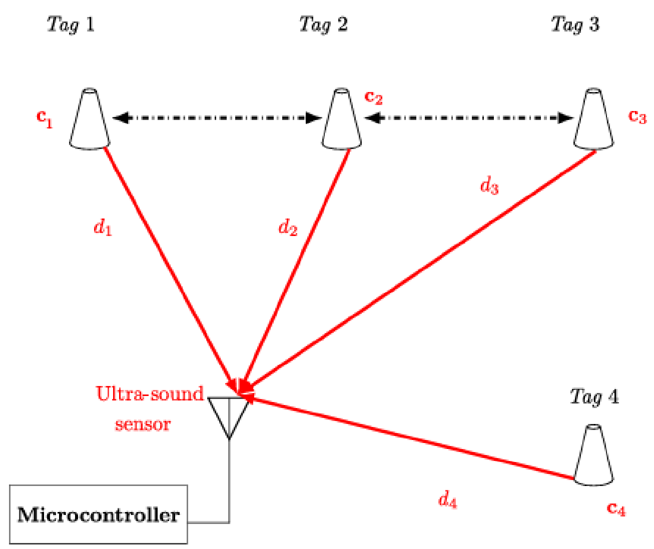


Figure 1: *Network of distributed tags with 4 tags.*

Practical Work Report

In424 – Distributed Information Systems

3D calibration of a ultra-sound sensor system 4 H

Students:
Simon Lignac
Steeve Mbock

Instructor:
Samy Labsir

Promotion 2027 – Aero 4
May 2026
Academic Year 2025–2026

Table of Contents

Part I – Simulation of the Measurement System

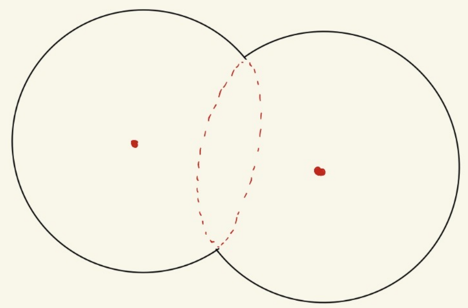
Question 1 – Minimum number of tags required

The unknown to be estimated is the position $\mathbf{p} = [p_x, p_y, p_z]^\top \in \mathbb{R}^3$, which has 3 degrees of freedom. Each tag i provides a single scalar measurement z_i, which constrains the position of the micro-controller to lie on a sphere of radius d_i centered at $\mathbf{c}_i$:

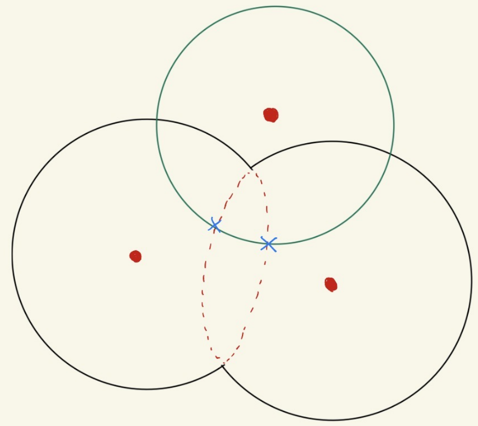
$$\|\mathbf{p} - \mathbf{c}_i\|_2 = d_i.$$

Geometrically:

- **1 tag** – the locus of solutions is a sphere in $\mathbb{R}^3$: the system is underdetermined (infinitely many solutions).
- **2 tags** – the intersection of two spheres is a circle: still infinitely many solutions.
- **3 tags** – the intersection of three spheres is generically a finite set of points (at most 2), yielding a nearly determined system.



For 2 tags —> Numbers of solutions equals a circle



For 3 tags —> Numbers of solutions equals one or two possible points

Therefore, at least 3 tags are necessary to reduce the solution set to a finite number of candidates and make the calibration tractable. In practice, a 4th tag (as used in this work) removes the remaining ambiguity and improves robustness against noise.

Question 2 – Expression of d_i

The distance $d_i \in \mathbb{R}$ between the micro-controller at position $\mathbf{p} = [p_x, p_y, p_z]^\top$ and tag i at position $\mathbf{c}_i = [c_{i,x},\, c_{i,y},\, c_{i,z}]^\top$ is given by the Euclidean norm:

$$d_i = \|\mathbf{p} - \mathbf{c}_i\|_2 = \sqrt{(p_x - c_{i,x})^2 + (p_y - c_{i,y})^2 + (p_z - c_{i,z})^2}. \tag{I.1}$$

Question 3 – Noisy measurement model

The ultra-sound sensor is subject to thermal noise. The noisy measurement z_i is modeled as the true distance d_i corrupted by an additive Gaussian noise $n_i \sim \mathcal{N}(0,\, \sigma_i^2)$:

$$z_i = d_i + n_i, \qquad n_i \sim \mathcal{N}(0,\, \sigma_i^2). \tag{I.2}$$

This additive noise model is standard for sensor calibration: the expected value of z_i equals the true distance d_i, and the measurement uncertainty is characterised by the variance σ_i^2.

Question 4 – Likelihood of the observations

From the measurement model (I.2), the conditional distribution of a single observation is:

$$z_i \,|\, \mathbf{p} \;\sim\; \mathcal{N}(d_i(\mathbf{p}),\; \sigma_i^2),$$

so:

$$p(z_i \,|\, \mathbf{p}) = \frac{1}{\sqrt{2\pi\, \sigma_i^2}} \exp\left(-\frac{(z_i - d_i)^2}{2\, \sigma_i^2}\right).$$

Since the noise terms $\{n_i\}_{i=1}^N$ are mutually independent, the joint likelihood of the full observation vector $\mathbf{z}$ factorises as a product of marginals:

$$\boxed{p(\mathbf{z} \,|\, \mathbf{p}) = \prod_{i=1}^{N} p(z_i \,|\, \mathbf{p}) = \prod_{i=1}^{N} \frac{1}{\sqrt{2\pi\, \sigma_i^2}} \exp\left(-\frac{\big(z_i - \|\mathbf{p} - \mathbf{c}_i\|_2\big)^2}{2\, \sigma_i^2}\right).} \tag{I.3}$$

Question 5 – Simulation of the observations

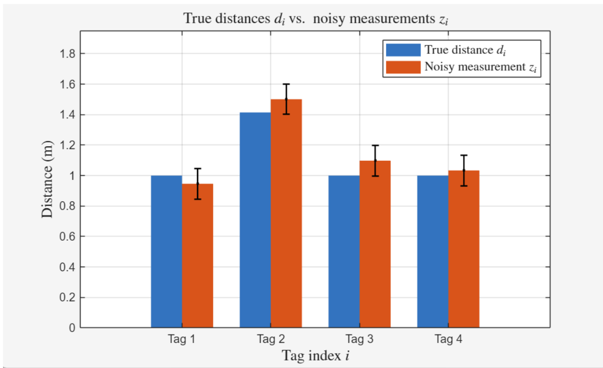
We implemented the distances and the noisy measurement on Matlab

```
% Compute true distances
d = zeros(N, 1);
for i = 1:N
    d(i) = norm(p - C(:,i));
end

% Generate noisy measurements
rng(42);
n = sigma * randn(N, 1);
z = d + n;
```

With the following simulation parameters, we got these values :

Simulation parameters	
Micro-controller position	$\mathbf{p} = [1, 1, 1]^\top$
σ_i	$0.1\,\text{m}$
Number of tags	$N = 4$
Tag 1	$\mathbf{c}_1 = [1, 1, 0]^\top$
Tag 2	$\mathbf{c}_2 = [1, 0, 0]^\top$
Tag 3	$\mathbf{c}_3 = [1, 0, 1]$
Tag 4	$\mathbf{c}_4 = [0, 1, 1]^\top$



The simulated measurements z_i are in close agreement with the true distances d_i, as expected given the low noise level $\sigma = 0.1$ m. The deviations $|z_i - d_i| = |n_i|$ remain small relative to the measured distances, which confirms that the Gaussian noise model is well-suited to represent the thermal noise of the ultra-sound sensor.

Part II – Estimation problem

II.1 Case I: Tag positions $\{c_i\}_{i=1}^N$ known

question 1:

if we take :

$$p = \frac{1}{\sqrt{2\pi\sigma^2}} \prod_i^N \exp \frac{-(z_i - d_i)^2}{2\sigma^2} \tag{II.1}$$

only one part depend of p. we only need to minimise : $\sum_i^N (z_i - d_i)^2$.

$$\sum_i^N (z_i - d_i)^2 = \sum_i^N (z_i - ||c_i - p||)^2 \tag{II.2}$$

So if we assume $\phi(p) = z_i - ||c_i - p||)^2$. we obtain that

$$\hat{p} = \underset{p \in \mathbb{R}^3}{\arg\min}\ ||\phi(p)||^2$$

question 2:

we start with:

$$\delta^{(l+1)} = \underset{\delta \in \mathbb{R}^3}{\arg\min}\ ||\phi(\hat{p}^{(l)}) + J_\phi^{(1,l)} \delta||^2 \tag{II.3}$$

we alraydy know the answers at this problem $Ax = B$. We identify $B = \phi(\hat{p}^{(l)})$, $A = J_\phi^{(1,l)}$ and $x = \delta$.

$$A^T A x = A^T B \tag{II.4}$$

we can deduce:

$$\left((J_\phi^{(1,l)})^T J_\phi^{(1,l)} \right)\ \delta^{(l+1)} = -(J_\phi^{(1,l)})^T\ \phi(\hat{p}^{(l)}) \tag{II.5}$$

question 3:

we have :

$$\phi(p) = z_i - ||c_i - p|| \tag{II.6}$$

In this case, only p is a variable, so the rest can be considered as constants. Therefore, z_i disappears.

$$\begin{aligned} J_\phi = d(||c_i - p||) &= d\left(\sqrt{(c_{i,1} - p_1)^2 + (c_{i,2} - p_2)^2 + (c_{i,3} - p_3)^2}\right) \\ &= \frac{-2(c_i - p)}{2||c_i - p||} \\ &= \frac{-(c_i - p)}{||c_i - p||} \end{aligned} \tag{II.7}$$

question 4:

```
    function [crit,err] = algorithm_1(p_true,d,z,C,iteration)

    crit = zeros(1,iteration);
    err  = zeros(1,iteration);
    N = length(d);
    p = mean(C,2);
    for l = 1:iteration


        phi = zeros(N,1);
        for i = 1:N
            di = norm(p - C(:,i));
            phi(i) = z(i) - di;
        end


        J = zeros(N,3);
        for i = 1:N
            di = norm(p - C(:,i));
            J(i,:) = -(C(:,i)-p)./di;
        end


        crit(l) = norm(phi)^2;
        err(l)  = norm(p - p_true);


        delta = (J.'*J) \ (J.' * phi);
        p = p + delta;

    end
end
```

question 5:

[H] Here we have the performance of the algorithm: [H]

question 6:

We chose to put an obstacle in front of the six tag, which means the variance of its noise is 30 times higher than that of the other one. we use the same algorithm than before here the

```
    sigma3 = 0.1*ones(1,4);
    sigma3(3) = 3;
```

result.

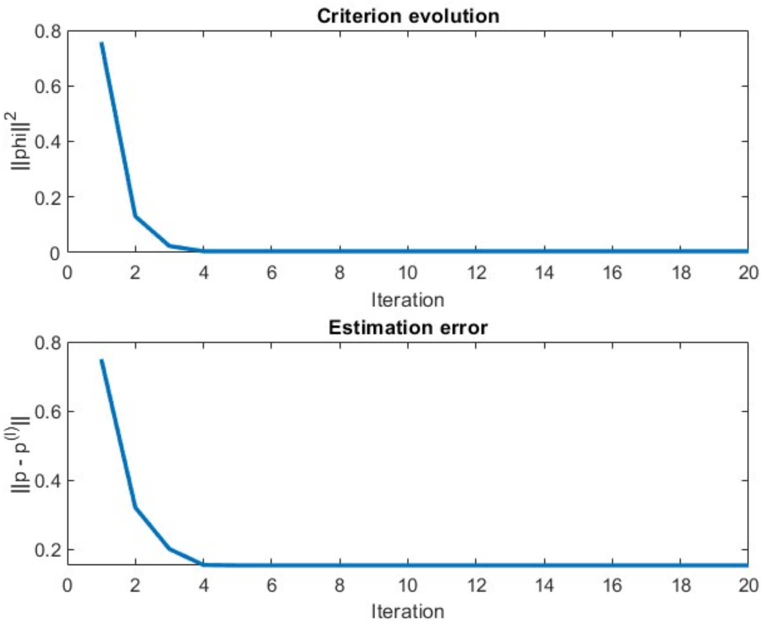


Figure II.1: Criterion and estimation error with all tags known

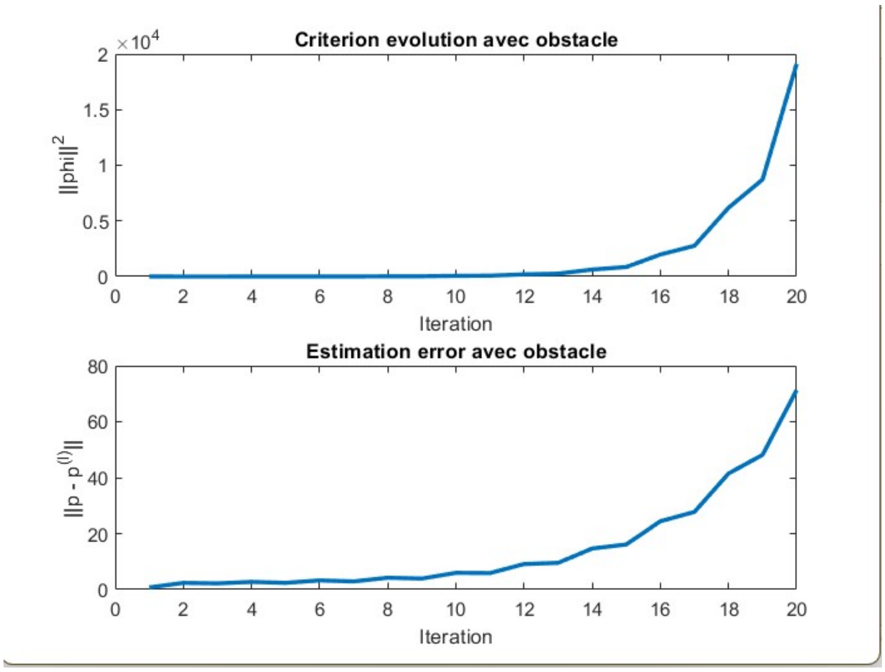


Figure II.2: Criterion and error estimation with obstacle

question 7

We observe that the algorithm does not converge due to the fact that there is a wrong signal with a high power than the other, all the tags have the same ponderation so the power of the noise have more impact. Our results are distorted by the intensity of the noise. Moreover for the previous simulation we obtain a mean error of $m\_err1 = 0.1776$ and for this $m_err1 = 14.9621$. that's confirm why the algorithm fails to converge.

question 8

We assume that the measurements $z = (z_1, \ldots, z_N)^T$ are corrupted by independent Gaussian noises with zero mean and variances σ_i^2. Conditionally on p, we have

$$z_i = d_i(p) + w_i, \quad d_i(p) = \|p - c_i\|, \quad w_i \sim \mathcal{N}(0, \sigma_i^2),$$

and the noises w_i are independent.

Therefore, the likelihood of z given p is

$$p(z \mid p) = \prod_{i=1}^{N} \frac{1}{\sqrt{2\pi\sigma_i^2}} \exp\left(-\frac{1}{2\sigma_i^2}\big(z_i - d_i(p)\big)^2\right).$$

Taking the negative log-likelihood (and discarding constants independent of p), we obtain

$$-\log p(z \mid p) = \frac{1}{2}\sum_{i=1}^{N} \frac{\big(z_i - d_i(p)\big)^2}{\sigma_i^2} + \text{constant}.$$

Define the residuals

$$\phi_i(p) = z_i - d_i(p), \quad \phi(p) = \begin{pmatrix} \phi_1(p) \\ \vdots \\ \phi_N(p) \end{pmatrix},$$

and the covariance matrix

$$\Sigma = \begin{pmatrix} \sigma_1^2 & 0 & \cdots & 0 \\ 0 & \sigma_2^2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_N^2 \end{pmatrix}.$$

Then the negative log-likelihood can be written as

$$-\log p(z \mid p) = \frac{1}{2}\,\phi(p)^T \Sigma^{-1} \phi(p) + \text{constant}.$$

Maximizing the likelihood is therefore equivalent to minimizing $\phi(p)^T\Sigma^{-1}\phi(p)$, so the maximum likelihood estimator is

$$\hat{p}_{ML} = \arg\min_{p \in \mathbb{R}^3} \phi(p)^T \Sigma^{-1} \phi(p).$$

```
    function [crit,err] = algorithm_2(p_true,d,z,sigma_inv,C,
        iteration)

    crit = zeros(1,iteration);
    err  = zeros(1,iteration);
    N = length(d);
    p = mean(C,2);
    for l = 1:iteration


        phi = zeros(N,1);
        for i = 1:N
            di = norm(p - C(:,i));
            phi(i) = (z(i) - di);

        end


        J = zeros(N,3);
        for i = 1:N
            di = norm(p - C(:,i));
            J(i,:) = -(C(:,i)-p)./di;
        end


        crit(l) = norm(phi)^2;
        err(l)  = norm(p - p_true);


        delta = (J.'*sigma_inv*J) \ (J.' *sigma_inv* phi);
        p = p + delta;

    end
end
```

question 9

Let $\phi(p) = \phi(p)\Sigma^{-1}\phi(p)$
Here the new algorithm:
Here the result :

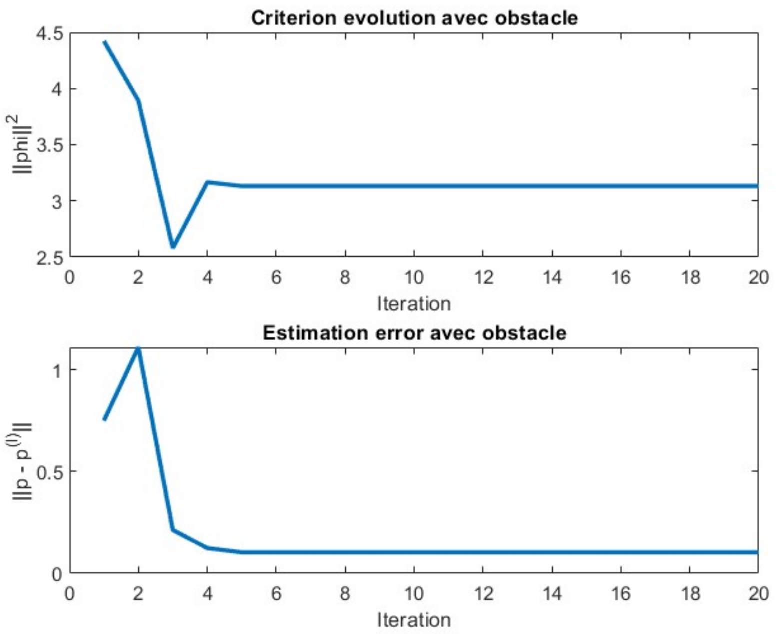


Figure II.3: criterion and estimaton error with new gauss newton

II.2 Case II: Tag positions $\{c_i\}_{i=1}^N$ unknown

II.2.1 question 1:

We consider the residual vector

$$\phi(p; \{c_i\}_{i=1}^N) = \begin{pmatrix} \phi_1(p, c_1) \\ \phi_2(p, c_2) \\ \vdots \\ \phi_N(p, c_N) \end{pmatrix}, \quad \phi_i(p, c_i) = z_i - \|p - c_i\|.$$

For each $i \in \{1, \ldots, N\}$, define

$$d_i(p, c_i) = \|p - c_i\|, \quad u_i = p - c_i.$$

Then

$$\phi_i(p, c_i) = z_i - d_i(p, c_i),$$

so the derivatives are

$$\frac{\partial \phi_i}{\partial p} = -\frac{\partial d_i}{\partial p} = -\frac{u_i^T}{\|u_i\|},$$

and

$$\frac{\partial \phi_i}{\partial c_i} = -\frac{\partial d_i}{\partial c_i} = +\frac{u_i^T}{\|u_i\|}.$$

The Jacobian $J^{(2)}$ of ϕ with respect to p and $\{c_i\}_{i=1}^N$ is thus an $N \times (3 + 3N)$ matrix of the form

$$J^{(2)}(p, \{c_i\}) = \begin{pmatrix} -\frac{(p-c_1)^T}{\|p-c_1\|} & \frac{(p-c_1)^T}{\|p-c_1\|} & \mathbf{0}_{1\times 3} & \cdots & \mathbf{0}_{1\times 3} \\ -\frac{(p-c_2)^T}{\|p-c_2\|} & \mathbf{0}_{1\times 3} & \frac{(p-c_2)^T}{\|p-c_2\|} & \cdots & \mathbf{0}_{1\times 3} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -\frac{(p-c_4)^T}{\|p-c_4\|} & \mathbf{0}_{1\times 3} & \mathbf{0}_{1\times 3} & \cdots & \frac{(p-c_4)^T}{\|p-c_4\|} \end{pmatrix}$$

where: - the first 3 columns correspond to the derivative with respect to p, - for each i, the block of 3 columns associated with c_i contains $\frac{(p-c_i)^T}{\|p-c_i\|}$ on row i and zeros elsewhere.

question 2

here the code for gauss newton:

```
    function [crit,err] = algorithm_3(p_true,d,z,C_true,iteration)

    N = size(C_true,2);

    % --- Initialisation ---
    p = p_true + randn(3,1);
    C = C_true + randn(3,N);

    crit = zeros(1,iteration);
    err  = zeros(1,iteration);

    for l = 1:iteration


        phi = zeros(N,1);
        for i = 1:N
            di = norm(p - C(:,i));
            phi(i) = z(i) - di;
        end

        % --- Construction de la Jacobienne ---
        J = zeros(N, 3 + 3*N);

        for i = 1:N
            diff = p - C(:,i);
            di   = norm(diff);


            J(i,1:3) = -(diff.'/di);
            col = 3 + (i-1)*3 + 1;
            J(i,col:col+2) = +(diff.'/di);
        end

        crit(l) = norm(phi)^2;
        err(l)  = norm(p - p_true);
        delta = J \ (-phi);
        p = p + delta(1:3);
        for i = 1:N
            col = 3 + (i-1)*3 + 1;
            C(:,i) = C(:,i) + delta(col:col+2);
        end
    end
end
```

question 3

here the evolution of error and criterion.

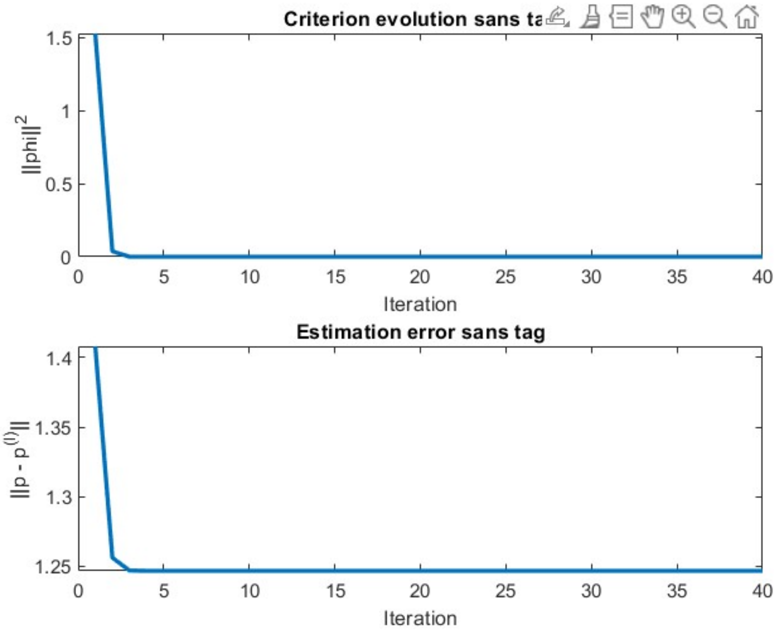


Figure II.4: Criterion and error

We obtain a mean error of m_(err) = 1.2508. This is an acceptable result when we consider the noise variance of one for each C parameter.

II.3 Case III: Tag positions $\{c_i\}_{i=1}^{N}$ partially known

Question 1 – Augmented criterion

We define the augmented residual vector by stacking the measurement residuals and the prior residuals:

$$\varphi^a(\mathbf{p}; \{\mathbf{c}_i\}) = \begin{bmatrix} \varphi(\mathbf{p}; \{\mathbf{c}_i\}) \\ \mathrm{c}_1 - \tilde{\mathbf{c}}_1 \\ \vdots \\ \mathrm{c}_N - \tilde{\mathbf{c}}_N \end{bmatrix} \in \mathbb{R}^{N+3N}, \tag{II.8}$$

and the augmented covariance matrix :

$$\mathbf{\Sigma}^a =$$

$$\begin{bmatrix} \mathrm{I}_N & \mathbf{0} \\ 0 & \boldsymbol{\Sigma}^c_{\text{block}} \end{bmatrix},$$

$\boldsymbol{\Sigma}^c_{\text{block}}$
$= \operatorname{diag}\big((\sigma^c_1)^2\mathbf{I}_3,\ \ldots,\ (\sigma^c_N)^2\mathbf{I}_3\big)$
$\in \mathbb{R}^{3N\times 3N}.$
(II.9)

Expanding the weighted norm:

$= \varphi^\top \mathbf{I}_N^{-1}\varphi$
$+ \sum_{i=1}^N$
$(\mathrm{c}_i - \tilde{\mathbf{c}}_i)^\top$
$[(\sigma^c_i)^2\mathbf{I}_3]^{-1}$
$(\mathrm{c}_i - \tilde{\mathbf{c}}_i)$

$= \|\varphi(\mathbf{p}; \{\mathbf{c}_i\})\|^2$
$+ \sum_{i=1}^N \|\mathbf{c}_i - \tilde{\mathbf{c}}_i\|^2_{\boldsymbol{\Sigma}^c},$

$\|\varphi^a\|^2_{\boldsymbol{\Sigma}^a}$
$= (\varphi^a)^\top(\boldsymbol{\Sigma}^a)^{-1}\varphi^a$

$= \varphi^\top \mathbf{I}_N^{-1}\varphi$
$+ \sum_{i=1}^N$
$(\mathrm{c}_i - \tilde{\mathbf{c}}_i)^\top$
$[(\sigma^c_i)^2\mathbf{I}_3]^{-1}$
$(\mathrm{c}_i - \tilde{\mathbf{c}}_i)$

$= \|\varphi(\mathbf{p}; \{\mathbf{c}_i\})\|^2$
$+ \sum_{i=1}^N \|\mathbf{c}_i - \tilde{\mathbf{c}}_i\|^2_{\boldsymbol{\Sigma}^c},$

Question 2 – Jacobian of φ^a

The variable vector is $\boldsymbol{\theta} = [\mathbf{p}^\top,\ \mathbf{c}_1^\top,\ \ldots,\ \mathbf{c}_N^\top]^\top \in \mathbb{R}^{3+3N}$. By the block structure of φ^a in (II.3) and the stacking rule for Jacobians:

$$J_{\varphi^a} = \frac{\partial \varphi^a}{\partial \boldsymbol{\theta}} = \begin{bmatrix} \dfrac{\partial \varphi}{\partial \boldsymbol{\theta}} \\ \dfrac{\partial}{\partial \boldsymbol{\theta}} \begin{bmatrix} \mathbf{c}_1 - \tilde{\mathbf{c}}_1 \\ \vdots \\ \mathbf{c}_N - \tilde{\mathbf{c}}_N \end{bmatrix} \end{bmatrix}.$$

The top block is by definition $J^{(2)}_\varphi$ (the Jacobian derived in Case II). For the bottom block, since $\tilde{\mathbf{c}}_i$ are constants:

$$\frac{\partial(\mathbf{c}_i - \tilde{\mathbf{c}}_i)}{\partial \mathbf{p}} = \mathbf{0}_{3\times 3}, \qquad \frac{\partial(\mathbf{c}_i - \tilde{\mathbf{c}}_i)}{\partial \mathbf{c}_j} = \begin{cases} \mathbf{I}_{3\times 3} & \text{if } j = i, \\ \mathbf{0}_{3\times 3} & \text{otherwise.} \end{cases}$$

Assembling these 3×3 blocks over all $i, j \in \{1, \ldots, N\}$ gives:

$$\frac{\partial}{\partial \boldsymbol{\theta}} \begin{bmatrix} \mathbf{c}_1 - \tilde{\mathbf{c}}_1 \\ \vdots \\ \mathbf{c}_N - \tilde{\mathbf{c}}_N \end{bmatrix} = \begin{bmatrix} \underbrace{\mathbf{0}_{3N\times 3}}_{\partial/\partial \mathbf{p}}, & \underbrace{\mathbf{I}_{3N\times 3N}}_{\partial/\partial\{\mathbf{c}_i\}} \end{bmatrix},$$

Question 3 – Simulation of $\{\mathbf{c}_i\}_{i=1}^N$

For each tag, a perturbed position is drawn as:

$$\mathbf{c}_i = \tilde{\mathbf{c}}_i + \mathbf{n}_i^c, \qquad \mathbf{n}_i^c \sim \mathcal{N}(\mathbf{0},\, (\sigma_i^c)^2\, \mathbf{I}_3).$$

This simulates the realistic scenario where the engineer has an approximate, but imperfect, knowledge of the tag positions.

Question 4 – New observation set

The observations are generated using the perturbed tag positions $\{\mathbf{c}_i\}$ from Question 3, following the same measurement model as Part I:

$$z_i = \|\mathbf{p} - \mathbf{c}_i\|_2 + n_i, \qquad n_i \sim \mathcal{N}(0,\, \sigma^2), \quad \sigma = 0.1 \text{ m}.$$

Question 5 – Gauss-Newton algorithm and performance analysis

The weighted Gauss-Newton descent direction at iteration l is obtained from:

$$\left(J_{\varphi^a}^{(l)\top}\, (\boldsymbol{\Sigma}^a)^{-1}\, J_{\varphi^a}^{(l)}\right) \delta^{(l+1)} = -J_{\varphi^a}^{(l)\top}\, (\boldsymbol{\Sigma}^a)^{-1}\, \varphi^a\left(\hat{\boldsymbol{\theta}}^{(l)}\right), \tag{II.10}$$

where $\hat{\boldsymbol{\theta}}^{(l)} = [\hat{\mathbf{p}}^{(l)\top}, \hat{\mathbf{c}}_1^{(l)\top}, \ldots, \hat{\mathbf{c}}_N^{(l)\top}]^\top$.

Figure II.5 shows the mean estimation error $\frac{1}{N_r}\sum_{n_r=1}^{N_r} \|\mathbf{p} - \hat{\mathbf{p}}^{(L,n_r)}\|$ as a function of σ_i^c, plotted alongside the constant reference levels obtained in Cases I and II.

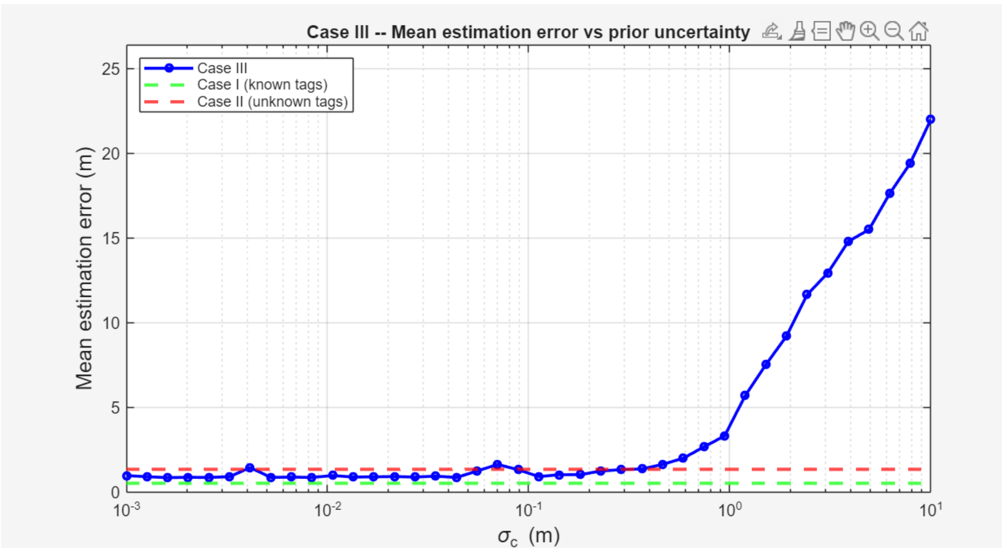


Figure II.5: Mean estimation error at the last iteration as a function of the prior uncertainty σ_i^c. Dashed lines: Case I (known tags, lower bound) and Case II (unknown tags, upper bound).

Interpretation. The curve exhibits the expected monotone behaviour: as σ_i^c decreases, the prior information becomes more constraining and the estimator approaches the performance of Case I (fully known tag positions). Conversely, as σ_i^c increases, the prior becomes uninformative and the error converges toward that of Case II (fully unknown tag positions). Case III therefore provides a continuous interpolation between the two extreme scenarios, allowing the engineer to incorporate partial knowledge in a principled Bayesian framework.