UiO : Department of Technology Systems
University of Oslo

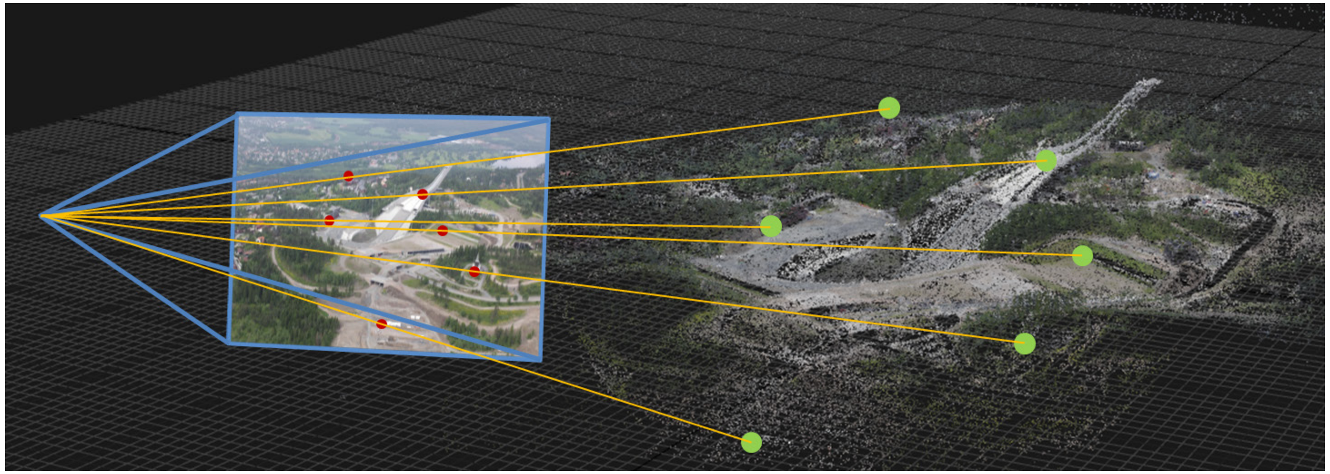
Lecture 6.2
An introduction to nonlinear least squares

Trym Vegard Haavardsholm

How can solve the indirect tracking problem?

Minimize **geometric error**

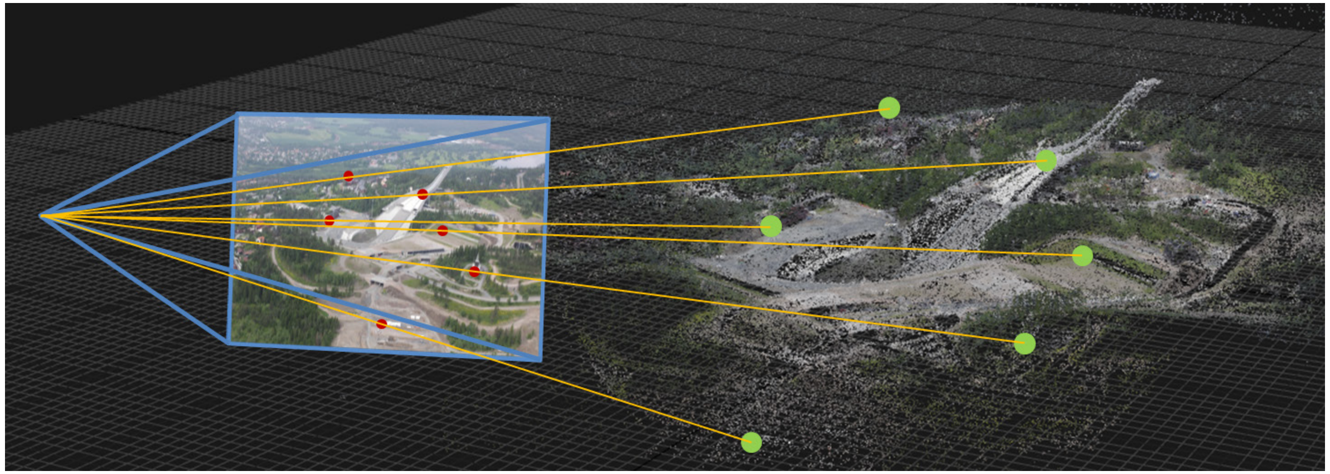
$$\mathbf{T}_{cw}^{*} = \underset{\mathbf{T}_{cw}}{\operatorname{argmin}} \sum_{i} \left\| \pi(\mathbf{T}_{cw}\tilde{\mathbf{x}}_{i}^{w}) - \mathbf{u}_{i} \right\|^{2}$$



How can solve the indirect tracking problem?

Minimize **geometric error** with **nonlinear least squares!**

$$\mathbf{T}_{cw}^{*} = \underset{\mathbf{T}_{cw}}{\operatorname{argmin}} \sum_{i} \left\| \pi(\mathbf{T}_{cw} \tilde{\mathbf{x}}_{i}^{w}) - \mathbf{u}_{i} \right\|^{2}$$



Problem formulation

Consider a set of *m* possibly nonlinear equations in *n* unknowns $\mathbf{x} = [x_1, \dots, x_n]^T$ written as

$$e_i(\mathbf{x}) = 0, \quad i = 1, \dots, m$$

$$e_i : \mathbb{R}^n \rightarrow \mathbb{R}$$

Problem formulation

Consider a set of *m* possibly nonlinear equations in *n* unknowns $\mathbf{x} = [x_1, \dots, x_n]^T$ written as

$$e_i(\mathbf{x}) = 0, \quad i = 1, \dots, m \qquad e_i : \mathbb{R}^n \rightarrow \mathbb{R}$$

*i*th equation

Problem formulation

Consider a set of *m* possibly nonlinear equations in *n* unknowns $\mathbf{x} = [x_1, \dots, x_n]^T$ written as

$$e_i(\mathbf{x}) = 0, \quad i = 1, \dots, m \qquad e_i : \mathbb{R}^n \to \mathbb{R}$$

*i*th *error* or *residual*

Problem formulation

Consider a set of m possibly nonlinear equations in n unknowns $\mathbf{x} = [x_1, \dots, x_n]^T$ written as

$$e_i(\mathbf{x}) = 0, \quad i = 1, \dots, m$$

$$e_i : \mathbb{R}^n \to \mathbb{R}$$

We can write these equations on vector form

$$e(\mathbf{x}) = \mathbf{0},$$

$$e : \mathbb{R}^n \to \mathbb{R}^m$$

where

$$e(\mathbf{x}) = \begin{bmatrix} e_1(\mathbf{x}) \\ \vdots \\ e_m(\mathbf{x}) \end{bmatrix}$$

Problem formulation

It is often not possible to find an exact solution to this problem.

We can instead seek an approximate solution
that minimizes the sum of squares of the residuals

$$f(\mathbf{x}) = e(\mathbf{x})^T e(\mathbf{x}) = \left\| e(\mathbf{x}) \right\|^2$$

Problem formulation

It is often not possible to find an exact solution to this problem.

We can instead seek an approximate solution
that minimizes the sum of squares of the residuals

$$\underbrace{f(\mathbf{x})} = e(\mathbf{x})^T e(\mathbf{x}) = \left\| e(\mathbf{x}) \right\|^2$$

The *objective function*

Problem formulation

It is often not possible to find an exact solution to this problem.

We can instead seek an approximate solution
that minimizes the sum of squares of the residuals

$$f(\mathbf{x}) = e(\mathbf{x})^T e(\mathbf{x}) = \left\| e(\mathbf{x}) \right\|^2$$

This means that we want to find the $\mathbf{x}$ that minimizes the objective function:

$$\mathbf{x}^* = \underset{\mathbf{x}}{\operatorname{argmin}}\, f(\mathbf{x}) = \underset{\mathbf{x}}{\operatorname{argmin}} \left\| e(\mathbf{x}) \right\|^2$$

Linear least squares

When the equations are linear,
we can obtain an objective function on the form

$$f(\mathbf{x}) = \left\| e(\mathbf{x}) \right\|^2 = \left\| \mathbf{A}\mathbf{x} - \mathbf{b} \right\|^2$$

A solution is required to have zero gradient:

$$\nabla f(\mathbf{x}^*) = 2\mathbf{A}^T \left(\mathbf{A}\mathbf{x}^* - \mathbf{b} \right) = \mathbf{0}$$

This results in the normal equations,

$$\mathbf{A}^T \mathbf{A}\mathbf{x}^* = \mathbf{A}^T \mathbf{b}$$

$$\mathbf{x}^* = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{b}$$

$$\mathbf{x}^* = \mathbf{A}^{\dagger} \mathbf{b}$$

which can be solved with Cholesky- or QR factorization.

Linear least squares

When the equations are linear,
we can obtain an objective function on the form

$$f(\mathbf{x}) = \|e(\mathbf{x})\|^2 = \|\mathbf{A}\mathbf{x} - \mathbf{b}\|^2$$

A solution is required to have zero gradient:

$$\nabla f(\mathbf{x}^*) = 2\mathbf{A}^T\left(\mathbf{A}\mathbf{x}^* - \mathbf{b}\right) = \mathbf{0}$$

This results in the normal equations,

$$\mathbf{A}^T\mathbf{A}\mathbf{x}^* = \mathbf{A}^T\mathbf{b}$$

$$\mathbf{x}^* = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{b}$$

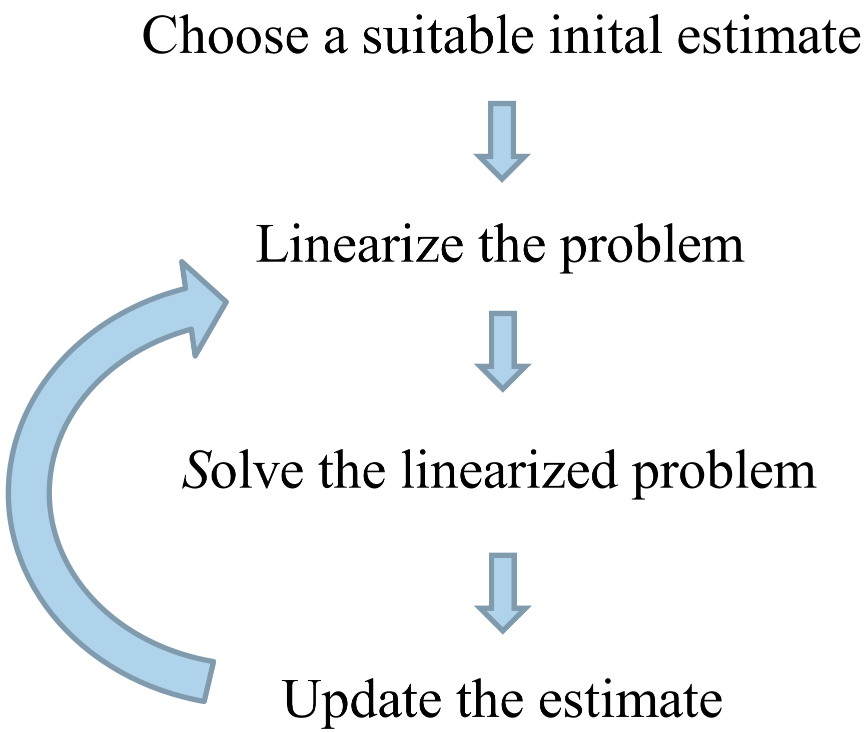
$$\mathbf{x}^* = \mathbf{A}^\dagger\mathbf{b}$$

which can be solved with Cholesky- or QR factorization.

Read more about LLS:
- http://vmls-book.stanford.edu/vmls.pdf

Nonlinear least squares

Nonlinear least squares problems cannot be solved directly,
but require an iterative procedure starting from a suitable initial estimate:



Nonlinear MAP inference for state estimation

We will use nonlinear least squares to solve **state estimation problems** based on **measurements** and corresponding **measurement models**

Let $X = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_k\}$ be the set of all state variables, and $Z = \{\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_m\}$ be the set of all measurements.

We say that X_i are the state variables involved in measurement $\mathbf{z}_i$.

We are interested in estimating the unknown state variables X, given the measurements Z.
The **Maximum a Posteriori estimate** is given by:

$$X^{MAP} = \underset{X}{\operatorname{argmax}}\ p(X \mid Z)$$

Nonlinear MAP inference for state estimation

Measurement model:

$$\mathbf{z}_i = h_i(X_i) + \eta, \qquad \eta \sim N(\mathbf{0}, \mathbf{\Sigma}_i)$$

Measurement prediction function:

$$\hat{\mathbf{z}}_i = h_i(X_i)$$

Measurement error function:

$$e_i(X_i) = h_i(X_i) - \mathbf{z}_i$$

Objective function:

$$f(X) = \sum_{i=1}^{m} \left\| h_i(X_i) - \mathbf{z}_i \right\|_{\mathbf{\Sigma}_i}^2$$

where $\left\|\mathbf{e}\right\|_{\mathbf{\Sigma}}^2 = \mathbf{e}^T \mathbf{\Sigma}^{-1} \mathbf{e}$ is the Mahalanobis norm

Nonlinear MAP inference for state estimation

Measurement model:

$$\mathbf{z}_i = h_i(X_i) + \eta, \qquad \eta \sim N(\mathbf{0}, \mathbf{\Sigma}_i)$$

Measurement prediction function:

$$\hat{\mathbf{z}}_i = h_i(X_i)$$

Measurement error function:

$$e_i(X_i) = h_i(X_i) - \mathbf{z}_i$$

Objective function:

$$f(X) = \sum_{i=1}^{m} \left\| h_i(X_i) - \mathbf{z}_i \right\|_{\mathbf{\Sigma}_i}^2$$

where $\left\|\mathbf{e}\right\|_{\mathbf{\Sigma}}^2 = \mathbf{e}^T \mathbf{\Sigma}^{-1} \mathbf{e}$ is the Mahalanobis norm.

This results in the nonlinear least squares problem:

$$X^* = \underset{X}{\operatorname{argmin}} \sum_{i=1}^{m} \left\| h_i(X_i) - \mathbf{z}_i \right\|_{\mathbf{\Sigma}_i}^2$$

It turns out that the nonlinear least squares solution to this problem is the MAP estimate!

Nonlinear MAP inference for state estimation

Measurement model:

$$\mathbf{z}_i = h_i(X_i) + \eta, \qquad \eta \sim N(\mathbf{0}, \boldsymbol{\Sigma}_i)$$

Measurement prediction function:

$$\hat{\mathbf{z}}_i = h_i(X_i)$$

Measurement error function:

$$e_i(X_i) = h_i(X_i) - \mathbf{z}_i$$

Objective function:

$$f(X) = \sum_{i=1}^{m} \left\| h_i(X_i) - \mathbf{z}_i \right\|_{\boldsymbol{\Sigma}_i}^2$$

where $\left\|\mathbf{e}\right\|_{\boldsymbol{\Sigma}}^2 = \mathbf{e}^T \boldsymbol{\Sigma}^{-1} \mathbf{e}$ is the Mahalanobis norm.

This results in the nonlinear least squares problem:

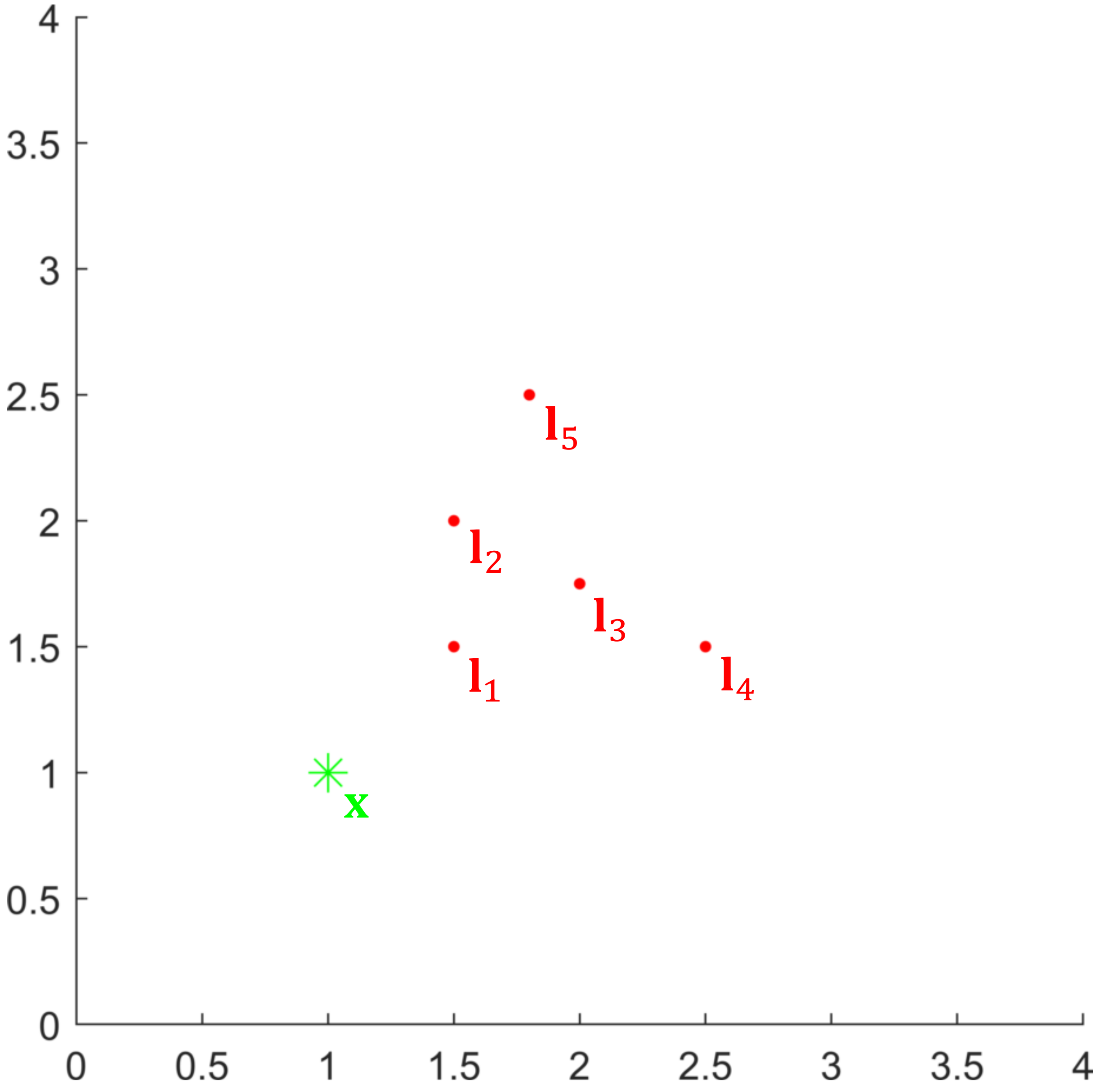
$$X^* = \underset{X}{\operatorname{argmin}} \sum_{i=1}^{m} \left\| h_i(X_i) - \mathbf{z}_i \right\|_{\boldsymbol{\Sigma}_i}^2$$

It turns out that the nonlinear least squares solution to this problem is the MAP estimate!

Assume for now that all $\boldsymbol{\Sigma}_i = \sigma \mathbf{I}$.
This simplifies our objective to:

$$X^* = \underset{X}{\operatorname{argmin}} \sum_{i=1}^{m} \left\| h_i(X_i) - \mathbf{z}_i \right\|^2$$

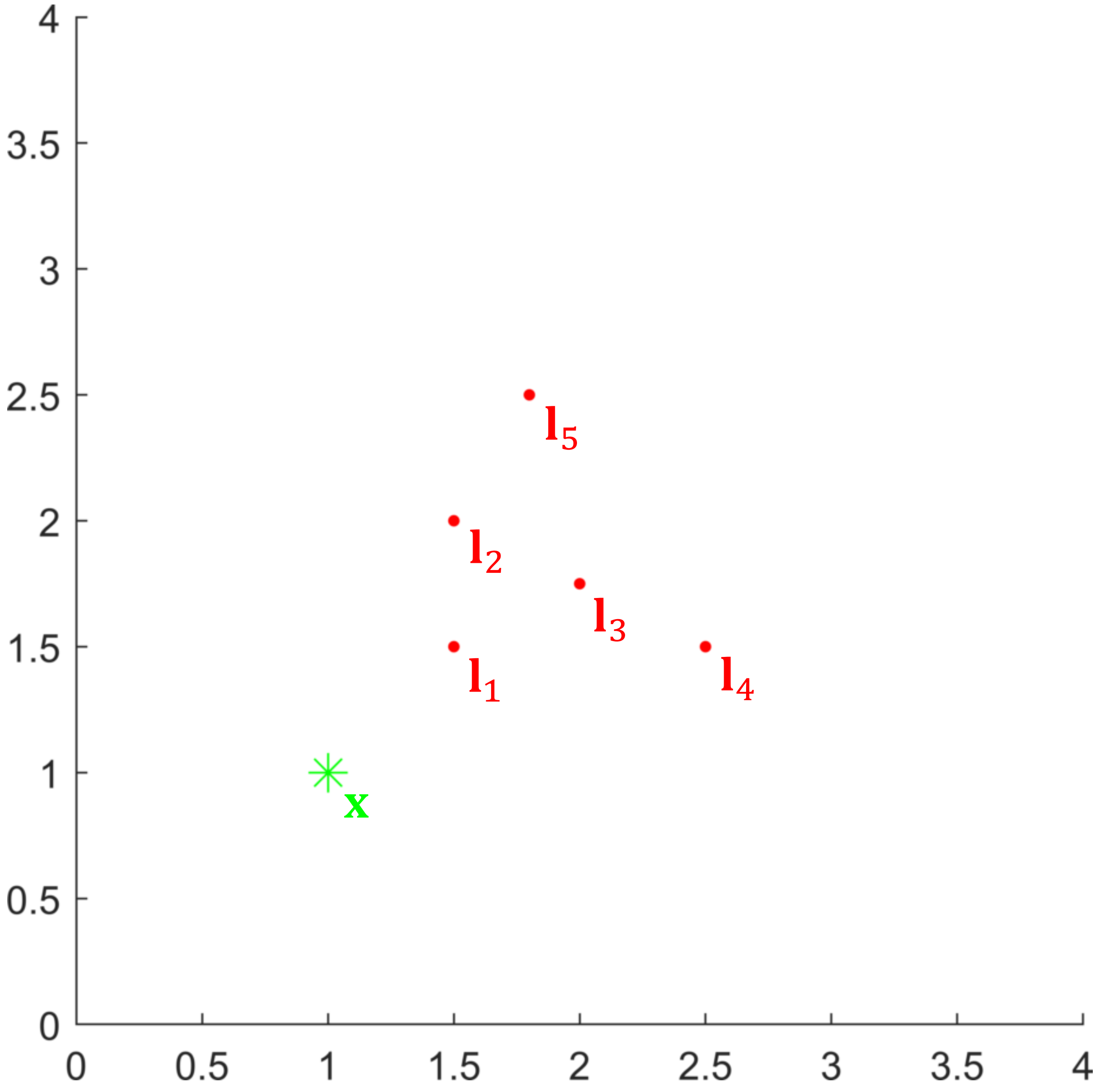
Example:
Range-based localization



Example:
Range-based localization

States: Our location

$$X = \mathbf{x}$$



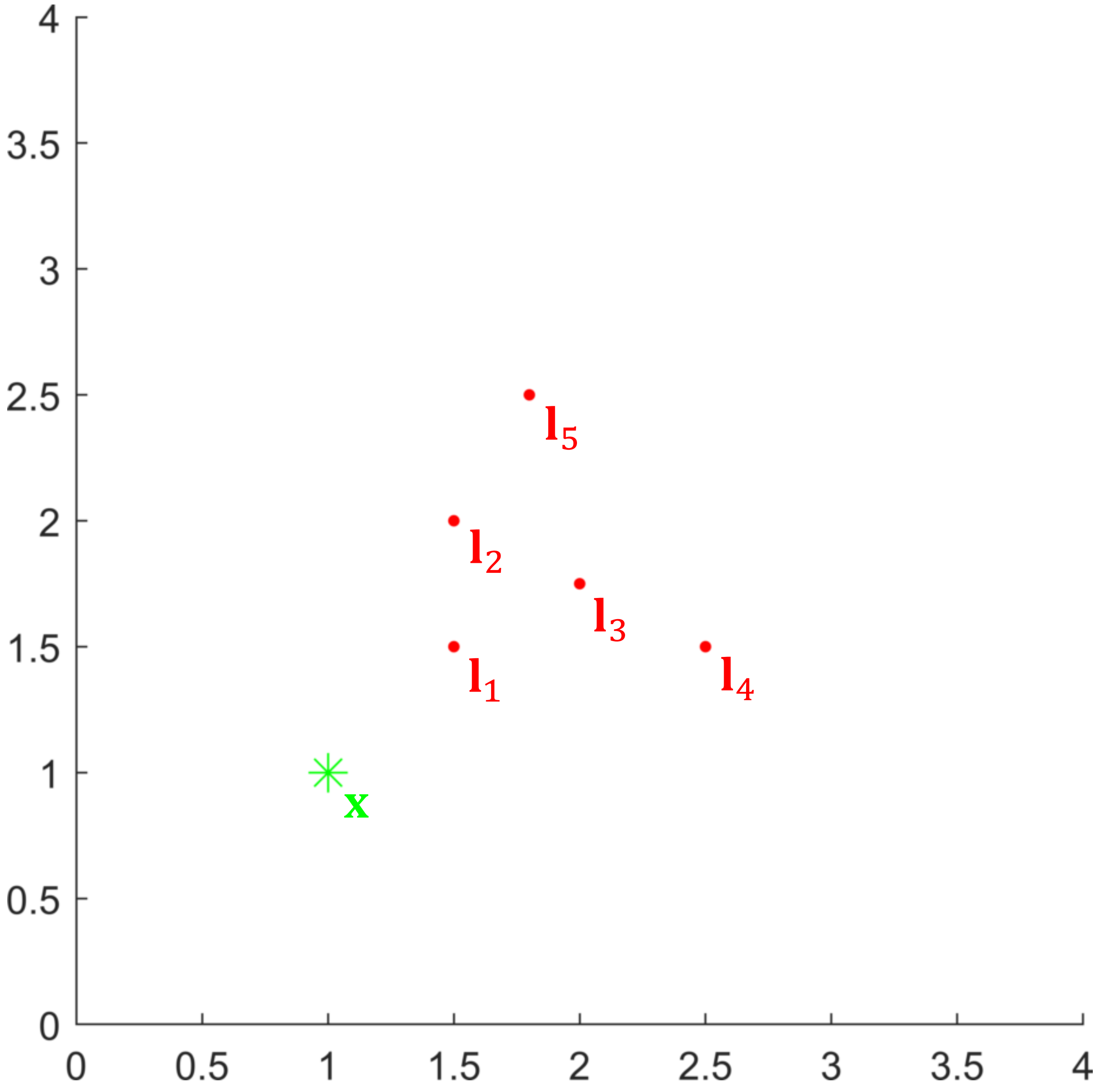
Example:
Range-based localization

States: Our location

$$X = \mathbf{x}$$

Measurements: Range to landmarks

$$Z = \{\rho_1, \ldots, \rho_m\}$$



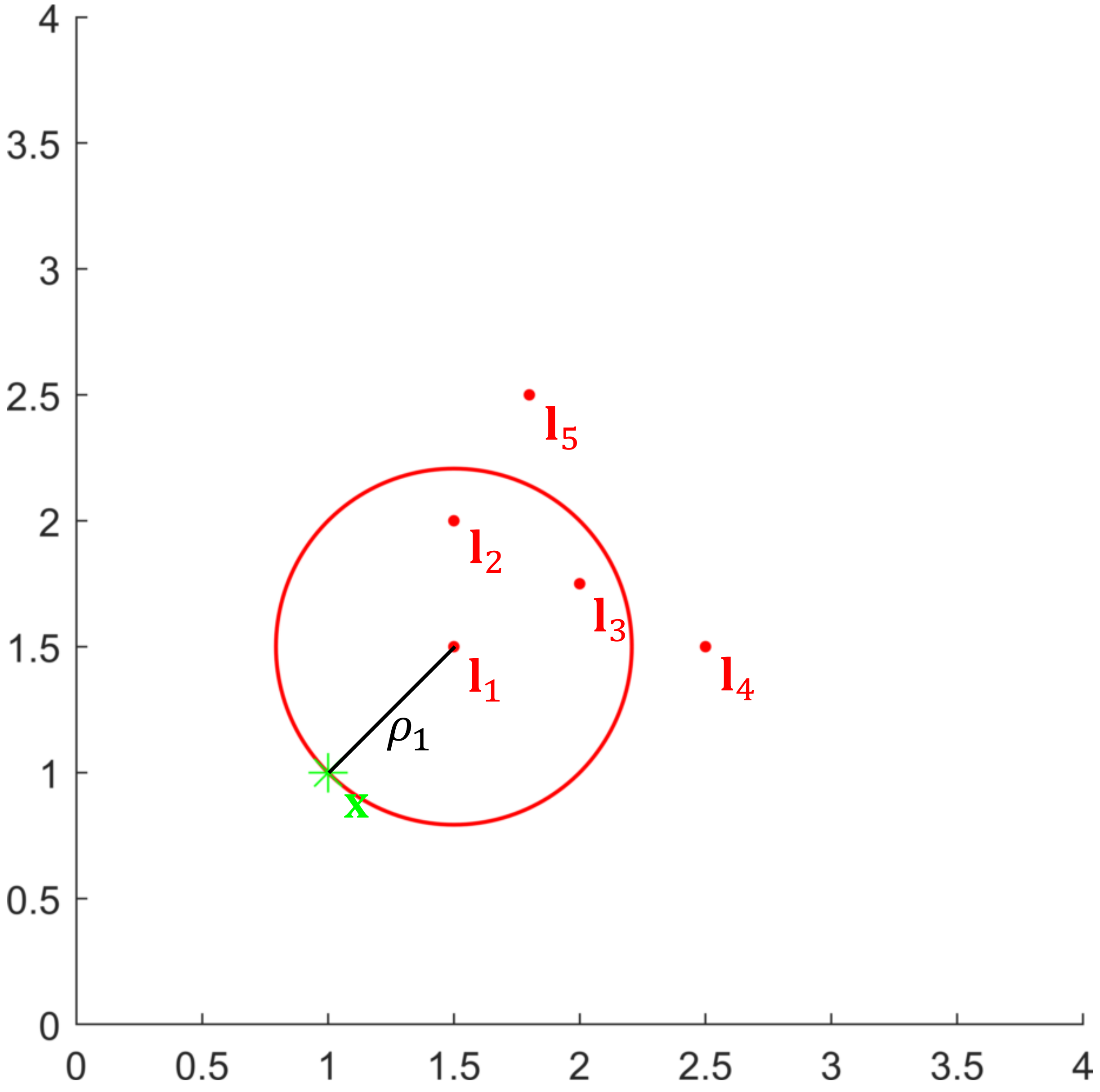
Example:
Range-based localization

States: Our location

$$X = \mathbf{x}$$

Measurements: Range to landmarks

$$Z = \{\rho_1, \ldots, \rho_m\}$$



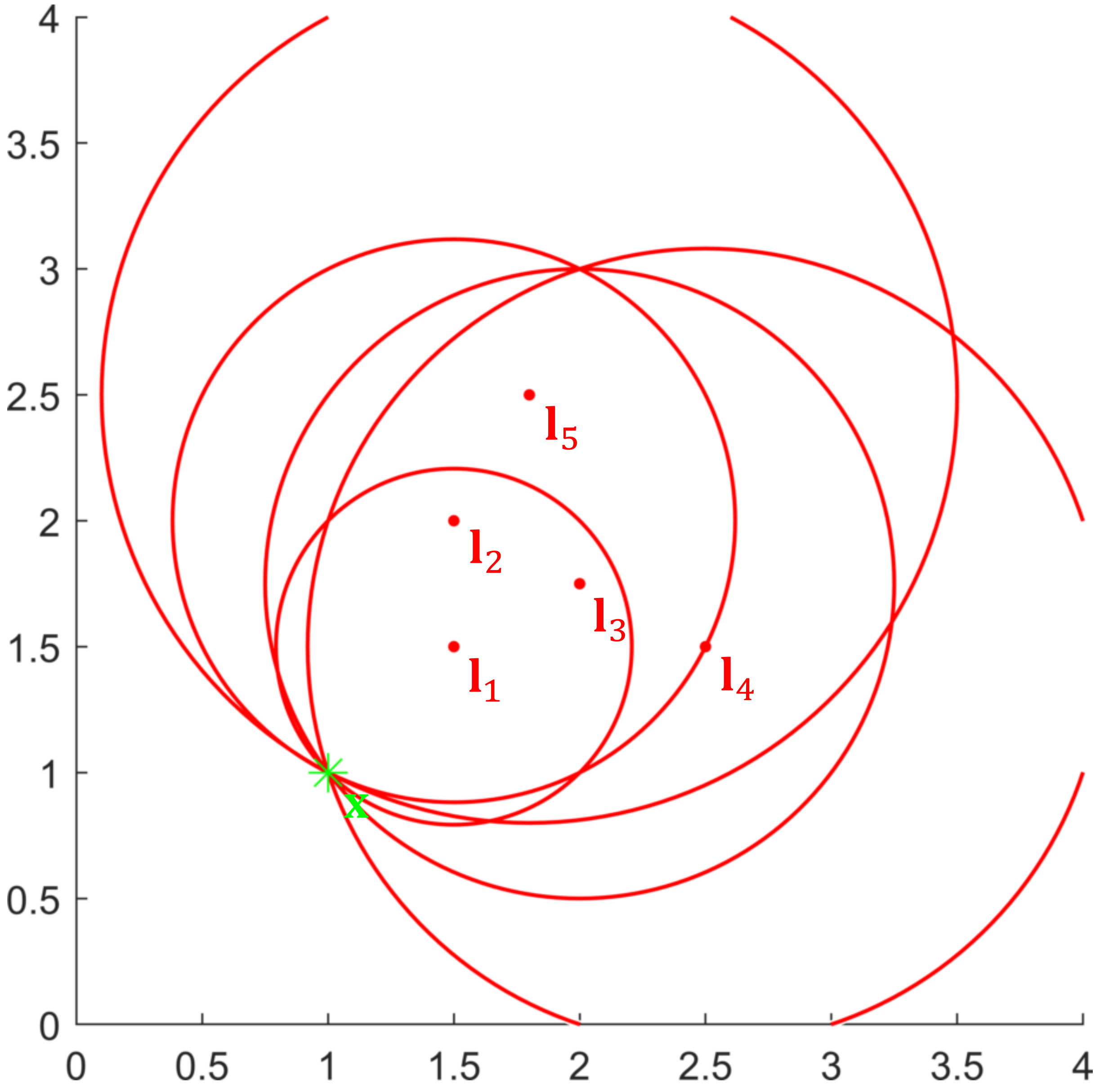
Example:
Range-based localization

States: Our location

$$X = \mathbf{x}$$

Measurements: Range to landmarks

$$Z = \{\rho_1, \ldots, \rho_m\}$$



Example:
Range-based localization

States: Our location

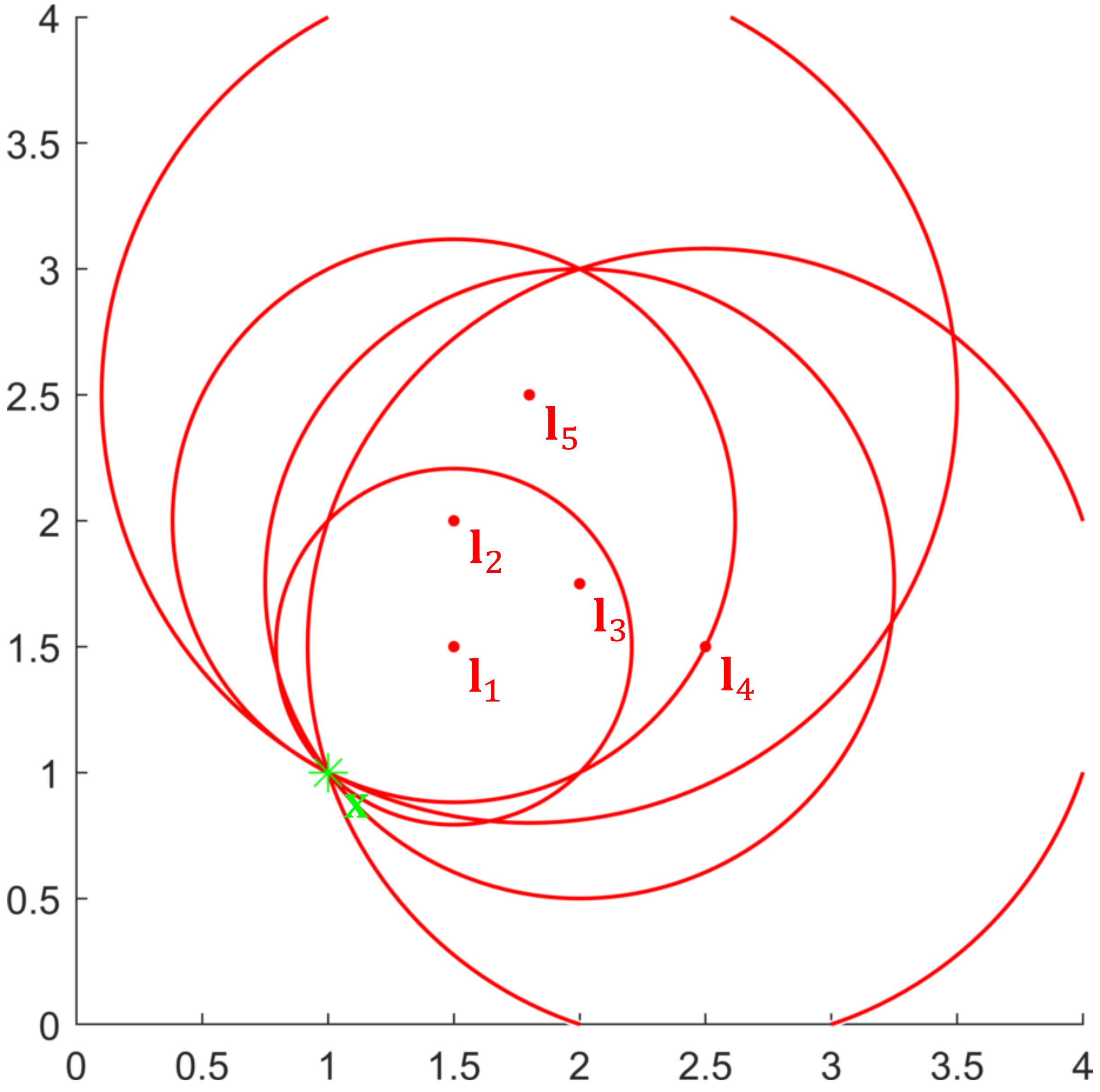
$$X = \mathbf{x}$$

Measurements: Range to landmarks

$$Z = \{\rho_1, \ldots, \rho_m\}$$

Measurement model:

$$\rho_i = \|\mathbf{x} - \mathbf{l}_i\| + \eta, \qquad \eta \sim N(0, \sigma^2)$$



Example:
Range-based localization

States: Our location

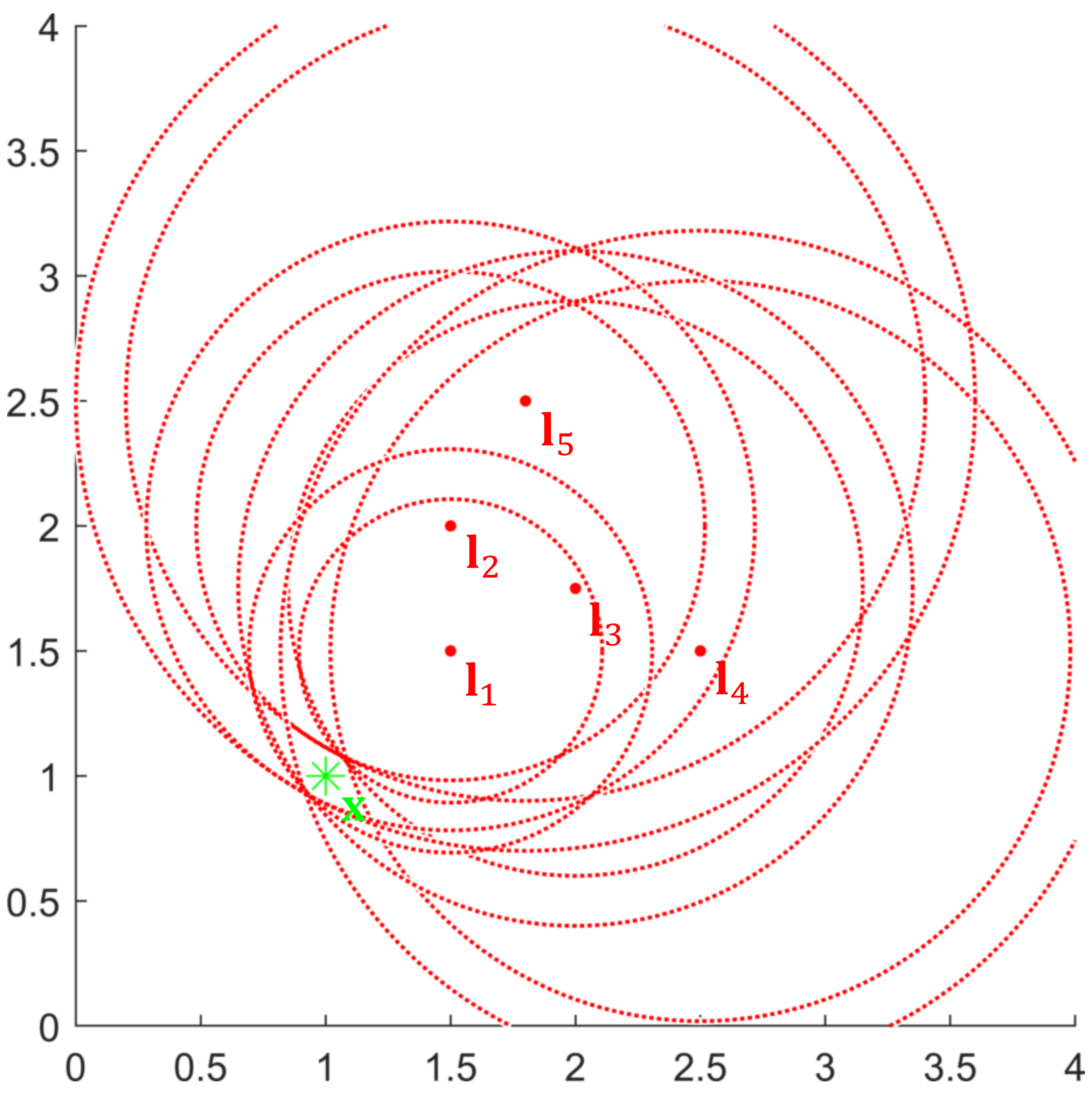
$$X = \mathbf{x}$$

Measurements: Range to landmarks

$$Z = \{\rho_1, \ldots, \rho_m\}$$

Measurement model:

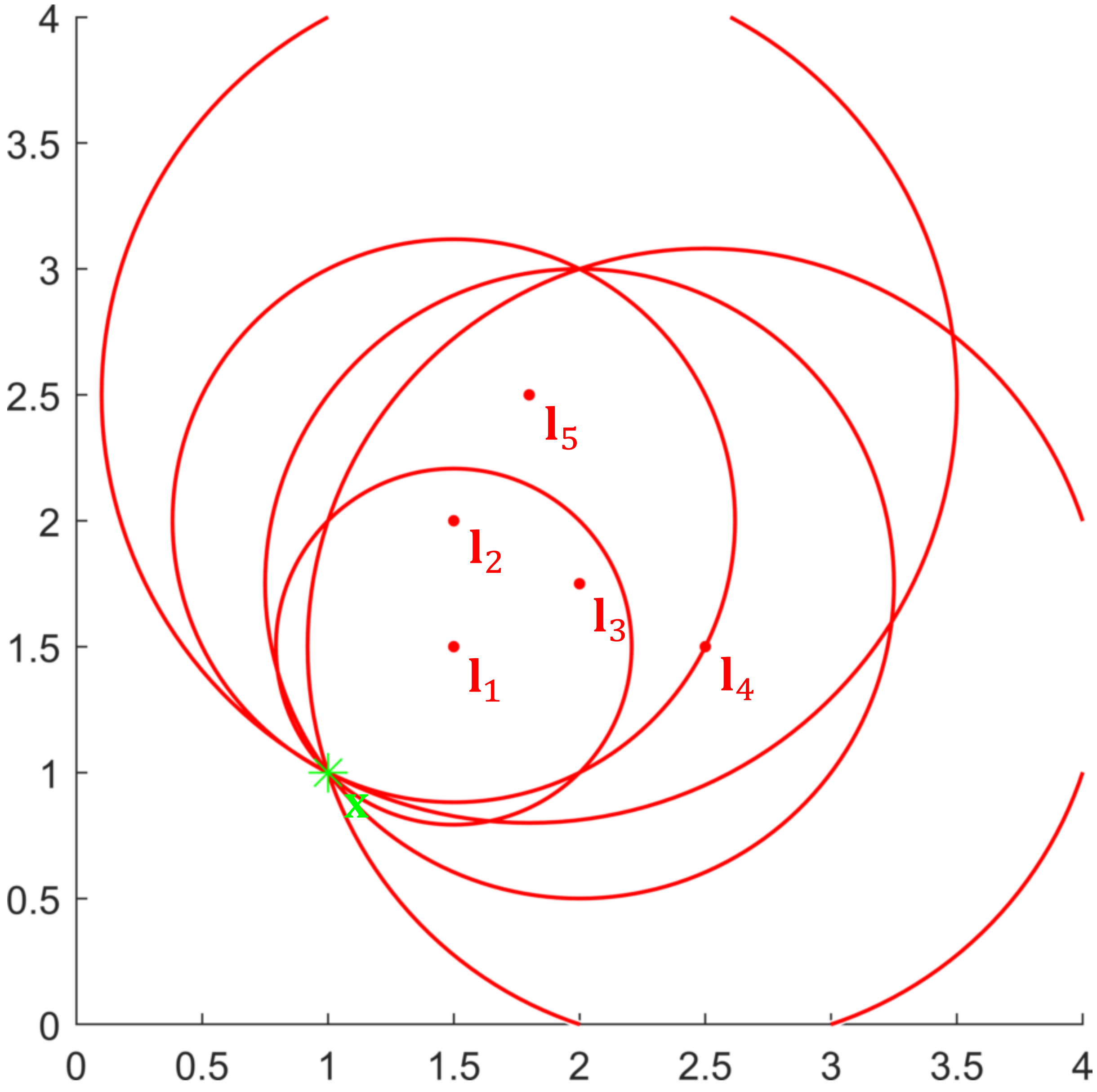
$$\rho_i = \|\mathbf{x} - \mathbf{l}_i\| + \eta, \qquad \eta \sim N(0, \sigma^2)$$



Example:
Range-based localization

Measurement prediction function:

$$\hat{\rho}_i = h(\mathbf{x};\mathbf{l}_i) = \left\|\mathbf{x} - \mathbf{l}_i\right\|$$



Example:
Range-based localization

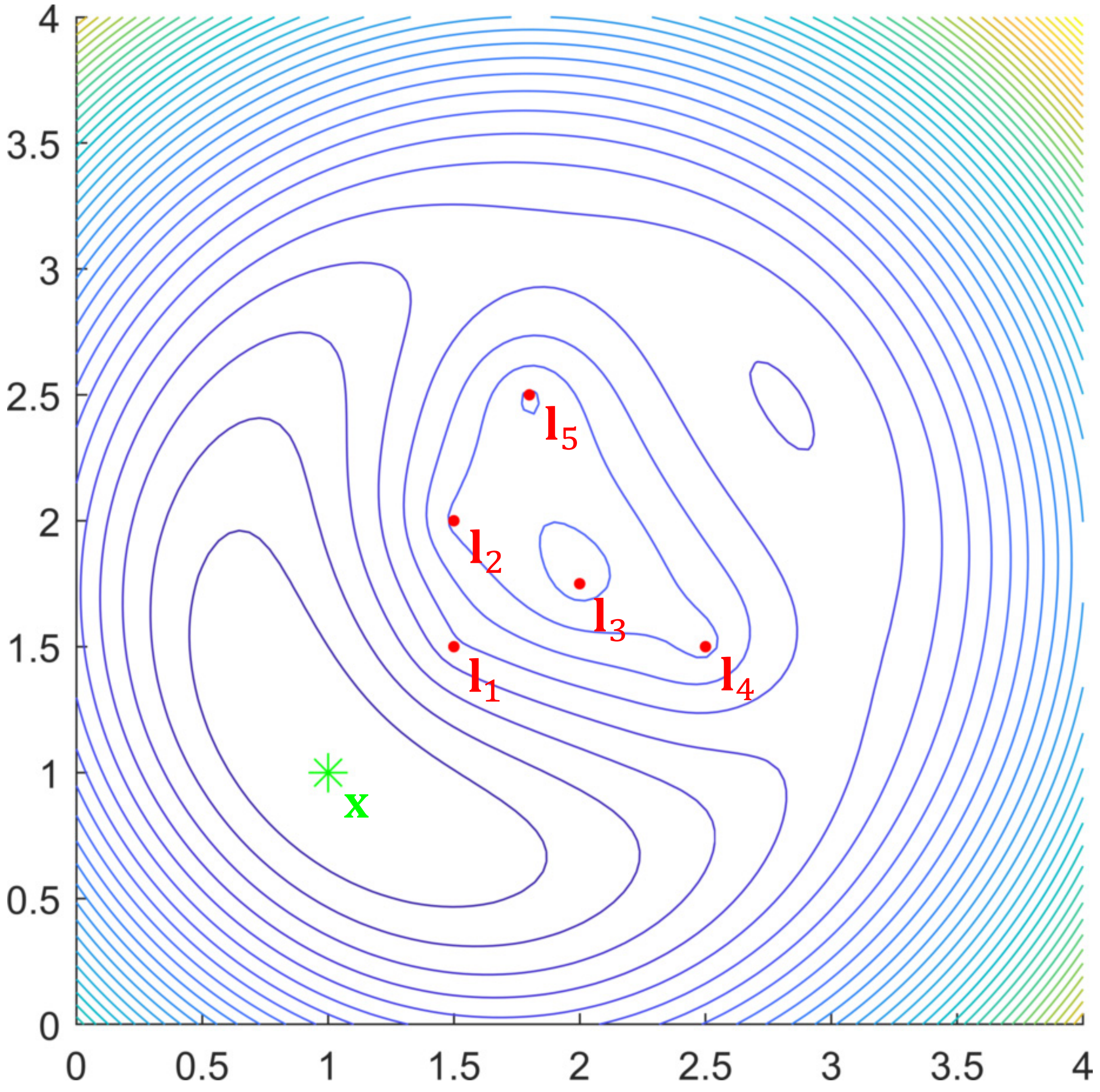
Measurement prediction function:

$$\hat{\rho}_i = h(\mathbf{x};\mathbf{l}_i) = \|\mathbf{x} - \mathbf{l}_i\|$$

Objective function:

$$f(\mathbf{x}) = \sum_{i=1}^{m} \|h(\mathbf{x};\mathbf{l}_i) - \rho_i\|^2$$

$$= \sum_{i=1}^{m} \left(\|\mathbf{x} - \mathbf{l}_i\| - \rho_i\right)^2$$



Example:
Range-based localization

Measurement prediction function:

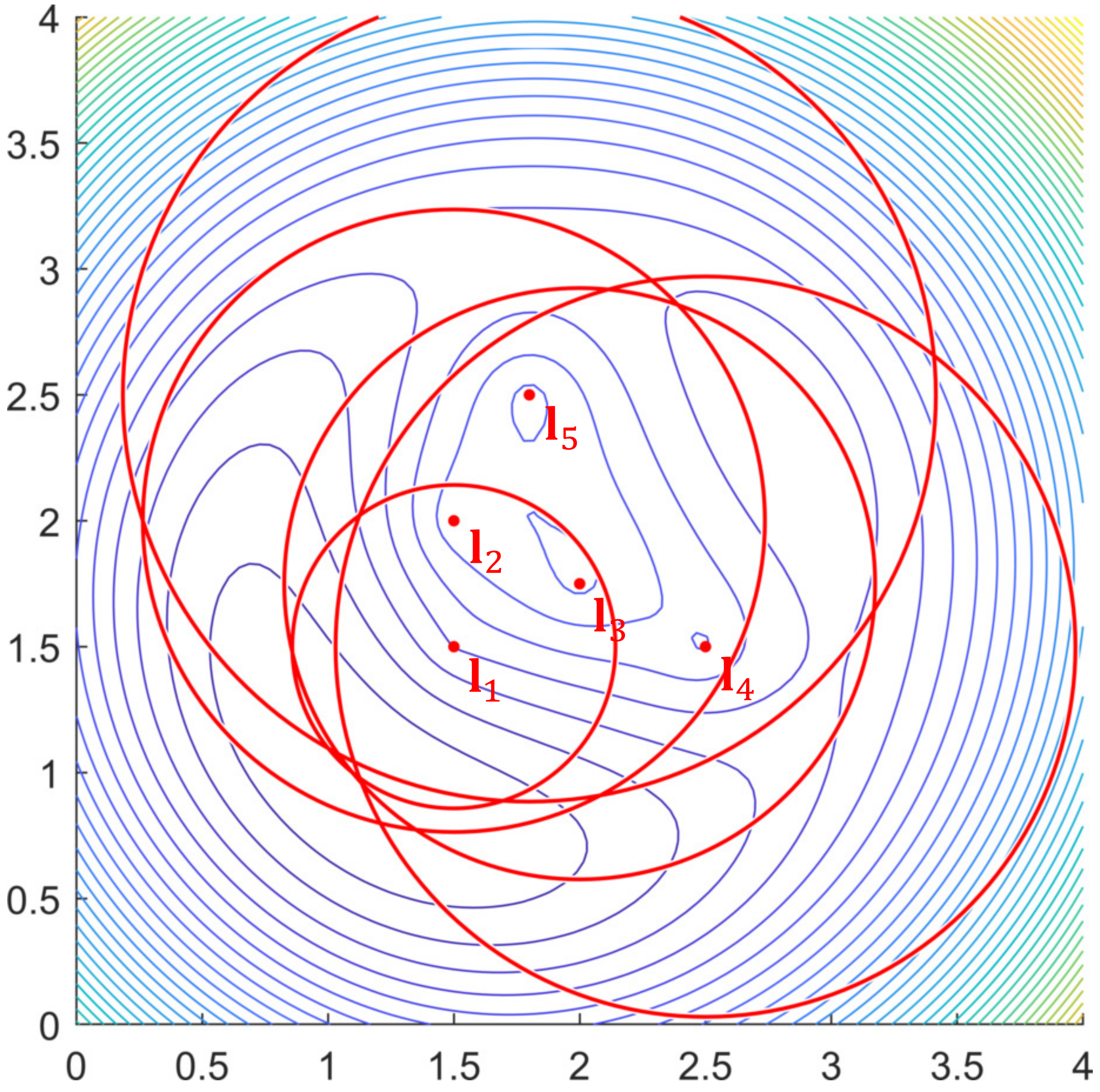
$$\hat{\rho}_i = h(\mathbf{x}; \mathbf{l}_i) = \left\| \mathbf{x} - \mathbf{l}_i \right\|$$

Objective function:

$$f(\mathbf{x}) = \sum_{i=1}^{m} \left\| h(\mathbf{x}; \mathbf{l}_i) - \rho_i \right\|^2$$
$$= \sum_{i=1}^{m} \left(\left\| \mathbf{x} - \mathbf{l}_i \right\| - \rho_i \right)^2$$

Nonlinear least squares problem:

$$\mathbf{x}^* = \underset{\mathbf{x}}{\operatorname{argmin}} \sum_{i=1}^{m} \left(\left\| \mathbf{x} - \mathbf{l}_i \right\| - \rho_i \right)^2$$



Linearization

We can linearize all **measurement prediction functions** $h_i(X_i)$ using a simple Taylor expansion at a suitable initial estimate X^0:

$$h_i(X_i) = h_i(X_i^0 + \mathbf{\Delta}_i) \approx h_i(X_i^0) + \mathbf{H}_i \mathbf{\Delta}_i$$

where the **measurement Jacobian** $\mathbf{H}_i$ is

$$\mathbf{H}_i \triangleq \left. \frac{\partial h_i(X_i)}{\partial X_i} \right|_{X_i^0}$$

and

$$\mathbf{\Delta}_i \triangleq X_i - X_i^0$$

is the **state update vector**.

Linearization

We can linearize all **measurement prediction functions** $h_i(X_i)$ using a simple Taylor expansion at a suitable initial estimate X^0:

$$h_i(X_i) = h_i(X_i^0 + \Delta_i) \approx h_i(X_i^0) + \mathbf{H}_i \Delta_i$$

where the **measurement Jacobian** $\mathbf{H}_i$ is

$$\mathbf{H}_i \triangleq \left. \frac{\partial h_i(X_i)}{\partial X_i} \right|_{X_i^0}$$

and

$$\Delta_i \triangleq X_i - X_i^0$$

is the **state update vector**.

The Jacobian matrix

$$f : \mathbb{R}^n \rightarrow \mathbb{R}^m \qquad \mathbf{x} \in \mathbb{R}^n \qquad f(\mathbf{x}) \in \mathbb{R}^m$$

$$\left. \frac{\partial f(\mathbf{x})}{\partial \mathbf{x}} \right|_{\mathbf{x}^t} = \begin{bmatrix} \frac{\partial f_1(\mathbf{x}^t)}{\partial x_1} & \cdots & \frac{\partial f_1(\mathbf{x}^t)}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m(\mathbf{x}^t)}{\partial x_1} & \cdots & \frac{\partial f_m(\mathbf{x}^t)}{\partial x_n} \end{bmatrix} \in \mathbb{R}^{m \times n}$$

Solving the linearized problem

This results in linear error functions $e_i(X_i^0 + \Delta)$,
and we obtain a **linear least squares** problem in the state update vector Δ:

$$
\begin{aligned}
\mathbf{\Delta}^* &= \underset{\mathbf{\Delta}}{\operatorname{argmin}} \sum_i \left\| h_i(X_i^0) + \mathbf{H}_i \mathbf{\Delta}_i - \mathbf{z}_i \right\|^2 \\
&= \underset{\mathbf{\Delta}}{\operatorname{argmin}} \sum_i \left\| \mathbf{H}_i \mathbf{\Delta}_i - \left\{ \mathbf{z}_i - h_i(X_i^0) \right\} \right\|^2 \\
&= \underset{\mathbf{\Delta}}{\operatorname{argmin}} \sum_i \left\| \mathbf{A}_i \mathbf{\Delta}_i - \mathbf{b}_i \right\|^2 \\
&= \underset{\mathbf{\Delta}}{\operatorname{argmin}} \left\| \mathbf{A} \mathbf{\Delta} - \mathbf{b} \right\|^2
\end{aligned}
$$

Which, as before, can be solved using the normal equations:

$$
\mathbf{A}^T \mathbf{A} \mathbf{\Delta}^* = \mathbf{A}^T \mathbf{b}
$$

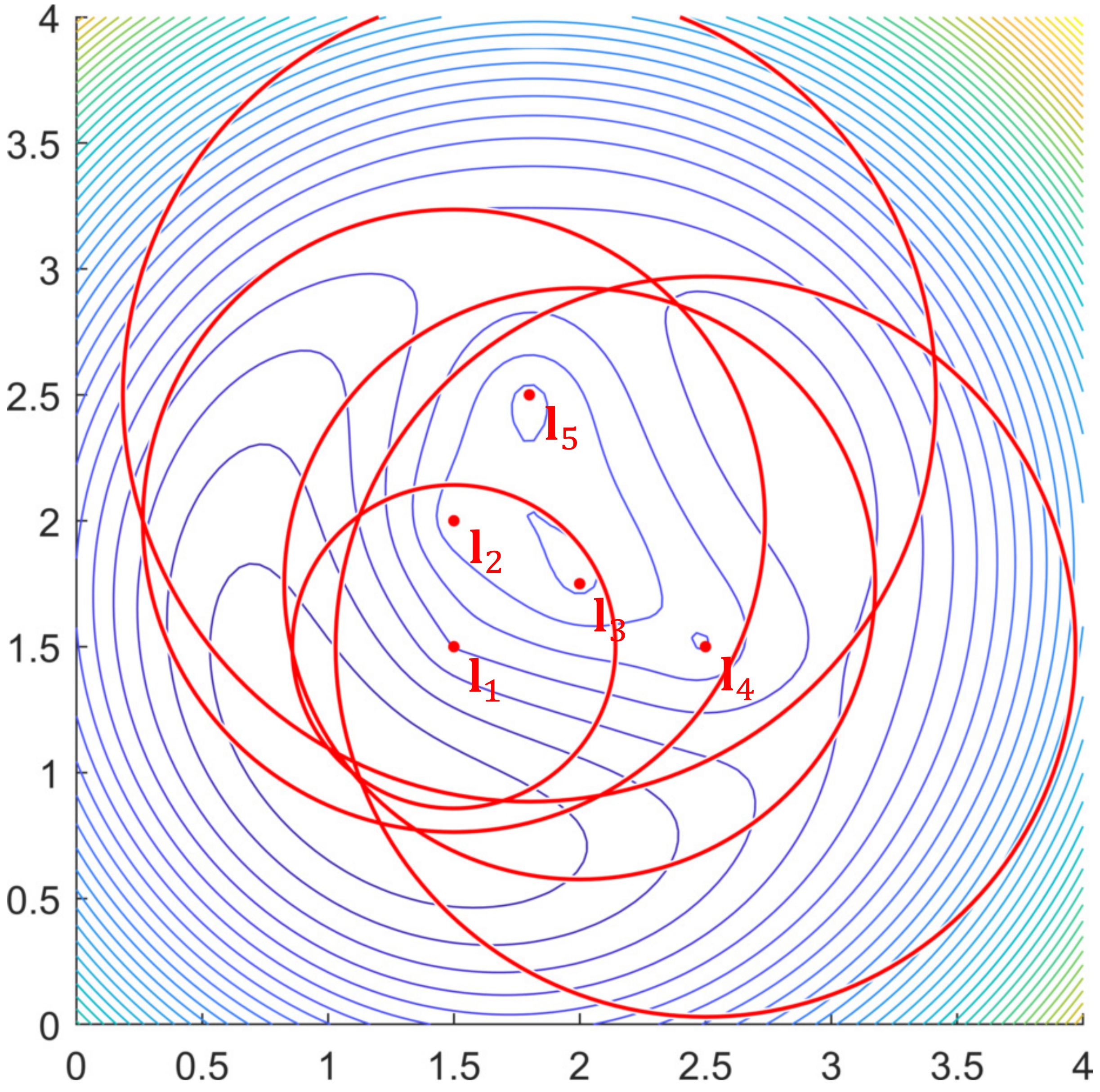
Example:
Range-based localization

Measurement prediction function:

$$\hat{\rho}_i = h(\mathbf{x}; \mathbf{l}_i) = \left\| \mathbf{x} - \mathbf{l}_i \right\|$$

Nonlinear least squares problem:

$$\mathbf{x}^* = \underset{\mathbf{x}}{\operatorname{argmin}} \sum_{i=1}^{m} \left(h(\mathbf{x}; \mathbf{l}_i) - \rho_i \right)^2$$



Example: Range-based localization

Nonlinear least squares problem:

$$\mathbf{x}^* = \underset{\mathbf{x}}{\operatorname{argmin}} \sum_{i=1}^{m} \left(h(\mathbf{x};\mathbf{l}_i) - \rho_i \right)^2$$

Linearized problem at $\mathbf{x}^0$:

$$\boldsymbol{\delta}^* = \underset{\boldsymbol{\delta}}{\operatorname{argmin}} \sum_{i=1}^{m} \left(h(\mathbf{x}^0;\mathbf{l}_i) + \mathbf{H}_i\boldsymbol{\delta} - \rho_i \right)^2$$

$$h(\mathbf{x};\mathbf{l}_i) = \left\| \mathbf{x} - \mathbf{l}_i \right\|$$

$$\mathbf{H}_i = \begin{bmatrix} \dfrac{\partial h(\mathbf{x}^0)}{\partial x_1} & \dfrac{\partial h(\mathbf{x}^0)}{\partial x_2} \end{bmatrix}$$

$$= \begin{bmatrix} \dfrac{x_1^0 - l_{i,1}}{\left\| \mathbf{x}^0 - \mathbf{l}_i \right\|} & \dfrac{x_2^0 - l_{i,2}}{\left\| \mathbf{x}^0 - \mathbf{l}_i \right\|} \end{bmatrix}$$

$$= \frac{\left(\mathbf{x}^0 - \mathbf{l}_i \right)^T}{\left\| \mathbf{x}^0 - \mathbf{l}_i \right\|}$$

Example:
Range-based localization

Nonlinear least squares problem:

$$\mathbf{x}^* = \underset{\mathbf{x}}{\operatorname{argmin}} \sum_{i=1}^{m} \left(h(\mathbf{x};\mathbf{l}_i) - \rho_i \right)^2$$

Linearized problem at $\mathbf{x}^0$:

$$\begin{aligned}
\boldsymbol{\delta}^* &= \underset{\boldsymbol{\delta}}{\operatorname{argmin}} \sum_{i=1}^{m} \left(h(\mathbf{x}^0;\mathbf{l}_i) + \mathbf{H}_i\boldsymbol{\delta} - \rho_i \right)^2 \\
&= \underset{\boldsymbol{\delta}}{\operatorname{argmin}} \sum_{i=1}^{m} \left(\mathbf{H}_i\boldsymbol{\delta} - \left\{ \rho_i - h(\mathbf{x}^0;\mathbf{l}_i) \right\} \right)^2
\end{aligned}$$

Example:
Range-based localization

Nonlinear least squares problem:

$$\mathbf{x}^* = \underset{\mathbf{x}}{\operatorname{argmin}} \sum_{i=1}^{m} \left(h(\mathbf{x}; \mathbf{l}_i) - \rho_i \right)^2$$

Linearized problem at $\mathbf{x}^0$:

$$\begin{aligned}
\boldsymbol{\delta}^* &= \underset{\boldsymbol{\delta}}{\operatorname{argmin}} \sum_{i=1}^{m} \left(h(\mathbf{x}^0; \mathbf{l}_i) + \mathbf{H}_i \boldsymbol{\delta} - \rho_i \right)^2 \\
&= \underset{\boldsymbol{\delta}}{\operatorname{argmin}} \sum_{i=1}^{m} \left(\mathbf{H}_i \boldsymbol{\delta} - \left\{ \rho_i - h(\mathbf{x}^0; \mathbf{l}_i) \right\} \right)^2 \\
&= \underset{\boldsymbol{\delta}}{\operatorname{argmin}} \sum_{i=1}^{m} \left(\mathbf{A}_i \boldsymbol{\delta} - \mathbf{b}_i \right)^2
\end{aligned}$$

Example:
Range-based localization

Nonlinear least squares problem:

$$\mathbf{x}^* = \underset{\mathbf{x}}{\operatorname{argmin}} \sum_{i=1}^{m} \left(h(\mathbf{x};\mathbf{l}_i) - \rho_i \right)^2$$

Linearized problem at $\mathbf{x}^0$:

$$\boldsymbol{\delta}^* = \underset{\boldsymbol{\delta}}{\operatorname{argmin}} \sum_{i=1}^{m} \left(h(\mathbf{x}^0;\mathbf{l}_i) + \mathbf{H}_i\boldsymbol{\delta} - \rho_i \right)^2$$

$$= \underset{\boldsymbol{\delta}}{\operatorname{argmin}} \sum_{i=1}^{m} \left(\mathbf{H}_i\boldsymbol{\delta} - \left\{ \rho_i - h(\mathbf{x}^0;\mathbf{l}_i) \right\} \right)^2$$

$$= \underset{\boldsymbol{\delta}}{\operatorname{argmin}} \sum_{i=1}^{m} \left(\mathbf{A}_i\boldsymbol{\delta} - \mathbf{b}_i \right)^2$$

$$\mathbf{l}_i = \left\{ \begin{bmatrix} 1.50 \\ 1.50 \end{bmatrix}, \begin{bmatrix} 1.50 \\ 2.00 \end{bmatrix}, \begin{bmatrix} 2.00 \\ 1.75 \end{bmatrix}, \begin{bmatrix} 2.50 \\ 1.50 \end{bmatrix}, \begin{bmatrix} 1.80 \\ 2.50 \end{bmatrix} \right\}$$

$$\rho_i = \left\{ 0.64, 1.23, 1.17, 1.47, 1.61 \right\}$$

$$\mathbf{x}^0 = \begin{bmatrix} 1.80 \\ 3.50 \end{bmatrix}$$

Example:
Range-based localization

Nonlinear least squares problem:

$$\mathbf{x}^* = \underset{\mathbf{x}}{\operatorname{argmin}} \sum_{i=1}^{m} \left(h(\mathbf{x};\mathbf{l}_i) - \rho_i \right)^2$$

Linearized problem at $\mathbf{x}^0$:

$$\begin{aligned} \boldsymbol{\delta}^* &= \underset{\boldsymbol{\delta}}{\operatorname{argmin}} \sum_{i=1}^{m} \left(h(\mathbf{x}^0;\mathbf{l}_i) + \mathbf{H}_i\boldsymbol{\delta} - \rho_i \right)^2 \\ &= \underset{\boldsymbol{\delta}}{\operatorname{argmin}} \sum_{i=1}^{m} \left(\mathbf{H}_i\boldsymbol{\delta} - \left\{ \rho_i - h(\mathbf{x}^0;\mathbf{l}_i) \right\} \right)^2 \\ &= \underset{\boldsymbol{\delta}}{\operatorname{argmin}} \sum_{i=1}^{m} \left(\mathbf{A}_i\boldsymbol{\delta} - \mathbf{b}_i \right)^2 \end{aligned}$$

$$\mathbf{l}_i = \left\{ \begin{bmatrix} 1.50 \\ 1.50 \end{bmatrix}, \begin{bmatrix} 1.50 \\ 2.00 \end{bmatrix}, \begin{bmatrix} 2.00 \\ 1.75 \end{bmatrix}, \begin{bmatrix} 2.50 \\ 1.50 \end{bmatrix}, \begin{bmatrix} 1.80 \\ 2.50 \end{bmatrix} \right\}$$

$$\rho_i = \left\{ 0.64, 1.23, 1.17, 1.47, 1.61 \right\}$$

$$\mathbf{x}^0 = \begin{bmatrix} 1.80 \\ 3.50 \end{bmatrix}$$

$$\mathbf{A}_1 = \mathbf{H}_1 = \frac{\left(\mathbf{x}^0 - \mathbf{l}_1\right)^T}{\left\|\mathbf{x}^0 - \mathbf{l}_1\right\|} = \frac{\left(\begin{bmatrix} 0.30 \\ 2.00 \end{bmatrix}\right)^T}{\left\|\begin{bmatrix} 0.30 \\ 2.00 \end{bmatrix}\right\|}$$

Example: Range-based localization

Nonlinear least squares problem:

$$\mathbf{x}^* = \underset{\mathbf{x}}{\operatorname{argmin}} \sum_{i=1}^{m} \left(h(\mathbf{x};\mathbf{l}_i) - \rho_i \right)^2$$

Linearized problem at $\mathbf{x}^0$:

$$\begin{aligned}\boldsymbol{\delta}^* &= \underset{\boldsymbol{\delta}}{\operatorname{argmin}} \sum_{i=1}^{m} \left(h(\mathbf{x}^0;\mathbf{l}_i) + \mathbf{H}_i\boldsymbol{\delta} - \rho_i \right)^2 \\ &= \underset{\boldsymbol{\delta}}{\operatorname{argmin}} \sum_{i=1}^{m} \left(\mathbf{H}_i\boldsymbol{\delta} - \left\{ \rho_i - h(\mathbf{x}^0;\mathbf{l}_i) \right\} \right)^2 \\ &= \underset{\boldsymbol{\delta}}{\operatorname{argmin}} \sum_{i=1}^{m} \left(\mathbf{A}_i\boldsymbol{\delta} - \mathbf{b}_i \right)^2\end{aligned}$$

$$\mathbf{l}_i = \left\{ \begin{bmatrix} 1.50 \\ 1.50 \end{bmatrix}, \begin{bmatrix} 1.50 \\ 2.00 \end{bmatrix}, \begin{bmatrix} 2.00 \\ 1.75 \end{bmatrix}, \begin{bmatrix} 2.50 \\ 1.50 \end{bmatrix}, \begin{bmatrix} 1.80 \\ 2.50 \end{bmatrix} \right\}$$

$$\rho_i = \left\{ 0.64, 1.23, 1.17, 1.47, 1.61 \right\}$$

$$\mathbf{x}^0 = \begin{bmatrix} 1.80 \\ 3.50 \end{bmatrix}$$

$$\mathbf{A}_1 = \mathbf{H}_1 = \frac{\left(\mathbf{x}^0 - \mathbf{l}_1\right)^T}{\left\|\mathbf{x}^0 - \mathbf{l}_1\right\|} = \frac{\left(\begin{bmatrix} 0.30 \\ 2.00 \end{bmatrix}\right)^T}{\left\|\begin{bmatrix} 0.30 \\ 2.00 \end{bmatrix}\right\|}$$

$$= \frac{\begin{bmatrix} 0.30 & 2.00 \end{bmatrix}}{2.02} = \begin{bmatrix} 0.15 & 0.99 \end{bmatrix}$$

Example:
Range-based localization

Nonlinear least squares problem:

$$\mathbf{x}^* = \underset{\mathbf{x}}{\operatorname{argmin}} \sum_{i=1}^{m} \left(h(\mathbf{x}; \mathbf{l}_i) - \rho_i \right)^2$$

Linearized problem at $\mathbf{x}^0$:

$$\begin{aligned} \boldsymbol{\delta}^* &= \underset{\boldsymbol{\delta}}{\operatorname{argmin}} \sum_{i=1}^{m} \left(h(\mathbf{x}^0; \mathbf{l}_i) + \mathbf{H}_i \boldsymbol{\delta} - \rho_i \right)^2 \\ &= \underset{\boldsymbol{\delta}}{\operatorname{argmin}} \sum_{i=1}^{m} \left(\mathbf{H}_i \boldsymbol{\delta} - \left\{ \rho_i - h(\mathbf{x}^0; \mathbf{l}_i) \right\} \right)^2 \\ &= \underset{\boldsymbol{\delta}}{\operatorname{argmin}} \sum_{i=1}^{m} \left(\mathbf{A}_i \boldsymbol{\delta} - \mathbf{b}_i \right)^2 \end{aligned}$$

$$\mathbf{l}_i = \left\{ \begin{bmatrix} 1.50 \\ 1.50 \end{bmatrix}, \begin{bmatrix} 1.50 \\ 2.00 \end{bmatrix}, \begin{bmatrix} 2.00 \\ 1.75 \end{bmatrix}, \begin{bmatrix} 2.50 \\ 1.50 \end{bmatrix}, \begin{bmatrix} 1.80 \\ 2.50 \end{bmatrix} \right\}$$

$$\rho_i = \{0.64, 1.23, 1.17, 1.47, 1.61\}$$

$$\mathbf{x}^0 = \begin{bmatrix} 1.80 \\ 3.50 \end{bmatrix}$$

$$\mathbf{A}_1 = \mathbf{H}_1 = \frac{\left(\mathbf{x}^0 - \mathbf{l}_1\right)^T}{\left\| \mathbf{x}^0 - \mathbf{l}_1 \right\|} = \frac{\left(\begin{bmatrix} 0.30 \\ 2.00 \end{bmatrix} \right)^T}{\left\| \begin{bmatrix} 0.30 \\ 2.00 \end{bmatrix} \right\|}$$

$$= \frac{\begin{bmatrix} 0.30 & 2.00 \end{bmatrix}}{2.02} = \begin{bmatrix} 0.15 & 0.99 \end{bmatrix}$$

$$\mathbf{b}_1 = \rho_1 - h(\mathbf{x}^0; \mathbf{l}_1) = 0.64 - 2.02 = -1.38$$

Example: Range-based localization

Nonlinear least squares problem:

$$\mathbf{x}^* = \underset{\mathbf{x}}{\operatorname{argmin}} \sum_{i=1}^{m} \left(h(\mathbf{x};\mathbf{l}_i) - \rho_i \right)^2$$

Linearized problem at $\mathbf{x}^0$:

$$\begin{aligned}
\boldsymbol{\delta}^* &= \underset{\boldsymbol{\delta}}{\operatorname{argmin}} \sum_{i=1}^{m} \left(h(\mathbf{x}^0;\mathbf{l}_i) + \mathbf{H}_i\boldsymbol{\delta} - \rho_i \right)^2 \\
&= \underset{\boldsymbol{\delta}}{\operatorname{argmin}} \sum_{i=1}^{m} \left(\mathbf{H}_i\boldsymbol{\delta} - \left\{ \rho_i - h(\mathbf{x}^0;\mathbf{l}_i) \right\} \right)^2 \\
&= \underset{\boldsymbol{\delta}}{\operatorname{argmin}} \sum_{i=1}^{m} \left(\mathbf{A}_i\boldsymbol{\delta} - \mathbf{b}_i \right)^2 \\
&= \underset{\boldsymbol{\delta}}{\operatorname{argmin}} \left\| \mathbf{A}\boldsymbol{\delta} - \mathbf{b} \right\|^2
\end{aligned}$$

$$\mathbf{l}_i = \left\{ \begin{bmatrix} 1.50 \\ 1.50 \end{bmatrix}, \begin{bmatrix} 1.50 \\ 2.00 \end{bmatrix}, \begin{bmatrix} 2.00 \\ 1.75 \end{bmatrix}, \begin{bmatrix} 2.50 \\ 1.50 \end{bmatrix}, \begin{bmatrix} 1.80 \\ 2.50 \end{bmatrix} \right\}$$

$$\rho_i = \{0.64, 1.23, 1.17, 1.47, 1.61\}$$

$$\mathbf{x}^0 = \begin{bmatrix} 1.80 \\ 3.50 \end{bmatrix}$$

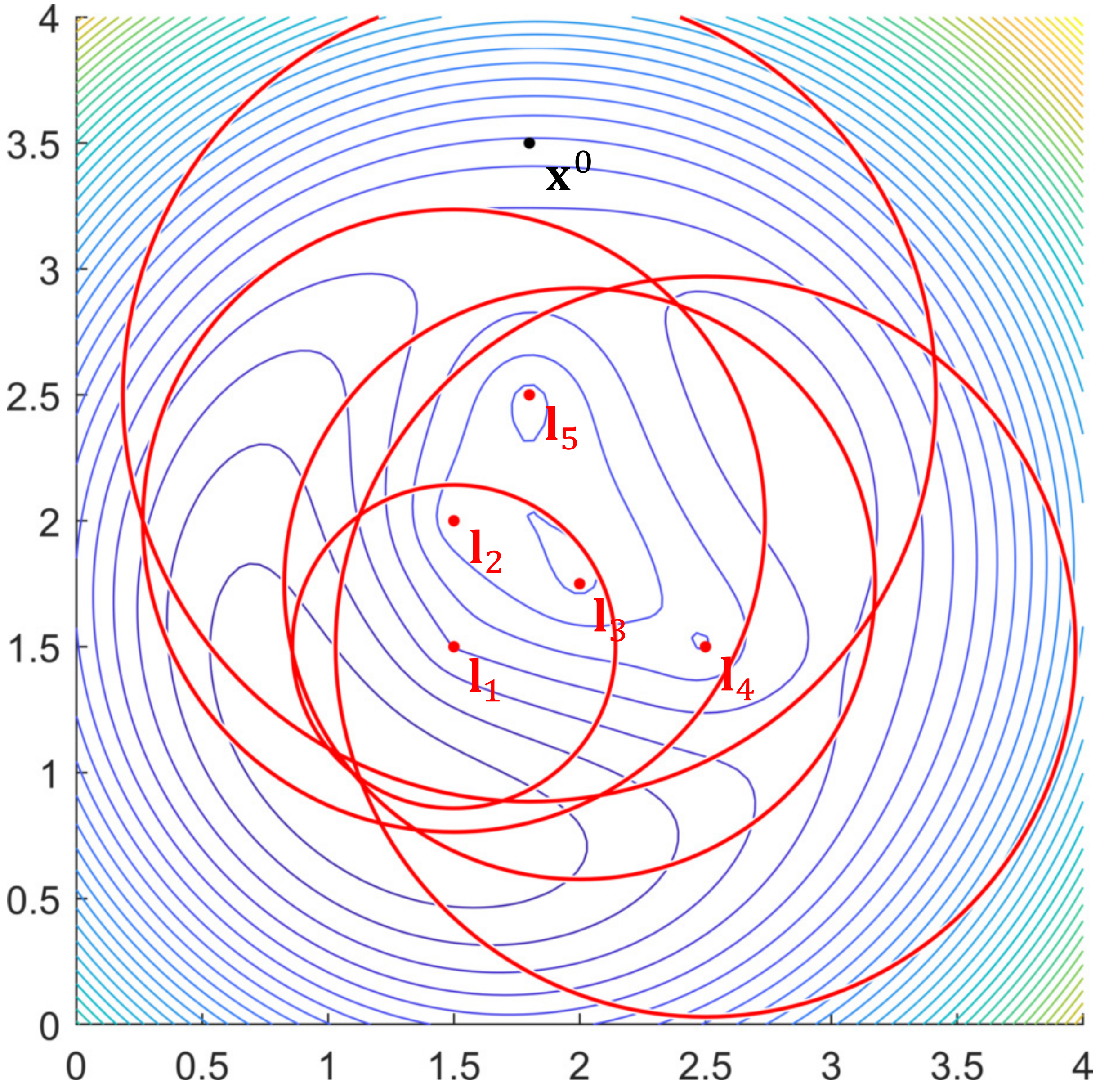
$$\mathbf{A} = \begin{bmatrix} 0.15 & 0.99 \\ 0.20 & 0.98 \\ -0.11 & 0.99 \\ -0.33 & 0.94 \\ 0 & 1.00 \end{bmatrix} \qquad \mathbf{b} = \begin{bmatrix} -1.38 \\ -0.29 \\ -0.59 \\ -0.65 \\ 0.62 \end{bmatrix}$$

Example: Range-based localization

Linearized problem at $\mathbf{x}^0$:

$$\boldsymbol{\delta}^* = \underset{\boldsymbol{\delta}}{\operatorname{argmin}} \left\| \mathbf{A}\boldsymbol{\delta} - \mathbf{b} \right\|^2$$

$$\mathbf{A} = \begin{bmatrix} 0.15 & 0.99 \\ 0.20 & 0.98 \\ -0.11 & 0.99 \\ -0.33 & 0.94 \\ 0 & 1.00 \end{bmatrix} \qquad \mathbf{b} = \begin{bmatrix} -1.38 \\ -0.29 \\ -0.59 \\ -0.65 \\ 0.62 \end{bmatrix}$$



Example:
Range-based localization

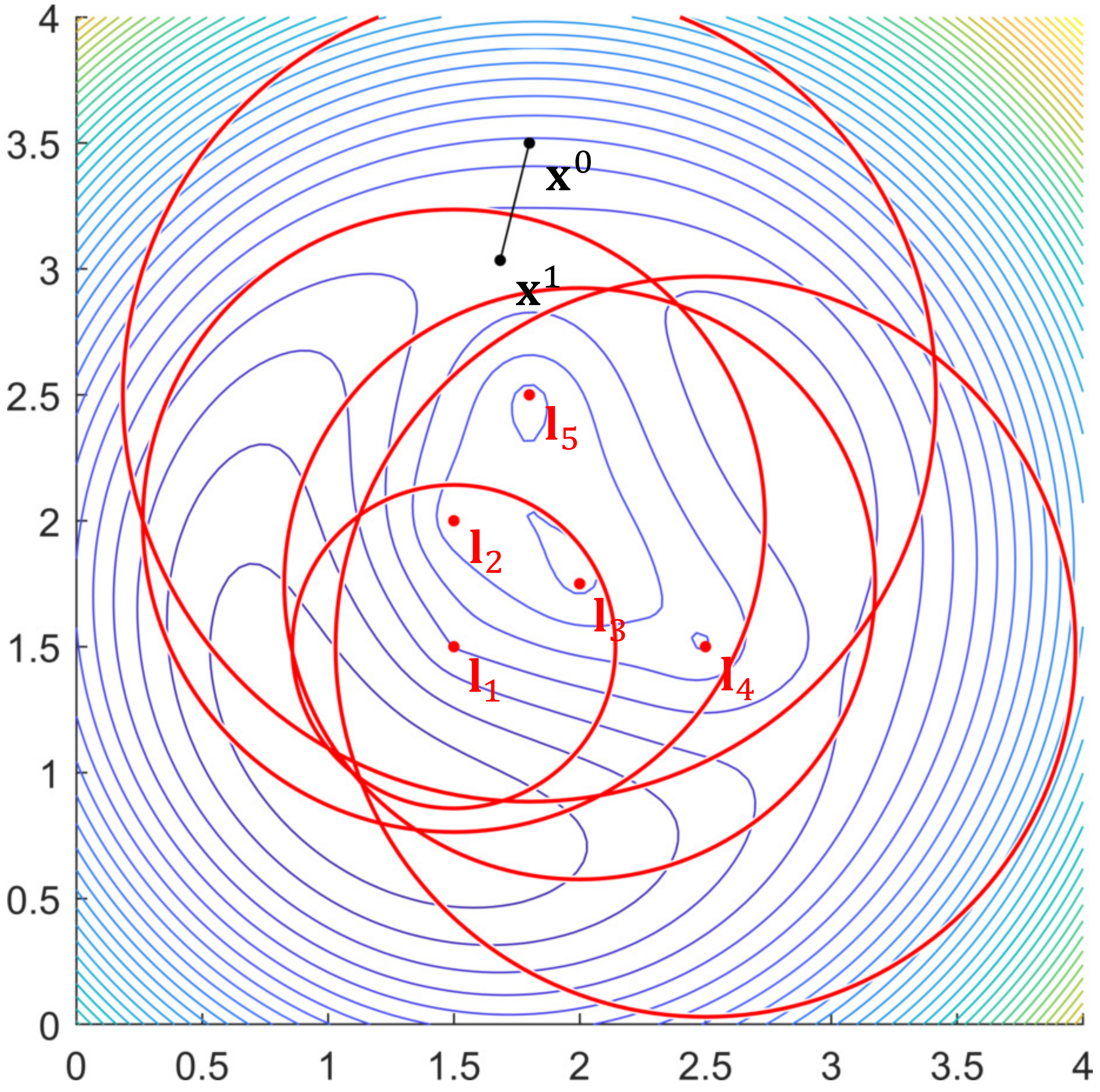
Linearized problem at $\mathbf{x}^0$:

$$\boldsymbol{\delta}^* = \underset{\boldsymbol{\delta}}{\operatorname{argmin}} \left\| \mathbf{A}\boldsymbol{\delta} - \mathbf{b} \right\|^2$$

$$\mathbf{A} = \begin{bmatrix} 0.15 & 0.99 \\ 0.20 & 0.98 \\ -0.11 & 0.99 \\ -0.33 & 0.94 \\ 0 & 1.00 \end{bmatrix} \qquad \mathbf{b} = \begin{bmatrix} -1.38 \\ -0.29 \\ -0.59 \\ -0.65 \\ 0.62 \end{bmatrix}$$

Solution to the normal equations $\mathbf{A}^T \mathbf{A} \boldsymbol{\delta}^* = \mathbf{A}^T \mathbf{b}$:

$$\boldsymbol{\delta}^* = \begin{bmatrix} -0.12 \\ -0.47 \end{bmatrix} \qquad \mathbf{x}^1 = \mathbf{x}^0 + \boldsymbol{\delta}^* = \begin{bmatrix} 1.68 \\ 3.03 \end{bmatrix}$$



Solving the nonlinear problem

We solve the nonlinear least-squares problem
by iteratively solving the linearized system:

$$\text{Choose a suitable inital estimate } X^0$$

$$\mathbf{A}, \mathbf{b} \leftarrow \text{Linearize at } X^t$$

$$\mathbf{\Delta}^* \leftarrow \text{Solve } \underset{\mathbf{\Delta}}{\operatorname{argmin}} \left\| \mathbf{A}\mathbf{\Delta} - \mathbf{b} \right\|^2$$

$$X^{t+1} \leftarrow X^t + \mathbf{\Delta}^*$$

The Gauss-Newton algorithm

Given an objective $f(X)$ and a good initial estimate X^0.

For $t = 0, 1, \dots, t^{max}$

$\mathbf{A}, \mathbf{b} \leftarrow$ Linearize $f(X)$ at X^t

$\mathbf{\Delta} \leftarrow$ Solve the linearized problem with $\mathbf{A}^T\mathbf{A}\mathbf{\Delta} = \mathbf{A}^T\mathbf{b}$

$X^{t+1} = X^t + \mathbf{\Delta}$

Terminate early if $f(X)$ is very small or $X^{t+1} \approx X^t$

The Gauss-Newton algorithm

Gauss-Newton actually approximates the Hessian of the objective $f(X)$ as

$$\left.\frac{\partial^2 f(\mathbf{x})}{\partial \mathbf{x} \partial \mathbf{x}^T}\right|_{\mathbf{x}^t} = \left(\left.\frac{\partial e(\mathbf{x})}{\partial \mathbf{x}}\right|_{\mathbf{x}^t} \right)^T \left(\left.\frac{\partial e(\mathbf{x})}{\partial \mathbf{x}}\right|_{\mathbf{x}^t} \right) + \sum_{i=1}^{m} e_i(\mathbf{x}^t) \left(\left.\frac{\partial^2 e_i(\mathbf{x})}{\partial \mathbf{x} \partial \mathbf{x}^T}\right|_{\mathbf{x}^t} \right) = \mathbf{A}^T \mathbf{A} + \mathbf{Q} \approx \mathbf{A}^T \mathbf{A}$$

This approximation is good if we are near the solution and the objective is nearly quadratic.

The Gauss-Newton algorithm

Gauss-Newton actually approximates the Hessian of the objective $f(X)$ as

$$\left.\frac{\partial^2 f(\mathbf{x})}{\partial\mathbf{x}\partial\mathbf{x}^T}\right|_{\mathbf{x}^t} = \left(\left.\frac{\partial e(\mathbf{x})}{\partial\mathbf{x}}\right|_{\mathbf{x}^t}\right)^T \left(\left.\frac{\partial e(\mathbf{x})}{\partial\mathbf{x}}\right|_{\mathbf{x}^t}\right) + \sum_{i=1}^{m} e_i(\mathbf{x}^t)\left(\left.\frac{\partial^2 e_i(\mathbf{x})}{\partial\mathbf{x}\partial\mathbf{x}^T}\right|_{\mathbf{x}^t}\right) = \mathbf{A}^T\mathbf{A} + \mathbf{Q} \approx \mathbf{A}^T\mathbf{A}$$

This approximation is good if we are near the solution and the objective is nearly quadratic.

When the approximation is good:

- The update direction is good
- The update step length is good
- We obtain almost quadratic convergence to a local minimum

The Gauss-Newton algorithm

Gauss-Newton actually approximates the Hessian of the objective $f(X)$ as

$$\left.\frac{\partial^2 f(\mathbf{x})}{\partial \mathbf{x} \partial \mathbf{x}^T}\right|_{\mathbf{x}^t} = \left(\left.\frac{\partial e(\mathbf{x})}{\partial \mathbf{x}}\right|_{\mathbf{x}^t}\right)^T \left(\left.\frac{\partial e(\mathbf{x})}{\partial \mathbf{x}}\right|_{\mathbf{x}^t}\right) + \sum_{i=1}^{m} e_i(\mathbf{x}^t) \left(\left.\frac{\partial^2 e_i(\mathbf{x})}{\partial \mathbf{x} \partial \mathbf{x}^T}\right|_{\mathbf{x}^t}\right) = \mathbf{A}^T \mathbf{A} + \mathbf{Q} \approx \mathbf{A}^T \mathbf{A}$$

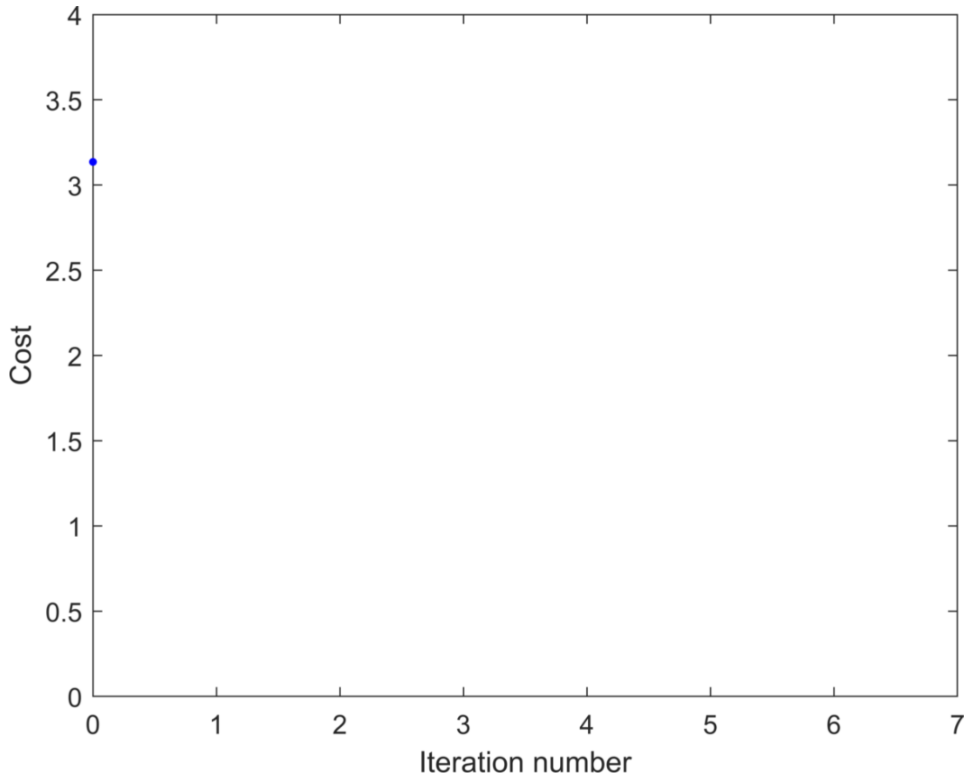
This approximation is good if we are near the solution and the objective is nearly quadratic.

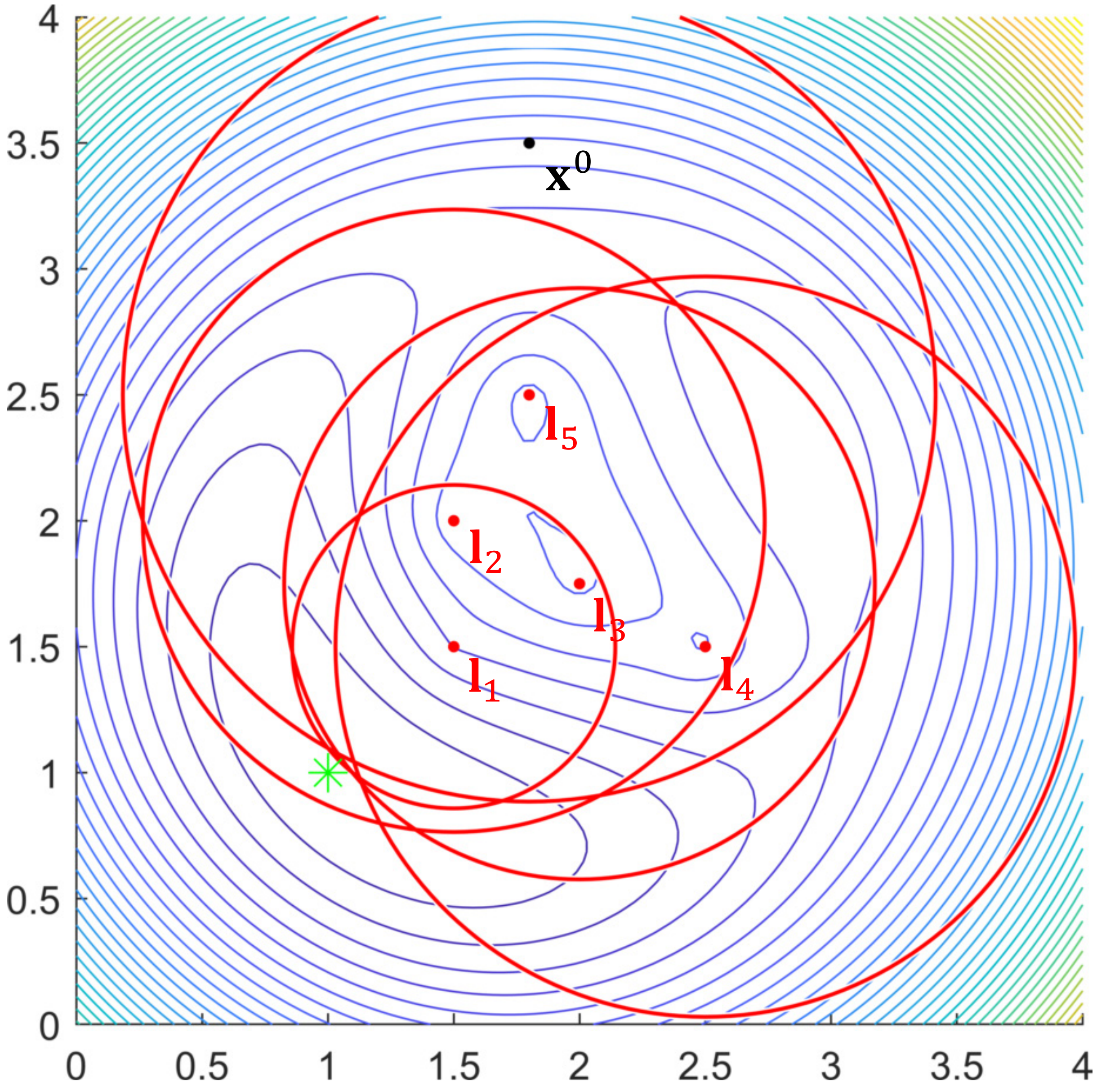
When the approximation is poor:

- The update direction is typically still decent
- The update step length may be bad
- The convergence is slower, and we may even diverge

Example:
Range-based localization

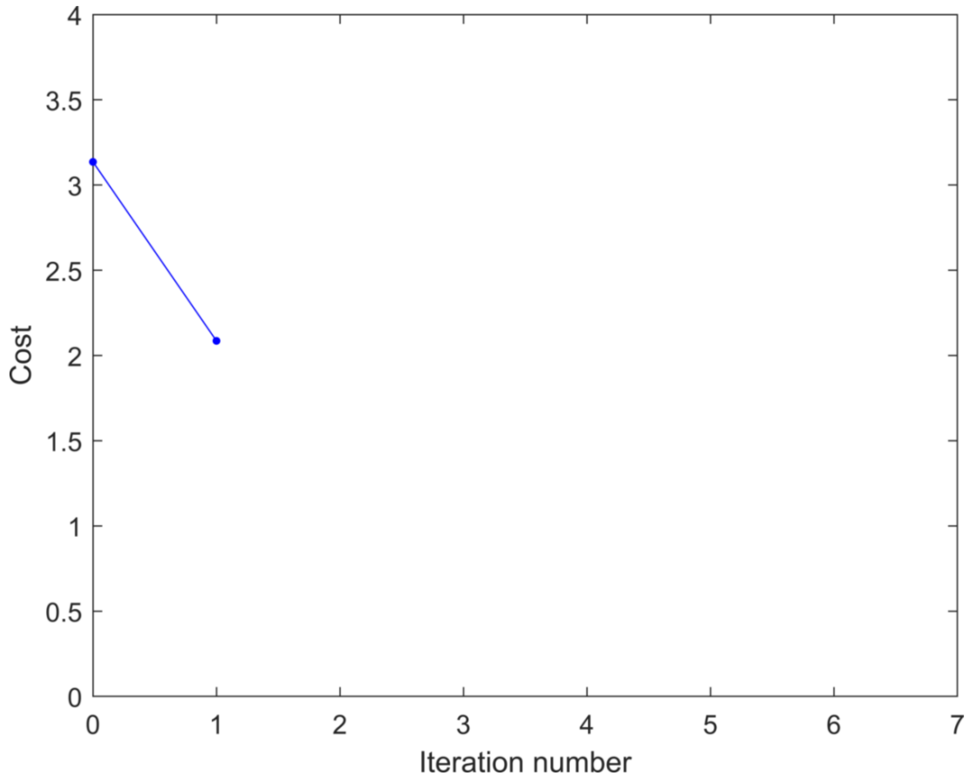
Gauss-Newton optimization

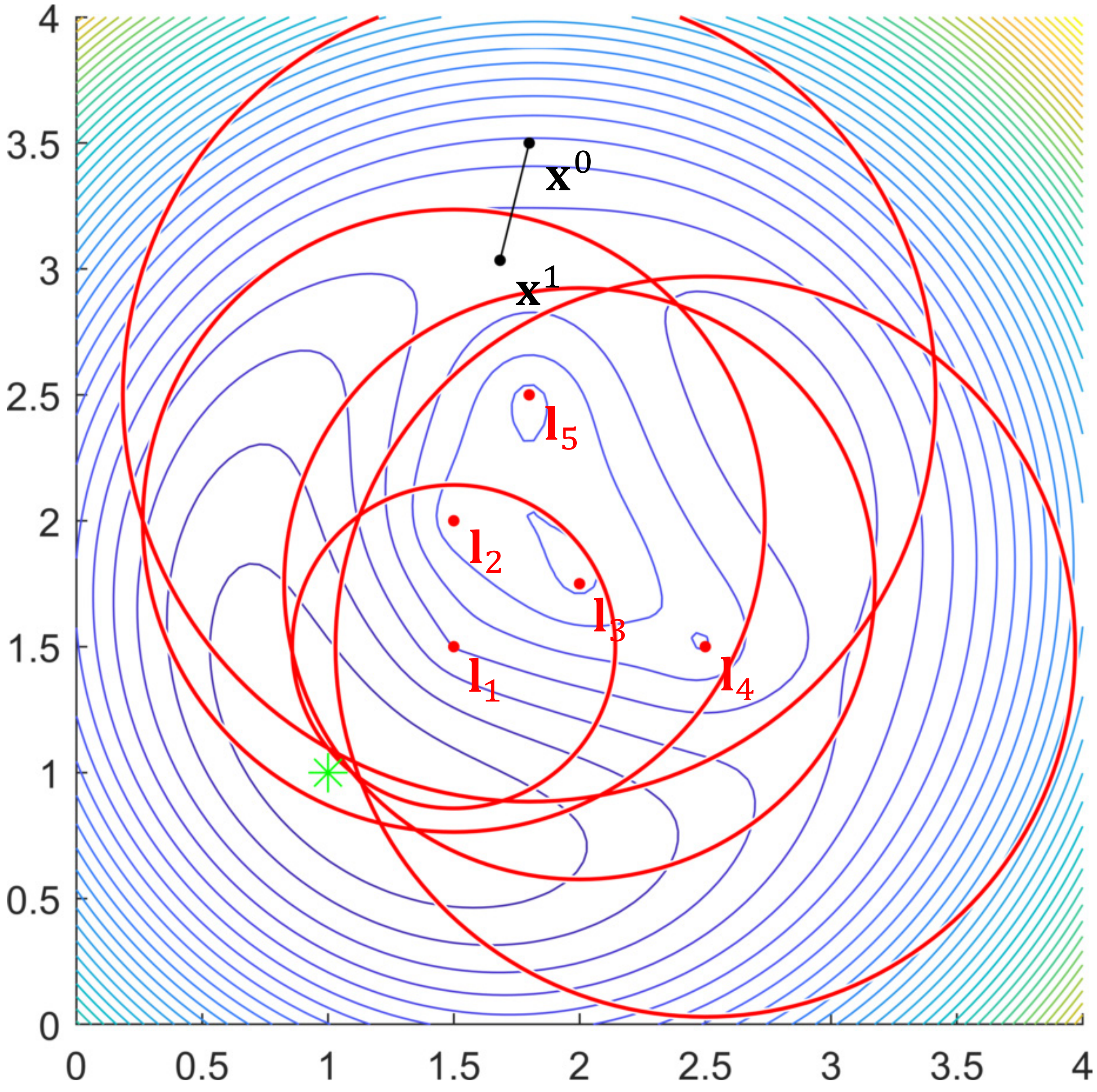




Example:
Range-based localization

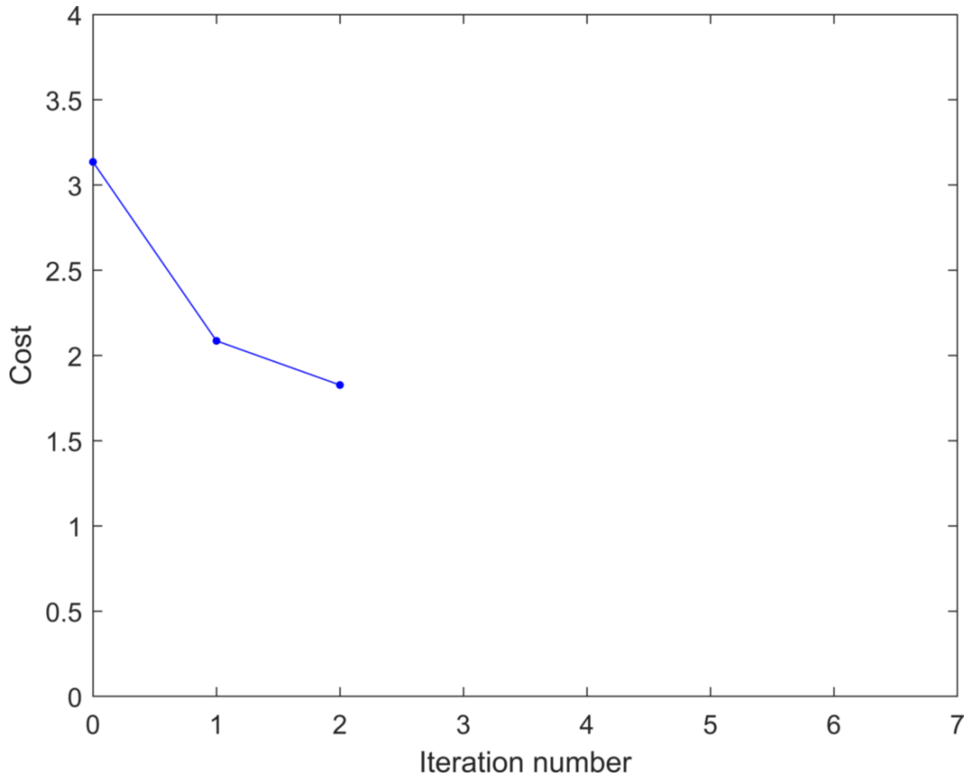
Gauss-Newton optimization

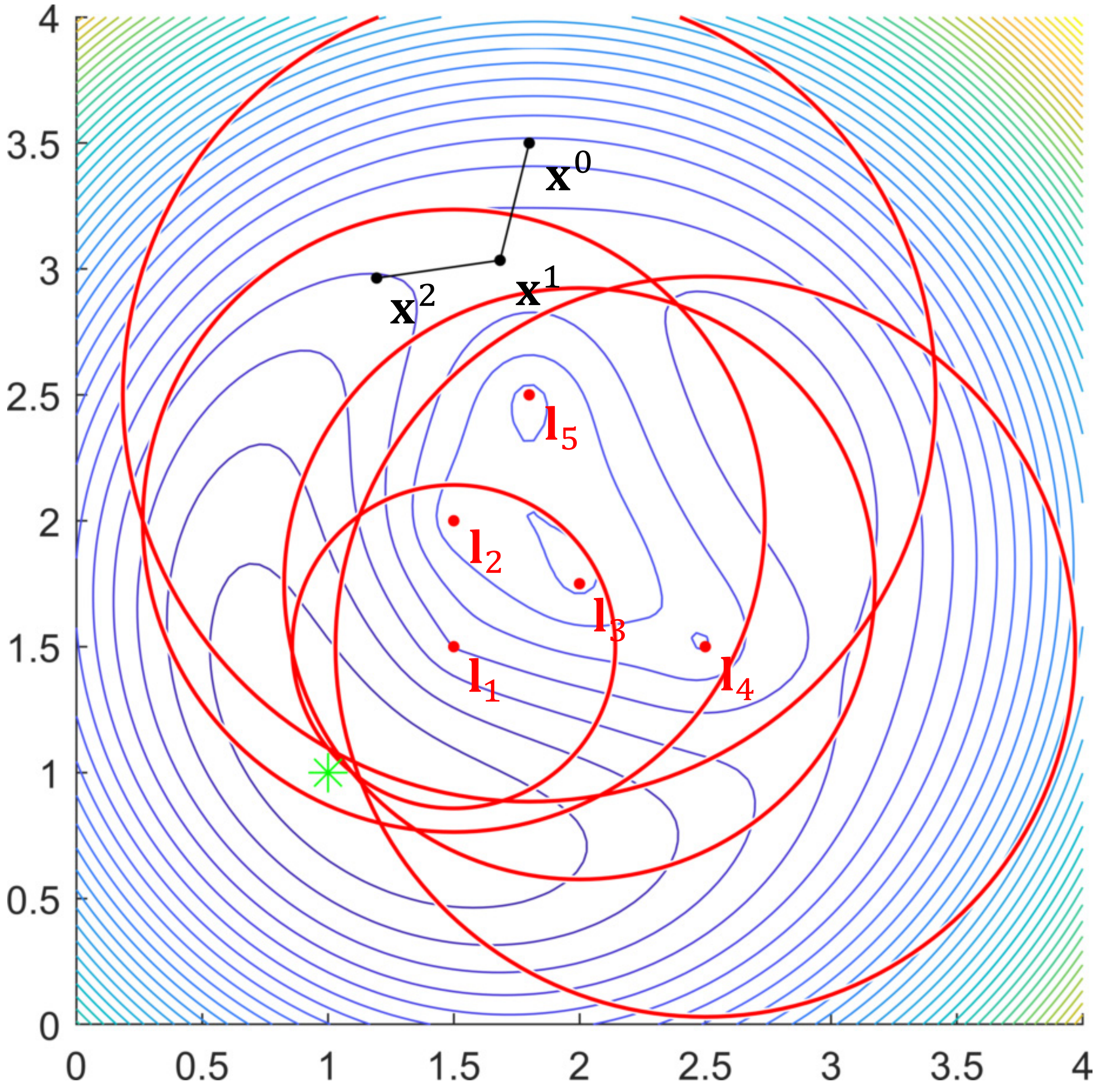




Example:
Range-based localization

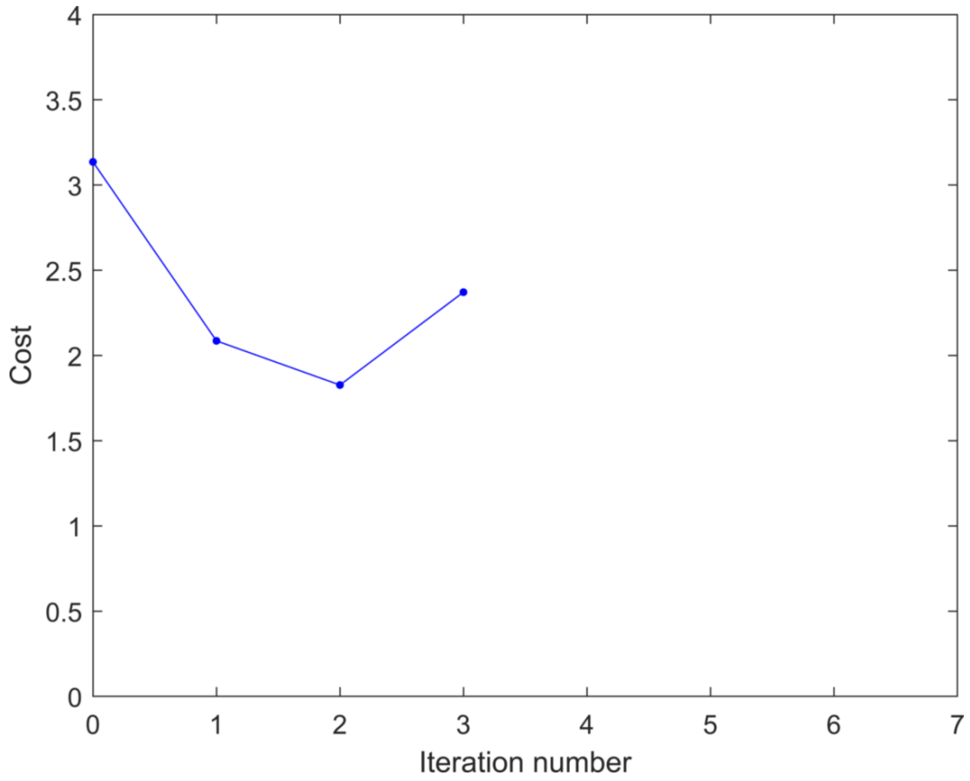
Gauss-Newton optimization

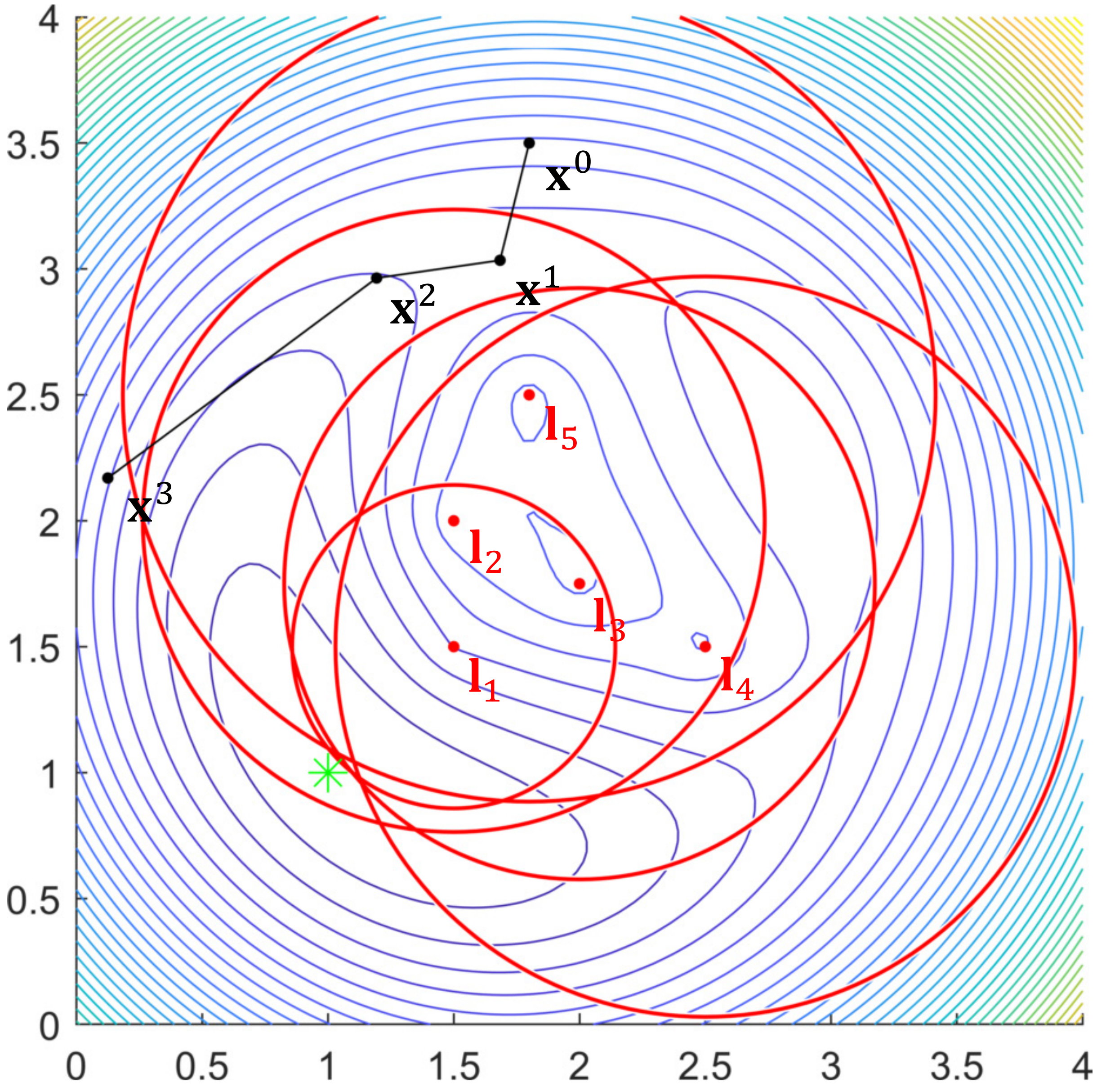




Example:
Range-based localization

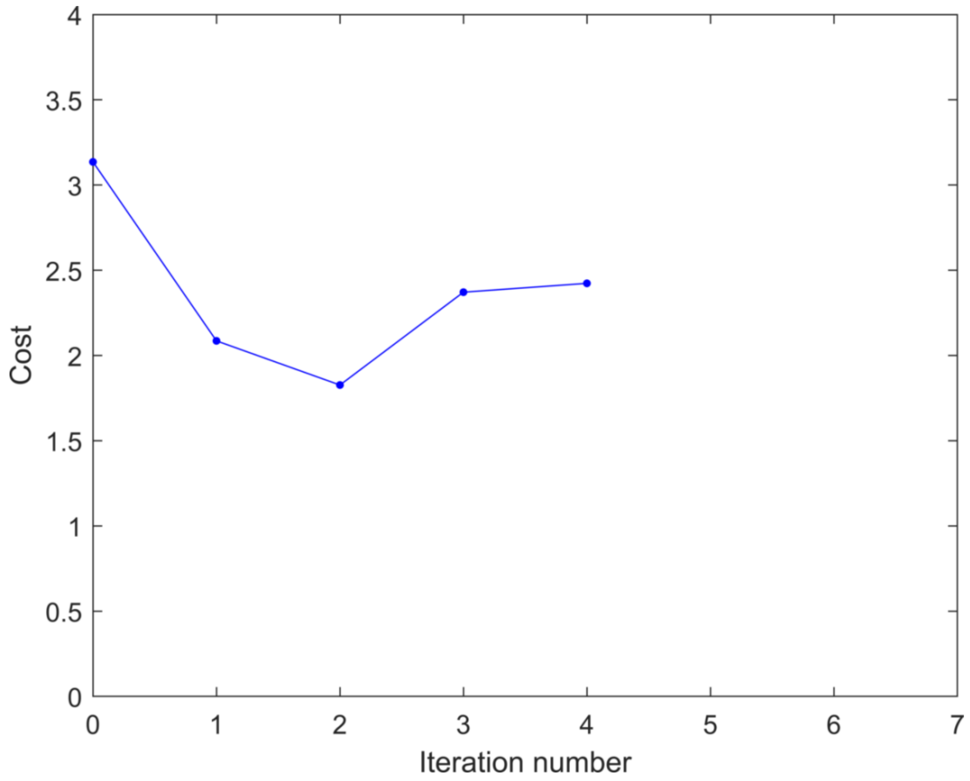
Gauss-Newton optimization

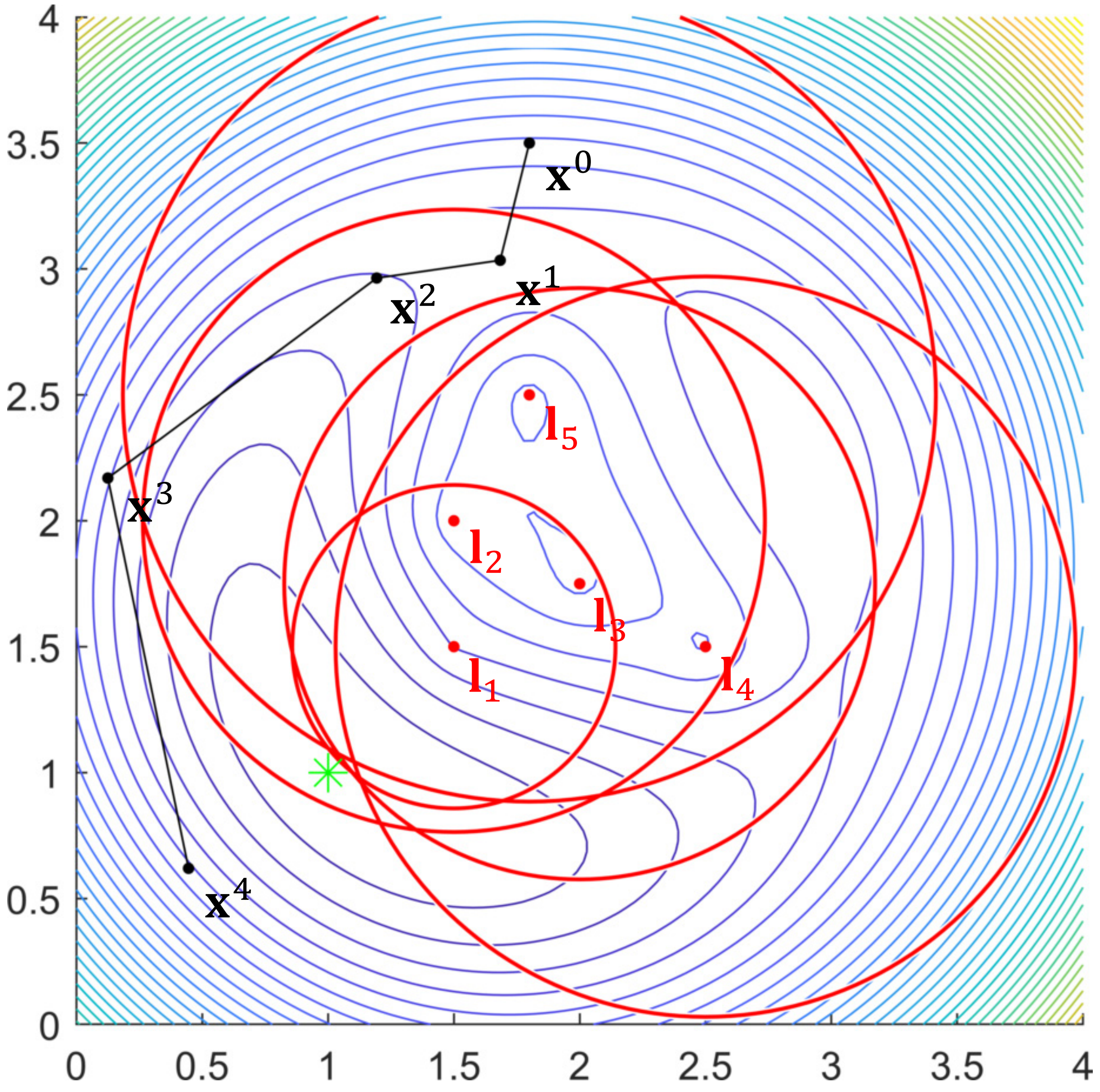




Example:
Range-based localization

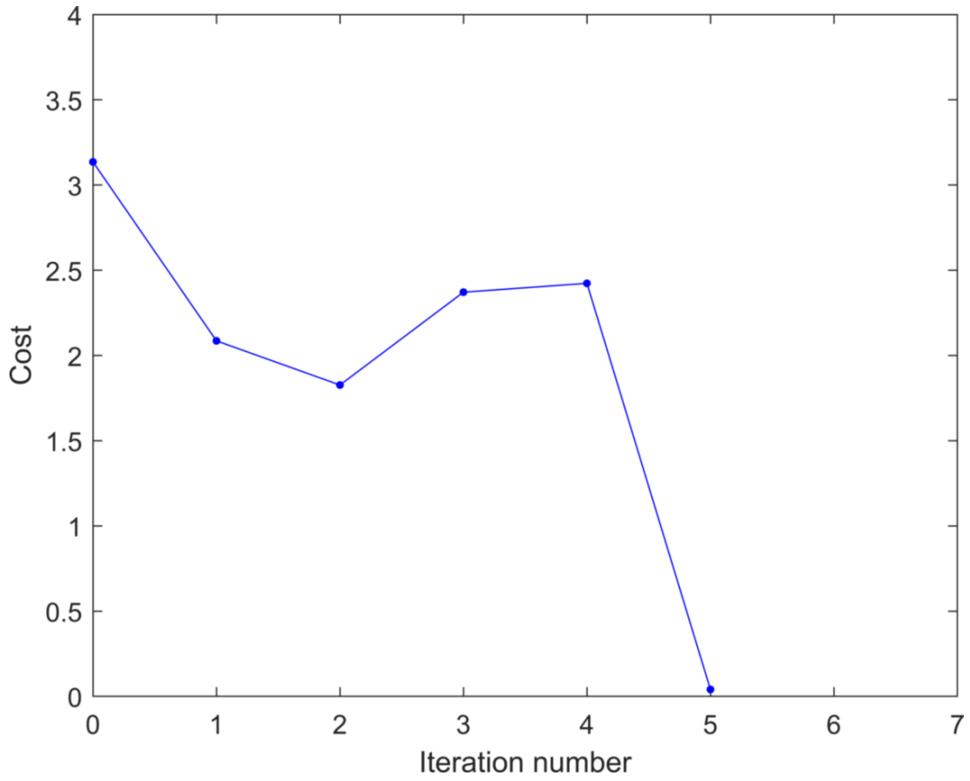
Gauss-Newton optimization

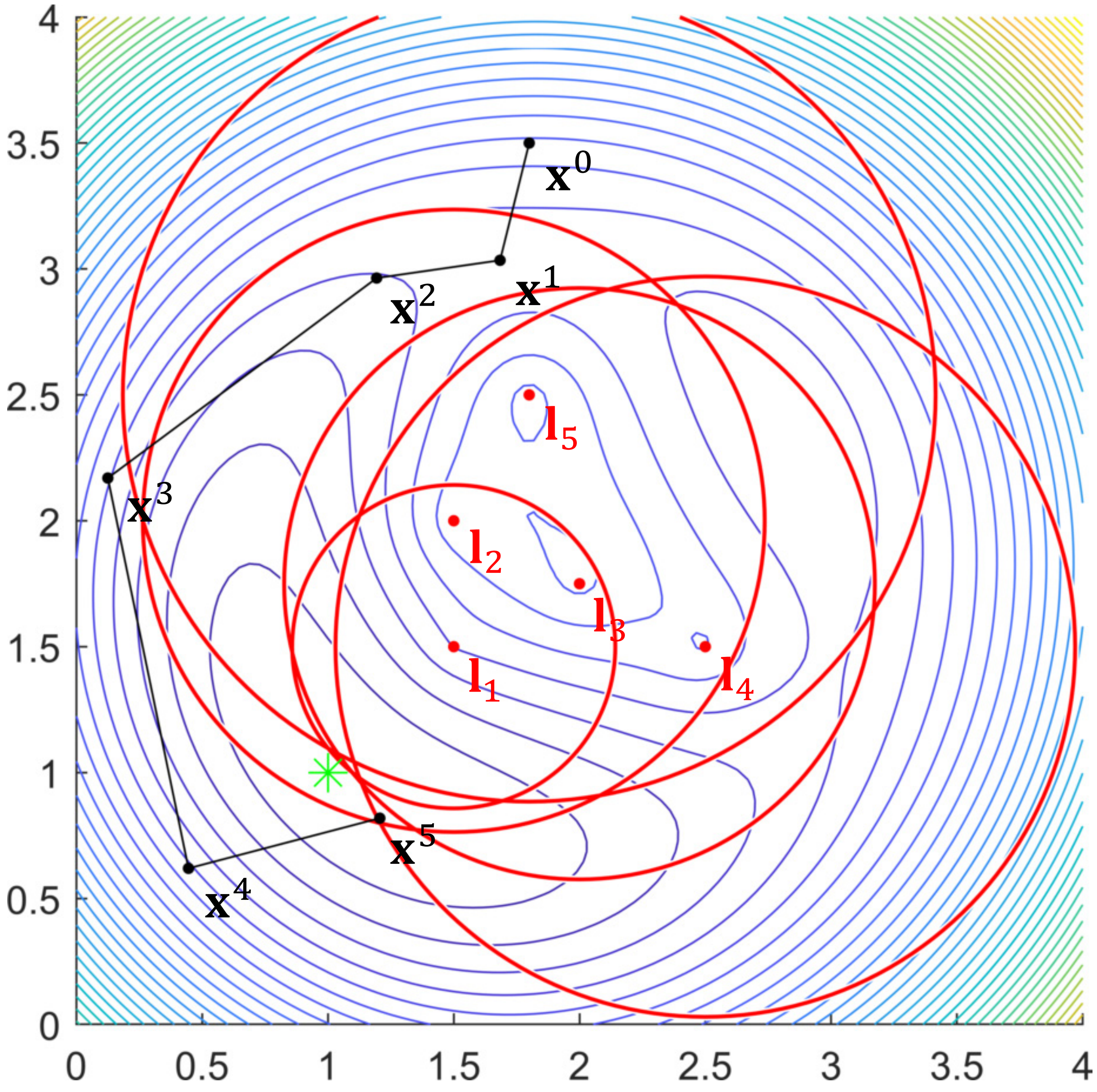




Example:
Range-based localization

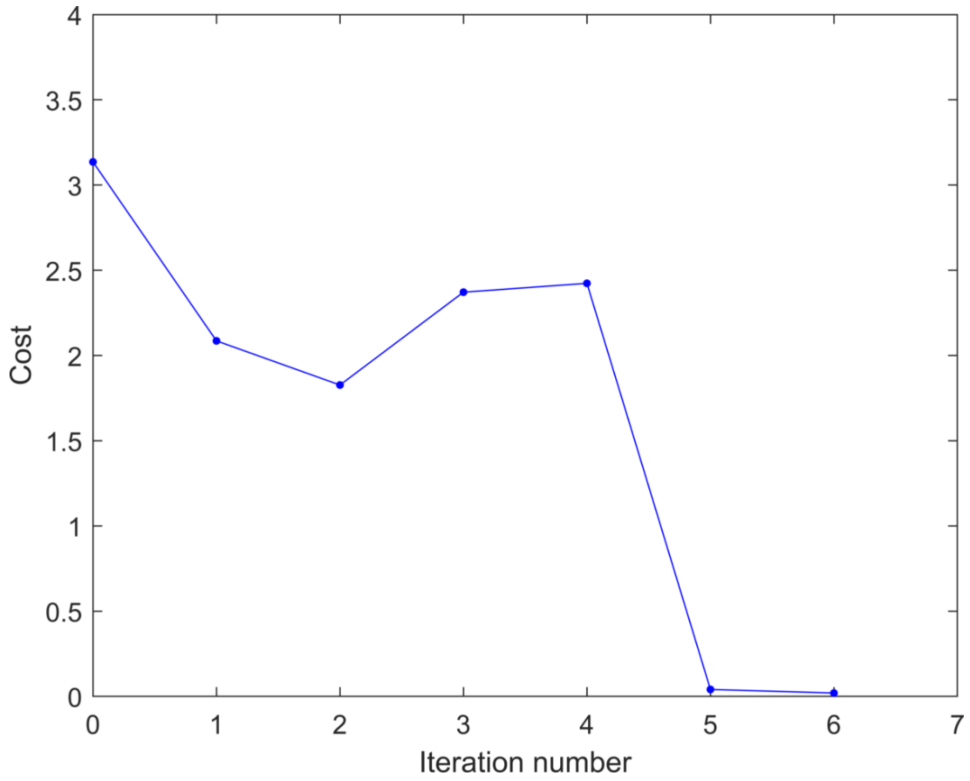
Gauss-Newton optimization

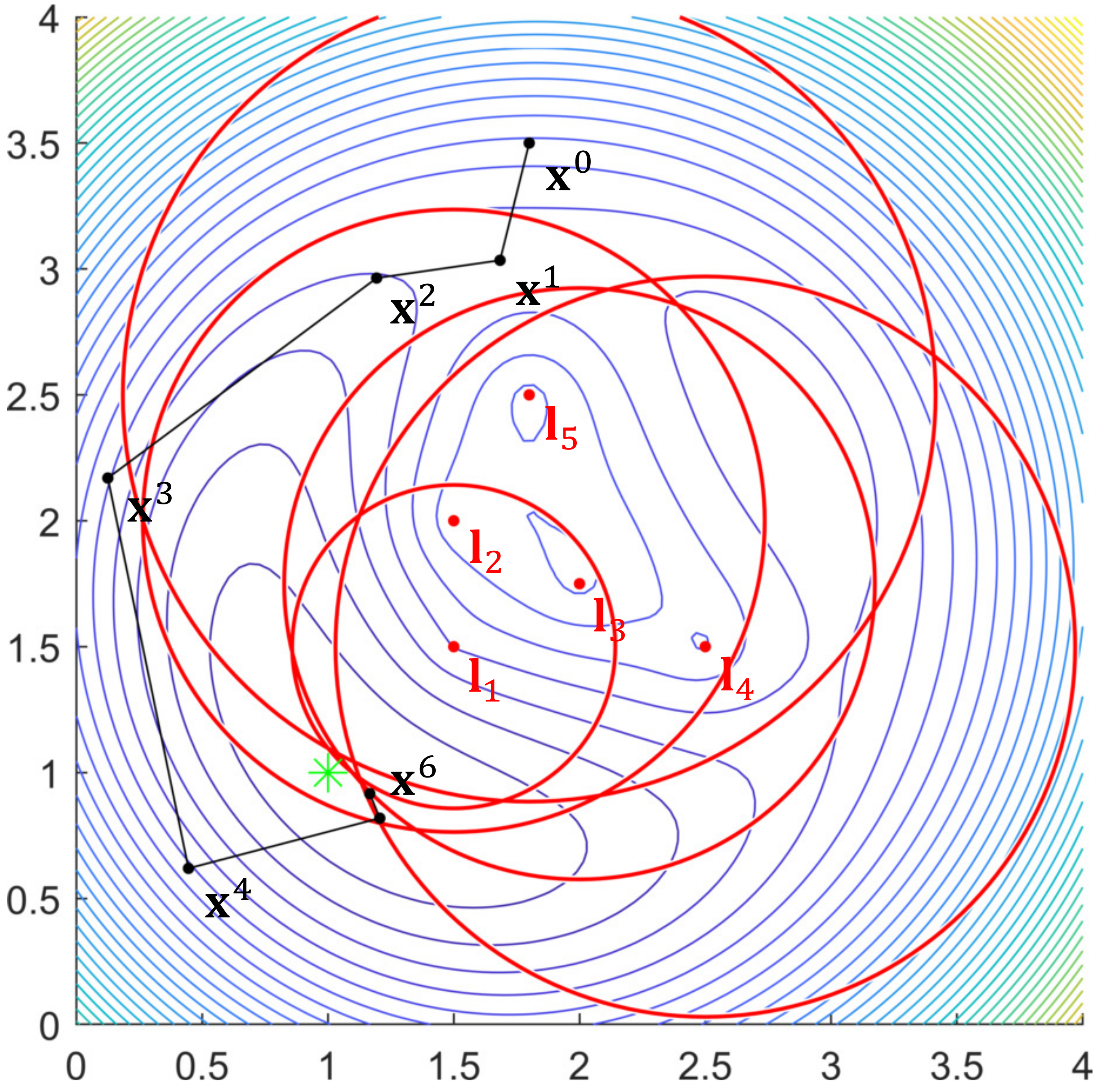




Example: Range-based localization

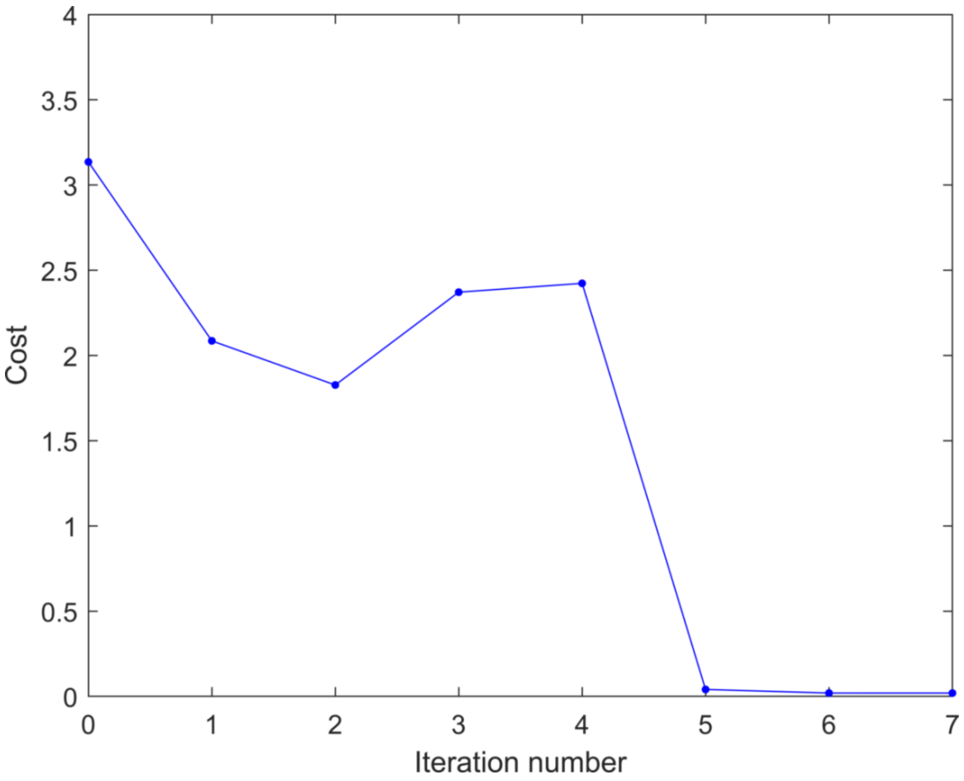
Gauss-Newton optimization

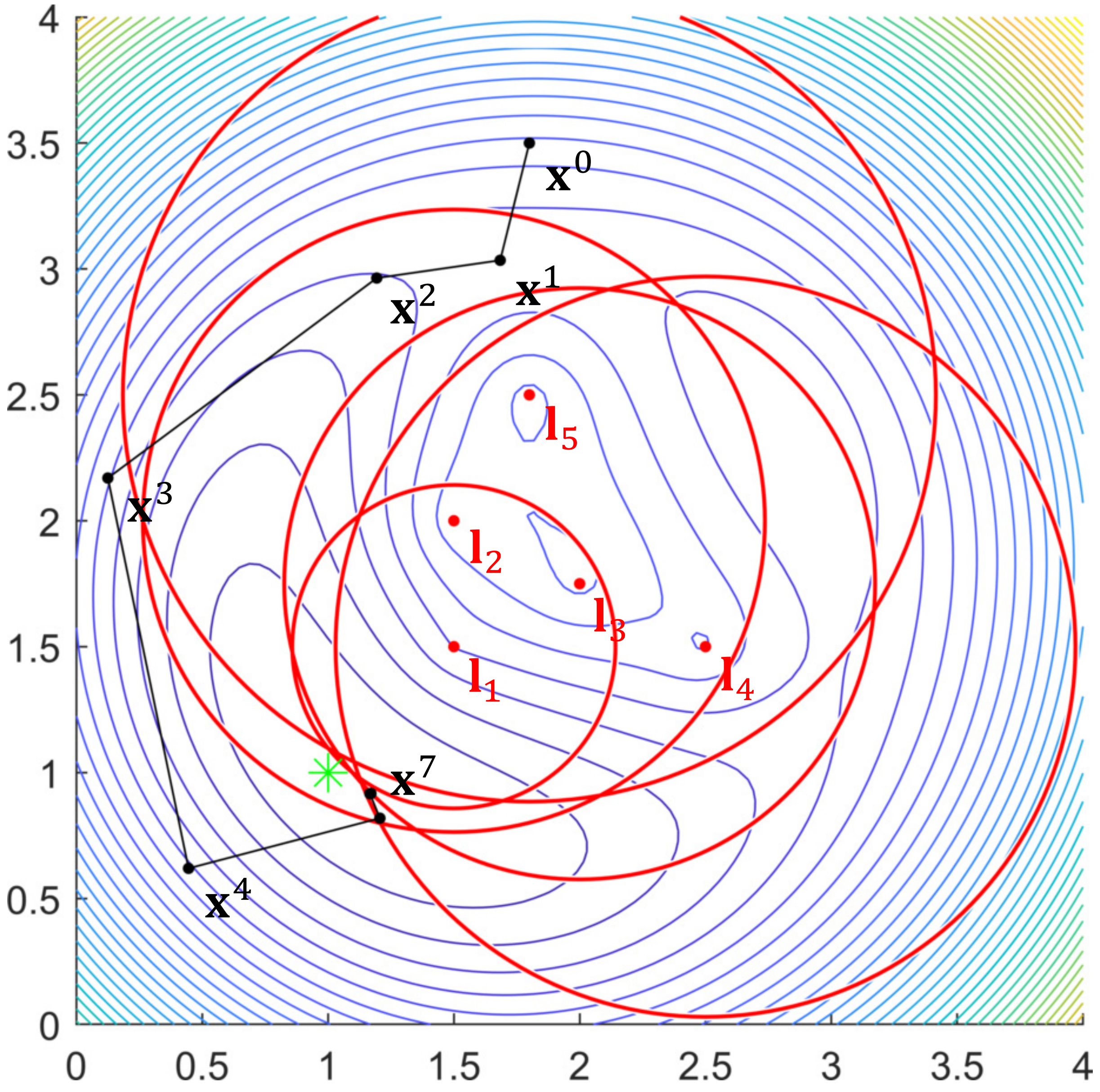




Example:
Range-based localization

Gauss-Newton optimization





Trust region

- The Gauss-Newton method is not guaranteed to converge because of the approximate Hessian matrix

- Since the update directions typically are decent, we can help with convergence by limiting the step sizes
  - More conservative towards robustness, rather than speed

- Such methods are often called **trust region methods**, and one example is **Levenberg-Marquardt**

The Levenberg–Marquardt algorithm

Given an objective $f(X)$ and a good initial estimate X^0.

$\lambda = 10^{-4}$

For $t = 0, 1, \ldots, t^{max}$

$\mathbf{A}, \mathbf{b} \leftarrow$ Linearize $f(X)$ at X^t

$\boldsymbol{\Delta} \leftarrow$ Solve the linearized problem with $(\mathbf{A}^T\mathbf{A} + \lambda \mathrm{diag}(\mathbf{A}^T\mathbf{A}))\boldsymbol{\Delta} = \mathbf{A}^T\mathbf{b}$

if $f(X^t + \boldsymbol{\Delta}) < f(X^t)$

$X^{t+1} = X^t + \boldsymbol{\Delta}$

$\lambda \leftarrow \lambda/10$

else

$X^{t+1} = X^t$

$\lambda \leftarrow \lambda * 10$

Terminate early if $f(X)$ is very small or $X^{t+1} \approx X^t$

The Levenberg–Marquardt algorithm

Given an objective $f(X)$ and a good initial estimate X^0.

$\lambda = 10^{-4}$

For $t = 0, 1, \dots, t^{max}$

$\mathbf{A}, \mathbf{b} \leftarrow$ Linearize $f(X)$ at X^t

$\mathbf{\Delta} \leftarrow$ Solve the linearized problem with $(\mathbf{A}^T\mathbf{A} + \lambda \mathrm{diag}(\mathbf{A}^T\mathbf{A}))\mathbf{\Delta} = \mathbf{A}^T\mathbf{b}$

if $f(X^t + \mathbf{\Delta}) < f(X^t)$

$X^{t+1} = X^t + \mathbf{\Delta}$

$\lambda \leftarrow \lambda/10$

Accept update, increase trust region

else

$X^{t+1} = X^t$

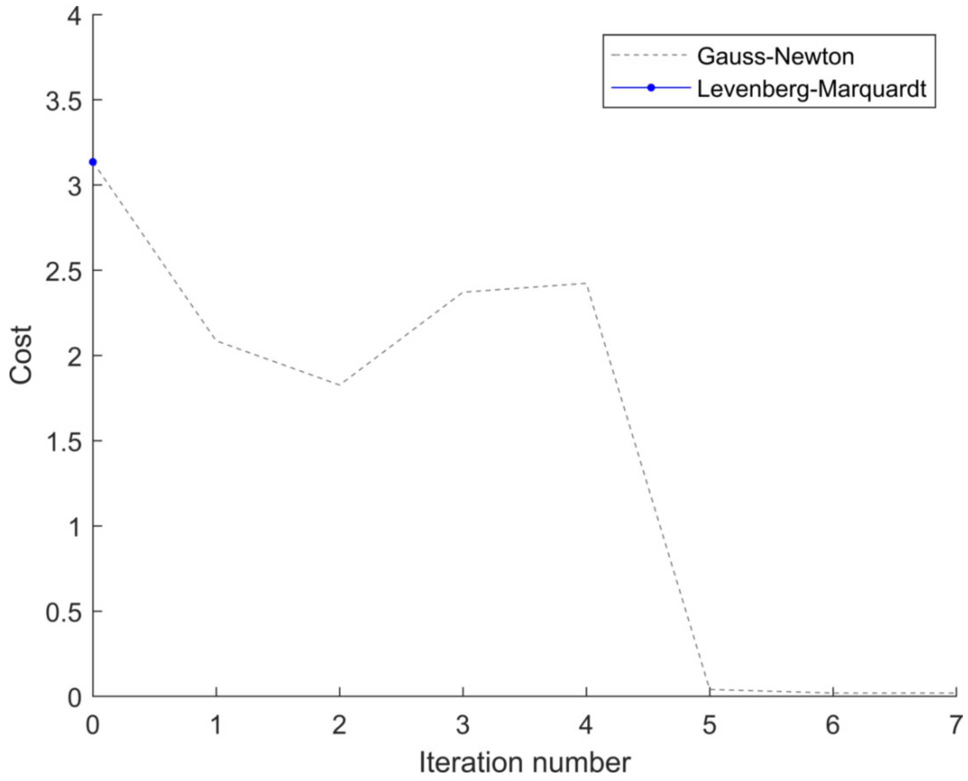
$\lambda \leftarrow \lambda * 10$

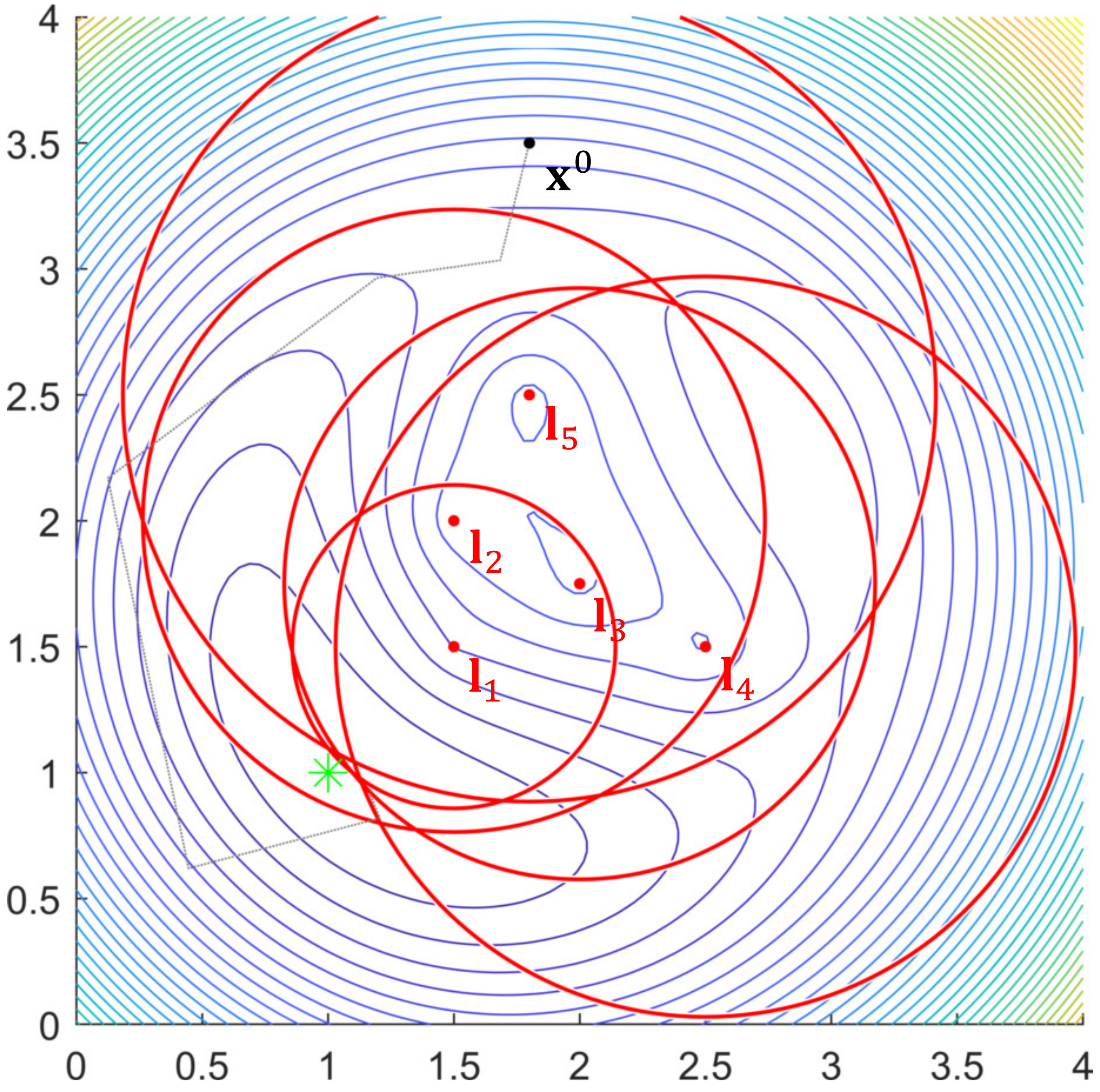
Reject update, reduce trust region

Terminate early if $f(X)$ is very small or $X^{t+1} \approx X^t$

Example:
Range-based localization

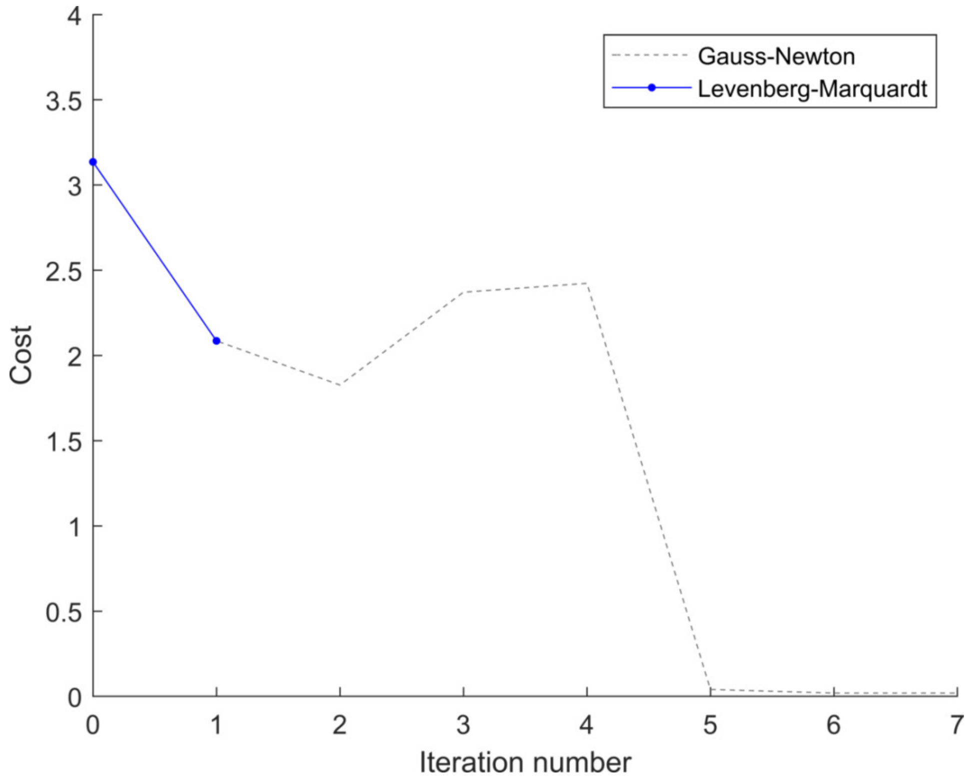
Levenberg–Marquardt optimization

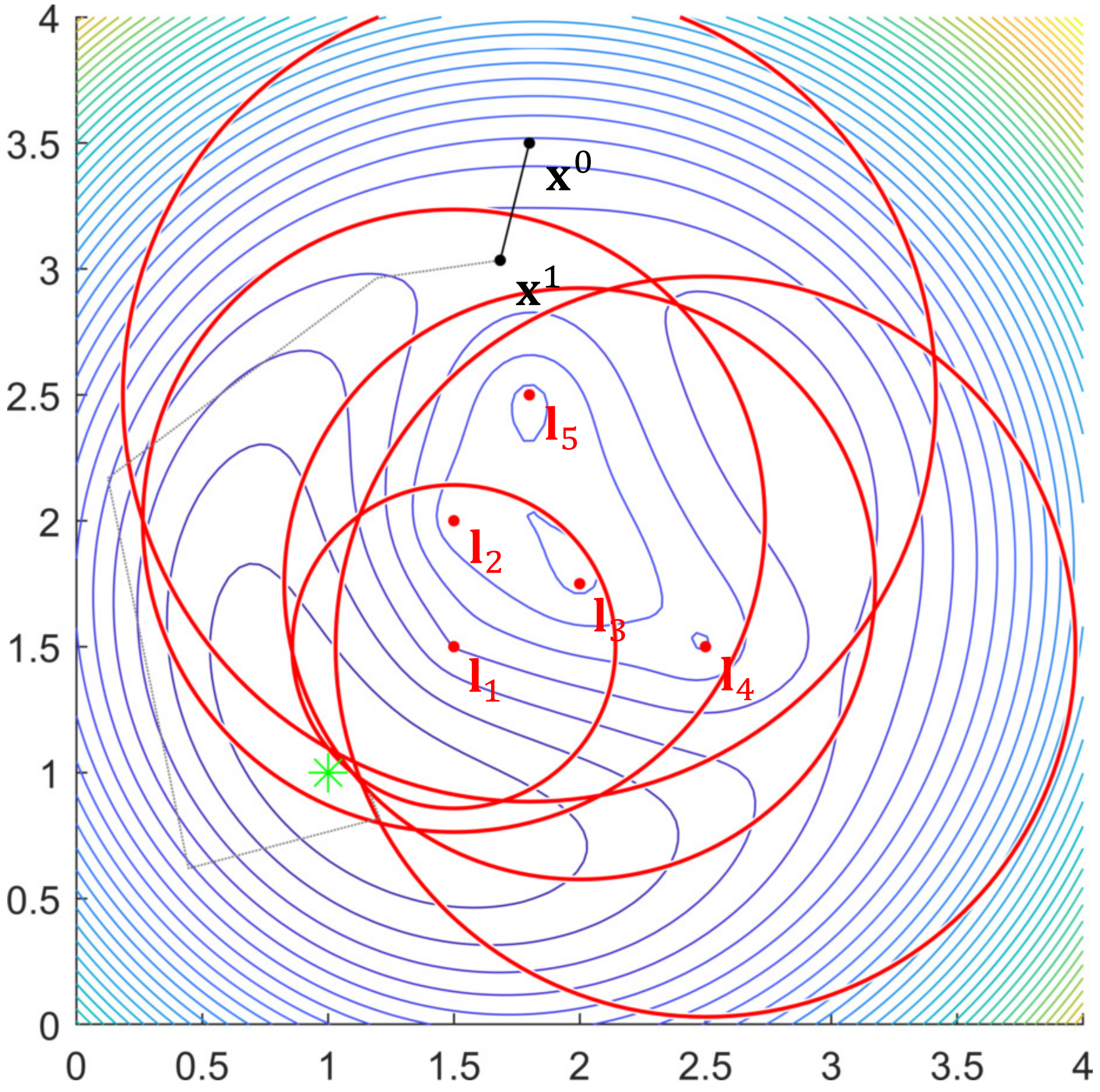




Example:
Range-based localization

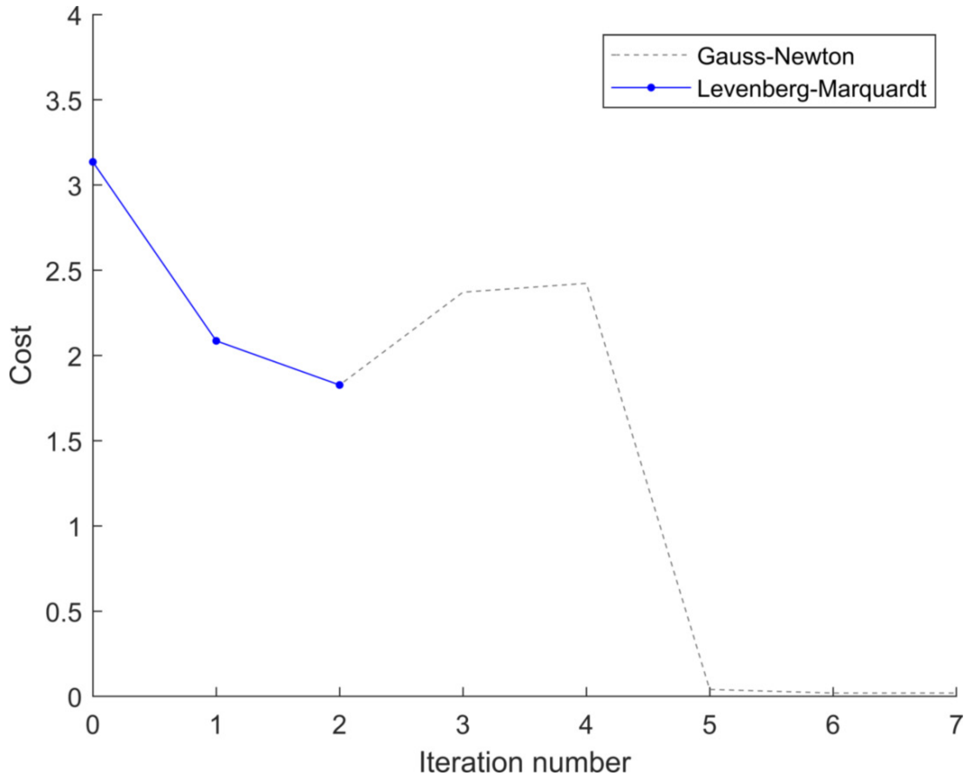
Levenberg–Marquardt optimization

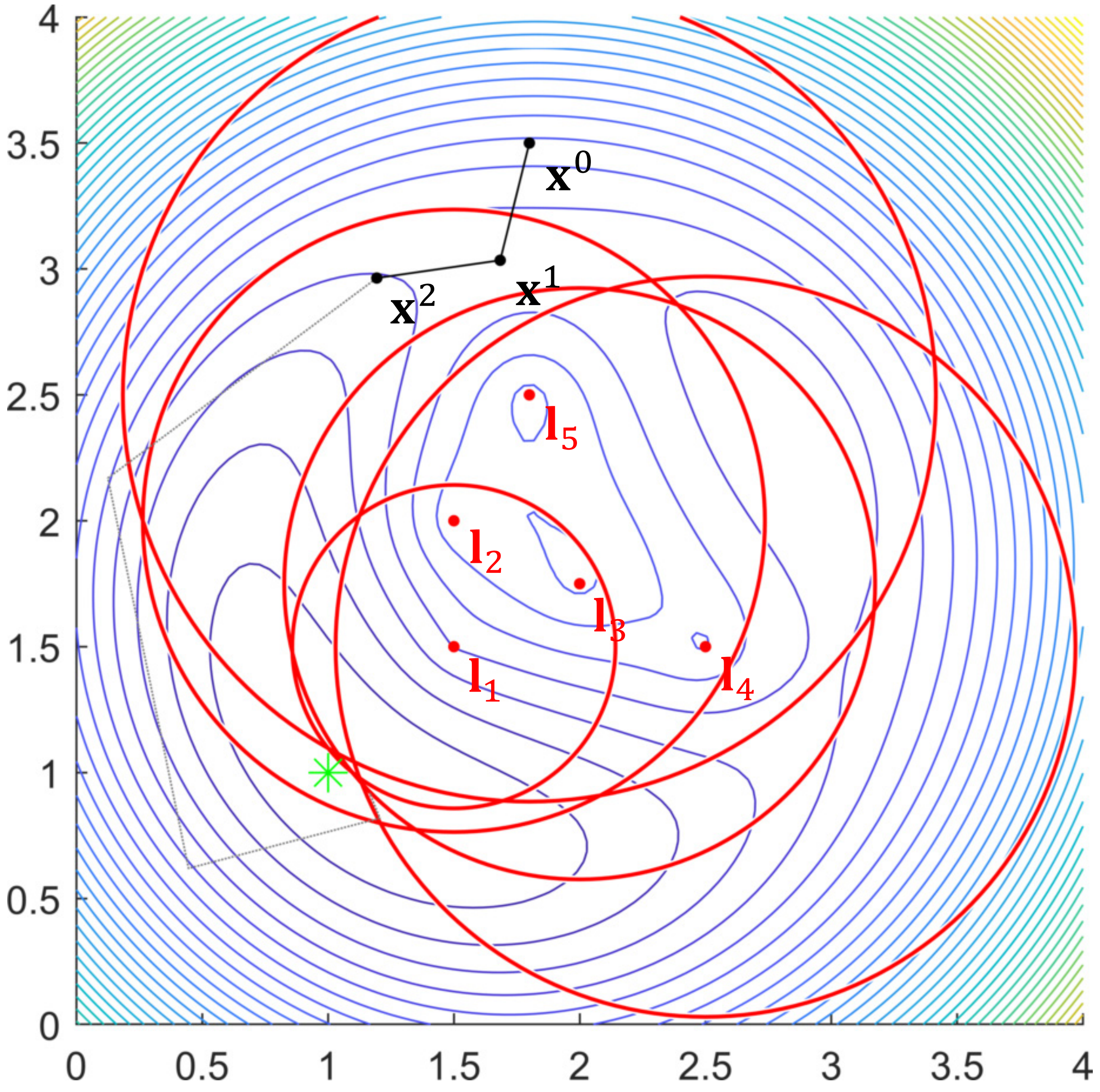




Example:
Range-based localization

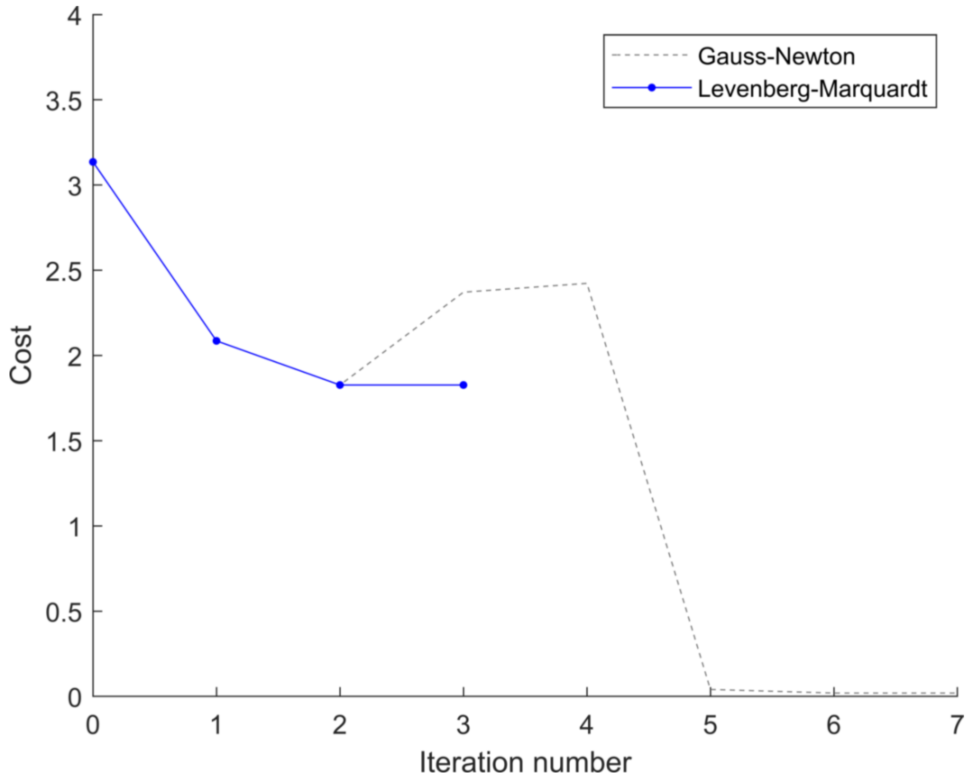
Levenberg–Marquardt optimization

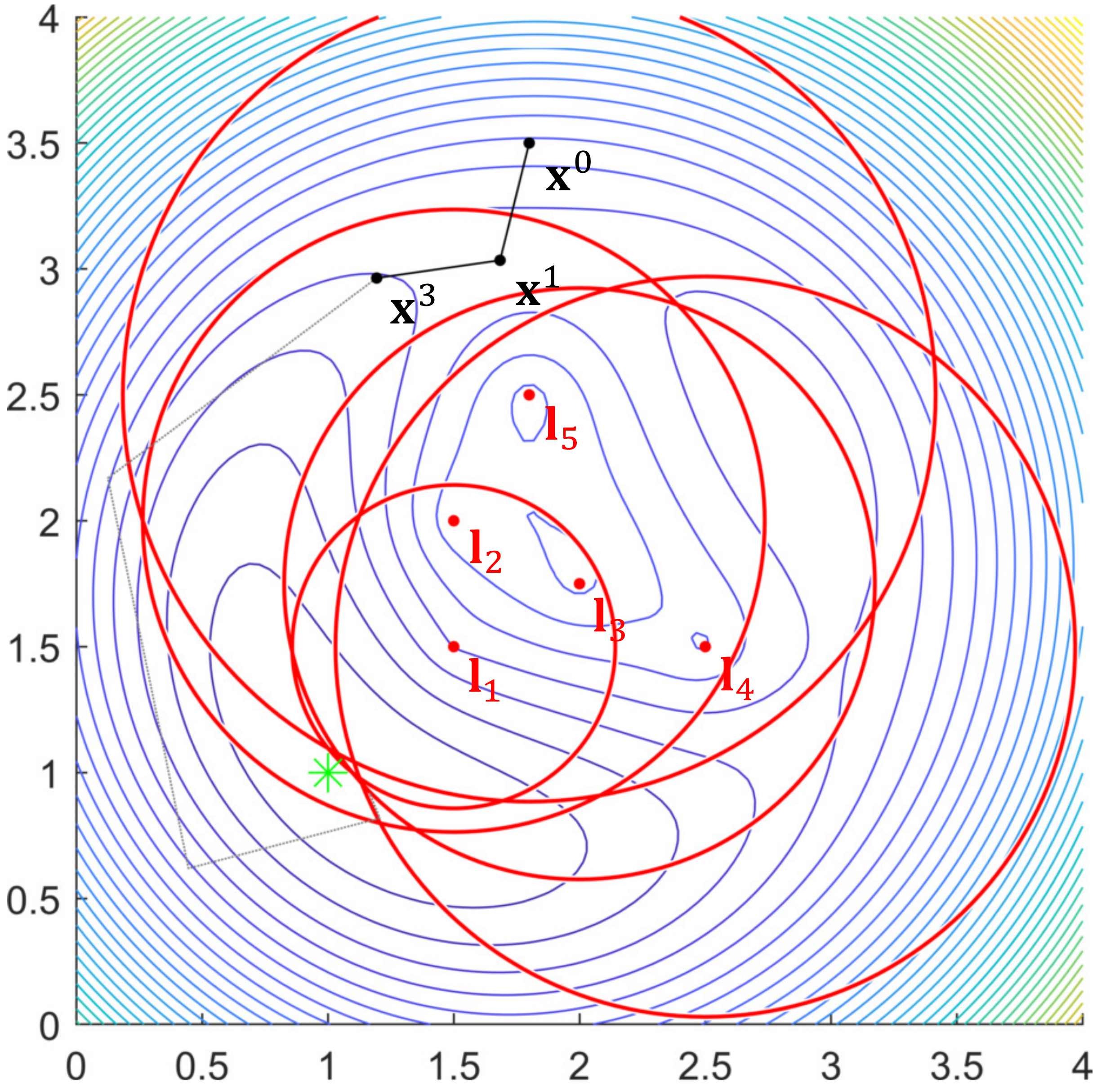




Example:
Range-based localization

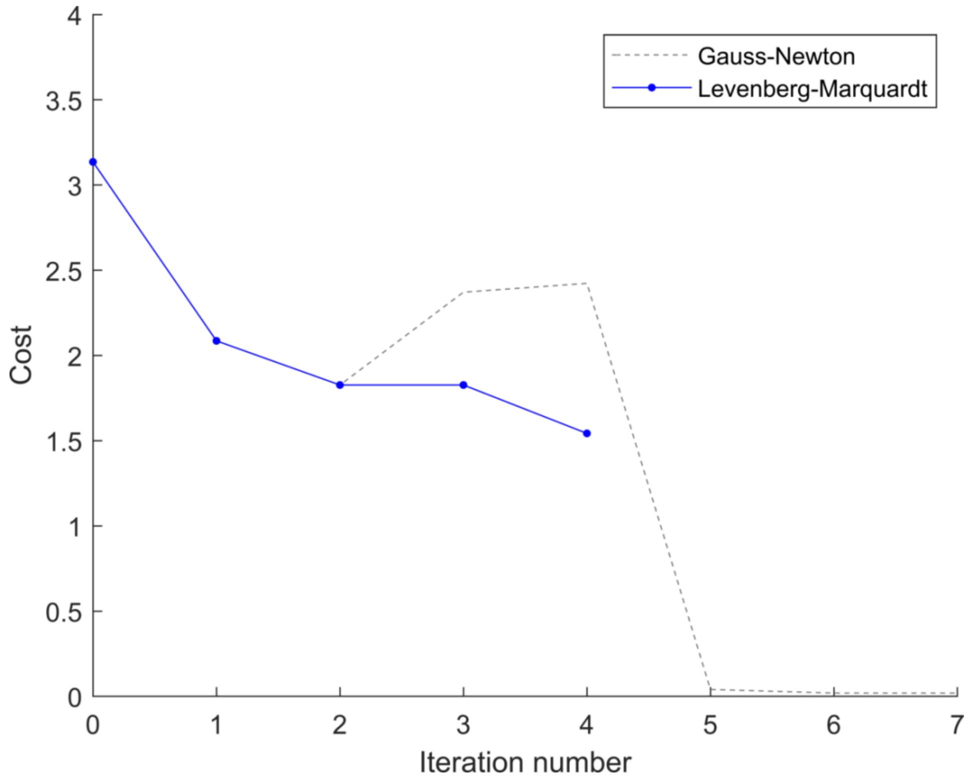
Levenberg–Marquardt optimization

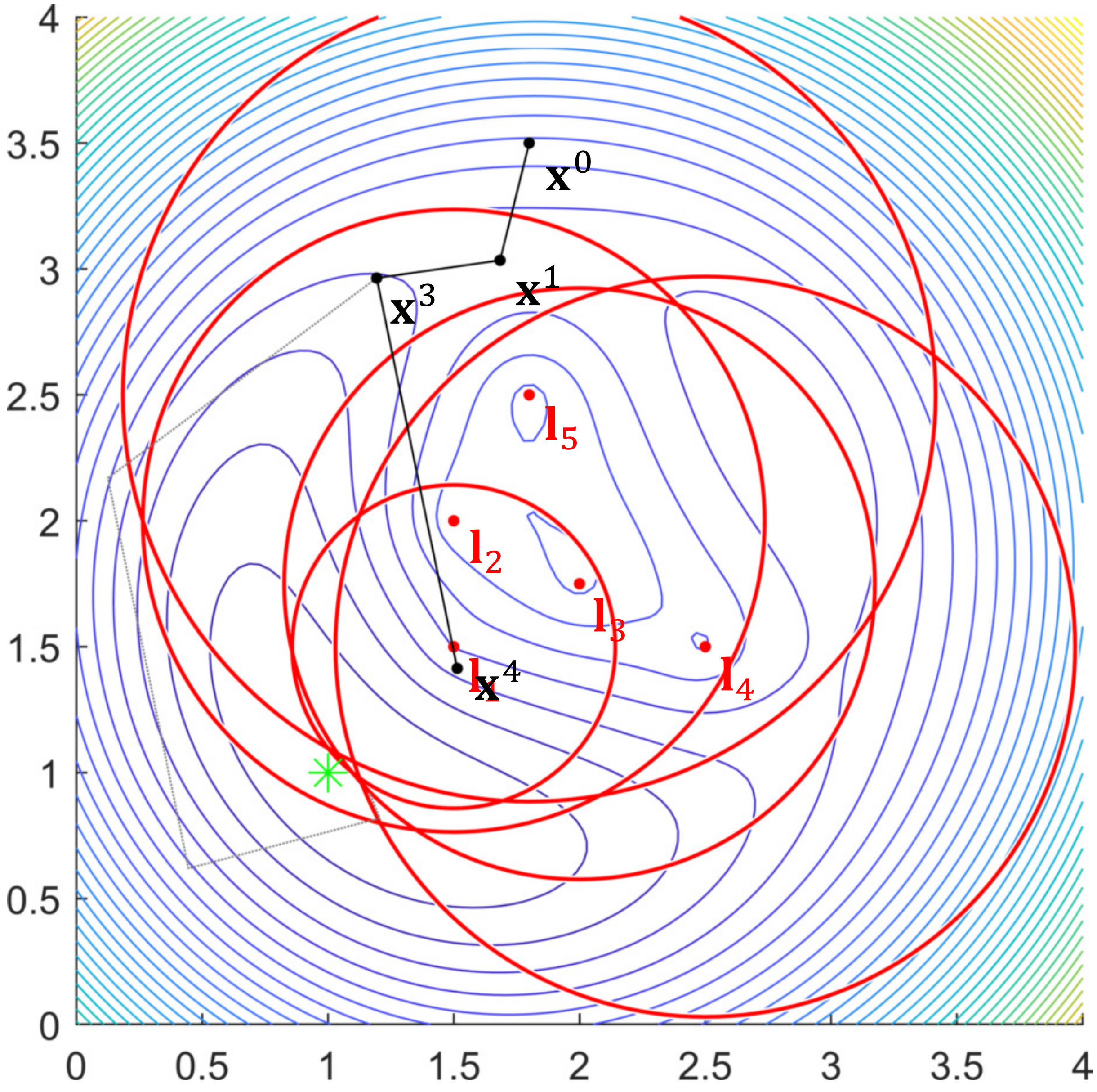




Example:
Range-based localization

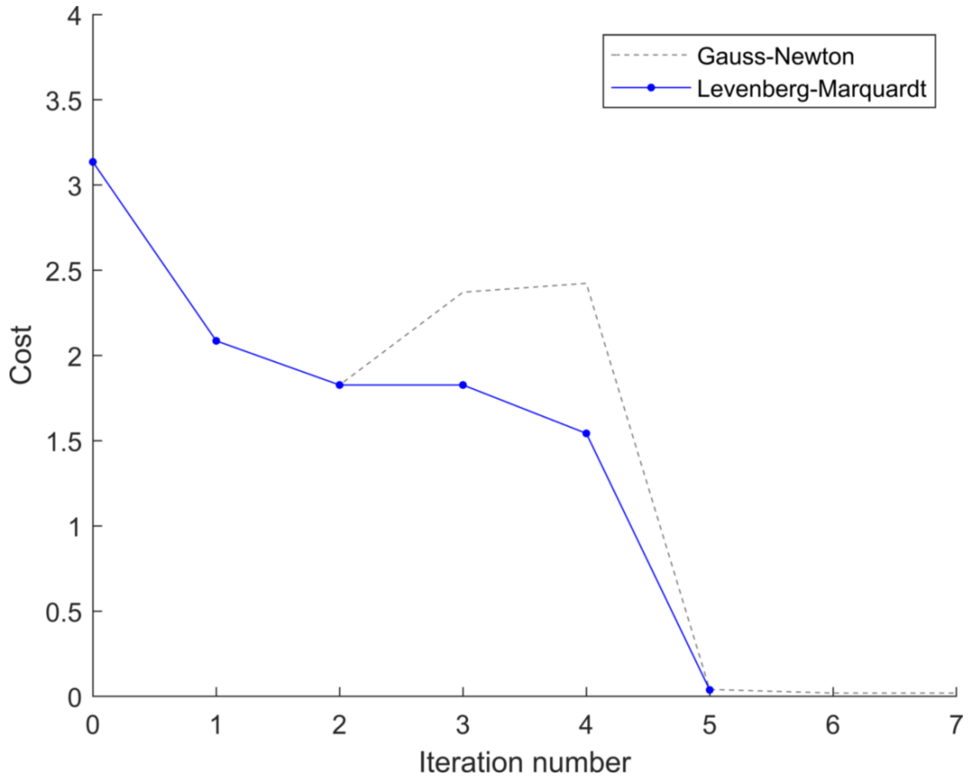
Levenberg–Marquardt optimization

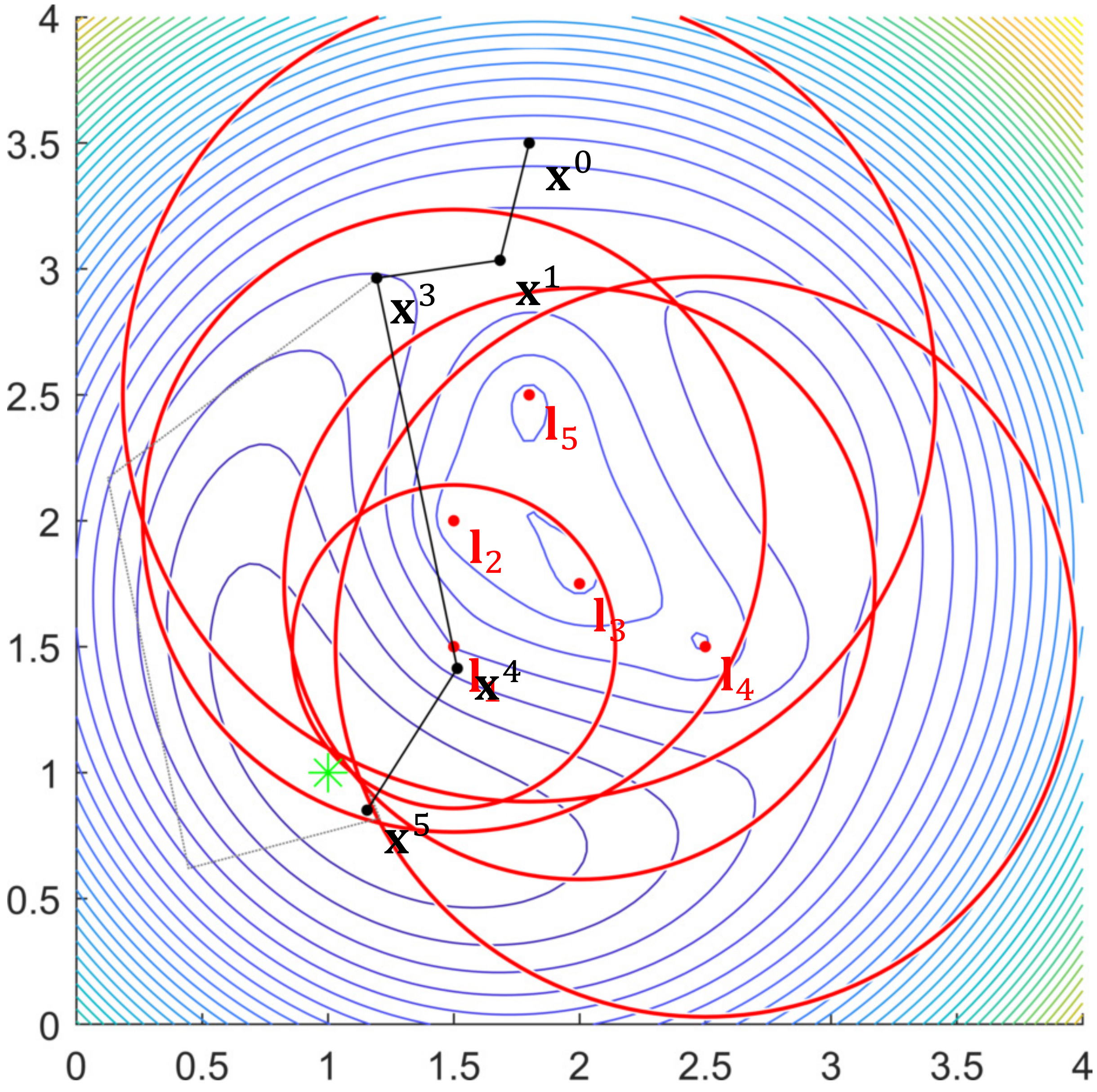




Example:
Range-based localization

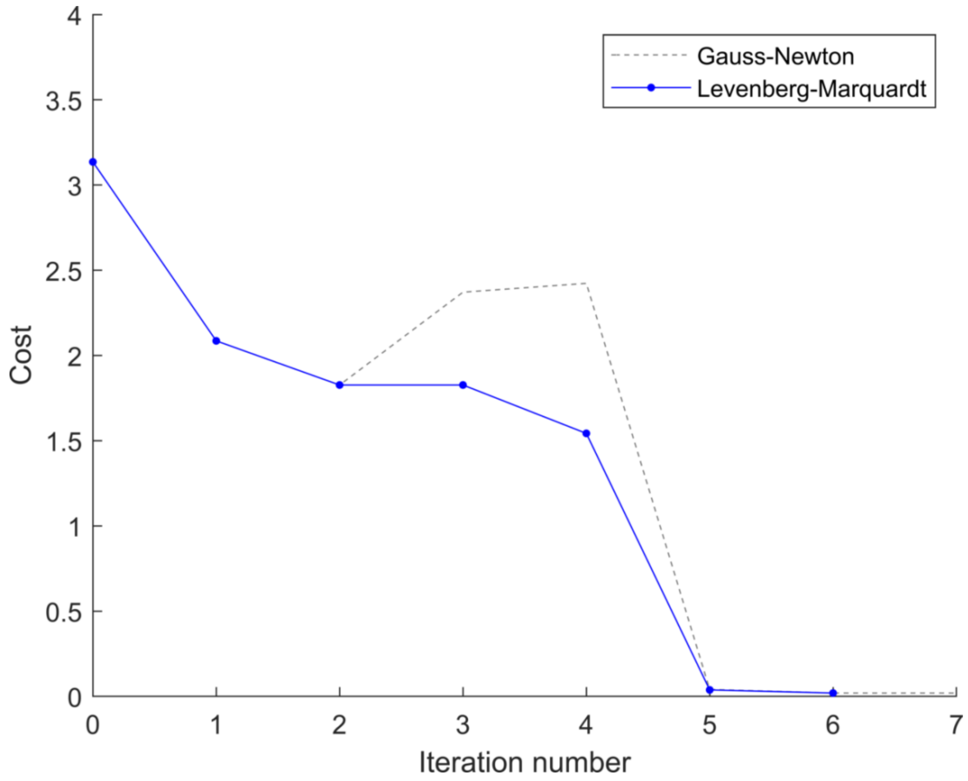
Levenberg–Marquardt optimization

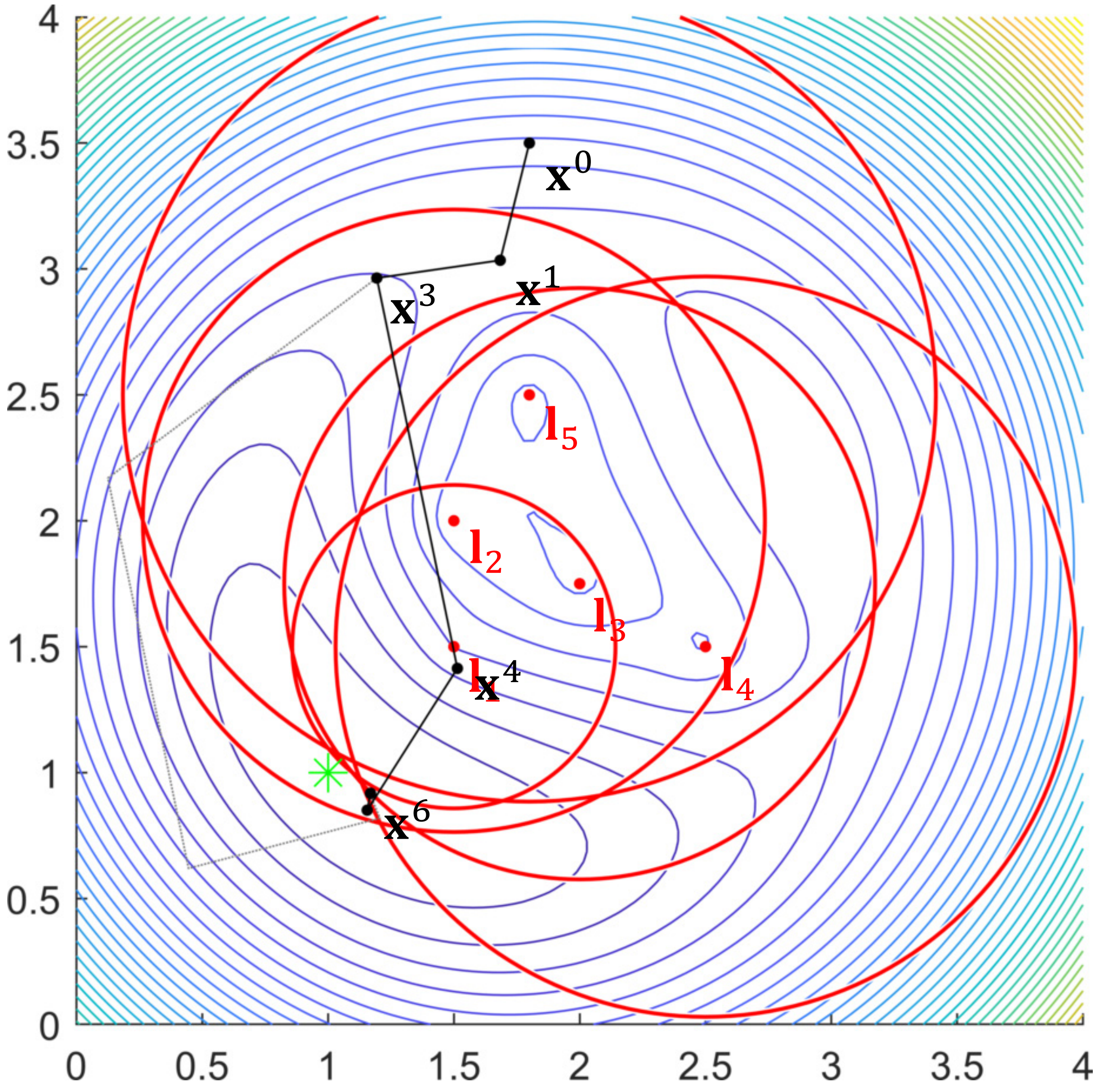




Example:
Range-based localization

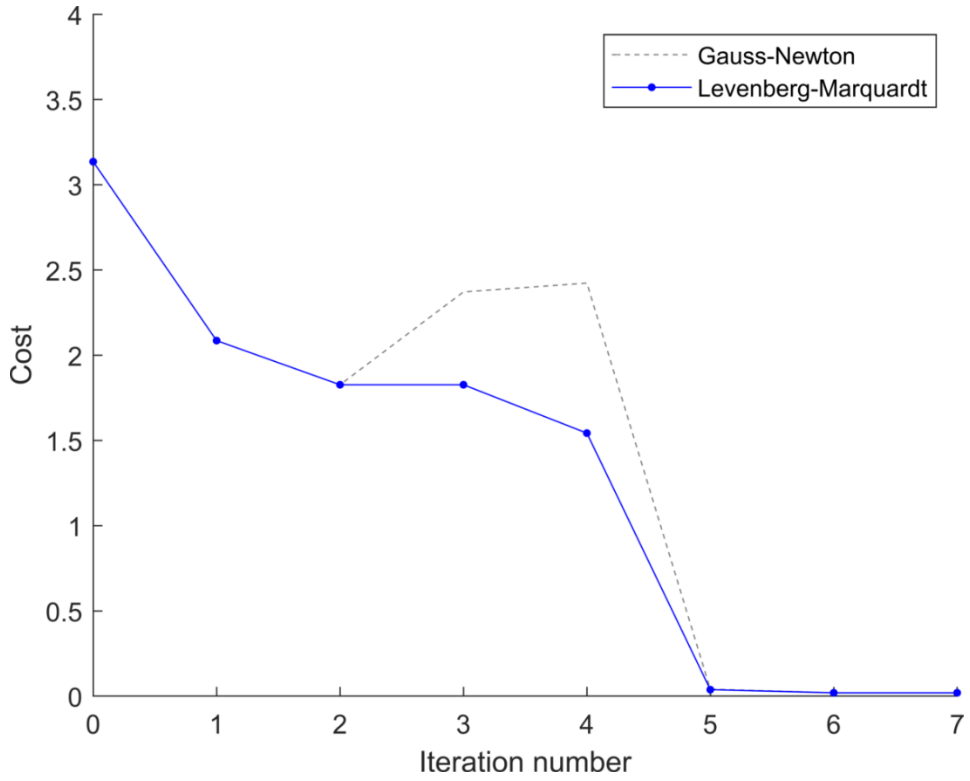
Levenberg–Marquardt optimization

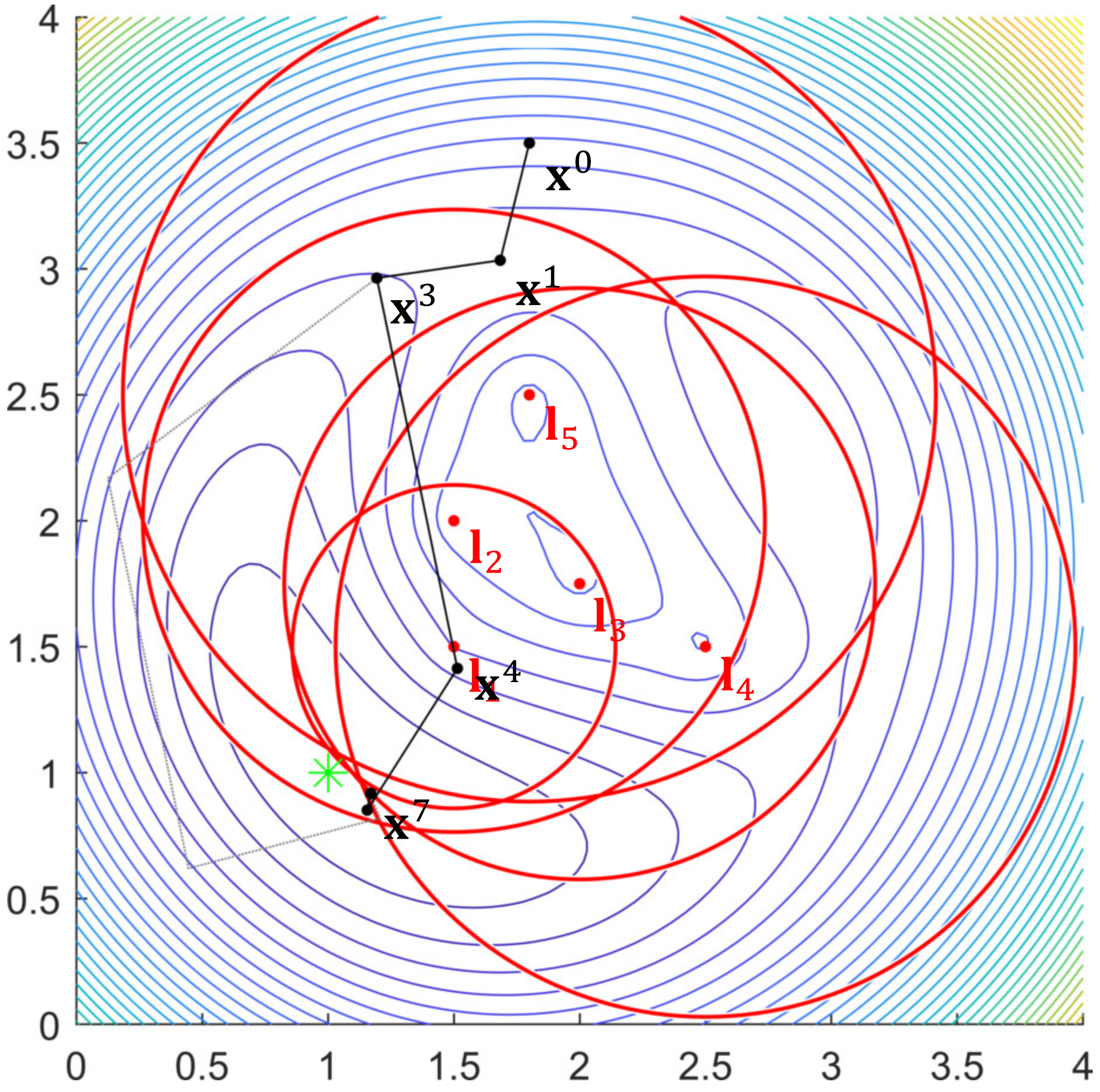




Example:
Range-based localization

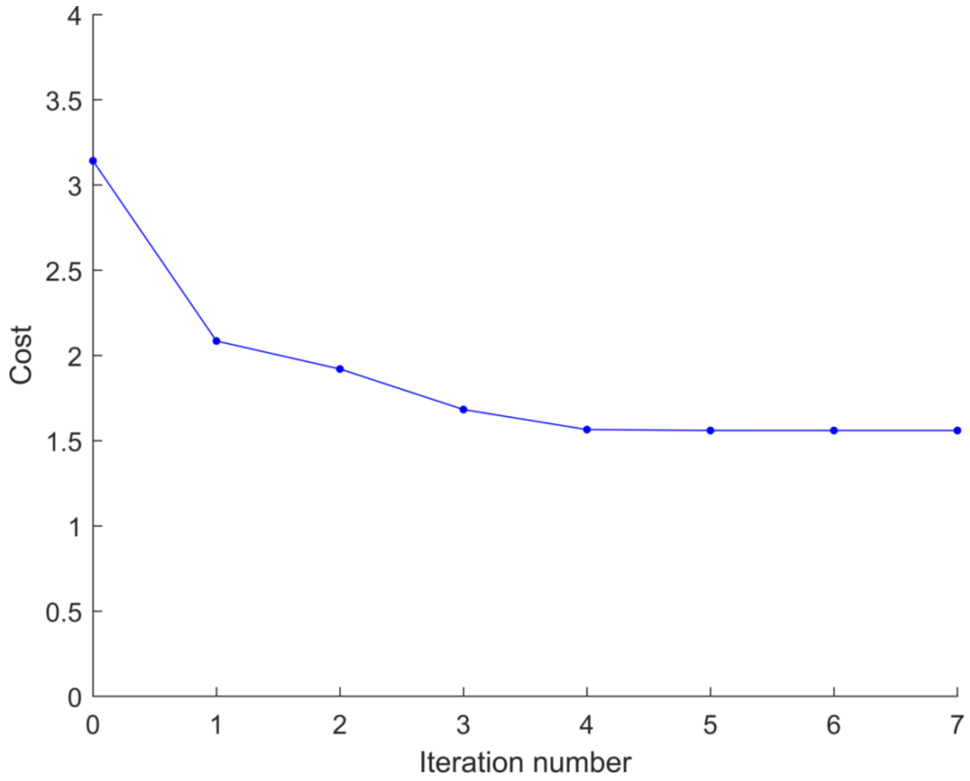
Levenberg–Marquardt optimization

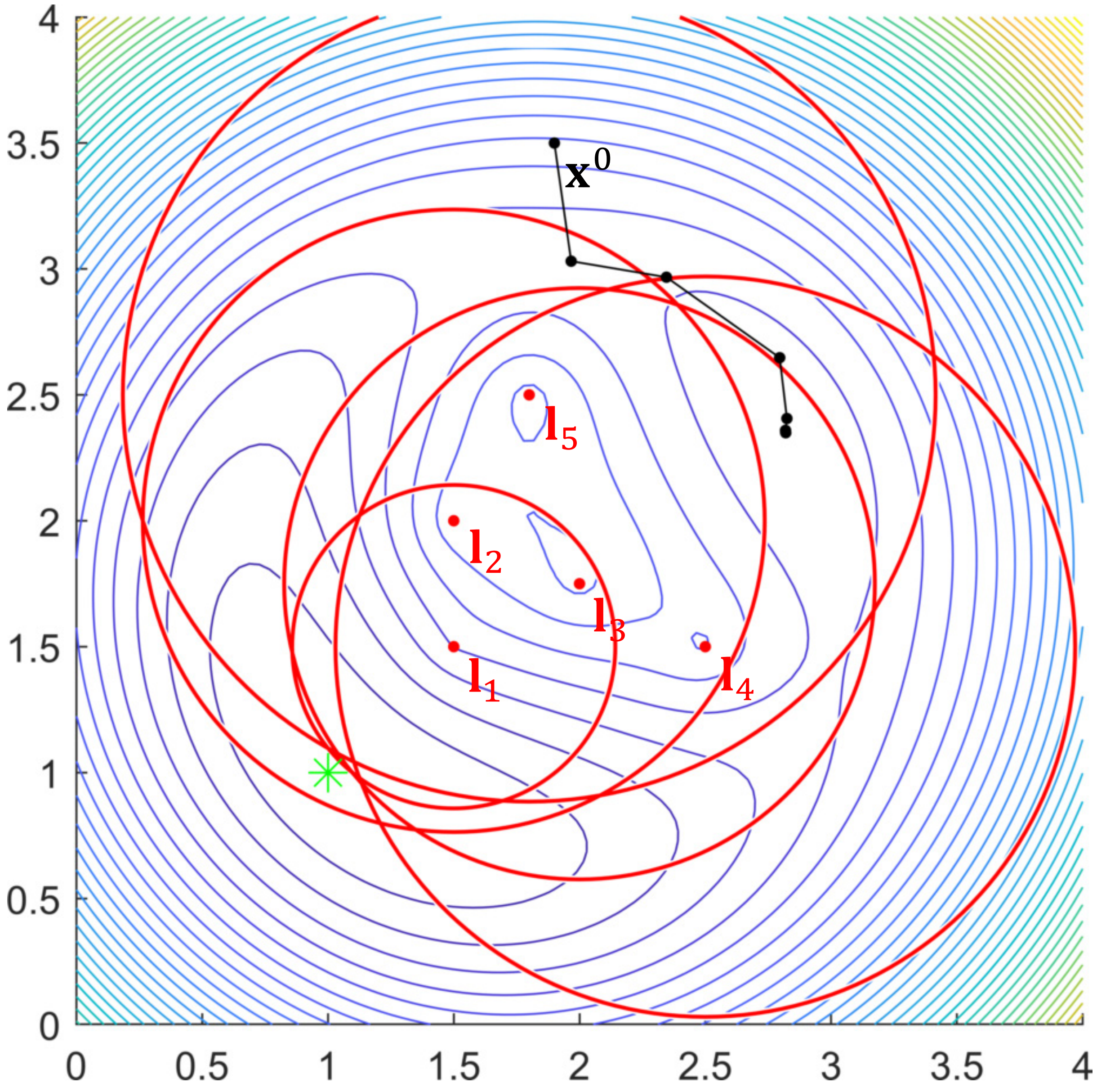




Example:
Range-based localization

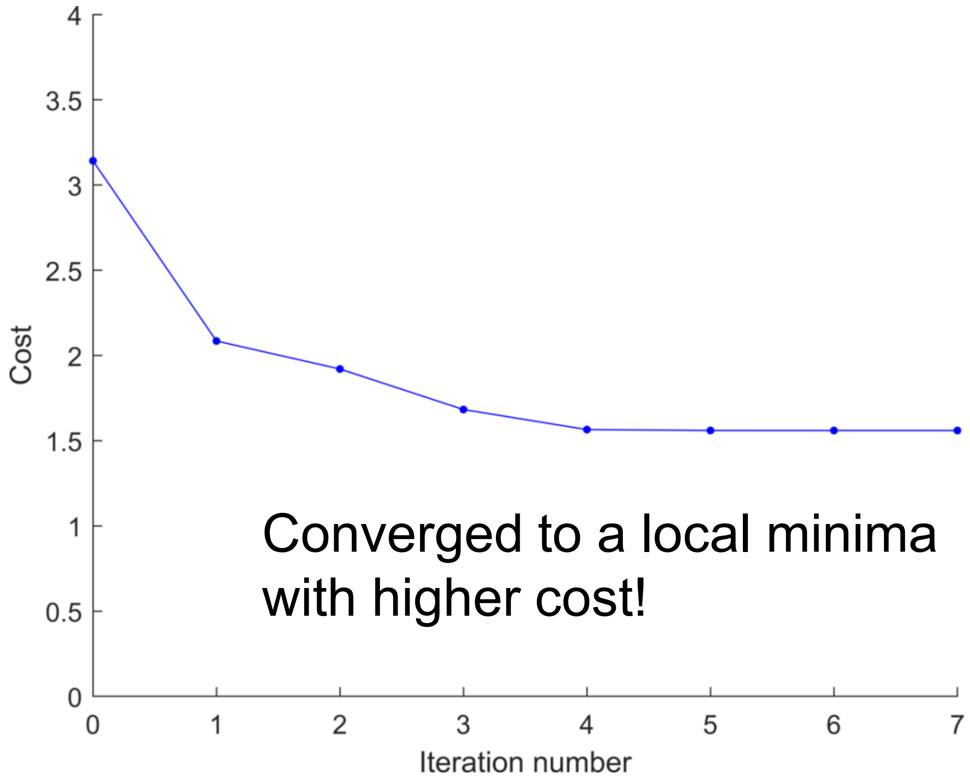
Levenberg–Marquardt optimization
- Slightly different initial estimate



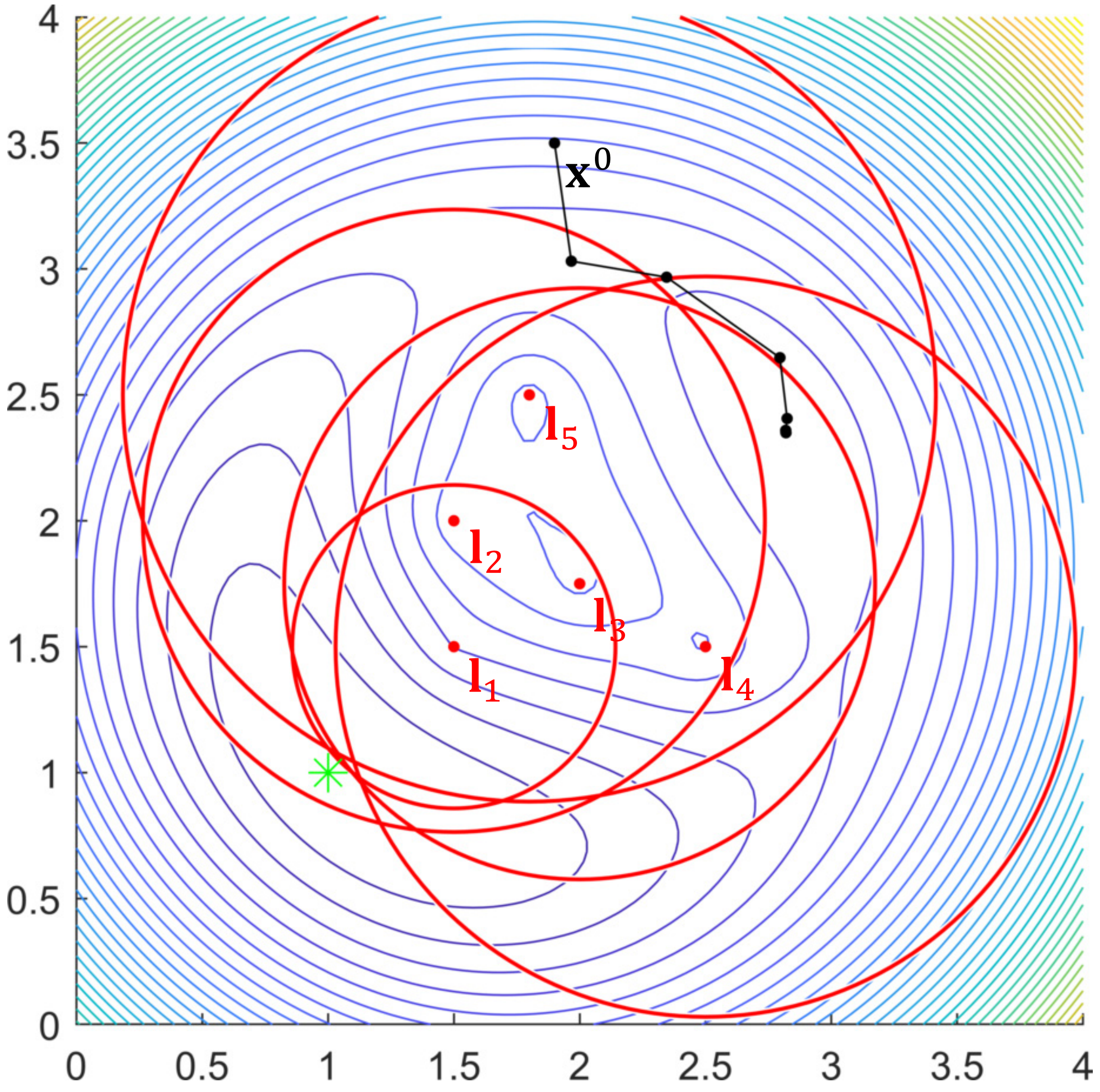


Example:
Range-based localization

Levenberg–Marquardt optimization
- Slightly different initial estimate



Converged to a local minima with higher cost!



Nonlinear MAP inference for state estimation

Measurement model:

$$\mathbf{z}_i = h(X_i) + \eta, \qquad \eta \sim N(\mathbf{0}, \boldsymbol{\Sigma}_i)$$

Measurement prediction function:

$$\hat{\mathbf{z}}_i = h(X_i)$$

Measurement error function:

$$e_i(X_i) = h_i(X_i) - \mathbf{z}_i$$

Objective function:

$$f(X) = \sum_{i=1}^{m} \left\| h_i(X_i) - \mathbf{z}_i \right\|_{\boldsymbol{\Sigma}_i}^2$$

where $\left\|\mathbf{e}\right\|_{\boldsymbol{\Sigma}}^2 = \mathbf{e}^T \boldsymbol{\Sigma}^{-1} \mathbf{e}$ is the Mahalanobis norm.

This results in the nonlinear least squares problem:

$$X^* = \underset{X}{\operatorname{argmin}} \sum_{i=1}^{m} \left\| h_i(X_i) - \mathbf{z}_i \right\|_{\boldsymbol{\Sigma}_i}^2$$

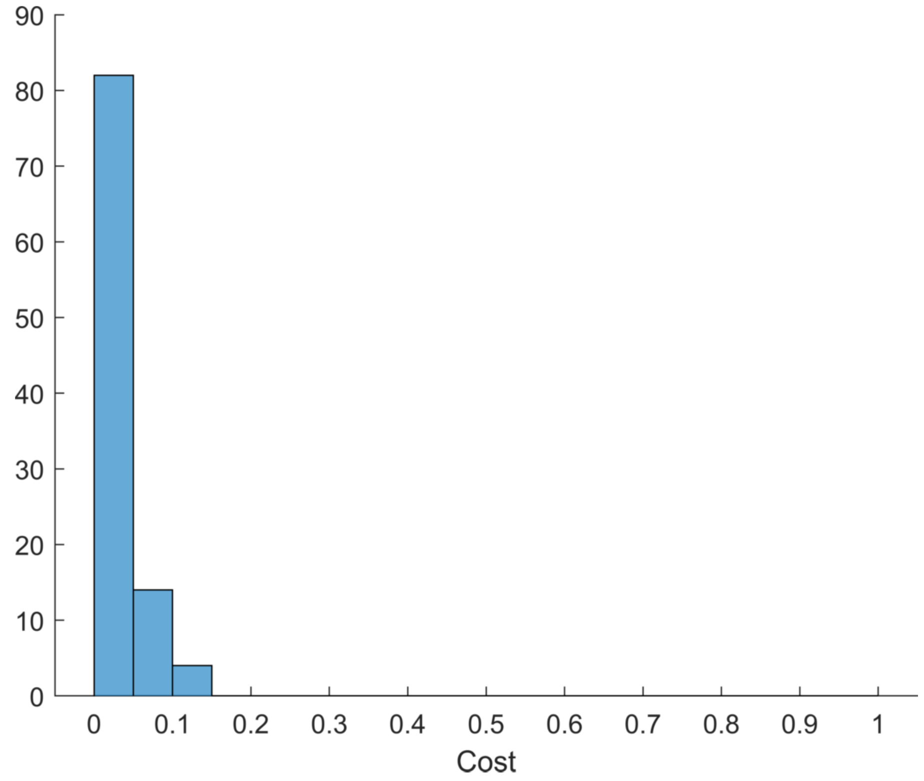
It turns out that the nonlinear least squares solution to this problem is the MAP estimate!

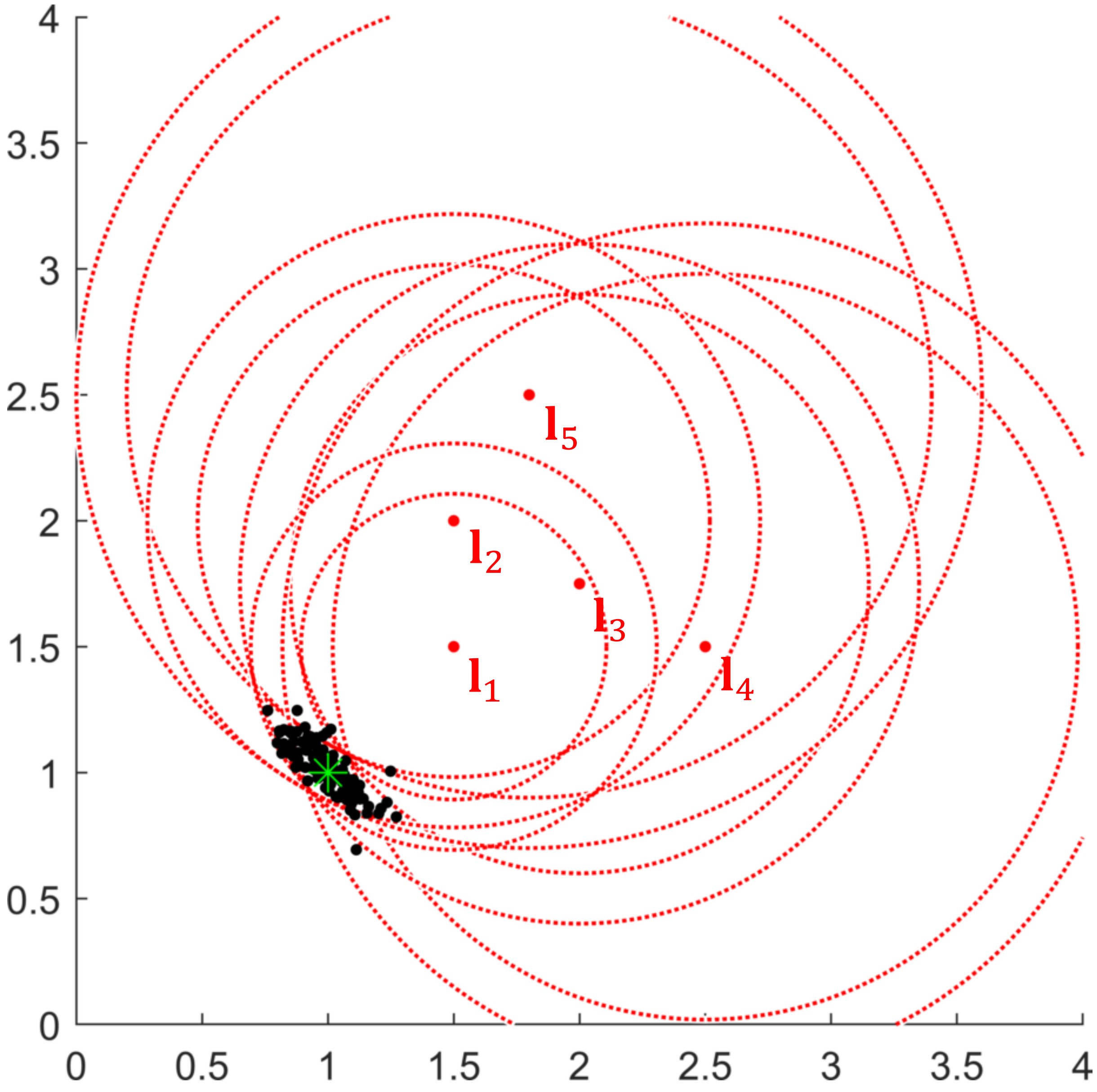
~~Assume for now that all $\boldsymbol{\Sigma}_i = \sigma \mathbf{I}$.
This simplifies our objective to:~~

$$X^* = \underset{X}{\operatorname{argmin}} \sum_{i=1}^{m} \left\| h_i(X_i) - \mathbf{z}_i \right\|^2$$

What about measurement noise?

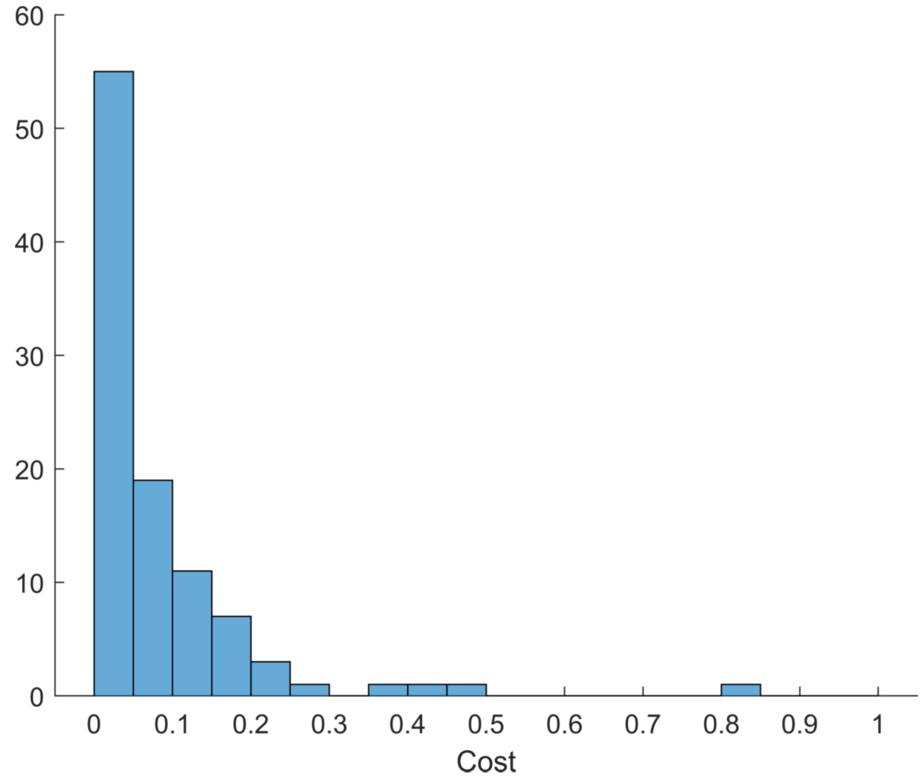
100 runs, $\sigma_i = 0.1$

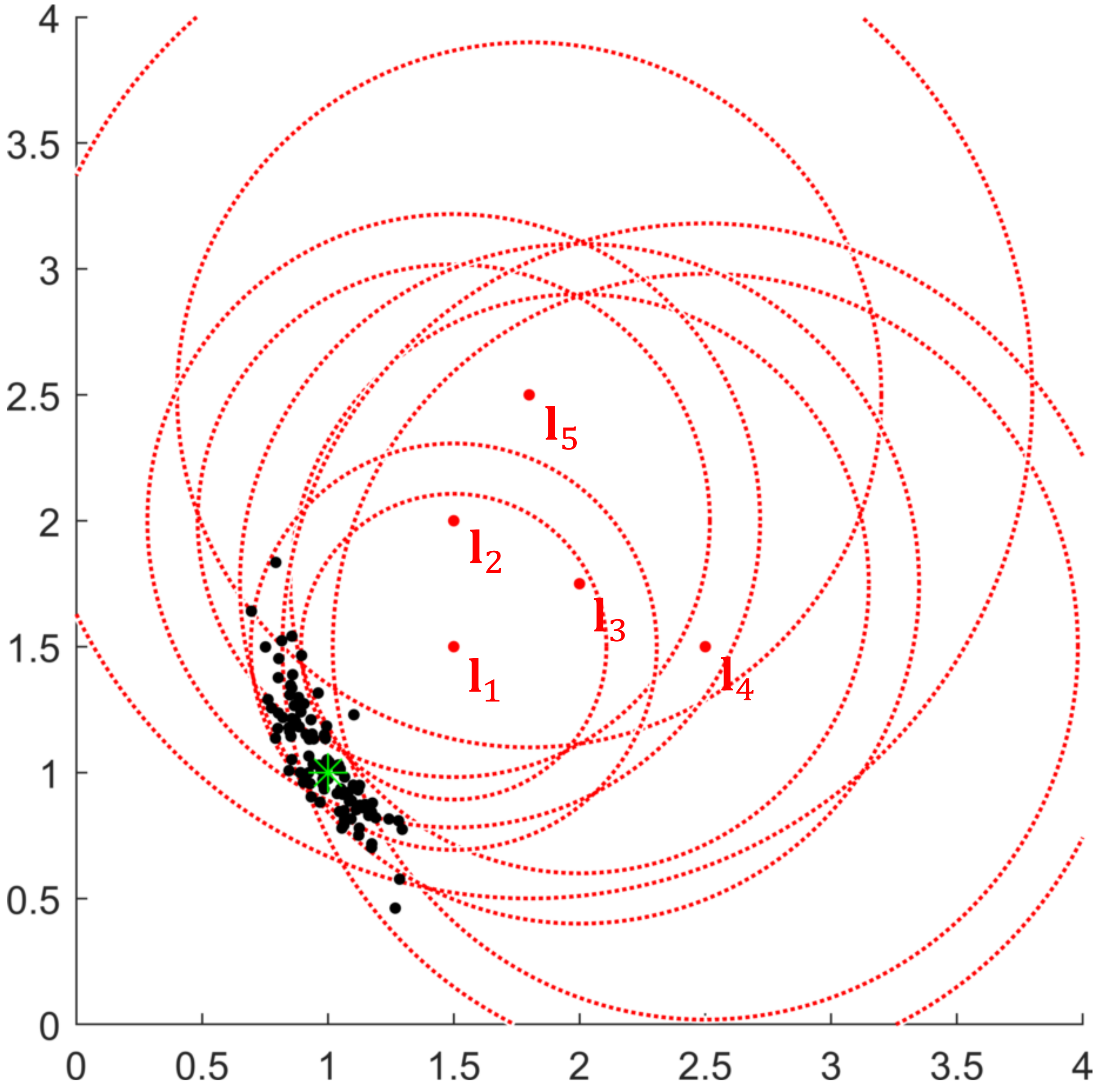




What about measurement noise?

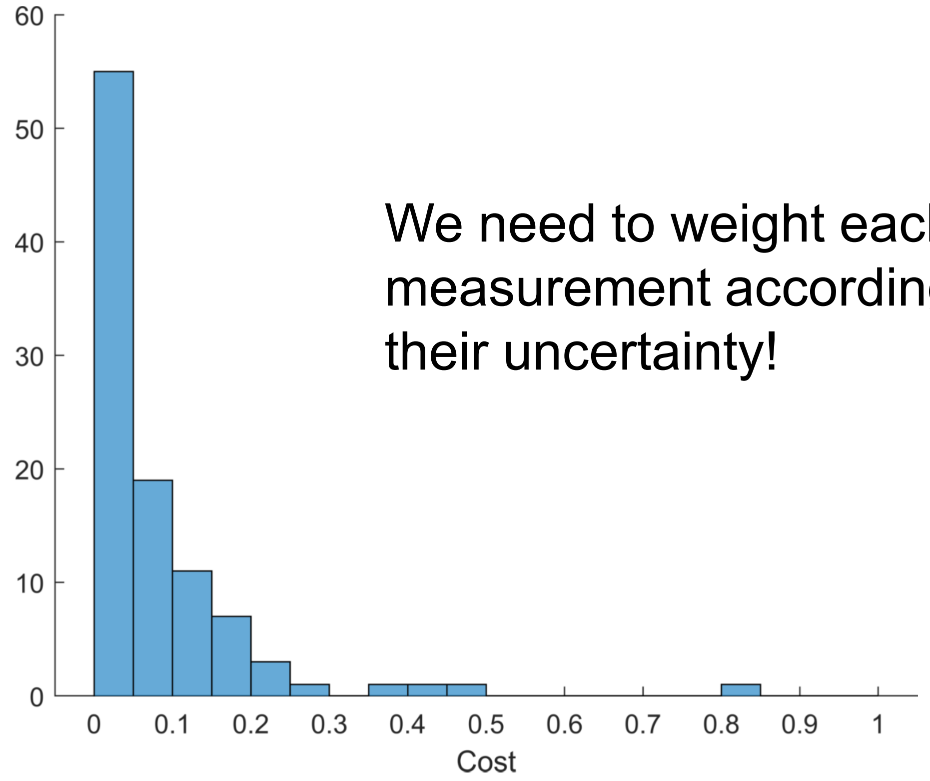
100 runs, $\sigma_1, \ldots, \sigma_4 = 0.1, \sigma_5 = 0.3$



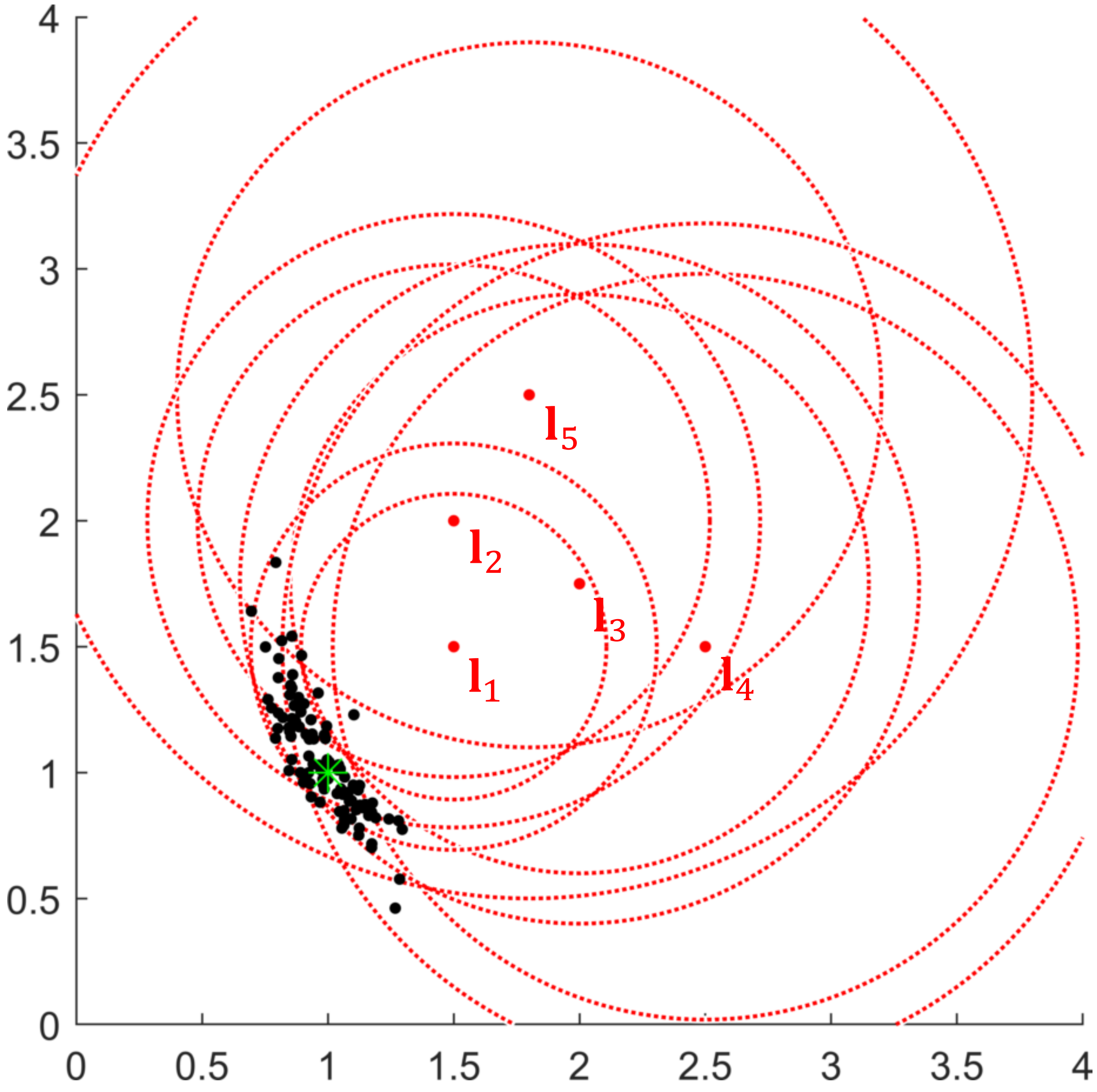


What about measurement noise?

100 runs, $\sigma_1, \dots, \sigma_4 = 0.1, \sigma_5 = 0.3$



We need to weight each measurement according to their uncertainty!



Weighted nonlinear least squares

We can rewrite the Mahalanobis norms as

$$\|\mathbf{e}\|_{\boldsymbol{\Sigma}}^2 \triangleq \mathbf{e}^T \boldsymbol{\Sigma}^{-1} \mathbf{e} = \left(\boldsymbol{\Sigma}^{-1/2}\mathbf{e}\right)^T \left(\boldsymbol{\Sigma}^{-1/2}\mathbf{e}\right) = \left\|\boldsymbol{\Sigma}^{-1/2}\mathbf{e}\right\|^2$$

Hence, we can eliminate the covariances
by weighting the Jacobian and the prediction error:

$$\mathbf{A}_i = \boldsymbol{\Sigma}_i^{-1/2}\mathbf{H}_i$$
$$\mathbf{b}_i = \boldsymbol{\Sigma}_i^{-1/2}\left(\mathbf{z}_i - h_i(X_i^0)\right)$$

This is a form of whitening,
which eliminates the units of the measurements

Weighted linearized problem

This results in linear error functions $e_i(X_i^0 + \Delta)$,
and we obtain a **linear least squares** problem in the state update vector Δ:

$$\begin{aligned}
\boldsymbol{\Delta}^* &= \underset{\boldsymbol{\Delta}}{\operatorname{argmin}} \sum_i \left\| h_i(X_i^0) + \mathbf{H}_i \boldsymbol{\Delta}_i - \mathbf{z}_i \right\|_{\boldsymbol{\Sigma}_i}^2 \\
&= \underset{\boldsymbol{\Delta}}{\operatorname{argmin}} \sum_i \left\| \mathbf{H}_i \boldsymbol{\Delta}_i - \left\{ \mathbf{z}_i - h_i(X_i^0) \right\} \right\|_{\boldsymbol{\Sigma}_i}^2 \\
&= \underset{\boldsymbol{\Delta}}{\operatorname{argmin}} \sum_i \left\| \mathbf{A}_i \boldsymbol{\Delta}_i - \mathbf{b}_i \right\|^2 \\
&= \underset{\boldsymbol{\Delta}}{\operatorname{argmin}} \left\| \mathbf{A} \boldsymbol{\Delta} - \mathbf{b} \right\|^2
\end{aligned}$$

$\longleftarrow$

$$\begin{aligned}
\mathbf{A}_i &= \boldsymbol{\Sigma}_i^{-1/2} \mathbf{H}_i \\
\mathbf{b}_i &= \boldsymbol{\Sigma}_i^{-1/2} \left(\mathbf{z}_i - h_i(X_i^0) \right)
\end{aligned}$$

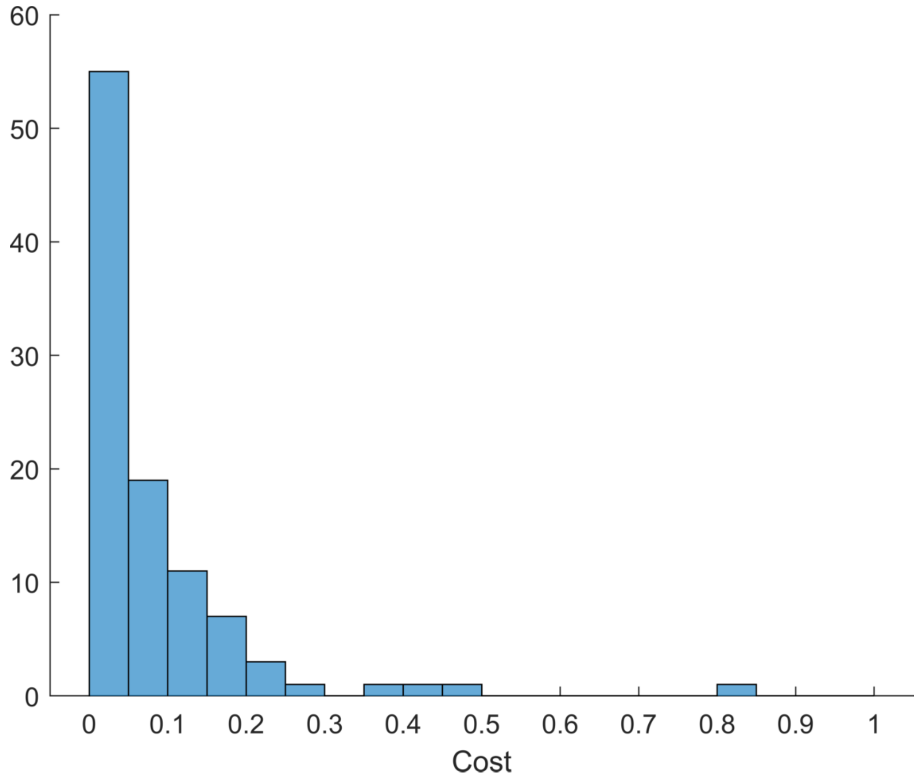
Which, as before, can be solved using the normal equations:

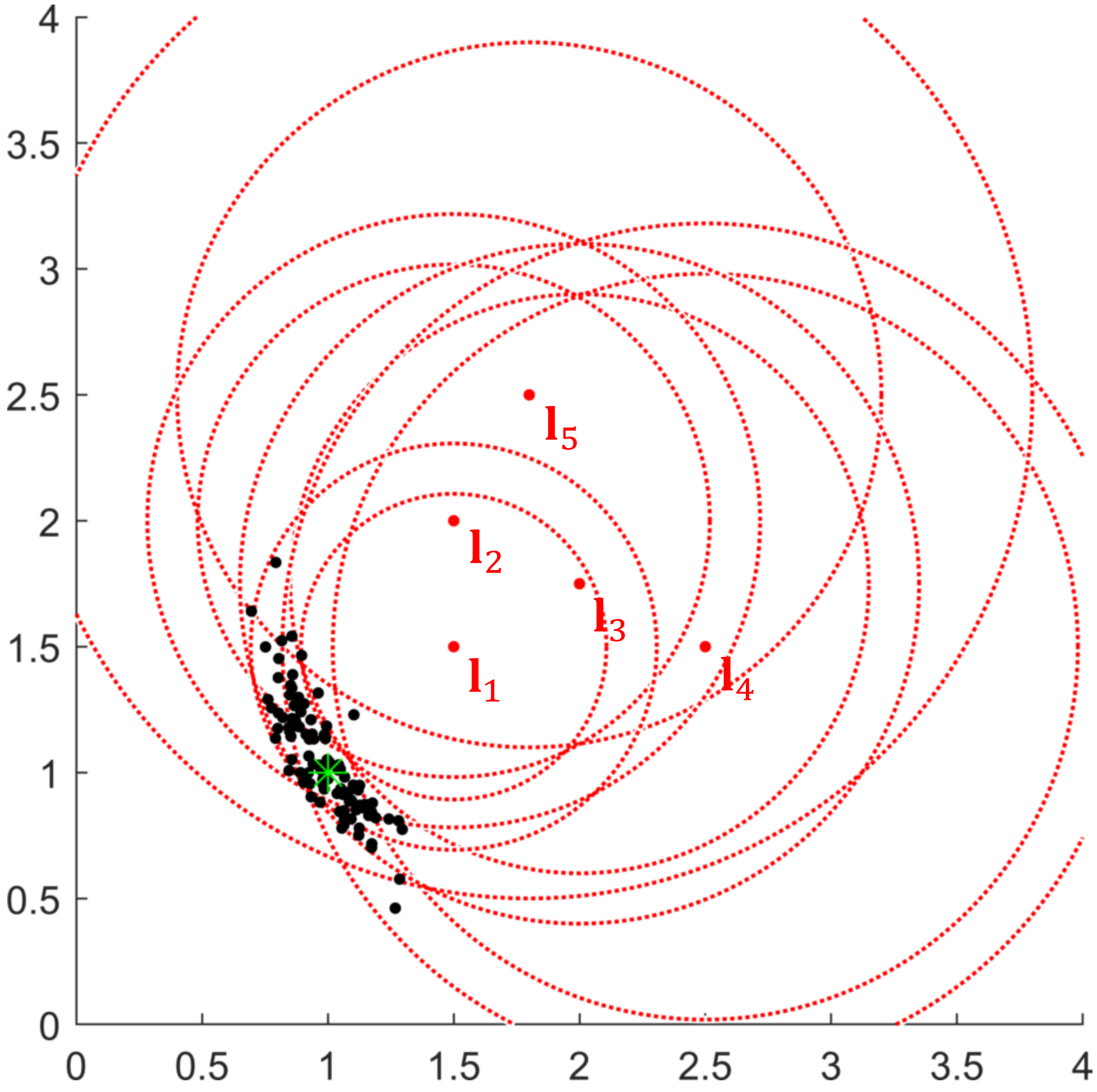
$$\mathbf{A}^T \mathbf{A} \boldsymbol{\Delta}^* = \mathbf{A}^T \mathbf{b}$$

What about measurement noise?

100 runs, $\sigma_1, \ldots, \sigma_4 = 0.1, \sigma_5 = 0.3$

Unweighted

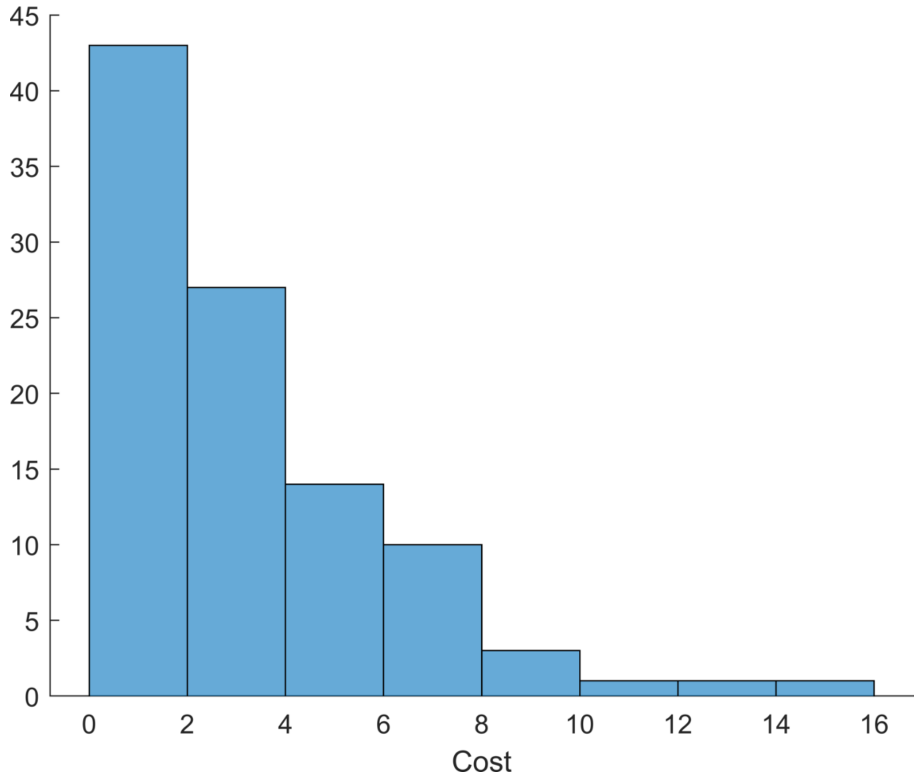


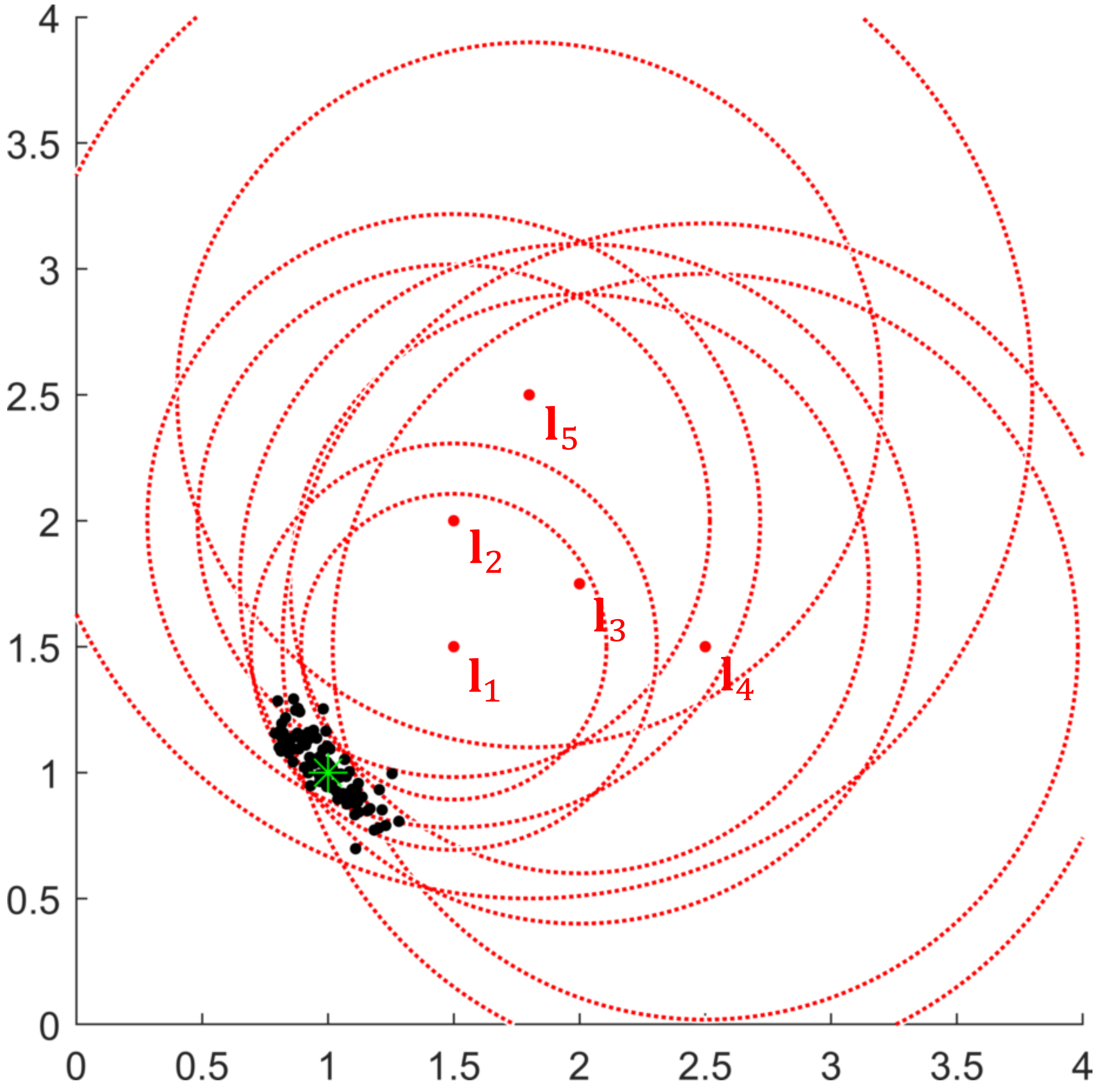


What about measurement noise?

100 runs, $\sigma_1, \dots, \sigma_4 = 0.1, \sigma_5 = 0.3$

Covariance weighted (whitened)





Estimating uncertainty in the MAP estimate

The Hessian at the solution for the weighted problem
is the inverse of the covariance matrix (the information matrix)!

$$\left.\frac{\partial^2 f(\mathbf{x})}{\partial \mathbf{x} \partial \mathbf{x}^T}\right|_{\mathbf{x}^*} = \Lambda = \Sigma^{-1}$$

Using our approximated Hessian,
we obtain a first order approximation of the true covariance for all states

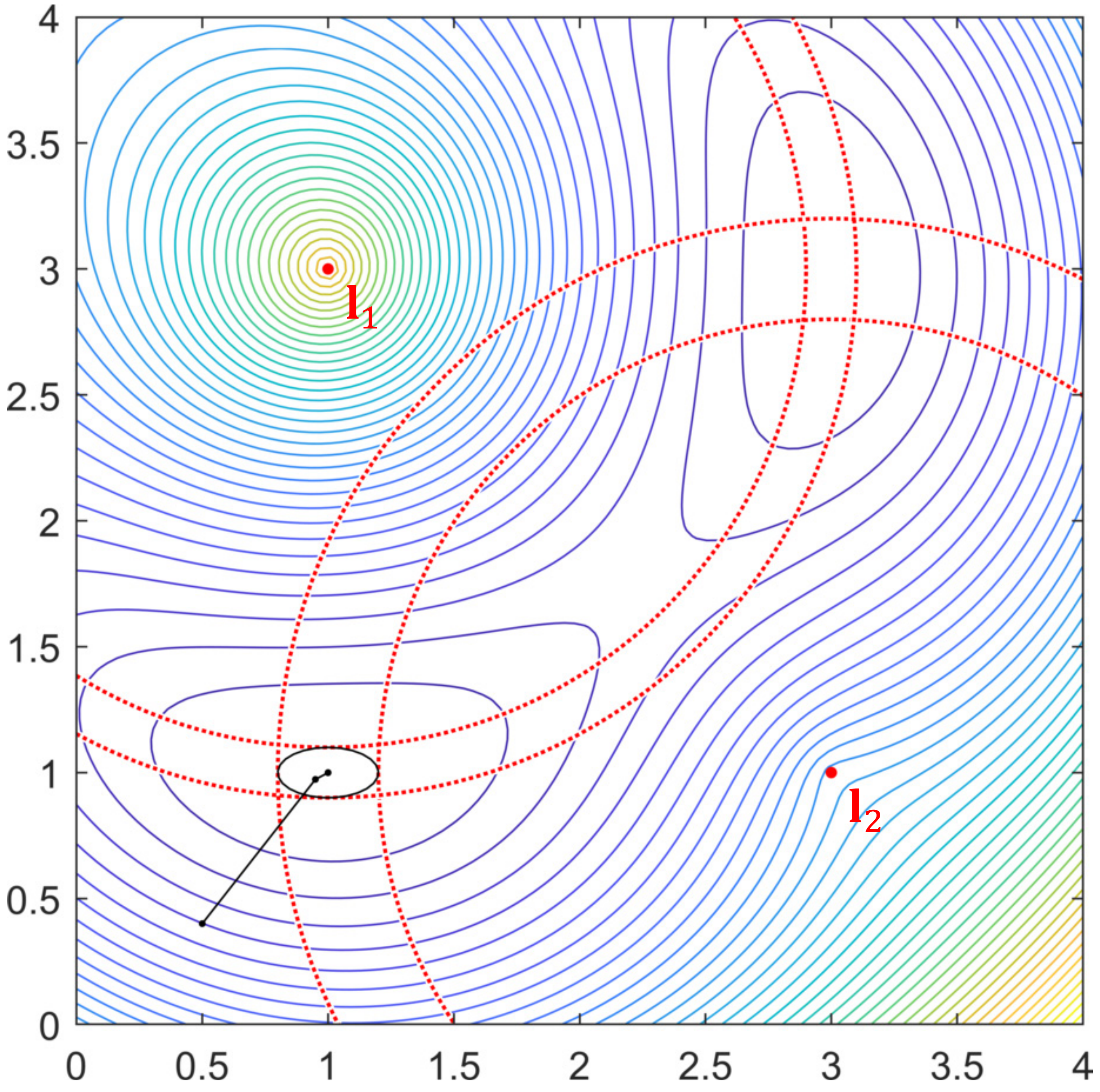
$$\Sigma_{X^*} \approx (\mathbf{A}_{X^*}^T \mathbf{A}_{X^*})^{-1}$$

Simple example: Two landmarks

(No noise added to measurements)

1σ covariance contours

$$\Sigma_{X^*} \approx (\mathbf{A}_{X^*}^T \mathbf{A}_{X^*})^{-1}$$

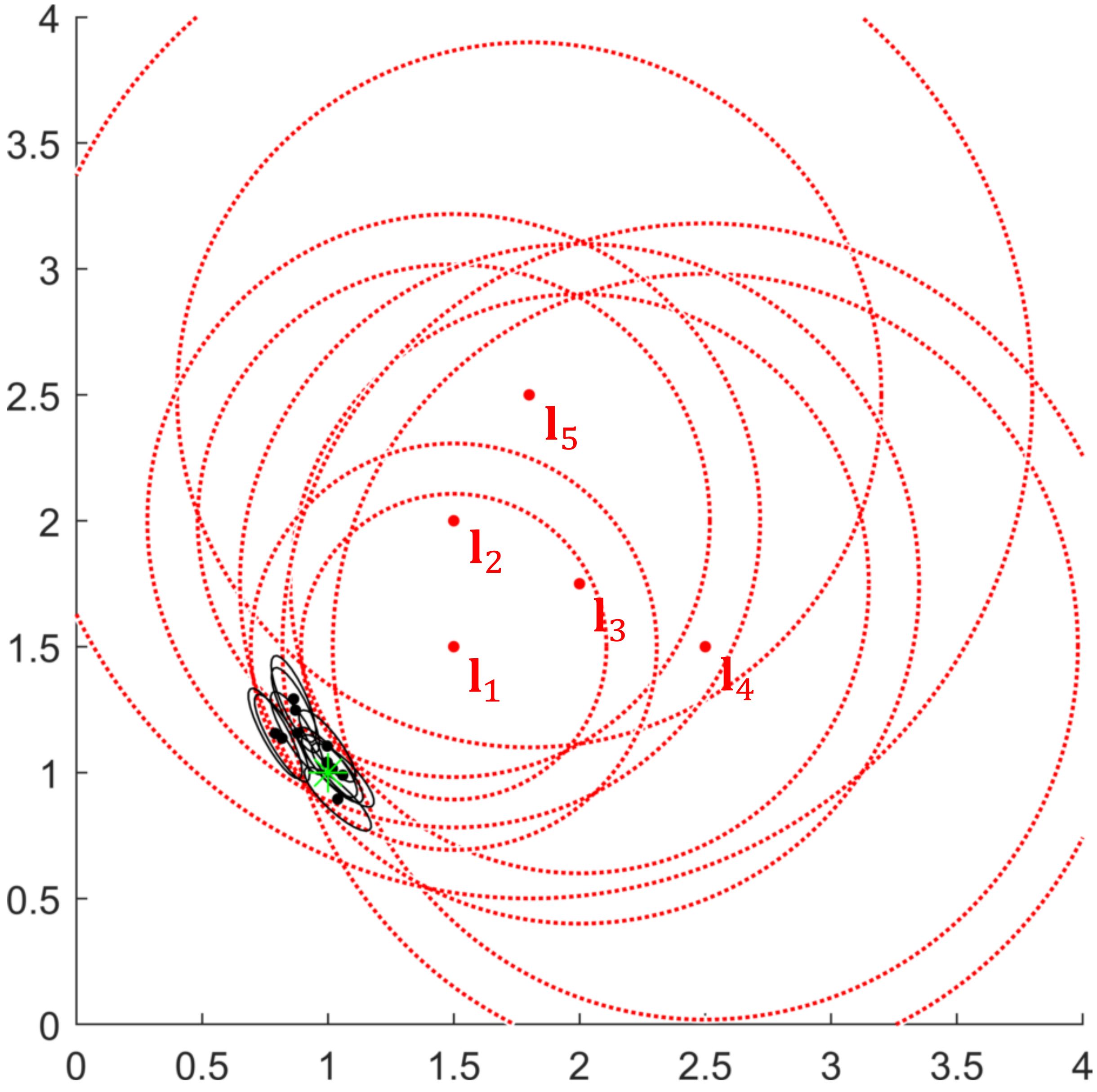


Example:
Range-based localization

10 runs, $\sigma_1, \ldots, \sigma_4 = 0.1, \sigma_5 = 0.3$

1σ covariance contours

$$\Sigma_{X^*} \approx (\mathbf{A}_{X^*}^T \mathbf{A}_{X^*})^{-1}$$



Summary

We have seen how we can find the MAP estimate
of our unknown states given measurements

$$X^{MAP} = \underset{X}{\operatorname{argmax}}\, p(X \mid Z)$$

by representing it as
a nonlinear least squares problem

$$X^{*} = \underset{X}{\operatorname{argmin}} \sum_{i=1}^{m} \left\| h_i(X_i) - \mathbf{z}_i \right\|_{\boldsymbol{\Sigma}_i}^{2}$$

Choose a suitable inital estimate X^{0}

⇓

$\mathbf{A}, \mathbf{b} \leftarrow \text{Linearize at } X^{t}$

⇓

$\boldsymbol{\Delta}^{*} \leftarrow \text{Solve } \underset{\boldsymbol{\Delta}}{\operatorname{argmin}} \left\| \mathbf{A}\boldsymbol{\Delta} - \mathbf{b} \right\|^{2}$

⇓

$X^{t+1} \leftarrow X^{t} + \boldsymbol{\Delta}^{*}$

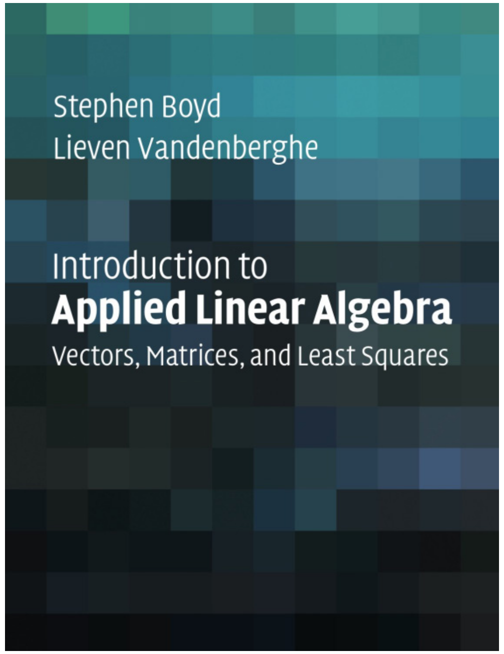
Summary

We are now almost ready to solve

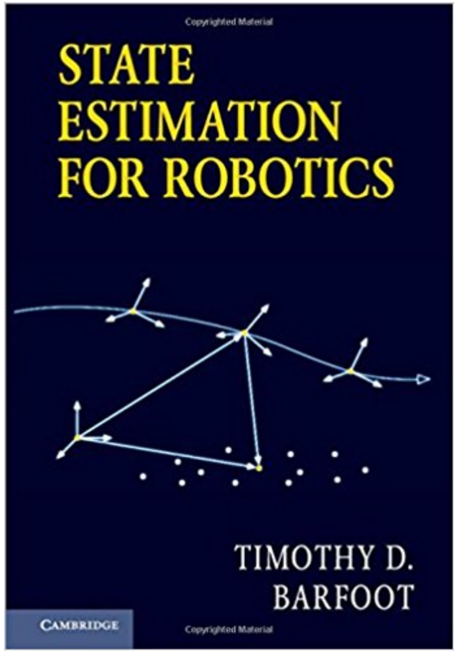
$$\mathbf{T}_{cw}^{*} = \underset{\mathbf{T}_{cw}}{\operatorname{argmin}} \sum_{i} \left\| \pi(\mathbf{T}_{cw}\tilde{\mathbf{x}}_i^w) - \mathbf{u}_i \right\|^2$$

We just need to know how to optimize over poses!

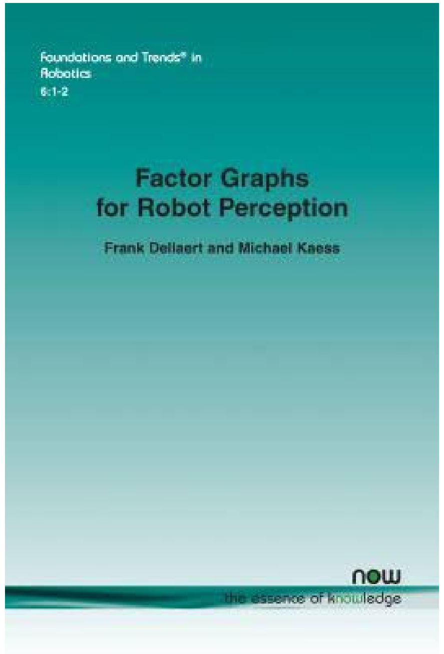
Further reading



http://vmls-book.stanford.edu/



http://asrl.utias.utoronto.ca/~tdb/
http://asrl.utias.utoronto.ca/~tdb/bib/barfoot_ser17.pdf



http://frc.ri.cmu.edu/~kaess/pub/Dellaert17fnt.pdf