Junseo Kim

Assignment 1

Summer 2023

1 Question 1

$\mathbf{x}$ is a random variable of length K:

$$\mathbf{x} = N(\mathbf{0}, \mathbf{1}).$$

a) What type of random variable is the following random variable?

$$\mathbf{y} = \mathbf{x}^\top \mathbf{x}.$$

This is chi-squared (of order K) random variable because this is sum of independent standard normal random variables.

b) Calculate the mean and variance of $\mathbf{y}$. Mean : K, Variance : 2K

$$E[X] = \int_0^\infty x fX(x)dx = \int_0^\infty x c x^{\frac{n}{2}-1} \exp\left(-\frac{1}{2}x\right) dx = c\int_0^\infty x^{\frac{n}{2}} \exp\left(-\frac{1}{2}x\right) dx = c\{\left[-x^{\frac{n}{2}} 2\exp\left(-\frac{1}{2}x\right)\right]_0^\infty +$$

$$\int_0^\infty \frac{n}{2} x^{\frac{n}{2}-1} 2\exp\left(-\frac{1}{2}x\right) dx\} = c\{(0-0) + n\int_0^\infty x^{\frac{n}{2}-1} \exp\left(-\frac{1}{2}x\right) dx\} = n\int_0^\infty c x^{\frac{n}{2}-1} \exp\left(-\frac{1}{2}x\right) dx = n\int_0^\infty fX(x)dx = n$$

$$E[X^2] = \int_0^\infty x^2 fX(x)dx = \int_0^\infty x^2 c x^{\frac{n}{2}-1} \exp\left(-\frac{1}{2}x\right) dx = c\int_0^\infty x^{\frac{n}{2}+1} \exp\left(-\frac{1}{2}x\right) dx = c\{\left[-x^{\frac{n}{2}+1} 2\exp\left(-\frac{1}{2}x\right)\right]_0^\infty +$$

$$\int_0^\infty (\frac{n}{2}+1) x^{\frac{n}{2}} 2\exp\left(-\frac{1}{2}x\right) dx\} = c\{(0-0) + (n+2)\int_0^\infty x^{\frac{n}{2}} \exp\left(-\frac{1}{2}x\right) dx\} = c(n+2)\{\int_0^\infty x^{\frac{n}{2}} \exp\left(-\frac{1}{2}x\right) dx\}$$

$$c(n+2)\{\left[-x^{\frac{n}{2}} 2\exp\left(-\frac{1}{2}x\right)\right]_0^\infty + \int_0^\infty \frac{n}{2} x^{\frac{n}{2}-1} 2\exp\left(-\frac{1}{2}x\right) dx\} = c(n=2)\{(0-0) + n\int_0^\infty x^{\frac{n}{2}-1} \exp\left(-\frac{1}{2}x\right) dx\}$$

$$(n+2)n\int_0^\infty c x^{\frac{n}{2}-1} \exp\left(-\frac{1}{2}x\right) dx = (n+2)n\int_0^\infty fX(x)dx = (n+2)n$$

$$Var[X] = E[X^2] - E[X]^2 = (n+2)n - n^2 = n(n+2-2) = 2n$$

c) Using Python, plot the PDF of $\mathbf{y}$ for K=1, 2, 3, 10, 100.

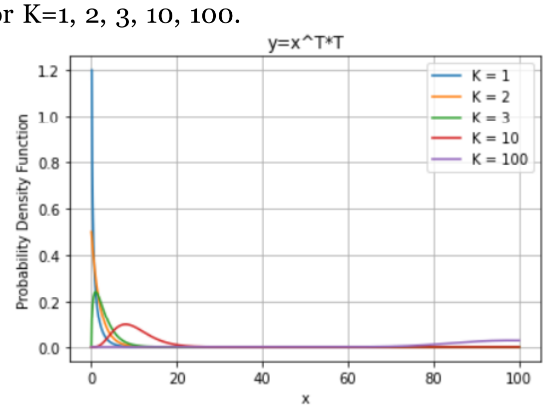

Figure 1: PDF of chi-squared distribution

Detailed Code source can be obtained from

https://github.com/JunseoKim19/State_estimation/blob/main/Probability_Density_Function_chi-square.py

2 Question 2

$\mathbf{x}$ is a random variable of length N:

$$\mathbf{x} = N(\mu, \boldsymbol{\Sigma})$$

a) Assume $\mathbf{x}$ is transformed linearly, i.e. $\mathbf{y} = \mathbf{Ax}$, where $\mathbf{A}$ is an N x N matrix. Calculate the mean and covariance of $\mathbf{y}$. Show the derivations. Mean: $\boldsymbol{A\mu}$, Covariance: $\mathbf{A\Sigma A^T}$

$$\boldsymbol{E[y] = E[Ax] = AE[x], E[x] = \mu, E[y] = A\mu}$$

$$\boldsymbol{cov[y] = E[(y - E[y])(y - E[y])^T] = E[(Ax - A\mu)(Ax - A\mu)^T] = AE[(x - \mu)(x - \mu)^T]A^T}$$

$$\boldsymbol{cov[y] = A\Sigma A^T}$$

b) Repeat **a)**, when $\mathbf{y} = \mathbf{A_1x} + \mathbf{A_2x}$. Mean: $\boldsymbol{A_1\mu + A_2\mu}$, Covariance: $\boldsymbol{A_1\Sigma A_1^T + A_2\Sigma A_2^T}$

$$\boldsymbol{E[y] = E[A_1x + A_2x] = A_1E[x] + A_2E[x] = A_1\mu + A_2\mu}$$

$$\boldsymbol{cov[y] = A_1E[(x - \mu)(x - \mu)^T]A_1^T + A_2E[(x - \mu)(x - \mu)^T]A_2^T = A_1\Sigma A_1^T + A_2\Sigma A_2^T}$$

c) If $\mathbf{x}$ is transformed by a nonlinear differentiable function, i.e. $\mathbf{y} = \mathbf{f(x)}$,compute the covariance matrix of $\mathbf{y}$. Show the derivation

Using Taylor expansion of the function $f(x)$, and Jacobian matrix of f evaluated at μ

$$f(x) = f(\mu) + J_f(\mu)(x - \mu)$$

$$E[y] = E[f(x)] = f(\mu)$$

$$cov[y] = E[\left(f(\mu) + J_f(\mu)(x - \mu) - f(\mu)\right)\left(f(\mu) + J_f(\mu)(x - \mu) - f(\mu)^T\right] = J_f(\mu)\Sigma J_f(\mu)^T$$

d) Apply **c)** when

$$x = \begin{bmatrix}\rho \\ \theta\end{bmatrix}, \boldsymbol{\Sigma} = \begin{bmatrix}\boldsymbol{\sigma^2}_{\boldsymbol{\rho\rho}} & \boldsymbol{\sigma^2}_{\boldsymbol{\rho\theta}} \\ \boldsymbol{\sigma^2}_{\boldsymbol{\rho\theta}} & \boldsymbol{\sigma^2}_{\boldsymbol{\theta\theta}}\end{bmatrix}, and\ y = \begin{bmatrix}\rho cos\theta \\ \rho sin\theta\end{bmatrix}.$$

$$f(x) = \begin{bmatrix}\rho cos\theta \\ \rho sin\theta\end{bmatrix}, J_f(x) = \begin{bmatrix}\frac{\vartheta(\rho cos\theta)}{\vartheta\rho} & \frac{\vartheta(\rho cos\theta)}{\vartheta\theta} \\ \frac{\vartheta(\rho sin\theta)}{\vartheta\rho} & \frac{\vartheta(\rho sin\theta)}{\vartheta\theta}\end{bmatrix} = \begin{bmatrix}cos\theta & -\rho sin\theta \\ sin\theta & \rho cos\theta\end{bmatrix}$$

$$cov[y] = \begin{bmatrix}cos\theta & -\rho sin\theta \\ sin\theta & \rho cos\theta\end{bmatrix}\begin{bmatrix}\boldsymbol{\sigma^2}_{\boldsymbol{\rho\rho}} & \boldsymbol{\sigma^2}_{\boldsymbol{\rho\theta}} \\ \boldsymbol{\sigma^2}_{\boldsymbol{\rho\theta}} & \boldsymbol{\sigma^2}_{\boldsymbol{\theta\theta}}\end{bmatrix}\begin{bmatrix}cos\theta & sin\theta \\ -\rho sin\theta & \rho cos\theta\end{bmatrix}$$

Compute the covariance of $\mathbf{y}$ analytically. This models how range-bearing measurements in the polar coordinate frame are converted to a Cartesiancoordinate frame.

e) Simulate **d)** using the Monte Carlo simulation, i.e. assume

$$\boldsymbol{x} = \begin{bmatrix} \boldsymbol{1m} \\ \boldsymbol{0.5°} \end{bmatrix}, \boldsymbol{\Sigma} = \begin{bmatrix} \boldsymbol{0.01} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0.005} \end{bmatrix}$$

Sample 1000 points from this distribution and plot the transformed results on x-y coordinates. Plot the uncertainty ellipse, calculated from part **d)**. Overlay the ellipse on the point samples.

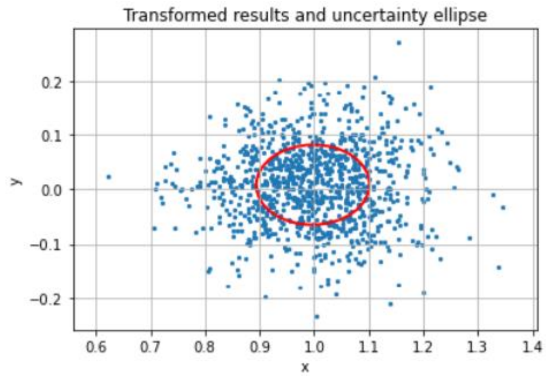

Figure 2: Monte Carlo simulation Case 1

f) Repeat part e), for the following values:

$$\boldsymbol{x} = \begin{bmatrix} \boldsymbol{1m} \\ \boldsymbol{0.5°} \end{bmatrix}, \boldsymbol{\Sigma} = \begin{bmatrix} \boldsymbol{0.01} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0.005} \end{bmatrix} - \boldsymbol{Refer\ to\ Figure\ 2}$$

$$\boldsymbol{x} = \begin{bmatrix} \boldsymbol{1m} \\ \boldsymbol{0.5°} \end{bmatrix}, \boldsymbol{\Sigma} = \begin{bmatrix} \boldsymbol{0.01} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0.1} \end{bmatrix}$$

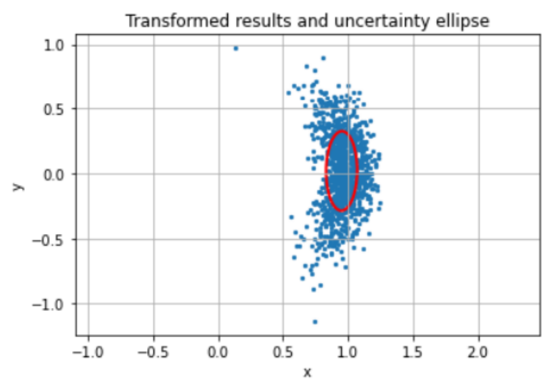

Figure 3: Monte Carlo simulation Case2

$$\boldsymbol{x} = \begin{bmatrix} \boldsymbol{1m} \\ \boldsymbol{0.5°} \end{bmatrix}, \boldsymbol{\Sigma} = \begin{bmatrix} \boldsymbol{0.01} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0.5} \end{bmatrix}$$

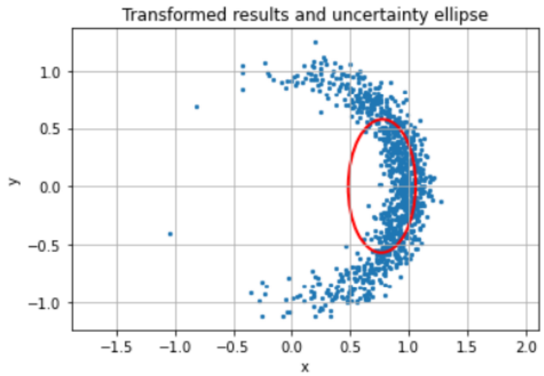

Figure 4: Monte Carlo simulation Case3

$$\boldsymbol{x} = \begin{bmatrix} \mathbf{1}\boldsymbol{m} \\ \mathbf{0.5}° \end{bmatrix}, \boldsymbol{\Sigma} = \begin{bmatrix} \mathbf{0.01} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} \end{bmatrix}$$

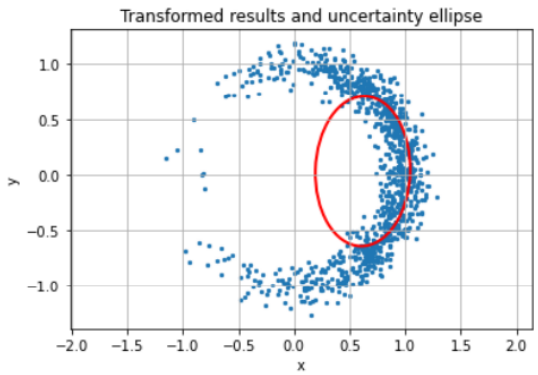

Figure 5: Monte Carlo simulation Case4

Detailed Code source can be obtained from
https://github.com/JunseoKim19/State_estimation/blob/main/MonteCarlo_Simulation_Cartesian.py