Assignment 4

Computational Intelligence, SS2017

Team Members		
Last name	First name	Matriculation Number
Gherman	Markus-Philipp	1431142
Moik	Fabian	1430095

Contents

1 Maximum Likelihood Estimation

1 Maximum Likelihood Estimation of Model Parameters

We assume that the true position - and the true distances - are known.

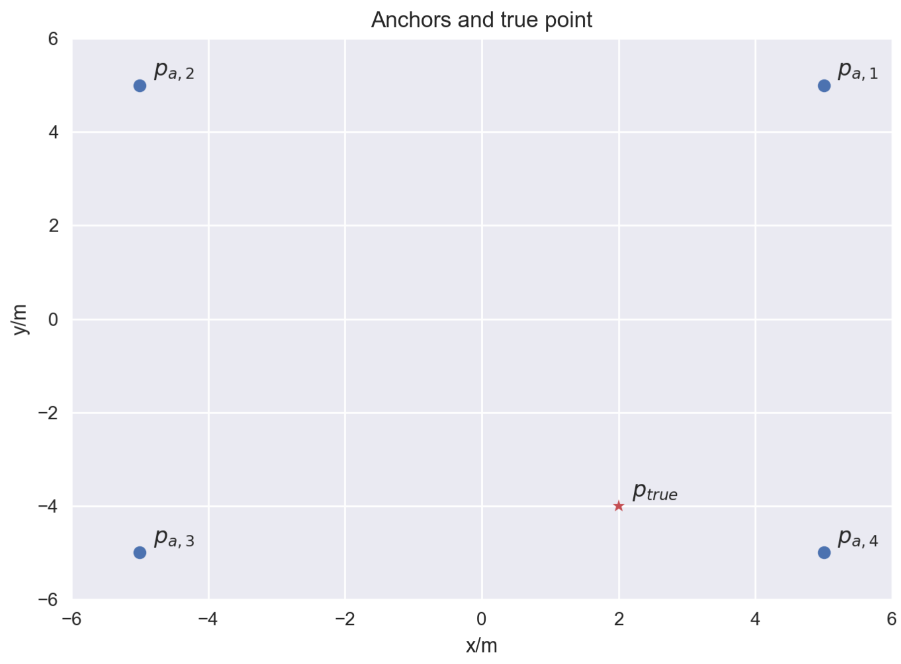


Figure 1.1: *Anchors and True point.*

Note that we increased the anchor indices by 1 for better understanding.

1.1 For scenario 2, find out which is the anchor with exponentially distributed measurements.

To find out which is the anchor with exponentially distributed measurements, we took all measurements for each anchor separately and plotted them in a distribution histogram.

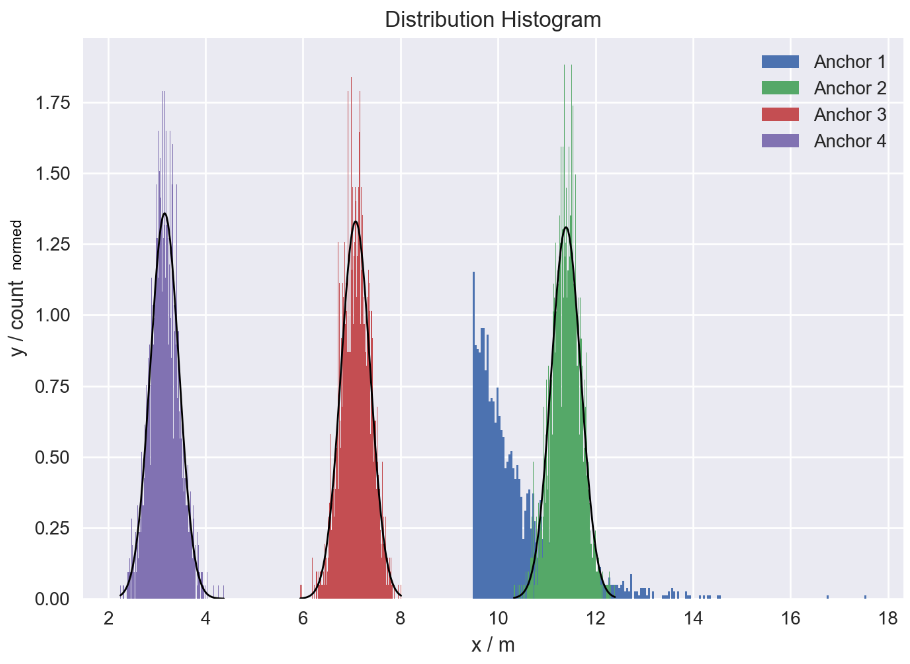


Figure 1.2: *Distribution Histogram of measurements per anchor for scenario 2.*

Using this histogram, we could conclude that Anchor 1 is the anchor with exponentially distributed measurements as seen in Figure 1.2

1.2 Analytically derive the maximum likelihood solution for the exponential distribution.

The maximum likelihood estimator is given as

$$\hat{\mathbf{p}}_{ML} = argmax \prod_{i=1}^{N} p(r_i|\mathbf{p})$$

and the exponential distribution is given as

$$p(r_i|\mathbf{p}) = \lambda_i \mathrm{e}^{-\lambda_i(r_i - d_i(\mathbf{p}))} \quad for \quad r_i \geq d_i(\mathbf{p}), \quad else \quad 0.$$

The analytical derivation of the maximum likelihood solution for the exponential distribution is performed as followed:

$$\hat{\mathbf{p}}_{ML} = argmax \prod_{i=1}^{N} p(r_i|\mathbf{p})$$
$$= argmax \quad \lambda_i^N \mathrm{e}^{-\lambda_i \sum\limits_{i=1}^{N} (r_i - d_i(\mathbf{p}))}$$

We now perform an ln operation, as ln is monoton rising, and therefor not changing anything for the argmax function.

$$\begin{aligned} ln(\hat{\mathbf{p}}_{ML}) &= argmax \quad ln(\lambda_i^N) + -\lambda_i \sum_{i=1}^{N}(r_i - d_i(\mathbf{p})) \\ &= argmax \quad N ln(\lambda_i) - \lambda_i \sum_{i=1}^{N}(r_i - d_i(\mathbf{p})) \end{aligned}$$

To find the optimal value for λ_i, we derive $ln(\hat{\mathbf{p}}_{ML})$ for λ_i. This derivation should equal 0, so that we get the maximum.

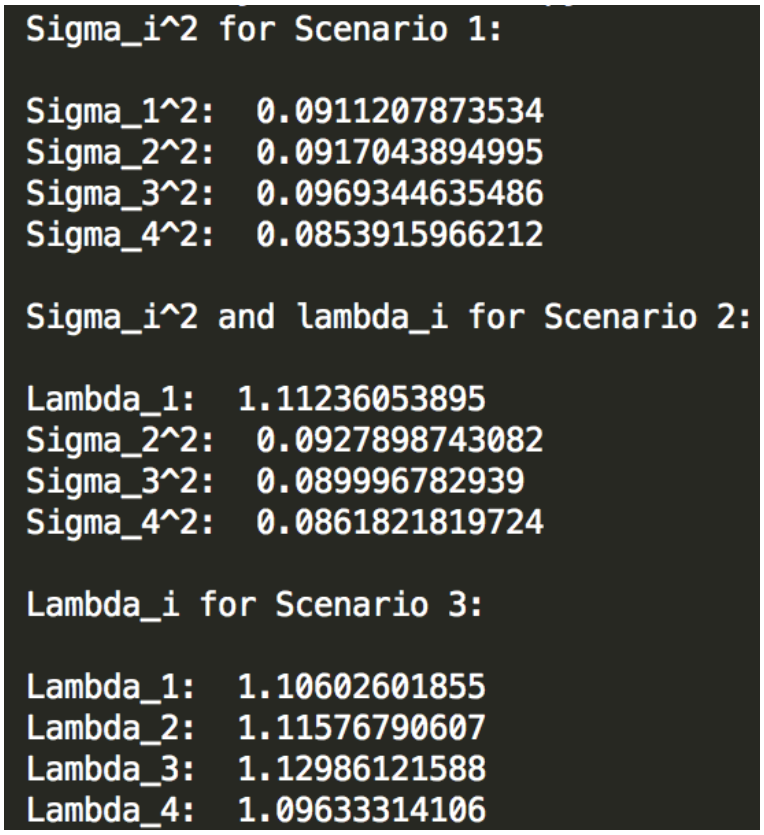
$$\frac{\partial ln(\hat{\mathbf{p}}_{ML})}{\partial \lambda_i} = \frac{N}{\lambda_i} - \sum_{i=1}^{N}(r_i - d_i(\mathbf{p})) \overset{!}{=} 0$$

From here we conclude our final solution for λ_i:

$$\lambda_i = \frac{N}{\sum\limits_{i=1}^{N}(r_i - d_i(\mathbf{p}))}$$

1.3 Estimate the parameters of the measurement models for all anchors in all 3 scenarios using the maximum likelihood method.

We estimated the parameters for all 3 scenarios and came to the following results:



```
Sigma_i^2 for Scenario 1:

Sigma_1^2:  0.0911207873534
Sigma_2^2:  0.0917043894995
Sigma_3^2:  0.0969344635486
Sigma_4^2:  0.0853915966212

Sigma_i^2 and lambda_i for Scenario 2:

Lambda_1:  1.11236053895
Sigma_2^2:  0.0927898743082
Sigma_3^2:  0.089996782939
Sigma_4^2:  0.0861821819724

Lambda_i for Scenario 3:

Lambda_1:  1.10602601855
Lambda_2:  1.11576790607
Lambda_3:  1.12986121588
Lambda_4:  1.09633314106
```

Figure 1.3: *Estimated parameters for all 3 scenarios.*

2 Least-Squares Estimation of the Position

2.1 Show analytically that for scenario 1 (joint likelihood for the distances is Gaussian), the least-squares estimator of the position is equivalent to the maximum likelihood estimator

The maximum likelihood estimator is given as:

$$\hat{\mathbf{p}}_{ML} = argmax \prod_{i=1}^{N} p(r_i|\mathbf{p})$$

We further perform an ln operation, as ln is monoton rising, and therefor not changing anything for the argmax function.

$$\begin{aligned}
\hat{\mathbf{p}}_{ML} &= argmax \prod_{i=1}^{N} p(r_i|\mathbf{p}) \\
&= argmax \sum_{i=1}^{N} ln(p(r_i|\mathbf{p})) \\
&= argmax \sum_{i=1}^{N} ln\left[\frac{1}{\sqrt{2\pi\sigma_i^2}} \mathrm{e}^{-\dfrac{(r_i - d_i(\mathbf{p}))^2}{2\sigma_i^2}}\right]
\end{aligned}$$

To continue, we use the following relation: $argmax f(x) = argmin - f(x)$

$$\begin{aligned}
\hat{\mathbf{p}}_{ML} &= argmax \sum_{i=1}^{N} ln\left[\frac{1}{\sqrt{2\pi\sigma_i^2}} \mathrm{e}^{-\dfrac{(r_i - d_i(\mathbf{p}))^2}{2\sigma_i^2}}\right] \\
&= argmin - \sum_{i=1}^{N} \left[ln\left(\frac{1}{\sqrt{2\pi\sigma_i^2}}\right) + ln\left(\mathrm{e}^{-\dfrac{(r_i - d_i(\mathbf{p}))^2}{2\sigma_i^2}}\right)\right] \\
&= argmin - \sum_{i=1}^{N} \left[ln\left(\frac{1}{\sqrt{2\pi\sigma_i^2}}\right) - \frac{(r_i - d_i(\mathbf{p}))^2}{2\sigma_i^2}\right]
\end{aligned}$$

Because a constant does not change anything for the argmin function, because it is changing every argument in a linear way, we can neglect the constant:

$$
\begin{aligned}
\hat{\mathbf{p}}_{ML} &= argmin - \sum_{i=1}^{N} \left[ln\left(\frac{1}{\sqrt{2\pi\sigma_i^2}} \right) - \frac{(r_i - d_i(\mathbf{p}))^2}{2\sigma_i^2} \right] \\
&\approx argmin - \sum_{i=1}^{N} \left[- \frac{(r_i - d_i(\mathbf{p}))^2}{2\sigma_i^2} \right] \\
&\approx argmin \sum_{i=1}^{N} \left[\frac{(r_i - d_i(\mathbf{p}))^2}{2\sigma_i^2} \right]
\end{aligned}
$$

Now, the constant can be again neglected, as it is not influencing the argmin function, and we can make the final step:

$$
\hat{\mathbf{p}}_{ML} \approx argmin \sum_{i=1}^{N} (r_i - d_i(\mathbf{p}))^2 \approx \hat{\mathbf{p}}_{LS}
$$

As one can see, the least-squares estimator of the position is equivalent to the maximum likelihood estimator (for the gaussian model).

2.2 Implement the Gauss-Newton algorithm to find the least-squares estimate for the position and write a function LeastSquaresGN. You may choose suitable values for the tolerance and the maximum number of iterations on your own. The output is the estimated position.

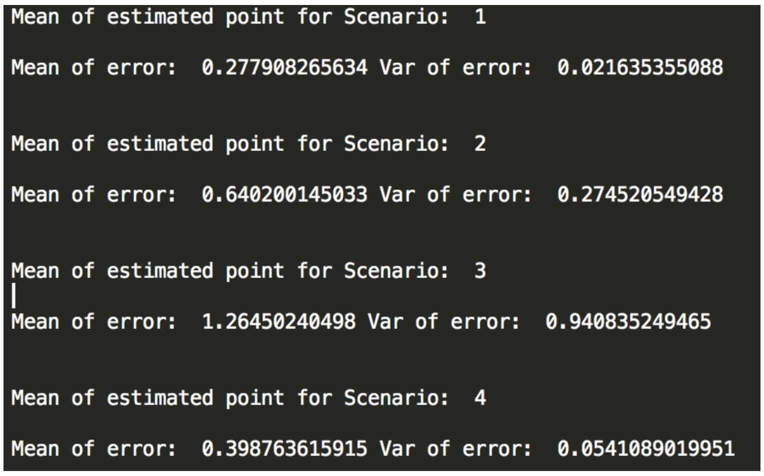
The functionality of the written function will be verified and used in Point 2.3. We set max_iter to 20 and tol to 10^{-3}. For the Jacobi Matrix, the following derivates were used:

$$
\frac{\partial(r_i - d_i(\mathbf{p}))}{\partial x} = \frac{x_i - x}{\sqrt{(y_i - y)^2 + (x_i - x)^2}}
$$

$$
\frac{\partial(r_i - d_i(\mathbf{p}))}{\partial y} = \frac{y_i - y}{\sqrt{(y_i - y)^2 + (x_i - x)^2}}
$$

2.3 For all three scenarios, evaluate your estimation algorithm using the provided data. For each of the N = 2000 independent measurements, choose the starting position p0 randomly according to a uniform distribution within the square spanned by the anchor points.

The mean and variance of the position estimation error.



```
Mean of estimated point for Scenario:  1

Mean of error:  0.277908265634 Var of error:  0.021635355088


Mean of estimated point for Scenario:  2

Mean of error:  0.640200145033 Var of error:  0.274520549428


Mean of estimated point for Scenario:  3

Mean of error:  1.26450240498 Var of error:  0.940835249465


Mean of estimated point for Scenario:  4

Mean of error:  0.398763615915 Var of error:  0.0541089019951
```

Figure 2.1: Mean and variance of the position estimation error for 4 scenarios, where the forth scenario only uses 3 gaussian anchors (exponential anchor was deleted).

Scatter plots of the estimated positions. Fit a two-dimensional Gaussian distribution to the point cloud of estimated positions and draw its contour lines. Do the estimated positions look Gaussian? What can you say about the probability of large estimation errors?

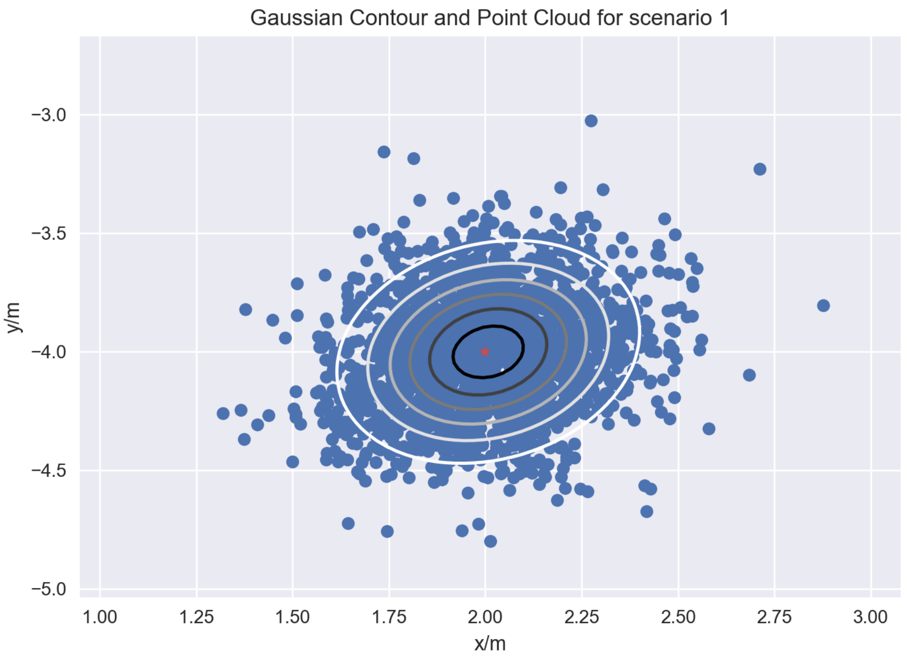


Figure 2.2: Gaussian contour and point cloud for scenario 1.

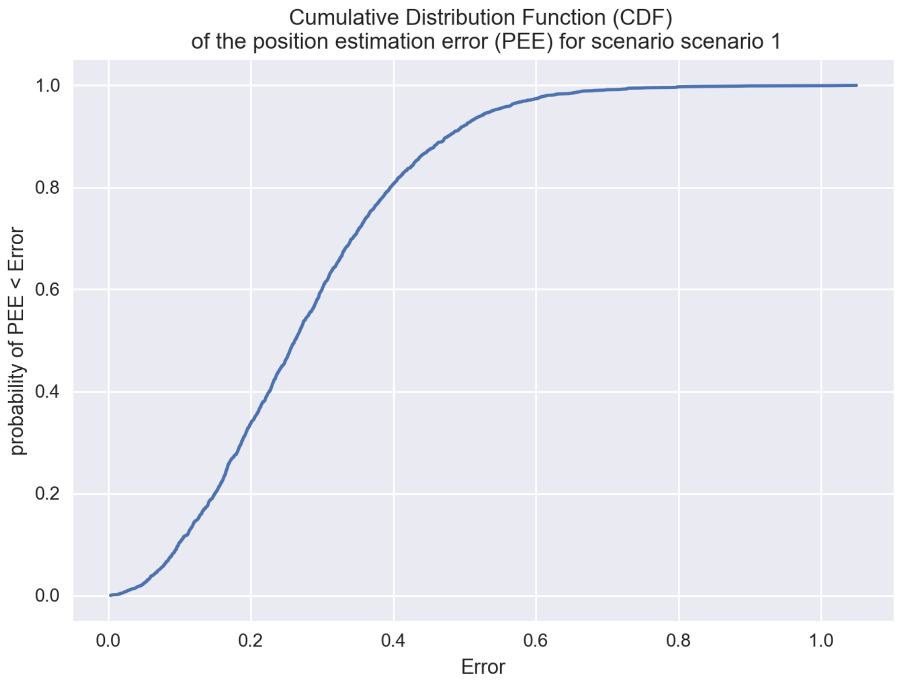


Figure 2.3: *Cumulative Distribution Function of the position estimation error for scenario 1.*

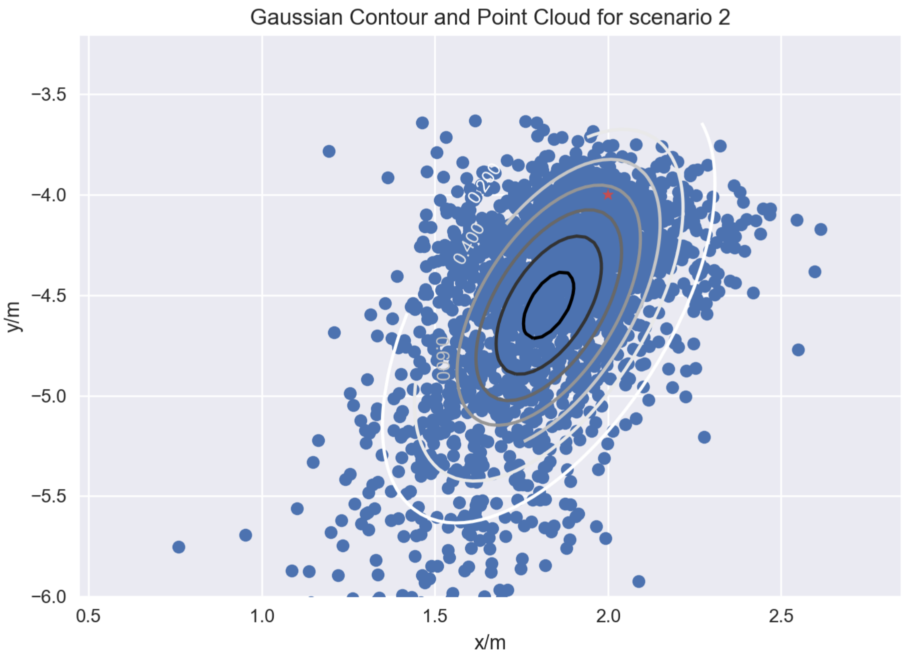


Figure 2.4: *Gaussian contour and point cloud for scenario 2.*

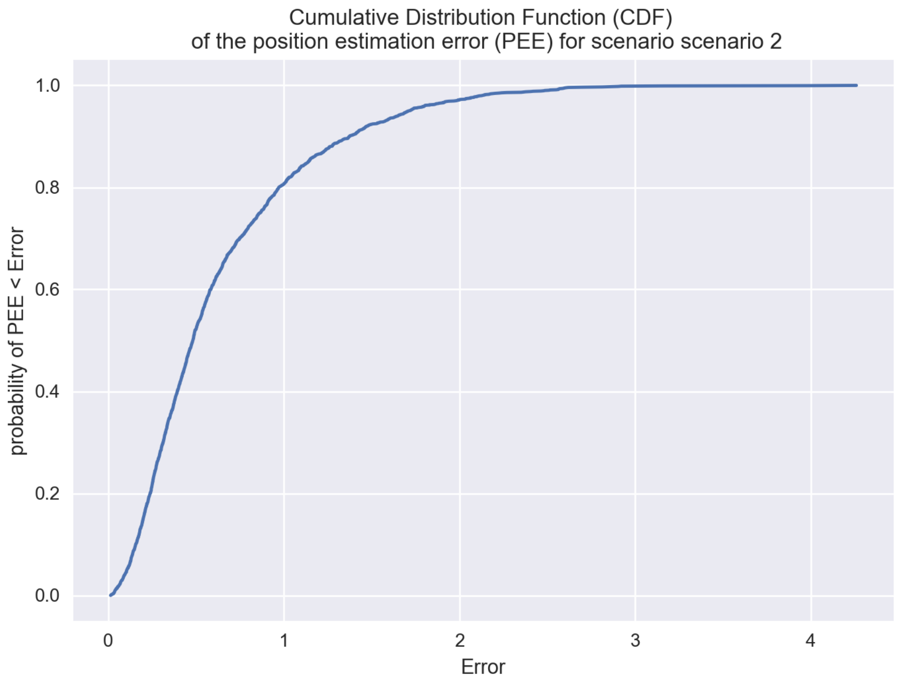


Figure 2.5: *Cumulative Distribution Function of the position estimation error for scenario 2.*

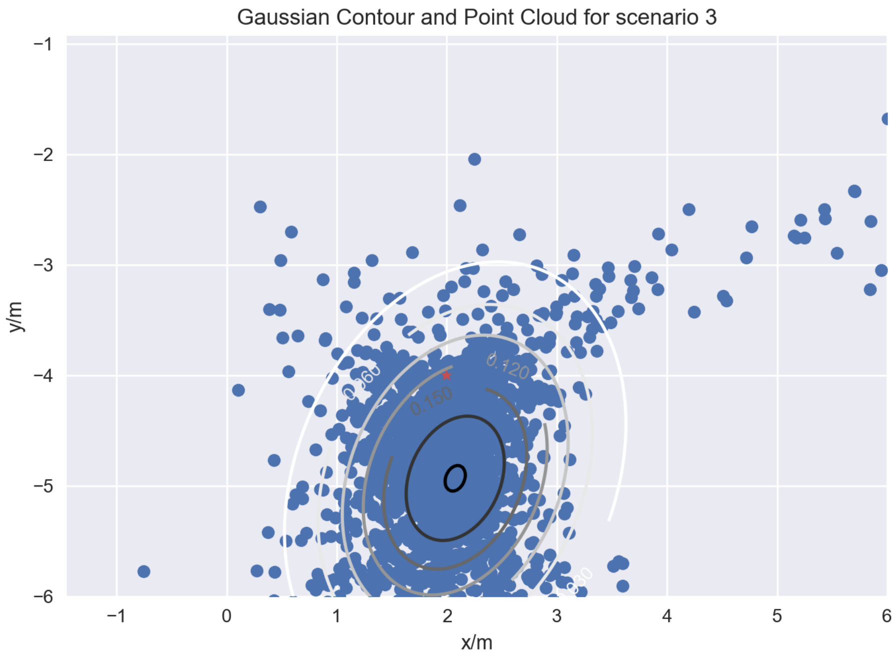


Figure 2.6: *Gaussian contour and point cloud for scenario 3.*

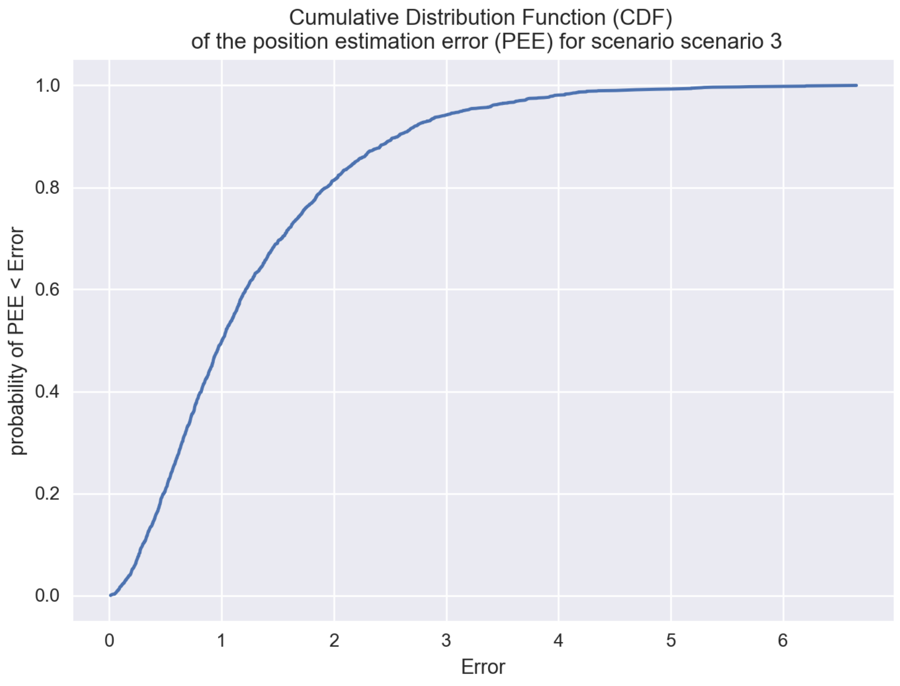


Figure 2.7: *Cumulative Distribution Function of the position estimation error for scenario 3.*

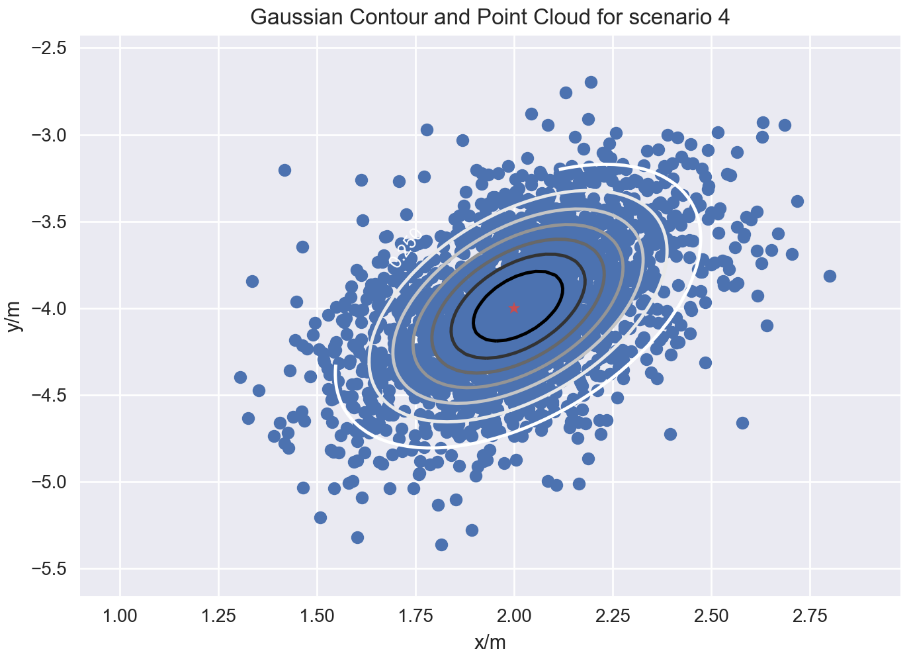


Figure 2.8: *Gaussian contour and point cloud for scenario 4.*

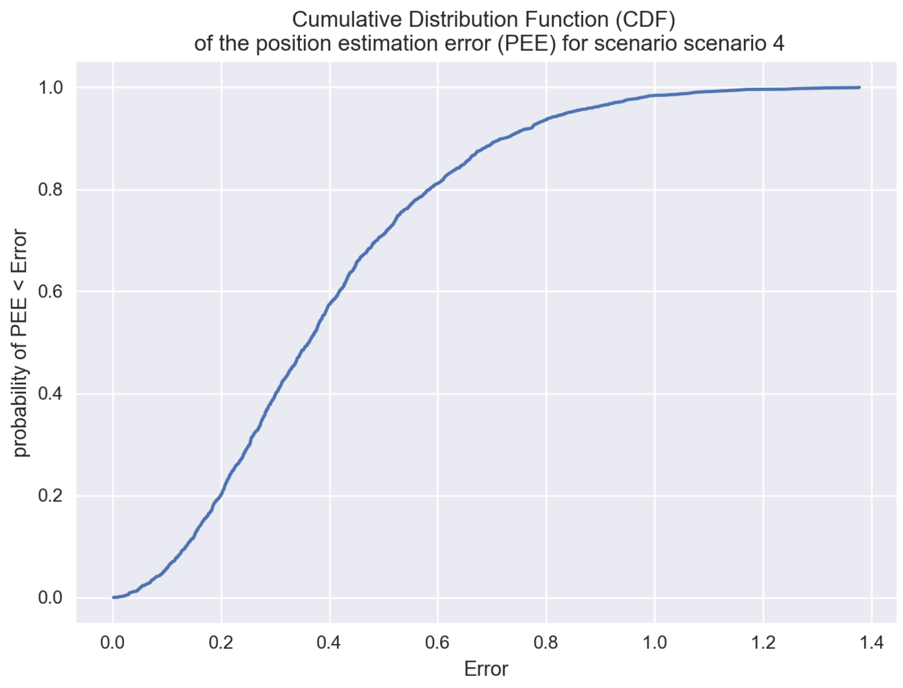


Figure 2.9: *Cumulative Distribution Function of the position estimation error for scenario 4.*

As one can see in the plots, the estimated points for scenario 1 and 4 look gaussian. This is because there are only gaussian anchors. The estimated points for scenario 2 and 3 do not really look gaussian, as there are a lot of points far away from the gaussian contour.

By looking at the CDF plots, we can see, that for scenario 1 and 4 (gaussian anchors) the probability for the error being smaller than 0.8 and 1.4 respectively is nearly 100%. For scenario 2 and 3 , where we have at least one exponential anchor, we can say with a high probability that our position estimation error is much greater than in the gaussian examples.

The following plot shows then mean of the estimated point cloud for each scenario.

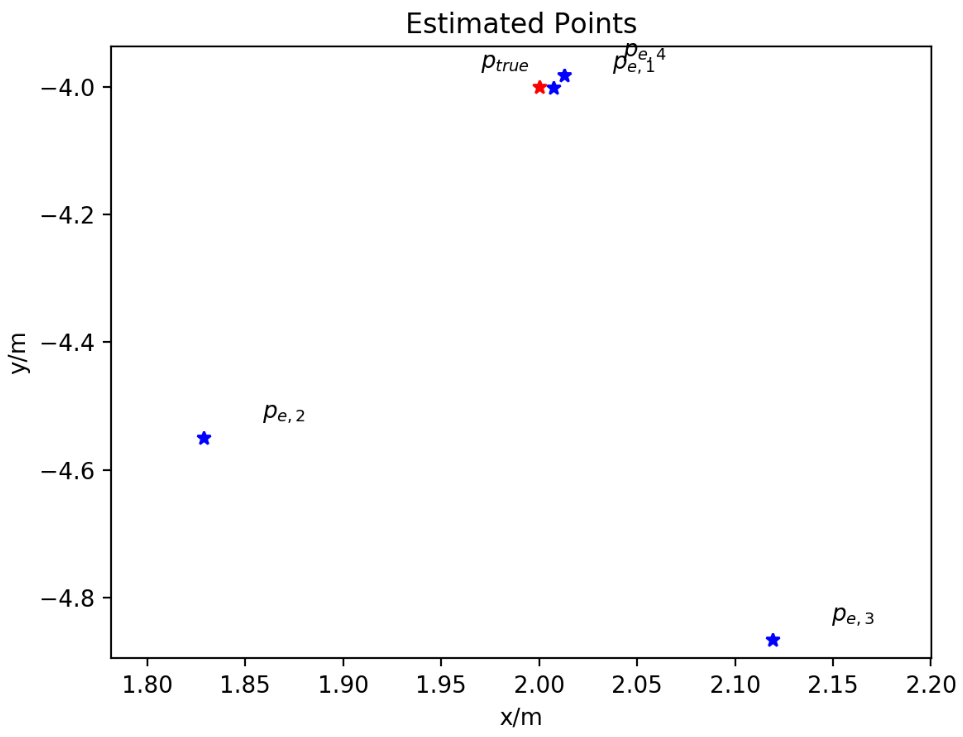


Figure 2.10: *Mean estimated points for each scenario (magnified).*

2.4 Compare the performance of scenario 2 with scenario 4. What can you observe?

By looking at our plots, we conclude that we get a lower position estimation error for scenario 4 compared to scenario 2. This effect can be seen again in Figure 2.1, where scenario 4 performs better because it has a lower mean and variance of estimation errors.

3 Numerical Maximum-Likelihood Estimation of the Position

3.1 For the first measurement (i.e. the first NA distance estimates), compute the joint likelihood function over a two dimensional grid with a resolution of 5 cm. Confine the evaluation to the square region that is enclosed by the anchors. Why might it be hard to find the maximum of this function with a gradient ascent algorithm using an arbitrary starting point within the evaluation region? Is the maximum at the true position?

First, we computed the joint likelihood function

$$\hat{\mathbf{p}}_{ML} = argmax \prod_{i=1}^{N} \lambda_i \mathrm{e}^{-\lambda_i (r_i - d_i(\mathbf{p}))}$$

and then we evaluated it for the first measurement of our given data.

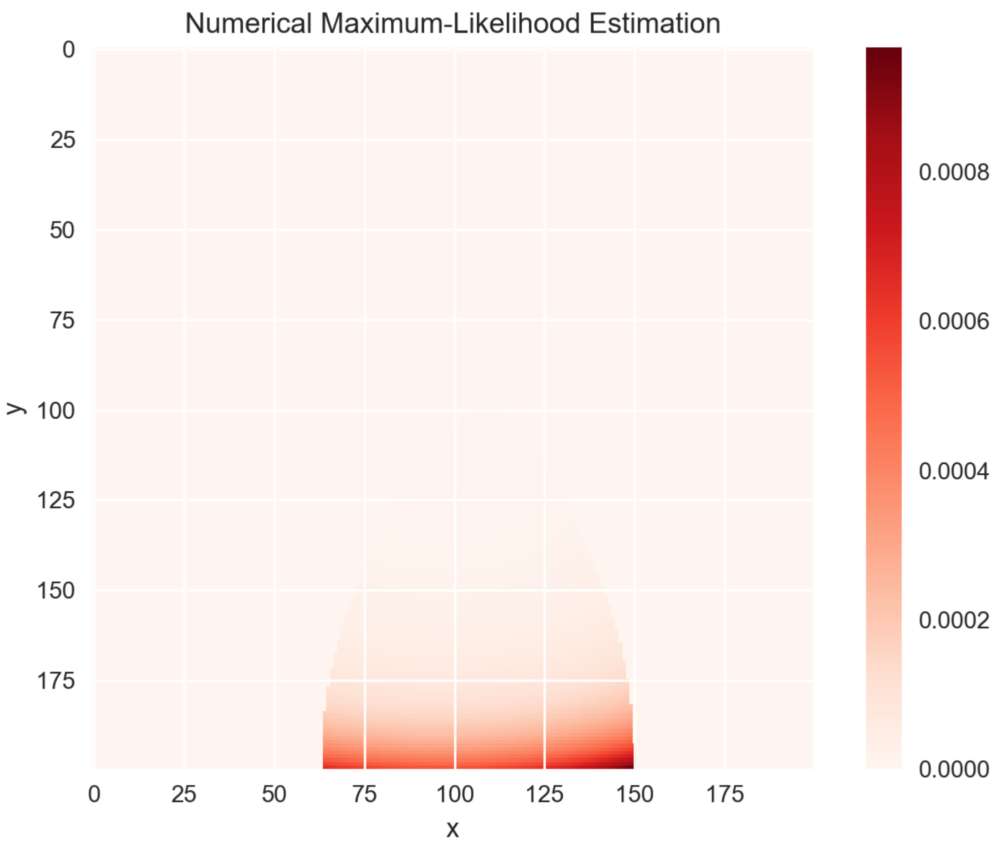


Figure 3.1: *Numerical maximum likelihood estimation.*

In this Figure we can see that the numerical maximum likelihood estimator is a non-convex function and therefore a gradient ascent algorithm would get stuck in local maxima quiet easily. The maximum is not at the true position (2,-4), instead it is at (2.5, -4.95).