POLITECNICO
MILANO 1863

DIPARTIMENTO DI ELETTRONICA
INFORMAZIONE E BIOINGEGNERIA

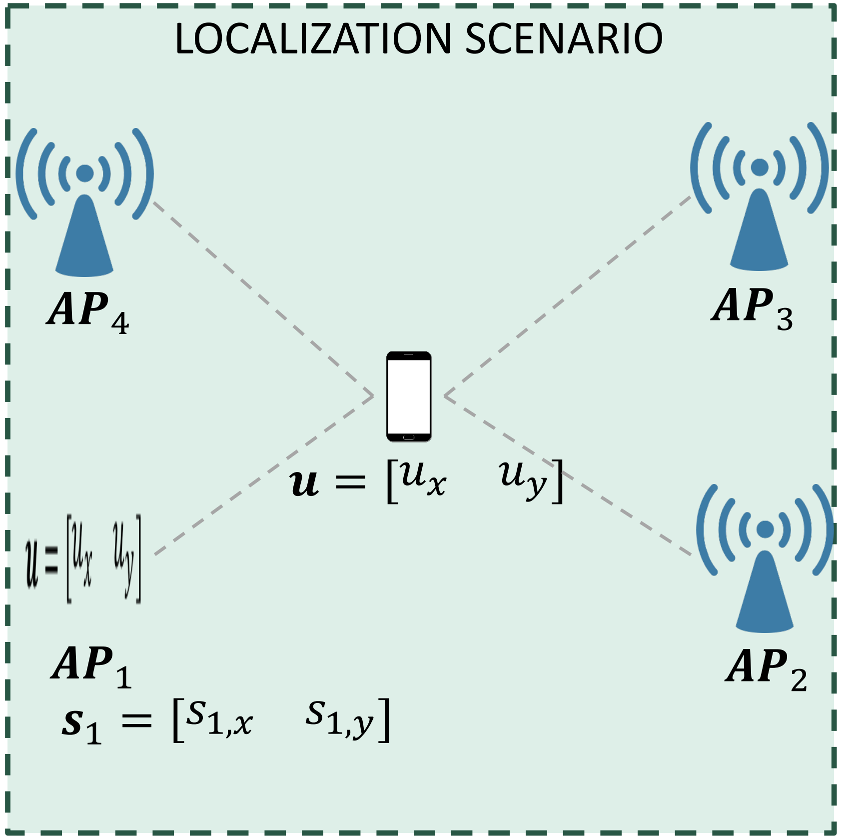
DEIB

Mattia Brambilla
mattia.brambilla@polimi.it

Lab 2: STATIC LOCALIZATION
Cramér-Rao Bound and iterative NLS

2D localization: problem formulation

GOAL: localize an unknown user (UE) $\boldsymbol{u}$ from a set of measurements $\boldsymbol{\rho}$ available at N_{AP} Access Points (APs).



$$\boldsymbol{\rho} = [\rho_1\ \rho_2\ \rho_3\ \rho_4]$$

The single measurement $\rho_i = h_i(\boldsymbol{u}, \boldsymbol{s}_i) + n_i$ is a non-linear function of the AP/UE states, corrupted by noise

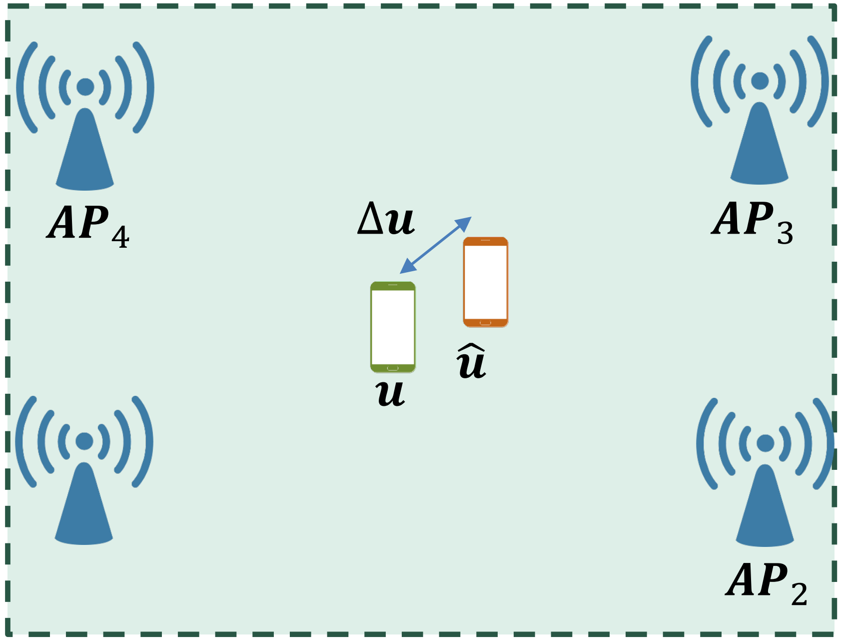
non-linear function states noise

CRB

Localization accuracy: definition

Given a UE **true** position $\boldsymbol{u}$ and its **estimate** $\hat{\boldsymbol{u}}$, the accuracy is defined as the error of the location estimate:

$$\Delta \boldsymbol{u} = \boldsymbol{u} - \hat{\boldsymbol{u}}$$



The Root Mean Square Error (RMSE) of the location estimate is:

$$\sigma_p = \sqrt{E[\Delta \boldsymbol{u}^2]} = \sqrt{\sigma_x^2 + \sigma_y^2} = \sqrt{\mathrm{tr}(\mathbf{C})}$$ (if unbiased estimator)

covariance matrix

Localization accuracy: lower bound

The covariance of the estimate $(\mathbf{C})$ is lower bounded by the Cramér-Rao bound $\boldsymbol{C}_{\mathrm{CRB}}$.

The CRB is the inverse of the Fisher Information Matrix (FIM).

The FIM is defined as:

$$\mathbf{I}(\boldsymbol{u}) = \mathbf{H}^{\mathrm{T}}(\boldsymbol{u})\mathbf{R}^{-1}\mathbf{H}(\boldsymbol{u}) \qquad \text{(Gaussian measurements)}$$

with

$$\mathbf{H}(\boldsymbol{u}) = \frac{\partial \mathbf{h}(\boldsymbol{u})}{\partial \boldsymbol{u}}$$

Thus $\mathbf{C} \geq \mathbf{C}_{\mathrm{CRB}} = \mathbf{I}^{-1}(\boldsymbol{u}) = \left(\mathbf{H}^{\mathrm{T}}(\boldsymbol{u})\mathbf{R}^{-1}\mathbf{H}(\boldsymbol{u})\right)^{-1}$

CRB: Gaussian measurements

For Gaussian measurements:

$$\mathbf{n} \sim \mathcal{N}(\mathbf{0},\mathbf{R}) \rightarrow \boldsymbol{\rho} \sim \mathcal{N}(\mathbf{h}(\mathbf{u}),\mathbf{R})$$

we get:

$$\mathbf{I}(\mathbf{u}_0) = \mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}$$

$$\mathbf{C}_{\mathrm{CRB}} = \left(\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}\right)^{-1}$$

where **H** is the NxM Jacobian matrix:

$$\mathbf{H} = \left.\frac{\partial \mathbf{h}(\mathbf{u})}{\partial \mathbf{u}}\right|_{\mathbf{u}=\mathbf{u}_0} = \left.\begin{bmatrix} \frac{\partial h_1(\mathbf{u})}{\partial u_1} & \cdots & \frac{\partial h_1(\mathbf{u})}{\partial u_M} \\ \vdots & \ddots & \vdots \\ \frac{\partial h_N(\mathbf{u})}{\partial u_1} & \cdots & \frac{\partial h_N(\mathbf{u})}{\partial u_M} \end{bmatrix}\right|_{\mathbf{u}=\mathbf{u}_0}$$

Proof:

$$\mathcal{L}(\boldsymbol{\rho}\,|\,\mathbf{u}) = -\frac{1}{2}\ln|\mathbf{R}| - \frac{1}{2}(\boldsymbol{\rho}-\mathbf{h}(\mathbf{u}))^T\mathbf{R}^{-1}(\boldsymbol{\rho}-\mathbf{h}(\mathbf{u}))$$

$$\frac{\partial \mathcal{L}(\boldsymbol{\rho}\,|\,\mathbf{u})}{\partial \mathbf{u}} = \mathbf{H}^T\mathbf{R}^{-1}(\boldsymbol{\rho}-\mathbf{h}(\mathbf{u}))$$

$$\mathbf{I}(\mathbf{u}) = E_\rho\left[\frac{\partial \mathcal{L}(\boldsymbol{\rho}\,|\,\mathbf{u})}{\partial \mathbf{u}}\frac{\partial \mathcal{L}(\boldsymbol{\rho}\,|\,\mathbf{u})^T}{\partial \mathbf{u}}\right] = \mathbf{H}^T\mathbf{R}^{-1}E_{\boldsymbol{\rho}}\left[(\boldsymbol{\rho}-\mathbf{h}(\mathbf{u}))(\boldsymbol{\rho}-\mathbf{h}(\mathbf{u}))^T\right]\mathbf{R}^{-1}\mathbf{H} = \mathbf{H}^T\mathbf{R}^{-1}\mathbf{R}\mathbf{R}^{-1}\mathbf{H}$$

Method	$[\mathbf{H}(\boldsymbol{u})]_i = \frac{\partial h_i(\boldsymbol{u})}{\partial \boldsymbol{u}}$
TOA	$\frac{u_x - s_{i,x}}{d_i}, \frac{u_y - s_{i,y}}{d_i}$
AOA	$-\frac{u_y - s_{i,y}}{{d_i}^2}, \frac{u_x - s_{i,x}}{{d_i}^2}$
RSS	$-\frac{10 n_p}{\ln 10}\frac{u_x - s_{i,x}}{{d_i}^2}, -\frac{10 n_p}{\ln 10}\frac{u_y - s_{i,y}}{{d_i}^2}$
TDOA	$\frac{u_x - s_{i,x}}{d_i} - \frac{u_x - s_{j,x}}{d_j}, \frac{u_y - s_{i,y}}{d_i} - \frac{u_y - s_{j,y}}{d_j}$

Geometric dilution of precision (GDOP)

Recalling

$$\mathbf{C}_{\mathrm{CRB}} = \mathbf{I}^{-1}(\boldsymbol{u})$$

$$\mathbf{C}_{\mathrm{CRB}} = \left(\mathbf{H}^{\mathrm{T}}(\boldsymbol{u})\mathbf{R}^{-1}\mathbf{H}(\boldsymbol{u})\right)^{-1}$$

if measurements are uncorrelated and with a same error σ_n

$$\mathbf{C}_{\mathrm{CRB}} = \sigma_n^2 \left(\mathbf{H}^{\mathrm{T}}(\boldsymbol{u})\, \mathbf{H}(\boldsymbol{u})\right)^{-1}$$

measurement accuracy

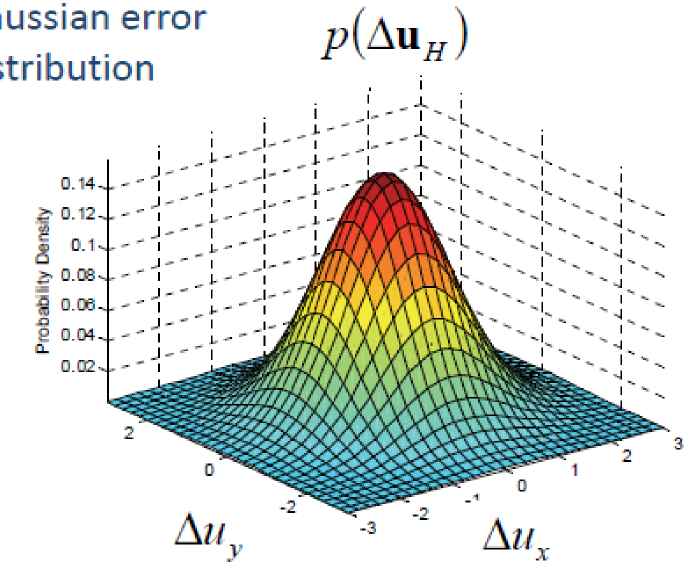
GDOP

$$\mathbf{G} = \left(\mathbf{H}^{\mathrm{T}}(\boldsymbol{u})\, \mathbf{H}(\boldsymbol{u})\right)^{-1}$$

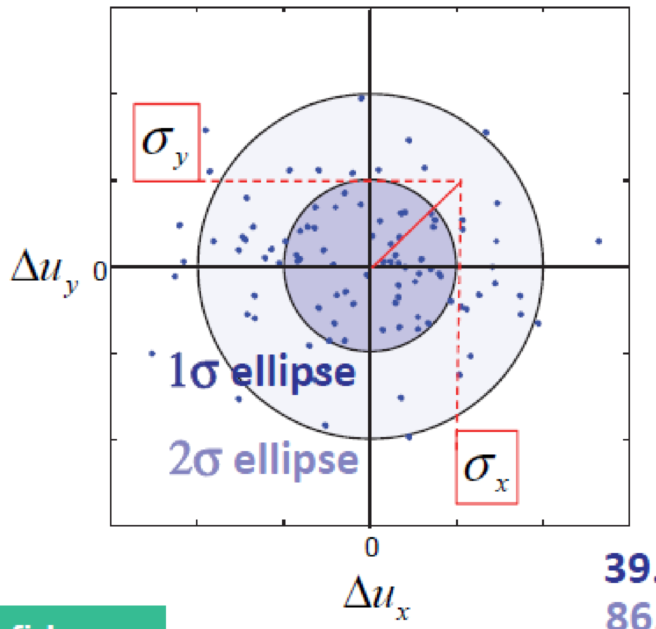
it describes the amplification of measurement error due to geometry

From the lecture notes

Assumption:
Gaussian error distribution



Error regions



39.35%
86.47%

Error ellipse	Confidence level
1σ	39.35%
2σ	86.47%
3σ	98.89%
4σ	99.97%

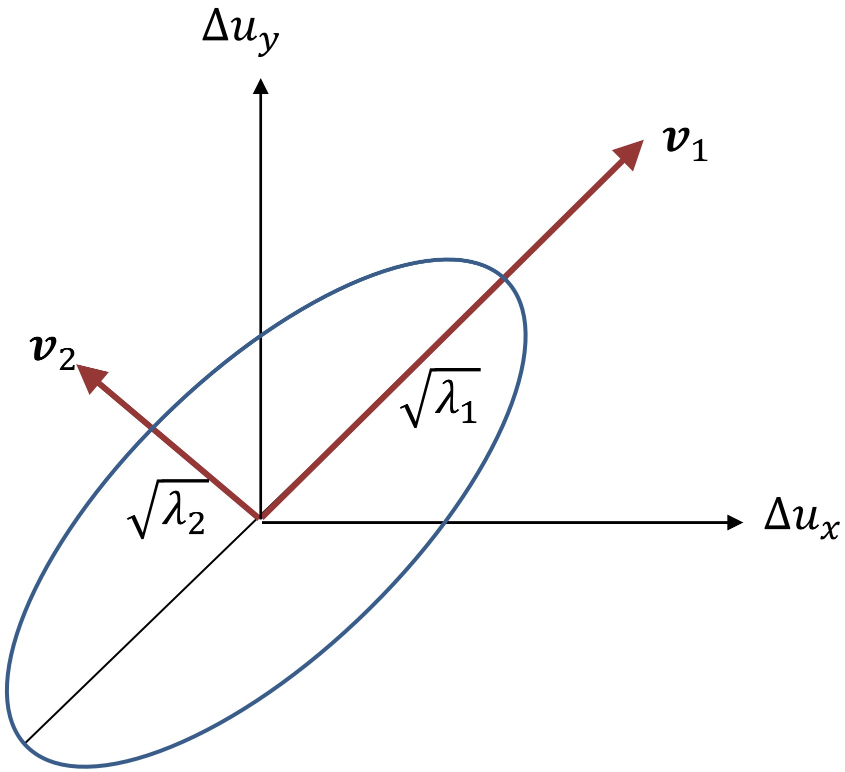
Error ellipse	Confidence level
1.18σ	50%
1.79σ	80%
2.15σ	90%
2.45σ	95%

For $k\sigma$ ellipse:

$$P = F_{\chi_2^2}(k^2)$$

$$k = \sqrt{F_{\chi_2^2}^{-1}(P)}$$

Error ellipse



$$\mathbf{C} = \begin{bmatrix} \sigma_x^2 & C_{xy} \\ C_{xy} & \sigma_y^2 \end{bmatrix}$$

How to get the orientation from $\mathbf{C}$: extract eigenvalues and corresponding eigenvector $\Rightarrow$ SVD in Matlab

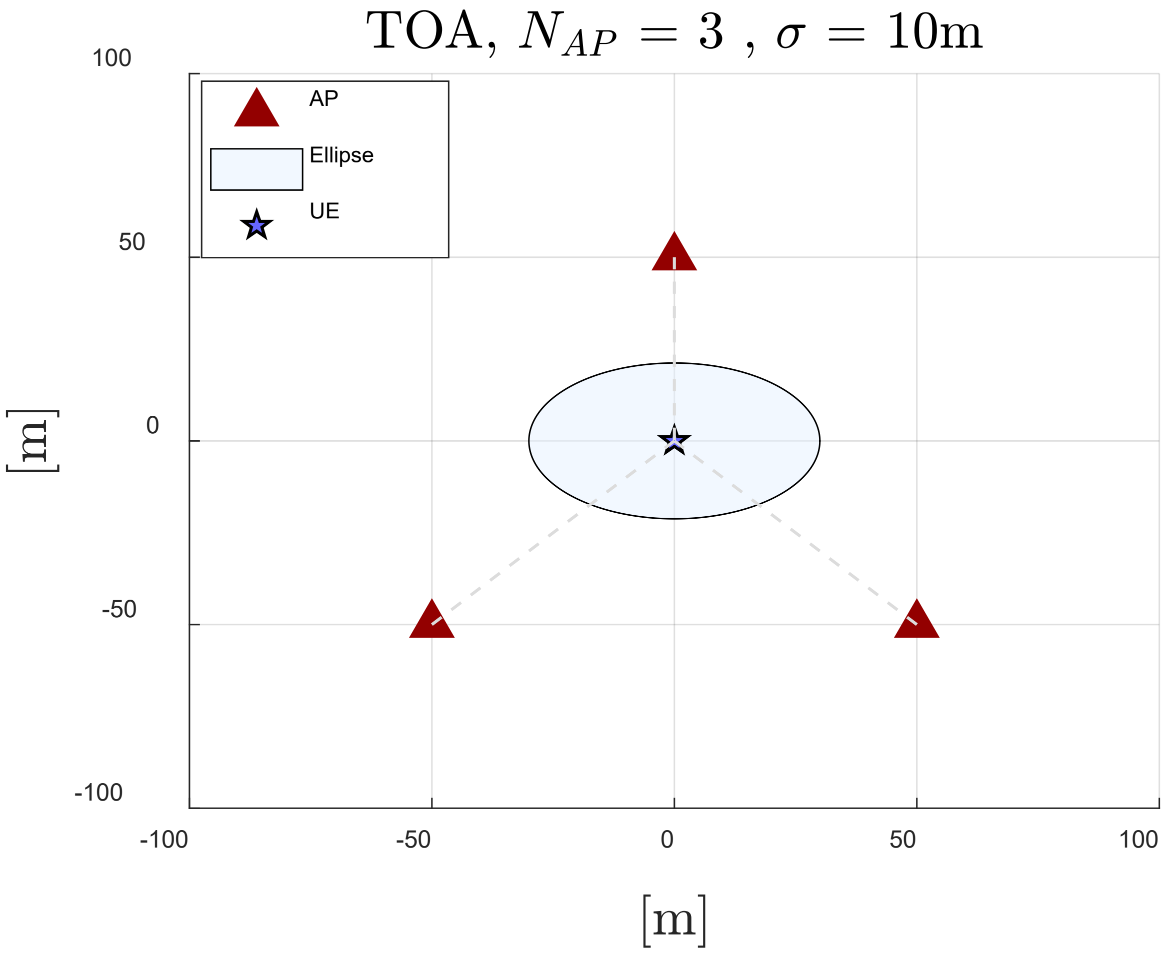
The axes of the ellipse are oriented as the eigenvectors, with length equal to the square root of eigenvalues.

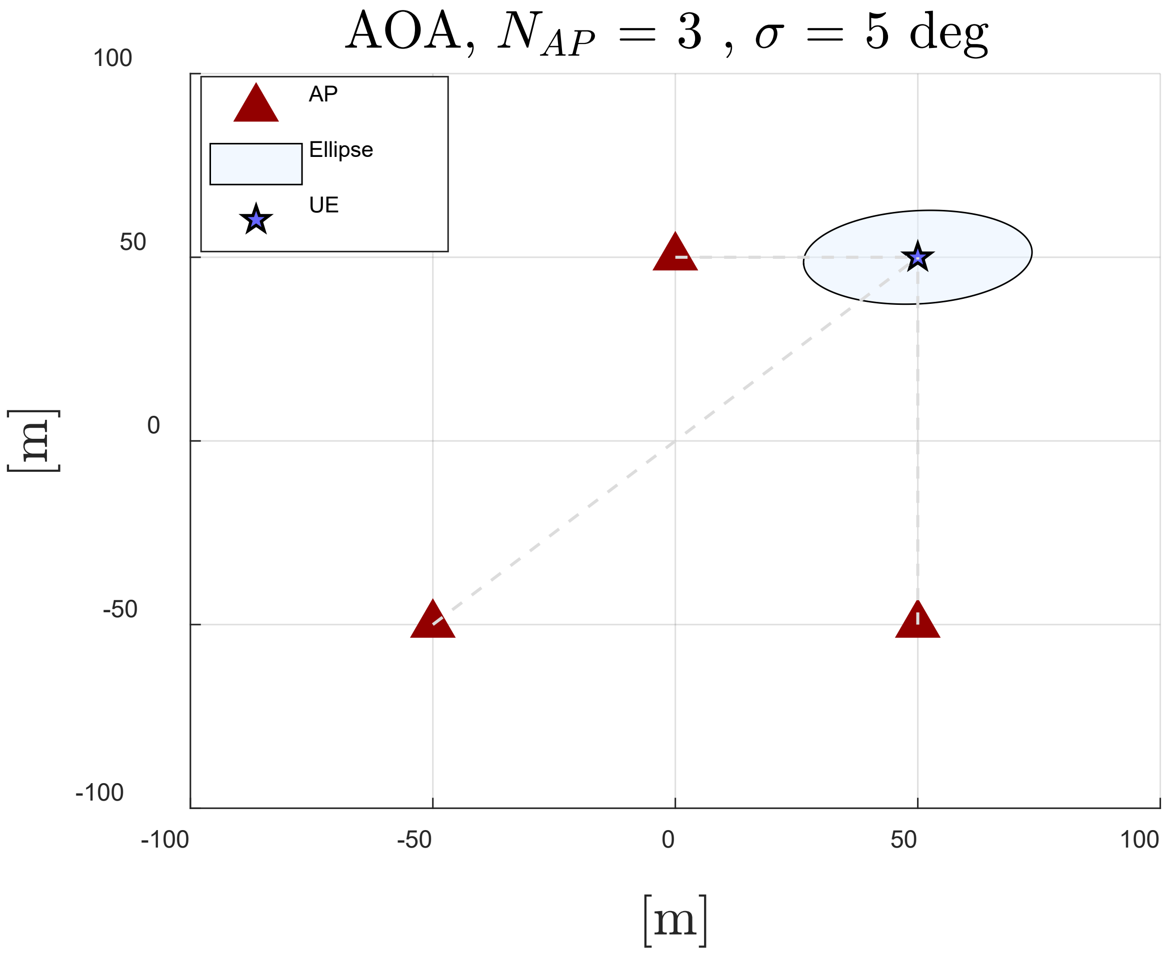
To do

1. Define the localization scenario (same as Lab 1)
2. Generate noisy measurements
3. Build covariance matrix $\mathbf{R}$
4. Build Jacobian matrix $\mathbf{H}(\boldsymbol{u})$
5. Calculate and plot the CRB ellipse

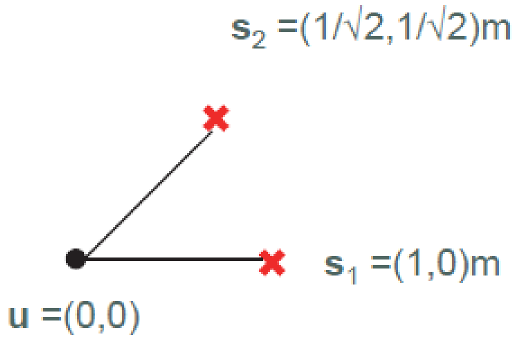
	Measurement	Jacobian
Method	$h_i(\boldsymbol{u})$	$[\mathbf{H}(\boldsymbol{u})]_i = \frac{\partial h_i(\boldsymbol{u})}{\partial \boldsymbol{u}}$
TOA	$d_i = \lvert \boldsymbol{u} - \boldsymbol{s}_i \rvert$	$\frac{u_x - s_{i,x}}{d_i}, \frac{u_y - s_{i,y}}{d_i}$
AOA	$\tan^{-1}\left(\frac{u_y - s_{i,y}}{u_x - s_{i,x}}\right)$	$-\frac{u_y - s_{i,y}}{{d_i}^2}, \frac{u_x - s_{i,x}}{{d_i}^2}$

Examples of expected results

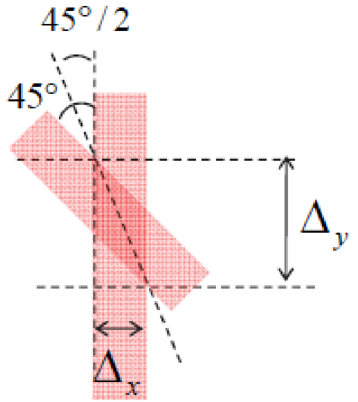




Example of the lecture notes



$\sigma_n = 1\text{cm}$



$$a_{x1} = \frac{s_{1x} - u_x}{|\mathbf{s}_1 - \mathbf{u}|} = 1$$

$$a_{y1} = \frac{s_{1y} - u_y}{|\mathbf{s}_1 - \mathbf{u}|} = 0$$

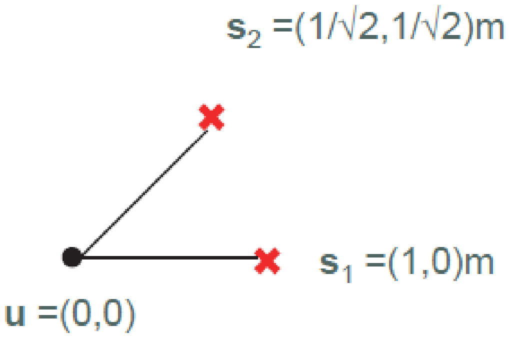
$$a_{x2} = \frac{s_{2x} - u_x}{|\mathbf{s}_2 - \mathbf{u}|} = \frac{1/\sqrt{2}}{1} = \frac{1}{\sqrt{2}}$$

$$a_{y2} = \frac{s_{2y} - u_y}{|\mathbf{s}_1 - \mathbf{u}|} = \frac{1/\sqrt{2}}{1} = \frac{1}{\sqrt{2}}$$

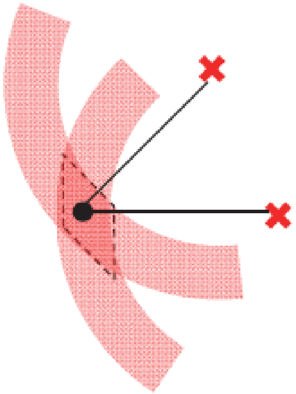
$$\Delta_x = \sigma_n$$

$$\Delta_y > \Delta_x$$

Example of the lecture notes



$\sigma_n = 1\text{cm}$



$$\mathbf{H} = \begin{bmatrix} -1 & 0 \\ -1/\sqrt{2} & -1/\sqrt{2} \end{bmatrix}$$

$$\mathbf{H}^T\mathbf{H} = \begin{bmatrix} -1 & -1/\sqrt{2} \\ 0 & -1/\sqrt{2} \end{bmatrix} \begin{bmatrix} -1 & 0 \\ -1/\sqrt{2} & -1/\sqrt{2} \end{bmatrix} = \begin{bmatrix} 3/2 & 1/2 \\ 1/2 & 1/2 \end{bmatrix} = \frac{1}{2}\begin{bmatrix} 3 & 1 \\ 1 & 1 \end{bmatrix}$$

$$\mathbf{G} = \left(\mathbf{H}^T\mathbf{H}\right)^{-1} = 2\begin{bmatrix} 1 & -1 \\ -1 & 3 \end{bmatrix}\frac{1}{3-1} = \begin{bmatrix} 1 & -1 \\ -1 & 3 \end{bmatrix} \qquad D_x = 1;\ D_y = 3$$

$$\mathbf{C} = \sigma_n^2\begin{bmatrix} 1 & -1 \\ -1 & 3 \end{bmatrix} \qquad \sigma_x = 1\text{cm};\ \sigma_y = 1.7\text{cm}$$

Example of the lecture notes

The semi-axes of the ellipse have length $\{\lambda_1,\lambda_2\}$ with $\{\lambda_1^2, \lambda_2^2\}$=eig [**C**].

Eigenvalue computation:

$$\mathbf{C} = \sigma_n^2 \begin{bmatrix} 1 & -1 \\ -1 & 3 \end{bmatrix}$$

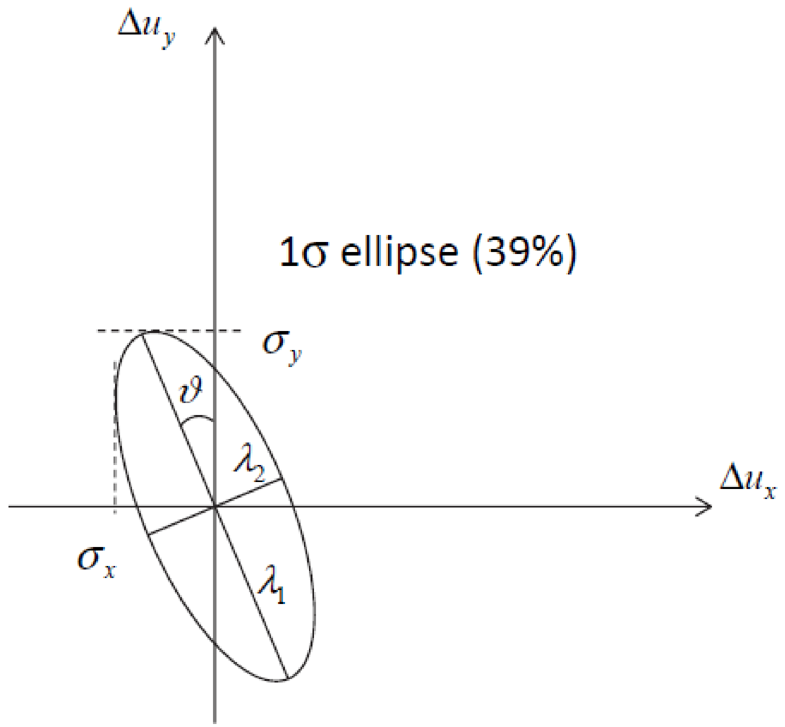
$$\det(\mathbf{C} - \delta \mathbf{I}) = 0$$

$$\det\left(\begin{bmatrix} 1-\delta & -1 \\ -1 & 3-\delta \end{bmatrix}\right) = 0$$

$$(1-\delta)(3-\delta) - 1 = 0$$

$$\delta_1 = 3.41 \rightarrow \text{ major semiaxis } \lambda_1 = \sqrt{3.41} = 1.8$$

$$\delta_2 = 0.58 \rightarrow \text{ minor semiaxis } \lambda_2 = \sqrt{0.58} = 0.7$$



Angle of rotation of the ellipse:

$$\vartheta = \frac{1}{2}\arctan\frac{2C_{xy}}{\sigma_x^2 - \sigma_y^2} = \frac{1}{2}\arctan\frac{-2}{1-3} = \frac{45}{2}\text{deg} = 22.5\,\text{deg}$$

ITERATIVE METHODS

Iterative Methods

We want to estimate the position $\hat{\boldsymbol{u}}$ from a set of measurements $\boldsymbol{\rho} = \mathbf{h}(\boldsymbol{u}) + \boldsymbol{n}$ of N_{AP}

$$\hat{\boldsymbol{u}} = argmin\,|\boldsymbol{\rho} - \mathbf{h}(\boldsymbol{u})|^2$$

$$= argmin\,\sum_{i=1}^{N_{AP}} \left(\rho_i - \mathrm{h}_i(\boldsymbol{u})\right)^2$$

Numerical search algorithms: iterative NLS

11

- In the general case, there is no closed-form solution to the non-linear localization problem.
- Numerical search methods are used. A good initialization is required to avoid convergence to a local minimum of the loss function F(**u**).
- In what follows we consider the iterative search of the NLS solution.

Iterative NLS

$$\hat{\mathbf{u}} = \arg\min_{\mathbf{u}} |\boldsymbol{\rho} - \mathbf{h}(\mathbf{u})|^2 = \arg\min_{\mathbf{u}} \sum_{i=1}^{N} (\rho_i - h_i(\mathbf{u}))^2$$

- Assume a solution is available from previous processing: $\hat{\mathbf{u}}^{(0)} = \begin{bmatrix} \hat{u}_x^{(0)} \\ \hat{u}_y^{(0)} \\ \hat{u}_z^{(0)} \end{bmatrix}$

- We linearize the system around the previous solution:

$$\mathbf{u} = \hat{\mathbf{u}}^{(0)} + \Delta\mathbf{u}; \quad \Delta\mathbf{u} = \begin{bmatrix} \Delta u_x \\ \Delta u_y \\ \Delta u_z \end{bmatrix}$$ correction w.r.t. the previous solution

$$h_i(u_x, u_y, u_y) \approx h_i\left(\hat{u}_x^{(0)}, \hat{u}_y^{(0)}, \hat{u}_z^{(0)}\right) + \left.\frac{\partial h_i(\mathbf{u})}{\partial u_x}\right|_{\hat{\mathbf{u}}_0} \cdot \Delta u_x + \left.\frac{\partial h_i(\mathbf{u})}{\partial u_y}\right|_{\hat{\mathbf{u}}_0} \cdot \Delta u_y + \left.\frac{\partial h_i(\mathbf{u})}{\partial u_z}\right|_{\hat{\mathbf{u}}_0} \cdot \Delta u_z$$

Prof. M. Nicoli, Localization, Navigation and Smart Mobility
POLITECNICO DI MILANO

Non-linear least square (NLS)

1. Start from an initial guess $\hat{\boldsymbol{u}}^{(0)}$
2. Compute $[\mathbf{H}(\boldsymbol{u})]_i = \frac{\partial \mathrm{h}_i(\boldsymbol{u})}{\partial \boldsymbol{u}}$ $\quad \forall i = 1, \dots, N_{AP}$
3. Evaluate $\Delta\rho_i^{(1)} = \rho_i - \mathrm{h}_i\left(\hat{\boldsymbol{u}}^{(0)}\right)$ $\quad \forall i = 1, \dots, N_{AP}$ $\longrightarrow$ $\Delta\boldsymbol{\rho}^{(1)} = \boldsymbol{\rho} - \mathrm{h}\left(\hat{\boldsymbol{u}}^{(0)}\right)$
4. Compute $\Delta\boldsymbol{u}^{(1)} = \left(\mathbf{H}^{(1)^{\mathrm{T}}}\mathbf{H} \quad\right)^{-1}\mathbf{H}^{(1)^{\mathrm{T}}}\Delta\boldsymbol{\rho}^{(1)}$
5. Update the estimate $\hat{\boldsymbol{u}}^{(1)} = \hat{\boldsymbol{u}}^{(0)} + \Delta\boldsymbol{u}^{(1)}$
6. Repeat 1-5 until a maximum number of iteration is reached or a threshold condition is satisfied

NLS – general case

1. Start from an initial guess at previous iteration $\hat{\boldsymbol{u}}^{(k-1)}$
2. Compute $[\mathbf{H}(\boldsymbol{u})]_i = \frac{\partial \mathrm{h}_i(\boldsymbol{u})}{\partial \boldsymbol{u}}$ $\quad \forall i = 1, \dots, N_{AP}$
3. Evaluate $\Delta\rho_i^{(k)} = \rho_i - \mathrm{h}_i\left(\hat{\boldsymbol{u}}^{(k-1)}\right)$ $\quad \forall i = 1, \dots, N_{AP}$ $\longrightarrow$ $\Delta\boldsymbol{\rho}^{(\mathrm{k})} = \boldsymbol{\rho} - \mathrm{h}\left(\hat{\boldsymbol{u}}^{(k-1)}\right)$
4. Compute the correction $\Delta\boldsymbol{u}^{(k)} = \left(\mathbf{H}^{(k)^{\mathrm{T}}}\mathbf{H}\right)^{-1} \mathbf{H}^{(k)^{\mathrm{T}}} \Delta\boldsymbol{\rho}^{(\mathrm{k})}$
5. Update the estimate $\hat{\boldsymbol{u}}^{(k)} = \hat{\boldsymbol{u}}^{(k-1)} + \Delta\boldsymbol{u}^{(k)}$
6. Repeat 1-5 until a maximum number of iteration is reached or a threshold condition is satisfied

Weighted non-linear least square - WNLS

1. Start from an initial guess at previous iteration $\hat{\boldsymbol{u}}^{(k-1)}$

2. Compute $[\mathbf{H}(\boldsymbol{u})]_i = \frac{\partial h_i(\boldsymbol{u})}{\partial \boldsymbol{u}}$ $\quad \forall i = 1, \dots, N_{AP}$

3. Evaluate $\Delta\rho_i^{(k)} = \rho_i - \mathrm{h}_i\left(\hat{\boldsymbol{u}}^{(k-1)}\right)$ $\quad \forall i = 1, \dots, N_{AP}$

4. Compute the correction $\Delta\boldsymbol{u}^{(k)} = \left(\mathbf{H}^{(k)^{\mathrm{T}}}\mathbf{R}^{-1}\mathbf{H}^{\mathrm{T}}\right)^{-1}\mathbf{H}^{(k)^{\mathrm{T}}}\mathbf{R}^{-1}\Delta\boldsymbol{\rho}^{(\mathrm{k})}$, with $\Delta\boldsymbol{\rho}^{(\mathrm{k})} = \left[\Delta\rho_i^{(k)}\right]_{i=1}^{N_{AP}}$

5. Update the estimate $\hat{\boldsymbol{u}}^{(k)} = \hat{\boldsymbol{u}}^{(k-1)} + \Delta\boldsymbol{u}^{(k)}$

6. Repeat 1-5 until a maximum number of iteration is reached or a threshold condition is satisfied

To do

1. Define the localization scenario (copy&paste Lab 2-CRB)

2. Generate noisy measurements (copy&paste Lab 2-CRB)

3. Implement NLS

    i. start from an initial guess $\hat{\boldsymbol{u}}^{(k)}$

    ii. compute Jacobian matrix $\mathbf{H}\left(\hat{\boldsymbol{u}}^{(k)}\right)$

    iii. compute $\mathrm{h}_i(\hat{\boldsymbol{u}}^{(k)})$

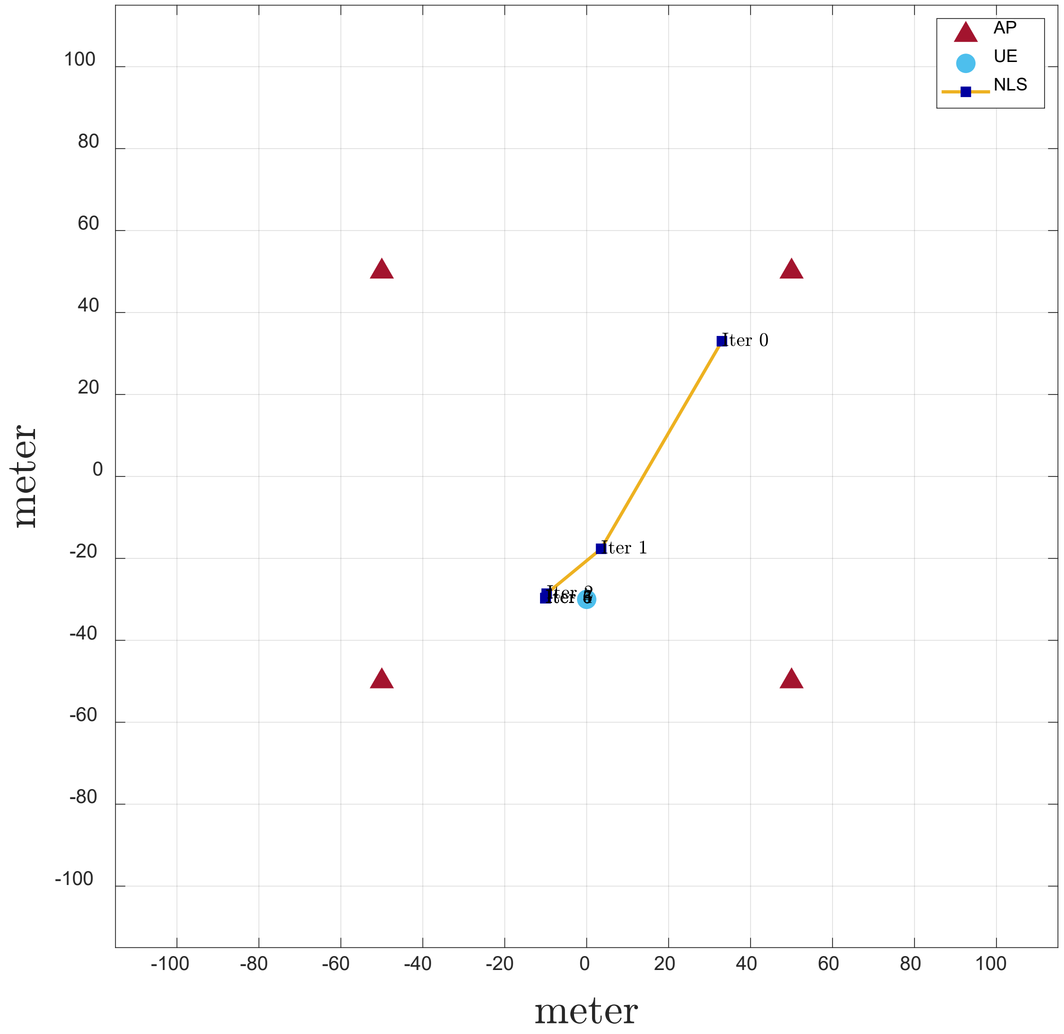
    iv. correct and update the estimate

4. Plot the NLS iterations

5. Perform Monte Carlo simulations

6. Check the CRB

Examples of expected results



Examples of expected results