of order 3 is considered to be the correct order. This is because, as the order increases further than 3, it remains almost constant.

2.5 Comparison of observations and estimated model

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

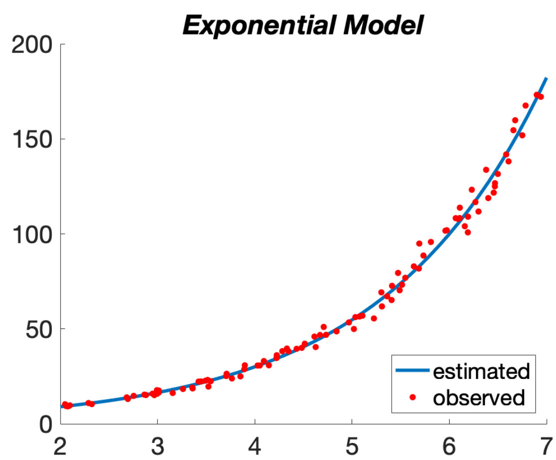
Solution: We created a function `compare` inside the function `plotGraph` which used the values of the parameters from the previous exercise i.e., when $p = 3$ for polynomial model and $p = 1$ for the rest of the models to compare with the observed values from the data set which was previously created.

MATLAB code:

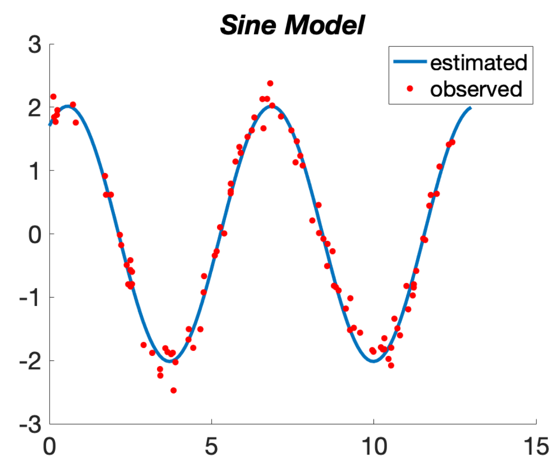
```
clear all
close all
clc

load dat3_1;
plotGraph('Exponential Model', xy1(:,1),xy1(:,2));
plotGraph('Sine Model', xy2(:,1),xy2(:,2));
plotGraph('Polynomial Model', xy3(:,1),xy3(:,2));
plotGraph('Power Model', xy4(:,1),xy4(:,2));

function plotGraph(model,observed_x,observed_y)
if strcmp(model,'Exponential Model') == 1
    expected_value = 2 : 0.1 : 7;
    expected_t = 2.714761*(1.823818.^expected_value);
    compare(expected_value, expected_t, observed_x, observed_y, model);
elseif strcmp(model,'Sine Model') == 1
    expected_value = 0 : 0.1 : 13;
    expected_t = 2.015541*sin(expected_value + 1.008430);
    compare(expected_value, expected_t, observed_x, observed_y, model);
elseif strcmp(model,'Polynomial Model') == 1
    expected_value = 0 : 0.1 : 5;
    expected_t = 4.709133 + 2.477967*expected_value ...
        + 1.18741*expected_value.^2 - 0.640213766*expected_value.^3;
    compare(expected_value, expected_t, observed_x, observed_y, model);
elseif strcmp(model,'Power Model') == 1
    expected_value = 0 : 0.1 : 5;
    expected_t = 1.354748*(expected_value .^ 0.499158);
    compare(expected_value, expected_t, observed_x, observed_y, model);
end
    function compare(expected_x,expected_y,observed_x,observed_y,model)
        figure()
        hold on
        plot(expected_x, expected_y,'LineWidth',3)
        plot(observed_x,observed_y,'.r','MarkerSize',15)
        hold off
```

```
        set(gca,'FontSize',22)
        title(model,'FontAngle', 'italic', 'FontWeight', 'bold')
        legend('estimated','observed')
    end


end
```

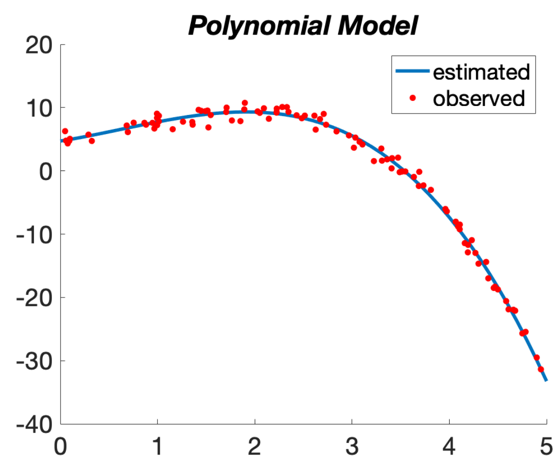
Output: The output of the code is shown below. The estimated value is shown by a line while the observed value is represented by dots.



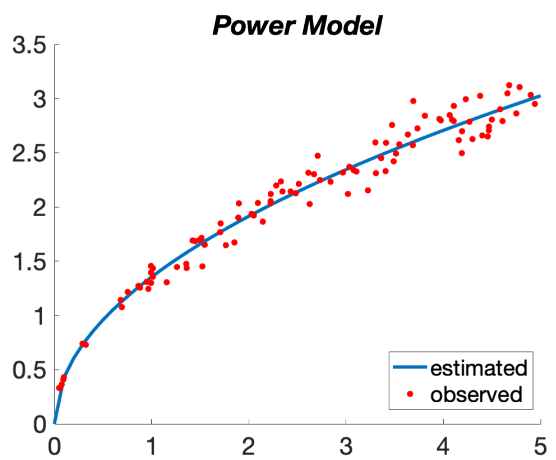
(a) Exponential model



(b) Sine model



(c) Polynomial model



(d) Power model

Figure 2.4: Comparison of observations and estimated models

Inference: We can infer that the observed value almost coincides with the estimated value.

2.6 LS-adjustment to a circle: estimating the centre

Task: 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 give from this the coordinates of the centre location.

Solution: `dat3_2` was loaded as mentioned in the task. It contains the matrix of data set xy. We estimate and find the centre of the circle to be located at (4,2).

MATLAB code:

```
clear all
close all
clc

load dat3_2;
figure()
hold on
plot(xy(:,1),xy(:,2),'.b','MarkerSize',28)
plot(4,2,'xr','MarkerSize',18)
hold off
set(gca,'FontSize',26)
title('LS - Adjustment to a circle: estimating the center', ...
    'FontAngle', 'italic', 'FontWeight', 'bold')
axis equal
```

Output: The output of the code is shown below.

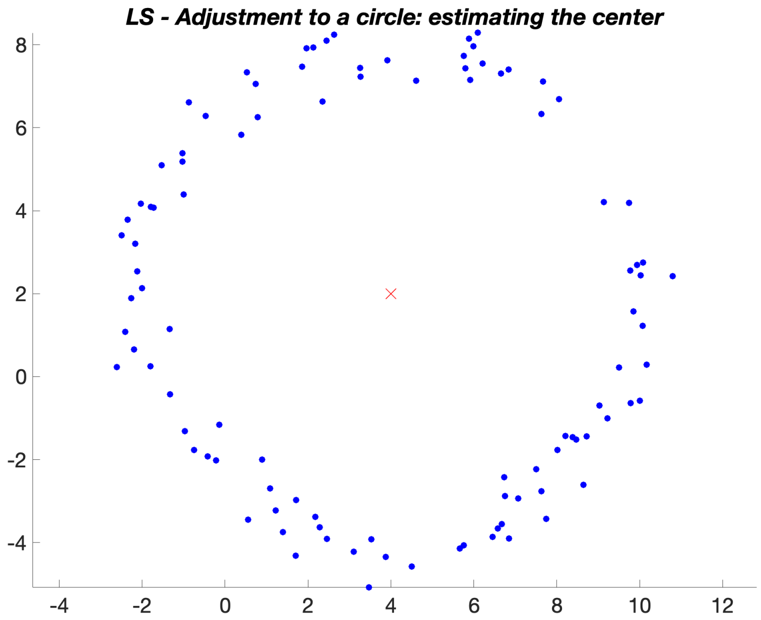


Figure 2.5: LS-adjustment to a circle: estimating the centre

Inference: The measurement points nearly takes the shape of a circle with centre at (4,2)

2.7 LS-Fit to a circle: iteration

Task: 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 $||(\tilde{X}_0(k), \tilde{Y}_0(k)) - (\tilde{X}_0(k-1), \tilde{Y}_0(k-1))|| < 10^{-10}$ is fulfilled, where $(\tilde{X}_0(k), \tilde{Y}_0(k))$ 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.

Solution: We carry a linear LS - Approach to a circle which is mentioned in topic 1.5. We wrote a function `LSE_circle` which outputs the LS-estimated circle fitting when given the measured points as the input. Maximum iteration time is added to control the termination incase the code does not converge.

MATLAB function LSE:

```
function [est_x0, est_y0, r] = LSE_circle(x, y, mean_x, mean_y)
err = 1;
count_max = 100;
count = 0;
while err > 10^-10  && (count < count_max)
    count = count + 1;
    for i = length(x) : -1 : 1
        c(i) = sqrt((x(i)-mean_x)^2 + (y(i)-mean_y)^2);
        a(i) = (x(i)-mean_x)/c(i);
        b(i) = (y(i)-mean_y)/c(i);
    end
    H = [ones(length(x),1) a' b'];
    V = (inv(H'*H))*H'*c';
    r = V(1);
    u0 = V(2);
    v0 = V(3);
    est_x0 = u0 + mean_x;
    est_y0 = v0 + mean_y;
    err = max(u0,v0);
    mean_x = est_x0;
    mean_y = est_y0;
end
end
```

MATLAB code:

```
clear all
close all
clc

load dat3_2;
x = mean(xy(:,1));
```

```
y = mean(xy(:,2));
[x0,y0,r] = LSE_circle(xy(:,1),xy(:,2),x,y);
figure()
hold on
plot(xy(:,1),xy(:,2),'.b','MarkerSize',25)
plot(x0,y0,'xr','MarkerSize',15)
hold off
axis equal
hold on
rectangle('Position',[x0-r,y0-r,2*r,2*r],'Curvature',[1,1],'LineWidth',3);
hold off
set(gca,'Title',text('String','LS-Fit to a circle', ...
    'FontAngle', 'italic','FontWeight', 'bold'), ...
        'xlabel',text('String', 'x', 'FontAngle','italic'),...
        'ylabel',text('String', 'y','FontAngle','italic'), ...
        'FontSize',28)
fprintf('\nThe fitted circle has:')
fprintf('\ncentre at (%f,%f)',x0,y0)
fprintf('\nradius = %f',r)
```

Output: The output of the code is shown below.

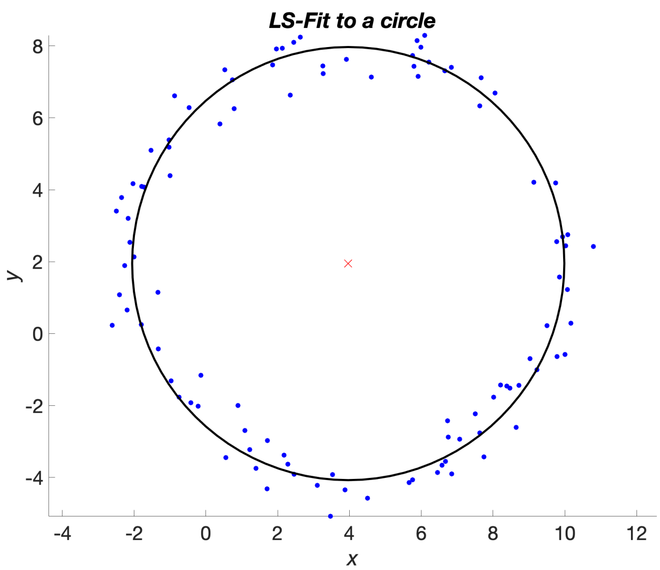


Figure 2.6: LS-Fit to a circle: iteration

The execution of the code gives us the following result.

```
The fitted circle has:
centre at (3.965483,1.947539)
radius = 6.020386>>
```

Inference: The measurement points nearly takes the shape of a circle with centre at (3.965,1.947) and radius of 6.020.

2.8 LLS-Fit to a circle: convergence

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

Solution: The function `LSE_circle` mentioned earlier is used again in this program. In this method, we require an initial guess of the centre of the circle. If this guess is poor, the method does not converge and no result can be obtained. **MATLAB code:**

```
clear all
close all
clc

load dat3_2;
x = xy(:,1);
y = xy(:,2);
[estx0, esty0, r] = LSE_circle(x, y, mean(x), mean(y));
mx = -4 : 1 : 12;
my = -6 : 1 : 10;
a = 1;
```

```
b = 1;
X_NCONV = [];
Y_NCONV = [];
X_CONV = [];
Y_CONV = [];
for i = 16 : -1 : 1
    for j = 1 : 17
        MX(i) = mx(i);
        MY(j) = my(j);
        [x0,y0,~] = LSE_circle(x,y,MX(i),MY(j));
        u0 = abs(estx0-x0);
        v0 = abs(esty0-y0);
        r1 = sqrt(u0.^2 + v0.^2);
        if r1 > r
            X_NCONV(a) = MX(i);
            Y_NCONV(a) = MY(j);
            a = a + 1;
        else
            X_CONV(b) = MX(i);
            Y_CONV(b) = MY(j);
            b = b+1;
        end
    end
end
figure()
plot(X_NCONV,Y_NCONV,'.r','MarkerSize',15)
hold on;
plot(X_CONV,Y_CONV,'.b','MarkerSize',15)
set(gca,'Title',text('String','LS-Fit to a circle: convergence', ...
    'FontAngle', 'italic','FontWeight', 'bold'), ...
        'xlabel',text('String', 'x', 'FontAngle','italic'),...
        'ylabel',text('String', 'y','FontAngle','italic'), ...
        'FontSize',28)
legend('not converging','converging');
rectangle('Position',[estx0-r,esty0-r,2*r,2*r],'Curvature',[1,1], ...
    'LineWidth',3);
```

```
axis equal;
```

Output: The output of the code is shown below. The blue dots represent convergence while the red represent non convergence. Thus, taking the blue points as initial guess will lead to convergence and vice versa.

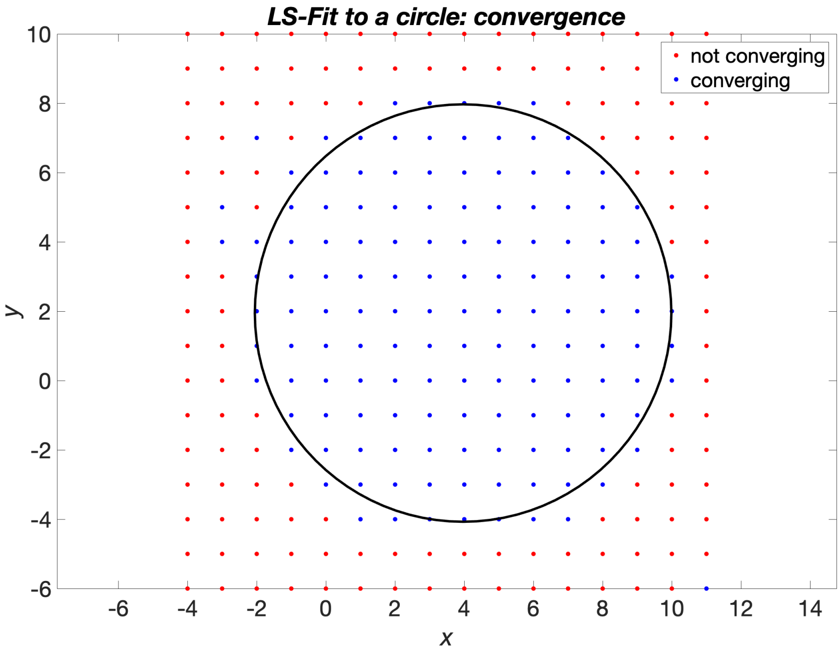


Figure 2.7: LS - adjustment to a circle: estimating the centre

Inference: Majority of the blue dots lie inside the circle leading to convergence. Few blue dots lie outside the circle which can also allow convergence. If the initial guess is placed where there is a red dot, the system will not converge and hence no result will be obtained.

Bibliography

[1] Prof. Dr.-Ing. Dieter Kraus, *"Stochastic Signals and Systems- Probability Theory* lecture notes.

[2] https://de.mathworks.com/help/matlab/

[3] Intuitive Probability and Random Processes using MATLAB, Steven M. Kay, Kluwer Academic Publishers, 2006