# From LSQ to NLSQ

**Michal Reinštein**

Czech Technical University in Prague
Faculty of Electrical Engineering, Department of Cybernetics
Center for Machine Perception
`http://cmp.felk.cvut.cz/~reinsmic,`
`reinstein.michal@fel.cvut.cz`

## Outline of the lecture:

- LSQ - Least Squares
- LSQ - The Proof
- WLSQ- Weighted LSQ
- NLSQ - Non-linear LSQ
- Exercise: Long Base-line 3D Navigation
- Exercise: NLSQ in MATLAB

# LSQ - Least Squares Estimation

Given measurements $\mathbf{z}$, we wish to solve for $\mathbf{x}$, assuming linear relationship:

$$\mathbf{H}\mathbf{x} = \mathbf{z}$$

If $\mathbf{H}$ is a square matrix with $\det \mathbf{H} \neq 0$ then the solution is trivial:

$$\mathbf{x} = \mathbf{H}^{-1}\mathbf{z},$$

otherwise (most commonly), we seek such solution $\hat{\mathbf{x}}$ that is closest (in Euclidean distance sense) to the ideal:

$$\hat{\mathbf{x}} = \operatorname*{argmin}_{x} ||\mathbf{H}\mathbf{x} - \mathbf{z}||^2 = \operatorname*{argmin}_{x} \left\{ (\mathbf{H}\mathbf{x} - \mathbf{z})^\top (\mathbf{H}\mathbf{x} - \mathbf{z}) \right\}$$

# LSQ - The Proof

Given the following matrix identities:

- $(\mathbf{AB})^\top = \mathbf{B}^\top \mathbf{A}^\top$
- $||\mathbf{x}||^2 = \mathbf{x}^\top \mathbf{x}$
- $\nabla_x \, \mathbf{b}^\top \mathbf{x} = \mathbf{b}$
- $\nabla_x \, \mathbf{x}^\top \mathbf{A} \mathbf{x} = 2\mathbf{A}\mathbf{x}$

We can derive the closed form solution[1]:

$$||\mathbf{Hx} - \mathbf{z}||^2 = \mathbf{x}^\top \mathbf{H}^\top \mathbf{H x} - \mathbf{x}^\top \mathbf{H}^\top \mathbf{z} - \mathbf{z}^\top \mathbf{H x} + \mathbf{z}^\top \mathbf{z}$$

$$\frac{\partial ||\mathbf{Hx} - \mathbf{z}||^2}{\partial \mathbf{x}} = 2\mathbf{H}^\top \mathbf{H x} - 2\mathbf{H}^\top \mathbf{z} = 0$$

$$\Rightarrow \quad \boxed{\mathbf{x} = (\mathbf{H}^\top \mathbf{H})^{-1} \mathbf{H}^\top \mathbf{z}}$$

[1] in MATLAB use the pseudo-inverse *pinv()*

# LSQ - Weighted Least Squares

If we have information about reliability of the measurements in $\mathbf{z}$, we can capture this as a covariance matrix $\mathbf{R}$ (diagonal terms only since the measurements are not correlated:

$$\mathbf{R} = \begin{bmatrix} \sigma_{z1}^2 & 0 & 0 \\ 0 & \sigma_{z2}^2 & \dots \\ \vdots & \vdots & \ddots \end{bmatrix}$$

In the error vector $\mathbf{e}$ defined as $\mathbf{e} = \mathbf{Hx} - \mathbf{z}$ we can weight each its element by uncertainty in each element of the measurement vector $\mathbf{z}$, i.e. by $\mathbf{R}^{-1}$. The optimization criteria then becomes:

$$\boxed{\hat{\mathbf{x}} = \operatorname*{argmin}_x ||\mathbf{R}^{-1}(\mathbf{Hx} - \mathbf{z})||^2}$$

Following the same derivation procedure, we obtain the weighted least squares:

$$\Rightarrow \quad \boxed{\mathbf{x} = (\mathbf{H}^\top\mathbf{R}^{-1}\mathbf{H})^{-1}\mathbf{H}^\top\mathbf{R}^{-1}\mathbf{z}}$$

# NLSQ - Non-linear Least Squares

Previous example concerned a linear observation model, however, in real world most of the models are rather a nonlinear function $\mathbf{h}(\mathbf{x})$. Measuring a Euclidean distance between two points, the task is reformulated:

$$\hat{\mathbf{x}} = \underset{x}{\operatorname{argmin}} \, ||(\mathbf{h}(\mathbf{x}) - \mathbf{z})||^2$$

# NLSQ - Non-linear Least Squares

The world is non-linear $\rightarrow$ nonlinear model function $\mathbf{h}(\mathbf{x})$ $\rightarrow$ non-linear LSQ[2]:

$$\hat{\mathbf{x}} = \underset{x}{\operatorname{argmin}} \, ||\mathbf{h}(\mathbf{x}) - \mathbf{z}||^2$$

- We seek such $\delta$ that for $\mathbf{x}_1 = \mathbf{x}_0 + \delta$ the $||\mathbf{h}(\mathbf{x}_1) - \mathbf{z}||^2$ is minimized.
- We use Taylor series expansion: $\mathbf{h}(\mathbf{x}_0 + \delta) = \mathbf{h}(\mathbf{x}_0) + \nabla \mathbf{H}_{\mathbf{x}0}\delta$

$$||\mathbf{h}(\mathbf{x}_1) - \mathbf{z}||^2 = ||\mathbf{h}(\mathbf{x}_0) + \nabla \mathbf{H}_{\mathbf{x}0}\delta - \mathbf{z}||^2 = ||\nabla \mathbf{H}_{\mathbf{x}0}\delta - (\mathbf{z} - \mathbf{h}(\mathbf{x}_0)||^2$$

where $\nabla \mathbf{H}_{\mathbf{x}0}$ is Jacobian of $\mathbf{h}(\mathbf{x})$:

$$\nabla \mathbf{H}_{\mathbf{x}0} = \frac{\partial \mathbf{h}}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial \mathbf{h}_1}{\partial \mathbf{x}_1} & \cdots & \frac{\partial \mathbf{h}_1}{\partial \mathbf{x}_m} \\ \vdots & & \vdots \\ \frac{\partial \mathbf{h}_n}{\partial \mathbf{x}_1} & \cdots & \frac{\partial \mathbf{h}_n}{\partial \mathbf{x}_m} \end{bmatrix}$$

[2]Note: We still measure the Euclidean distance between two points that we want to optimize over.

# NLSQ - Non-linear Least Squares

- We use Taylor series expansion: $\mathbf{h}(\mathbf{x}_0 + \delta) = \mathbf{h}(\mathbf{x}_0) + \nabla\mathbf{H}_{\mathbf{x}0}\delta$

$$||\mathbf{h}(\mathbf{x}_1) - \mathbf{z}||^2 = ||\mathbf{h}(\mathbf{x}_0) + \nabla\mathbf{H}_{\mathbf{x}0}\delta - \mathbf{z}||^2 = ||\underbrace{\nabla\mathbf{H}_{\mathbf{x}0}}_{\mathbf{A}}\delta - \underbrace{(\mathbf{z} - \mathbf{h}(\mathbf{x}_0))}_{\mathbf{b}}||^2$$

- We solve it as standard least squares $\mathbf{A}\delta = \mathbf{b}$ and hence by inspection:

$$\delta = (\nabla\mathbf{H}_{\mathbf{x}0}^\top\nabla\mathbf{H}_{\mathbf{x}0})^{-1}\nabla\mathbf{H}_{\mathbf{x}0}^\top(\mathbf{z} - \mathbf{h}(\mathbf{x}_0))$$

## NLSQ - Non-linear Least Squares



The extension of LSQ to the non-linear LSQ can be formulated as an algorithm:

1. Start with an initial guess $\hat{\mathbf{x}}$. [3]
2. Evaluate the LSQ expression for $\delta$ (update the $\nabla \mathbf{H}_{\hat{\mathbf{x}}}$ and substitute). [4]

$$\delta := (\nabla \mathbf{H}_{\hat{\mathbf{x}}}{}^\top \nabla \mathbf{H}_{\hat{\mathbf{x}}})^{-1} \nabla \mathbf{H}_{\hat{\mathbf{x}}}{}^\top [\mathbf{z} - \mathbf{h}(\hat{\mathbf{x}})]$$

3. Apply the $\delta$ correction to our initial estimate: $\hat{\mathbf{x}} := \hat{\mathbf{x}} + \delta$.[5]
4. Check for the stopping precision: if $||\mathbf{h}(\hat{\mathbf{x}}) - \mathbf{z}||^2 > \epsilon$ proceed with step $(2)$ or stop otherwise.[6]

[3] Note: We can usually set to zero.
[4] Note: This expression is obtained using the LSQ closed form and substitution from previous slide.
[5] Note: Due to these updates our initial guess should converge to such $\hat{\mathbf{x}}$ that minimizes the $||\mathbf{h}(\hat{\mathbf{x}}) - \mathbf{z}||^2$
[6] Note: $\epsilon$ is some small threshold, usually set according to the noise level in the sensors.

# Exercise: Assignment



## Long Base-line Navigation

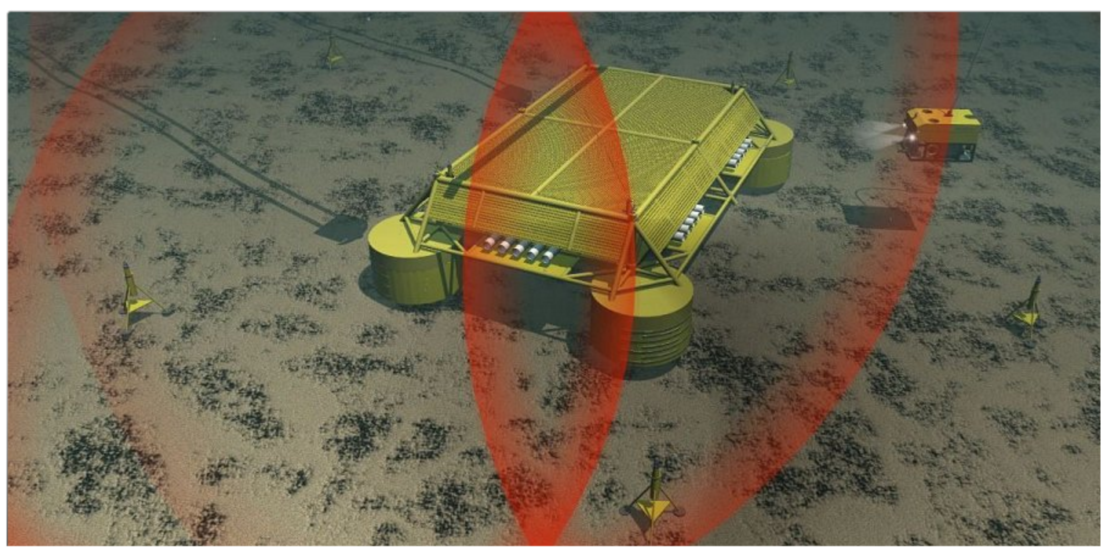
Assume an underwater robot operating within the range of 4 beacons and receiving time-of-flight measurements simultaneously and without delay.

We wish to find the LSQ estimate of robot position $\mathbf{x}_v = [x, y, z]^\top$ while each beacon $i$ is at known position $\mathbf{x}_{bi} = [x_{bi}, y_{bi}, z_{bi}]^\top$. We assume the transceiver operates at speed of sound $c$.

- Write NLSQ algorithm for estimating the robot position.
- Plot the precision vs iteration curve.
- Play with the algorithm by changing: initial position, measurements noise, stopping criteria.

# Exercise: Assignment

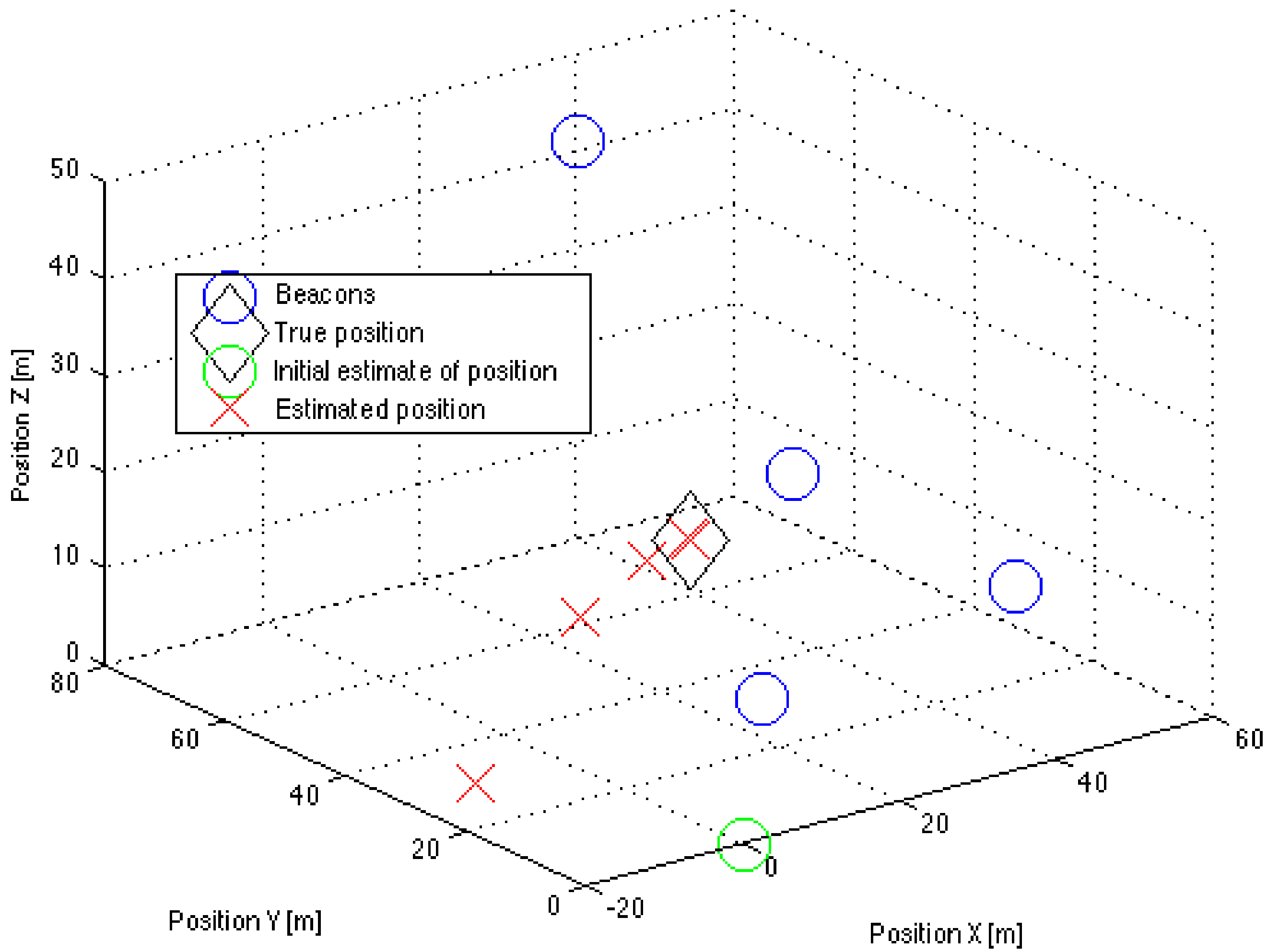
## Long Base-line Navigation SONARDYNE



# Exercise: Solution

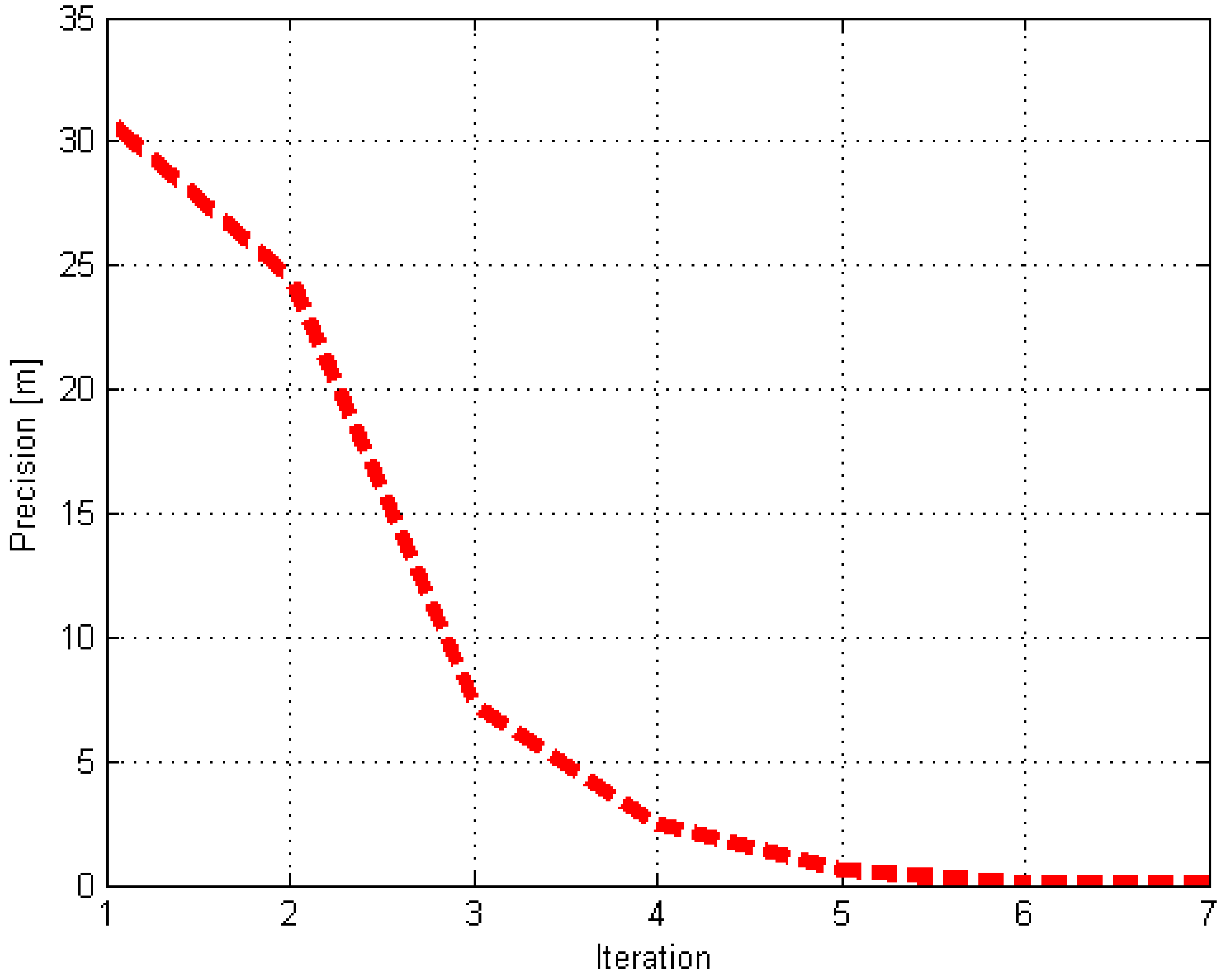


## Long Base-line Navigation



# Exercise: Solution

## Long Base-line Navigation



# Exercise: Solution

## Long Base-line Navigation

Assume an underwater robot operating within the range of 4 beacons and receiving time-of-flight measurements simultaneously and without delay.

We wish to find the LSQ estimate of robot position $\mathbf{x}_v = [x, y, z]^\top$ while each beacon $i$ is at known position $\mathbf{x}_{bi} = [x_{bi}, y_{bi}, z_{bi}]^\top$. The observation model is[7]:

$$\mathbf{z} = \begin{bmatrix} t_1 \\ t_2 \\ t_3 \\ t_4 \end{bmatrix} = h(\mathbf{x}_v) = \frac{2}{c} \begin{bmatrix} ||\mathbf{x}_{b1} - \mathbf{x}_v|| \\ ||\mathbf{x}_{b2} - \mathbf{x}_v|| \\ ||\mathbf{x}_{b3} - \mathbf{x}_v|| \\ ||\mathbf{x}_{b4} - \mathbf{x}_v|| \end{bmatrix}$$

where $t_i$ is the measured time-of-flight from beacon $i$.

[7] Note: We assume the transceiver operates at speed of sound $c$

# Exercise: Solution



## Long Base-line Navigation

We derive the $\nabla \mathbf{H}_{\mathbf{x}v}$ and plug it into the 4-step algorithm already introduced:

$$\nabla \mathbf{H}_{\mathbf{x}v} = -\frac{2}{c} \begin{bmatrix} \Delta_{x1} & \Delta_{y1} & \Delta_{z1} \\ \Delta_{x2} & \Delta_{y2} & \Delta_{z2} \\ \Delta_{x3} & \Delta_{y3} & \Delta_{z3} \\ \Delta_{x4} & \Delta_{y4} & \Delta_{z4} \end{bmatrix}$$

where:

$$\Delta_{xi} = (x_{bi} - x)/r_i, \Delta_{yi} = (y_{bi} - y)/r_i, \Delta_{zi} = (z_{bi} - z)/r_i$$

$$r_i = \sqrt{(x_{bi} - x)^2 + (y_{bi} - y)^2 + (z_{bi} - z)^2}$$

# Exercise: Solution

## Long Base-line Navigation

```
%% Non-linear least squares solution to the Long Base-line Navigation
precision_history = [];                          % initialization precision history [m]
desired_precision = 0.001;                       % desired precision of the estimated position [m]
c = 343;                                         % speed fo sound [mps]
dH = zeros(4,3);                                 % initial Jacobian values
Xb = [10 50 60 25; 10 20 70 60; 10 10 5 50];     % known beacon positions [m]
Xv_est = [0; 0; 0];                              % initial estimate of vehicle position [m]
Xv_true = [5.123; 15.456; 25.789];               % unknown true vehicle position [m]
% generating time-of-flight measurements (no sensor noise assumed):
Xdiff_true = Xb - repmat(Xv_true, 1, size(Xb, 2));
Ztof = 2*([norm(Xdiff_true(:,1)); norm(Xdiff_true(:,2)); norm(Xdiff_true(:,3)); norm(Xdiff_true(:,4))])/c;

Xdiff_est = Xb - repmat(Xv_est, 1, size(Xb, 2));
Hest = 2*([norm(Xdiff_est(:,1)); norm(Xdiff_est(:,2)); norm(Xdiff_est(:,3)); norm(Xdiff_est(:,4))])/c;
precision = 0.5*c*norm(Ztof - Hest);
while  precision > desired_precision
% updating the Jacobian
    for i=1:size(Xb,2)
        dH(i,:) = -2/c*transpose(Xdiff_est(:,i)./norm(Xdiff_est(:,i)));
    end
% updating the position estimate
Xv_est = Xv_est + pinv(dH'*dH)*dH'*(Ztof - Hest);
% propagating new estimate thrgough the observation model
Xdiff_est = Xb - repmat(Xv_est, 1, size(Xb, 2));
Hest = 2*([norm(Xdiff_est(:,1)); norm(Xdiff_est(:,2)); norm(Xdiff_est(:,3)); norm(Xdiff_est(:,4))])/c;
% updating the precision of the current estimate
precision = 0.5*c*norm(Ztof - Hest); %[m]
end
```