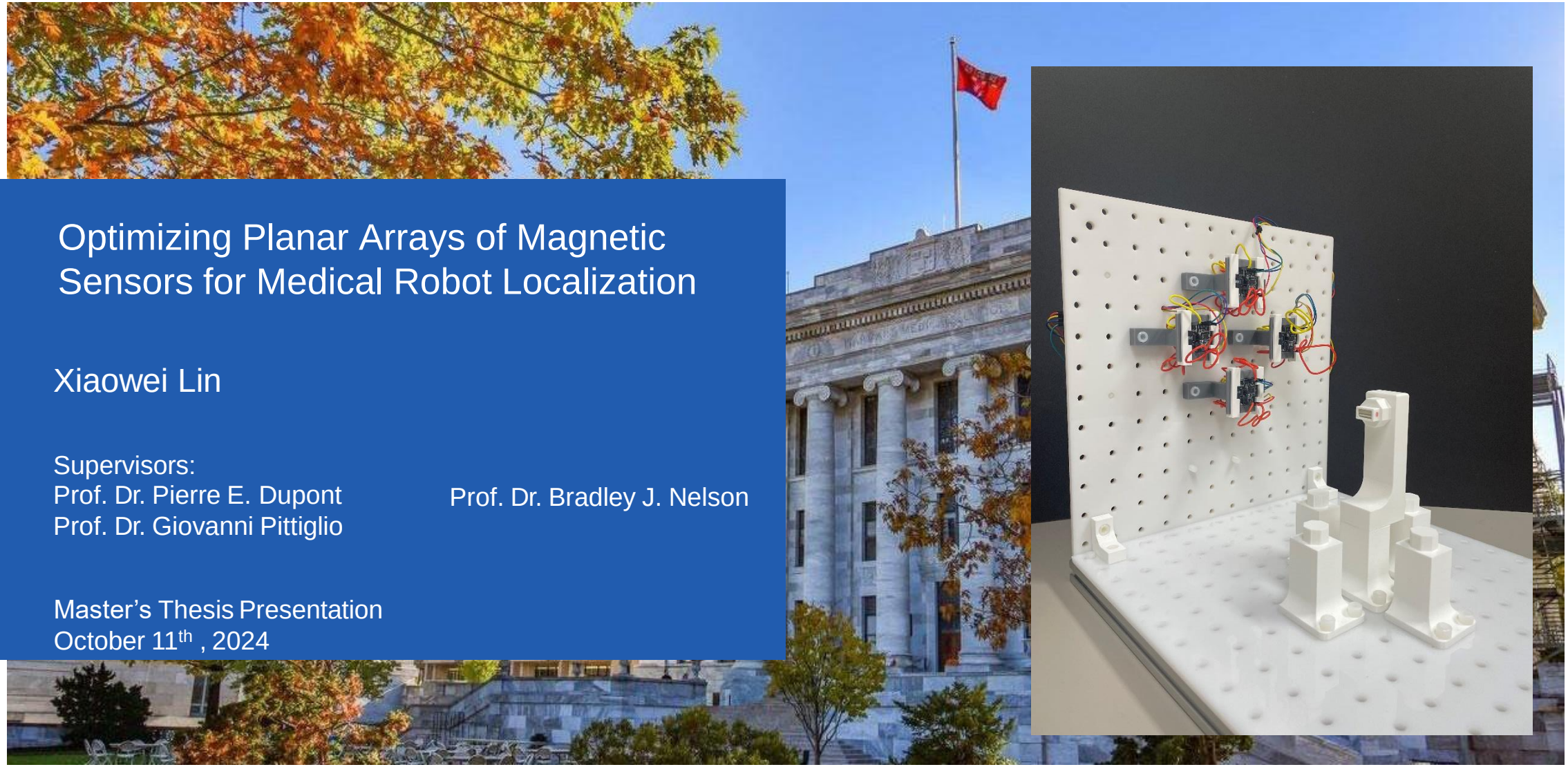
Optimizing Planar Arrays of Magnetic Sensors for Medical Robot Localization

Xiaowei Lin

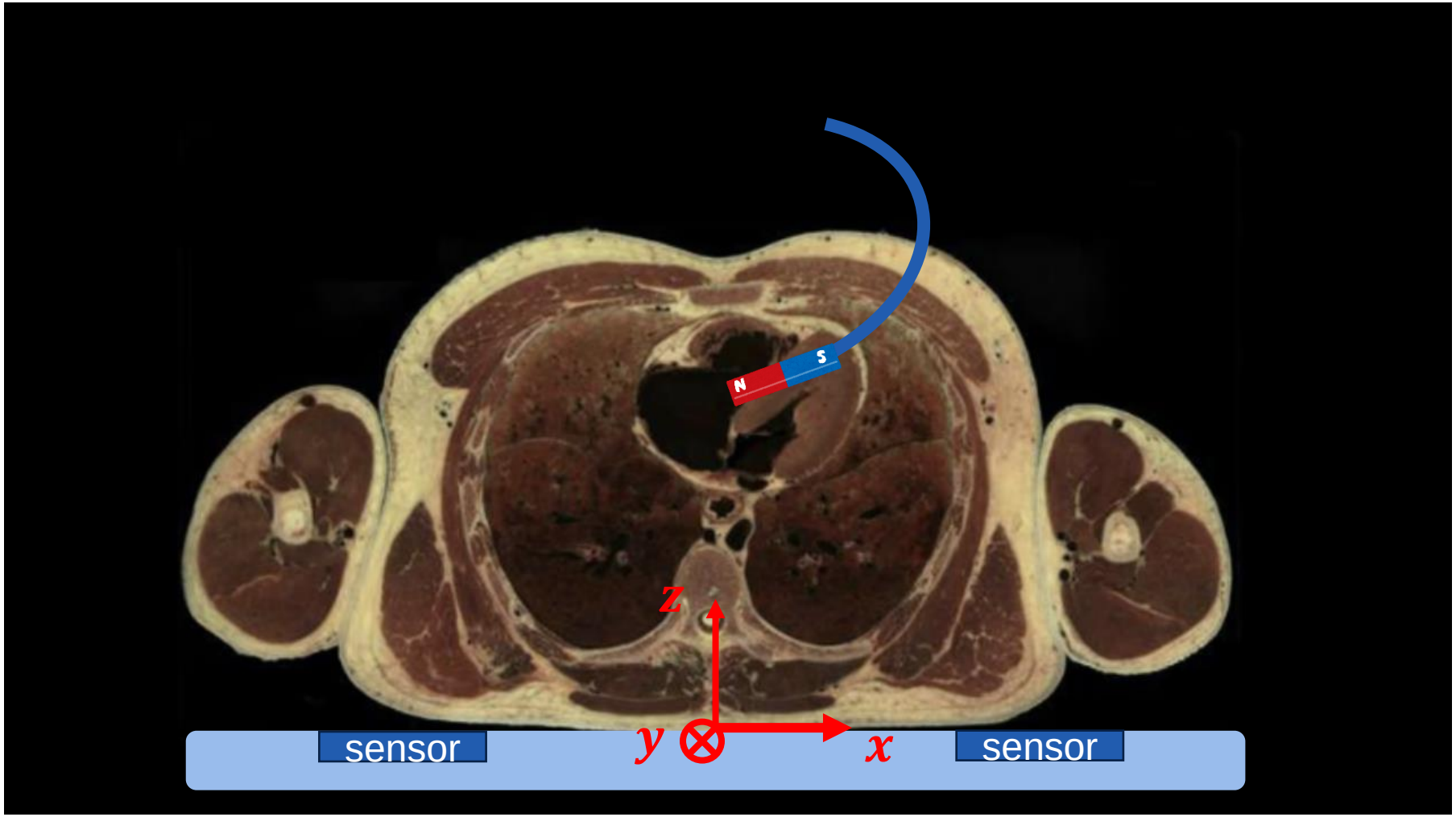
Supervisors:
Prof. Dr. Pierre E. Dupont
Prof. Dr. Giovanni Pittiglio
Prof. Dr. Bradley J. Nelson

Master's Thesis Presentation
October 11th , 2024

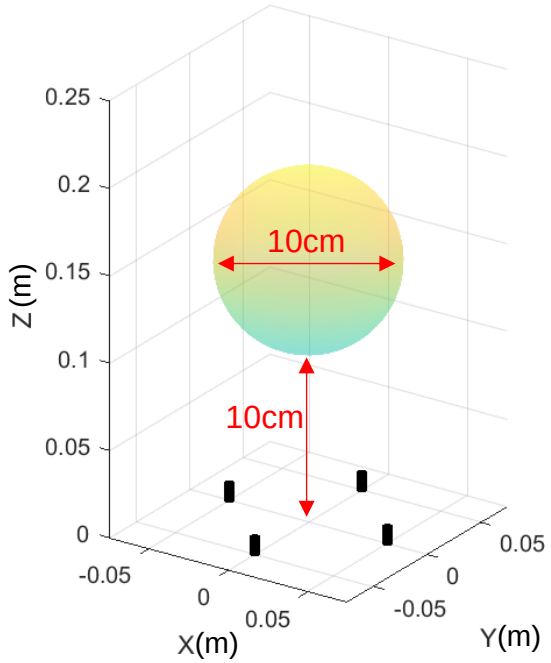


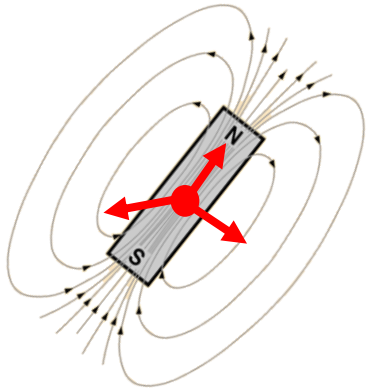
Index

Motivation

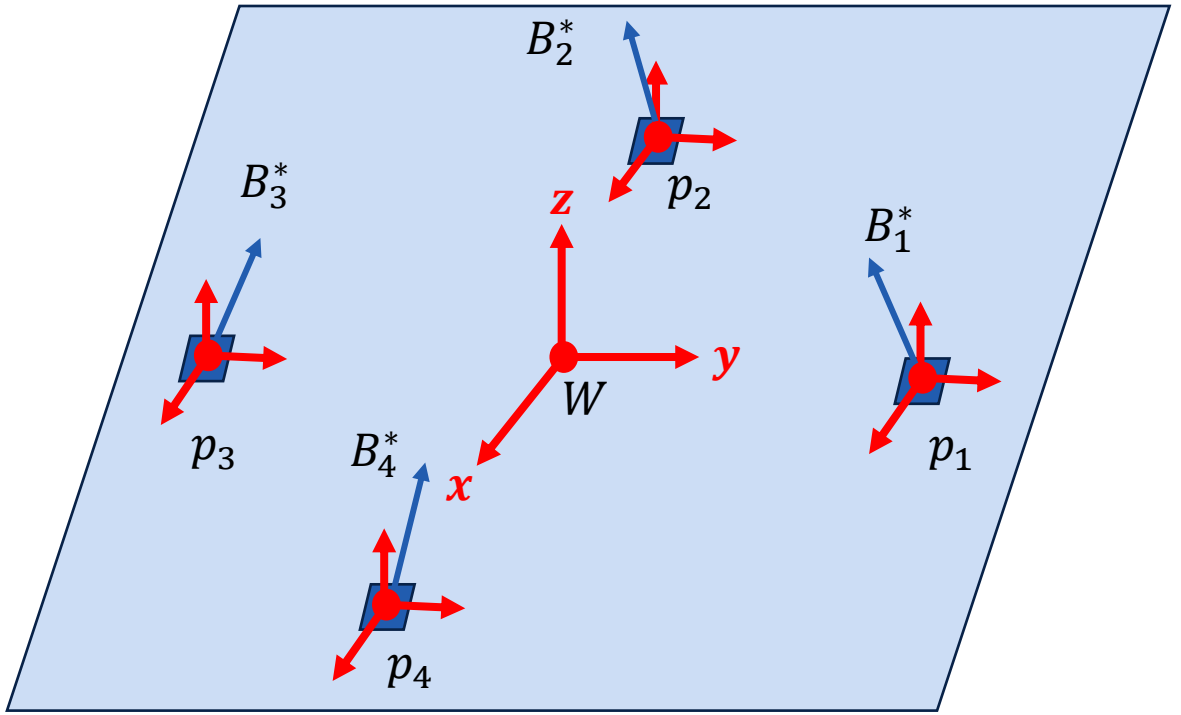


Problem definition



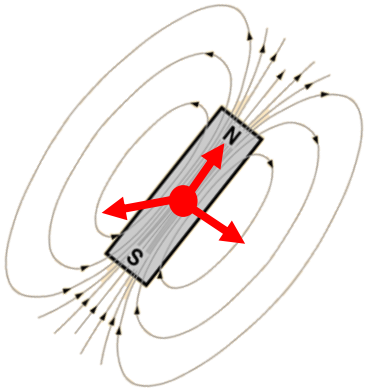


p, R

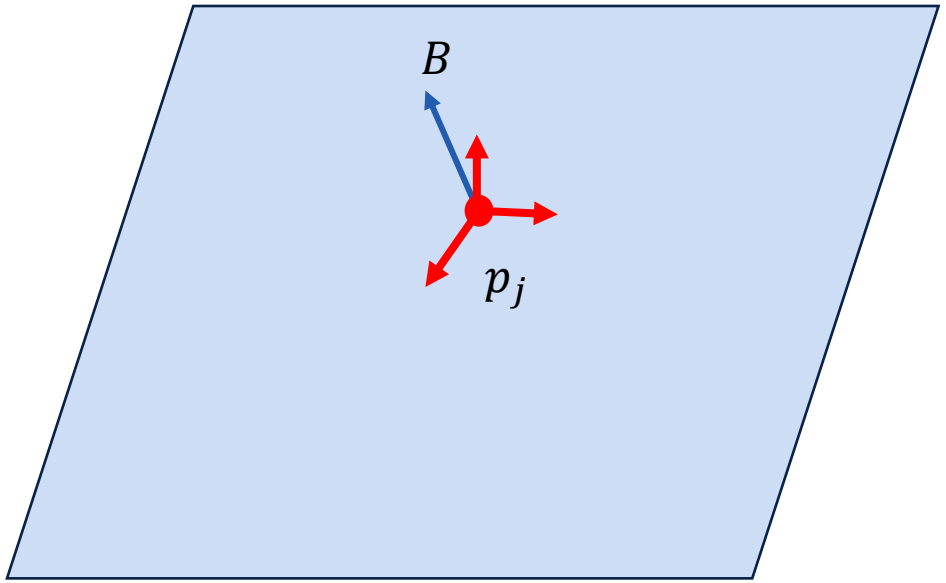


$$B \in \begin{bmatrix} B_1 \\ B_2 \\ B_3 \\ B_4 \end{bmatrix} \Rightarrow p_m, R_m$$

Magnetic dipole model and its linearization



p, R



Magnetic dipole model:

$$B(p,R) = \frac{B_r V}{4\pi|r|^3}(3\hat{r}\hat{r}^T - I)Re_z, r = p_j - p$$

Linearized model:

$$B(p^* + \Delta p, R^* + \Delta R) \approx B(p^*, R^*) + J_p(p^*, R^*)\Delta p + J_R(p^*, R^*)\Delta R$$

Linearize with respect to p:

$$J_p(p,R) = \frac{\partial B(p,R)}{\partial r}\frac{\partial r}{\partial p}$$

$$= -\frac{3B_r V}{4\pi|r|^4}((I - 5\hat{r}\hat{r}^T)\left(\hat{r}^T\widehat{Re_z}\right) + \widehat{Re_z}\hat{r}^T + \hat{r}\widehat{Re_z}^T)$$

Linearize with respect to orientation in $SO(3)$?

Linearize with respect to orientation

Parametrization of the orientation R: Exponential coordinates.

$$R(\omega)\colon\ \omega \in \mathbb{R}^3 \rightarrow\ R \in SO(3)$$

Rodrigues formula:

$$R(\omega) = \exp([\omega]_\times) = I + \frac{\sin|\omega|}{|\omega|}[\omega]_\times + \frac{1-\cos|\omega|}{|\omega|^2}[\omega]_\times^2$$

Problem with the global parametrization:

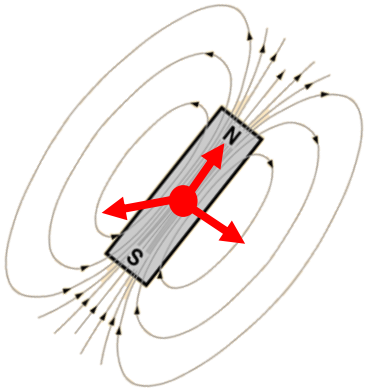
$$\mathbb{R}^3 \ncong SO(3)$$

$$null\left(\frac{\partial B}{\partial \omega}\right) \neq e_z$$

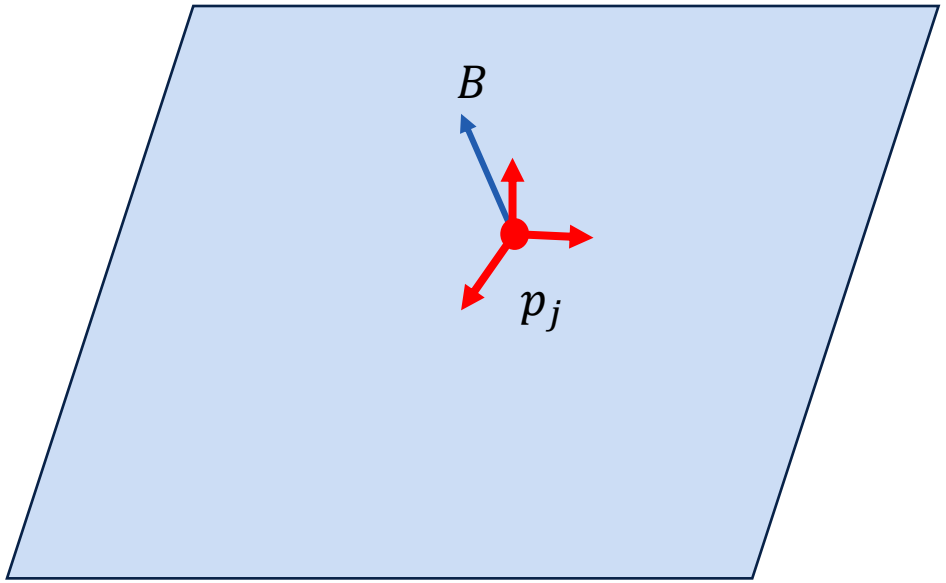
Local parametrization:

$$R(\omega) = \exp([\omega]_\times) \Rightarrow R(\omega) = R^* \exp([\omega]_\times)$$

Magnetic dipole model and its linearization



p, R



Magnetic dipole model:

$$B(p,R) = \frac{B_r V}{4\pi|r|^3}(3\hat{r}\hat{r}^T - I)Re_z, r = p_j - p$$

Linearized model:

$$B(p^* + \Delta p, R^* + \Delta R) \approx B(p^*, R^*) + J_p(p^*, R^*)\Delta p + J_R(p^*, R^*)\Delta R$$

Linearize with respect to p:

$$J_p(p,R) = \frac{\partial B(p,R)}{\partial r}\frac{\partial r}{\partial p}$$

$$= -\frac{3B_r V}{4\pi|r|^4}((I - 5\hat{r}\hat{r}^T)\left(\hat{r}^T\widehat{Re_z}\right) + \widehat{Re_z}\hat{r}^T + \hat{r}\widehat{Re_z}^T)$$

Linearize with respect to orientation:

$$J_\omega\left(p, R(\omega)\right)\Big|_{\omega=0} = \frac{B_r V}{4\pi|r|^3}(3\hat{r}\hat{r}^T - I)\frac{\partial R^* \exp([\omega]_\times)\, e_z}{\partial \omega}\Bigg|_{\omega=0}$$

$$= \frac{B_r V}{4\pi|r|^3}(3\hat{r}\hat{r}^T - I)R^*[e_z]_\times$$

Linearized magnetic dipole model - Summary

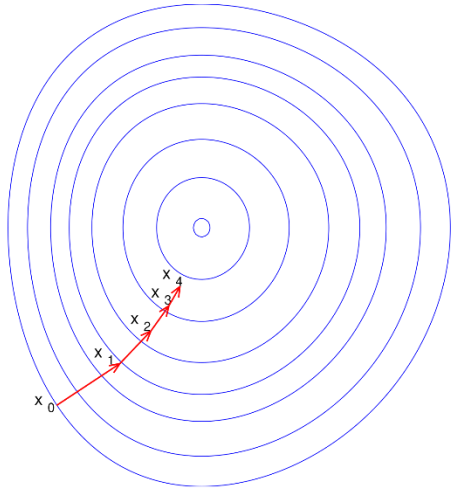
$$B(p^* + \delta p, R^* + \delta R) = B(p^*, R^*) + [J_p \quad J_\omega] \begin{bmatrix} \delta p \\ \delta \omega \end{bmatrix}$$
$$= B(p^*, R^*) + J \begin{bmatrix} \delta p \\ \delta \omega \end{bmatrix}$$

where $J_p(p^*, R^*) \in \mathbb{R}^{3\times3}$ and $J_\omega(p^*, R^*) \in \mathbb{R}^{3\times3}$

In the case of m sensors, J_p and J_ω are the stacked version of the Jacobians correspond to one sensor. As a results, $J_p \in \mathbb{R}^{3m\times3}$, $J_\omega \in \mathbb{R}^{3m\times3}$, and $J = [J_p \quad J_\omega] \in \mathbb{R}^{6m\times3}$:

$$B(p^* + \delta p, R^* + \delta R) = B(p^*, R^*) + J \begin{bmatrix} \delta p \\ \delta \omega \end{bmatrix}$$

Gauss-Newton on manifold algorithm



Given the measured magnetic field $\bar{B}$:

$$\hat{p}, \hat{\omega} = \min_{p,\omega} \|\bar{B} - B(p, \omega)\|_2^2$$
$$r(p, \omega) = \bar{B} - B(p, \omega)$$

In every step k of the algorithm, rewrite the residual function:

$$r(p, \omega) = \bar{B} - B\big(p, \exp([\omega]_\times)\big) \rightarrow \bar{B} - B(p, R_k \exp([\omega]_\times)$$

Linearize the residual function at $p = p_k$ and $\omega = 0$ using the derived Jacobians:

$$r(p, R_k \exp([\omega]_\times) \approx r(p_k, R_k) - [J_p \quad J_\omega] \begin{bmatrix} \Delta \mathrm{p} \\ \Delta\omega \end{bmatrix}$$

Solve the linearized least square problem for the update step:

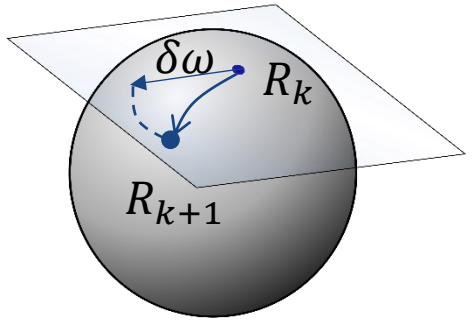
$$\begin{bmatrix} \delta p \\ \delta\omega \end{bmatrix} = [J_p \quad J_\omega]^\dagger r(p_k, R_k)$$

The position guess is updated normally:

$$p_{k+1} = p_k + \delta p$$

The orientation guess is updated in the tangent space of R_k and then map it back to $SO(3)$ using the exponential map:

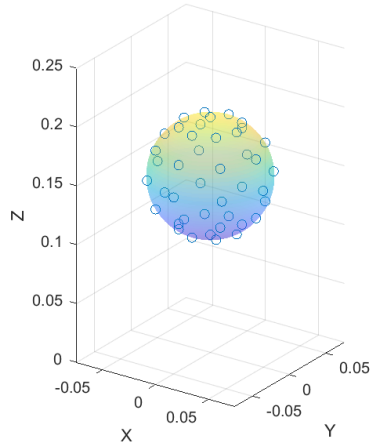
$$R_{k+1} = R_k \exp([0 + \delta\omega]_\times)$$



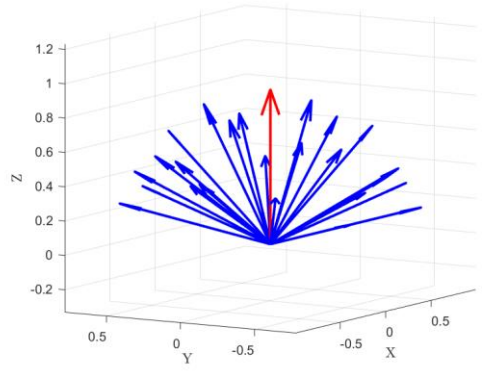
Comparison with MATLAB's algorithm

One can also use over-parametrization, e.g. quaternion q. But it requires additional constraint: $\|q\| = 1$. The minimization problem is turned into a constrained optimization problem and can be solved using the method of Lagrange multiplier and (quasi-)Newton algorithm.

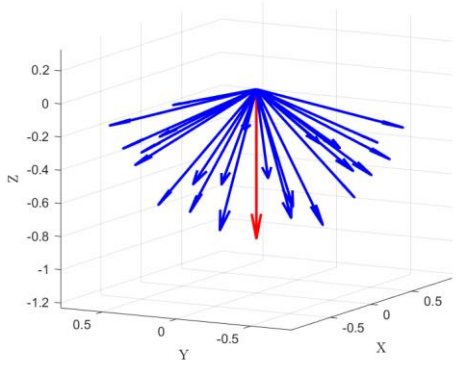
Test conditions: 4 sensors put on a circle of radius=25cm. Five different radii of sphere are sampled within the spherical workspace. On each surface, 42 positions are selected. At each position, 25 orientations in a cone are selected. 16 cones are tested. In total 84000 test configurations. The selected orientations cannot deviate from the initial guess more than $60°$. The initial guess for the position is the center of the sphere.



42 positions on one sphere radius



25 orientations selected at each position



Different cone orientations

Comparison with MATLAB's algorithm

	MATLAB's algorithm	Gauss-Newton on manifold
Convergence rate	100%	100%
Average time for solving 1 configuration	$34ms$	$0.32ms$

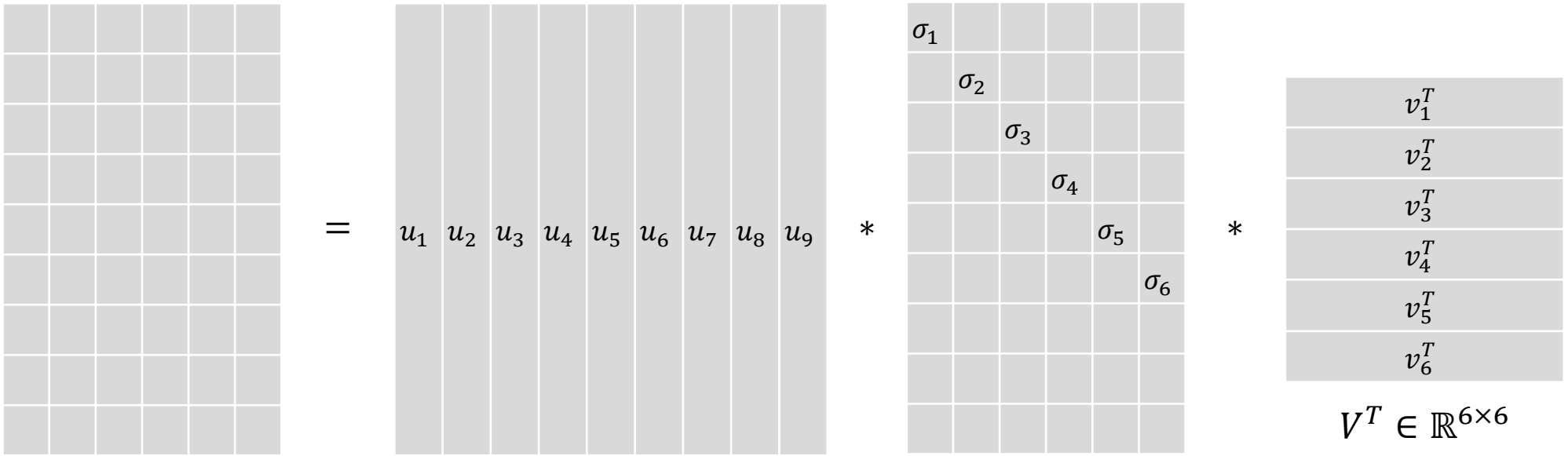
The Gauss-Newton on manifold algorithm is ~100 times faster than the MATLAB's algorithm.

Optimization criteria - Minimum singular value σ_{min}

The linear model at certain configuration p^*, R^*: $\Delta B = J\Delta x$

Singular value decomposition of $J \in \mathbb{R}^{3m\times 6}$:

$$J = U\Sigma V^T$$



$J \in \mathbb{R}^{9\times 6}$
3 sensors

$U \in \mathbb{R}^{9\times 9}$

$\Sigma \in \mathbb{R}^{9\times 6}$

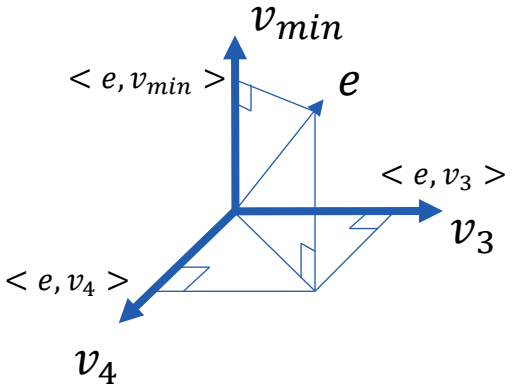
$\sigma_1 > \sigma_2 > \cdots \sigma_5 > \sigma_6 = 0$

$V^T \in \mathbb{R}^{6\times 6}$

$$\sigma_{min} \triangleq \sigma_5$$
$$v_{min} \triangleq v_5$$
$$USV^T * v_5 = \sigma_5 v_5$$

Optimization criteria - Minimum singular value σ_{min}

Hypothesis 1: For the Jacobian of a certain configuration, v_{min} represents the motion direction that‘s hardest to distinguish from noise. When the algorithm tries to solve the magnet‘s configuration, this direction is the hardest to resolve from noise. As a result, it contributes the most to the error.



Hypothesis 2: If σ_{min} is larger, the same amount of motion in the direction of v_{min} generates larger $||\Delta B||$. As a result, when the algorithm tries to solve for the magnet‘s configuration, the error is smaller.

Criterion 1: Maximize $mean_{x \in X} \sigma_{min}(x)$

Optimization criteria – Condition number κ

Condition number: $\kappa = \frac{\sigma_{max}}{\sigma_{min}} \in [1, +\infty]$

Every step in the Gauss-Newton algorithm requires taking the pseudo-inverse of the Jacobian:

$$\begin{bmatrix} \Delta p \\ \Delta \omega \end{bmatrix} = \Delta x = J(p_k, R_k)^{\dagger} r$$

There might be error in evaluating J and r, δJ and δr, resulting in the error computing Δx, δx:

$$\Delta x + \delta x = (J + \delta J)^{\dagger}(r + \delta r)$$

δJ and δr can come from numerical rounding, noise, etc.

Wedin, 1973, provides an upper bound on δx using the condition number κ under some assumptions:

- J and J + δJ have equal rank
- $\|\delta J\|_2 < \varepsilon \|A\|_2$
- $\varepsilon\kappa < 1$

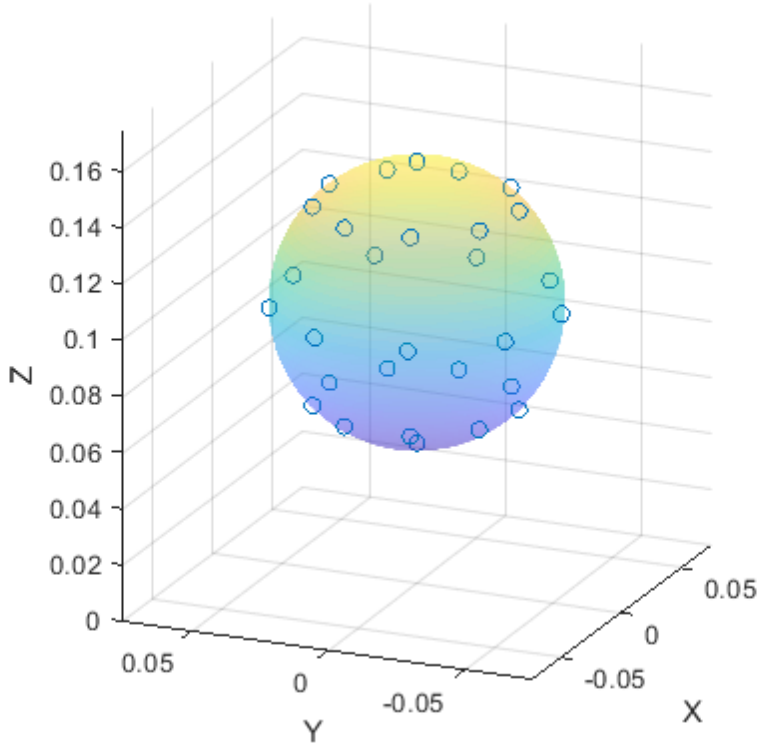
$$\frac{||\delta x||_2}{||\Delta x||_2} \leq \frac{\kappa}{1 - \varepsilon\kappa}\left(\varepsilon + \frac{||\delta r||_2}{||J||_2||\Delta x||_2} + \frac{\varepsilon\kappa||r_0||_2}{||J||_2||\Delta x||_2}\right) + \frac{\varepsilon||(JJ^T)^{\dagger}b||_2||J||_2}{||\Delta x||_2}$$

Optimization criteria – Condition number κ

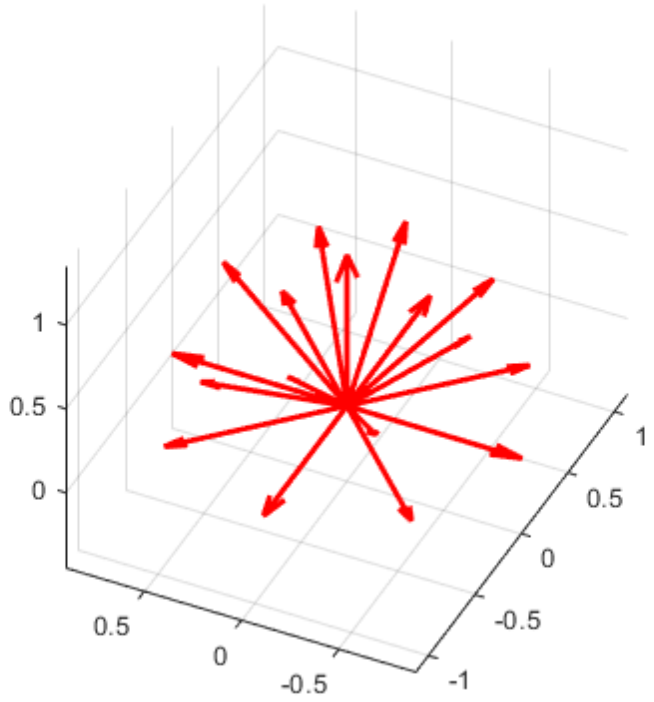
Hypothesis 3: The smaller κ for all the configuration, the less error in calculating the update step. The faster convergence for the algorithm.

Criterion 2: Maximize $\min_{x \in \mathcal{X}} \frac{1}{\kappa}(x)$

Optimized configuration



29 positions selected on the surface of the spherical workspace



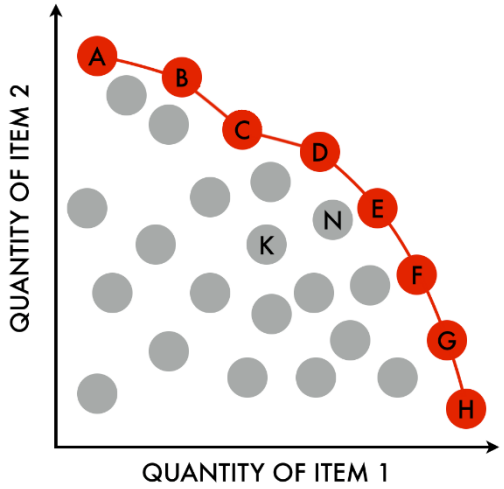
17 orientations selected on each position

493 configurations are considered for optimization

Optimization using the multi-objective genetic algorithm

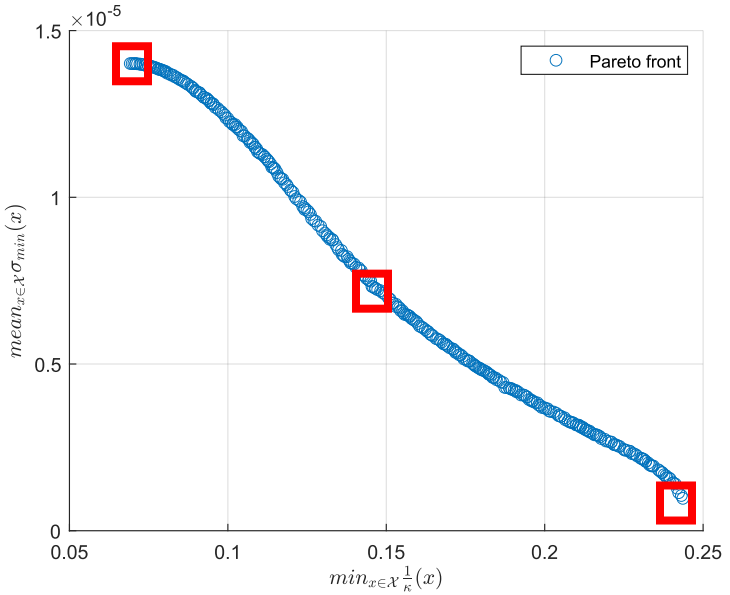
Genetic algorithm: An algorithm tries to find the optimum of the objective(s) based on the natural selection process that mimics biological evolution.

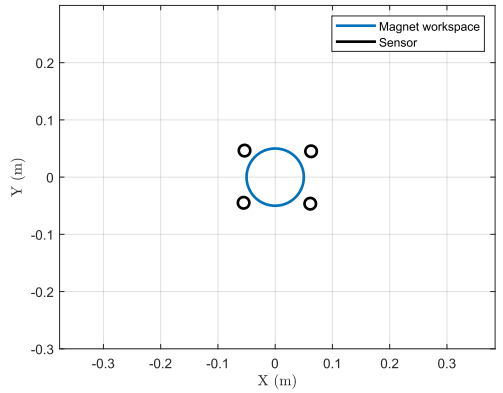
The Pareto front for multi-objective optimization: The candidates on the Pareto front is strictly dominating other candidates for at least one criteria.

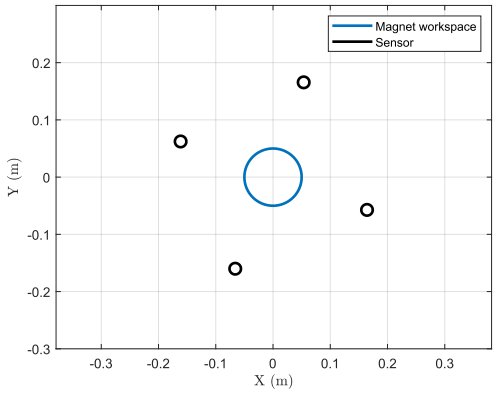


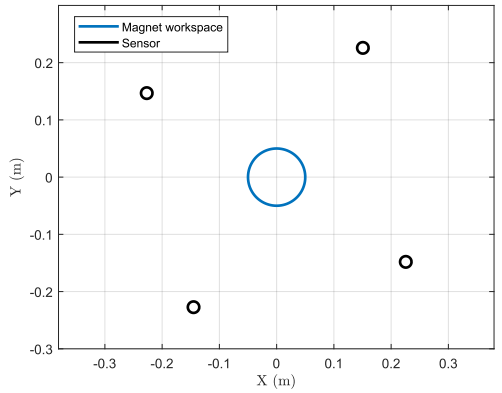
The optimization is run for different number of sensors $m \in \{4,\ 5,\ 9,\ 16\}$.

Optimization results – 4 sensors

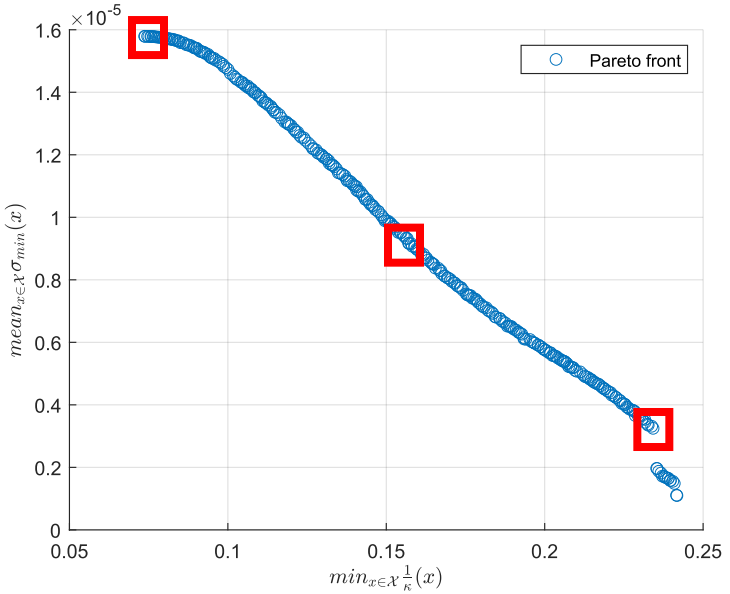


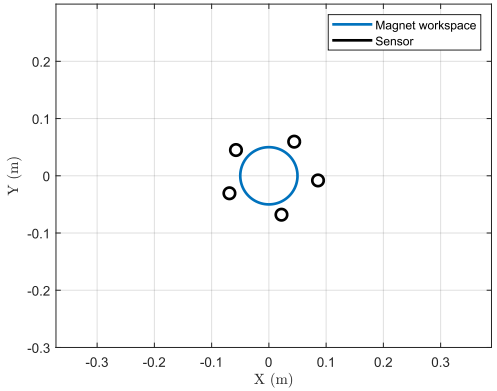


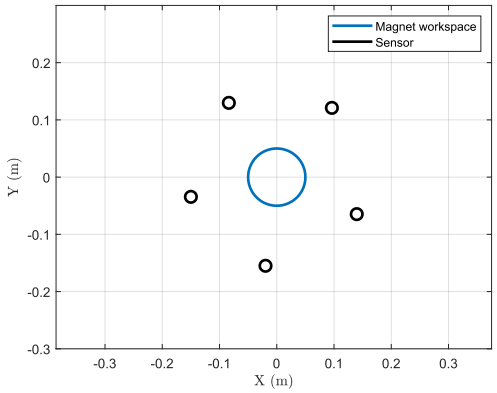


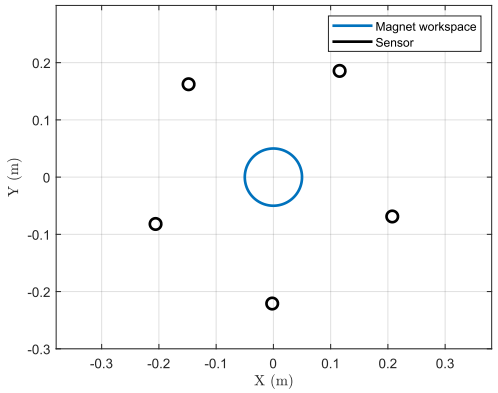


Optimization results – 5 sensors

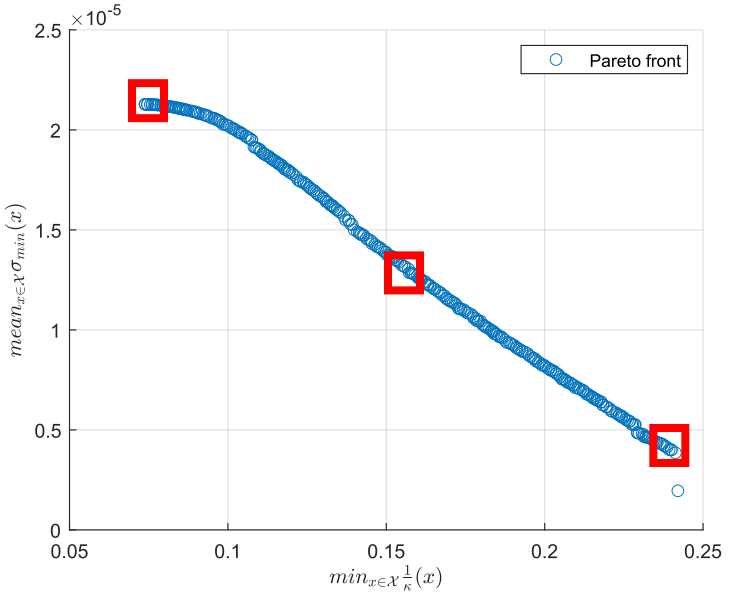


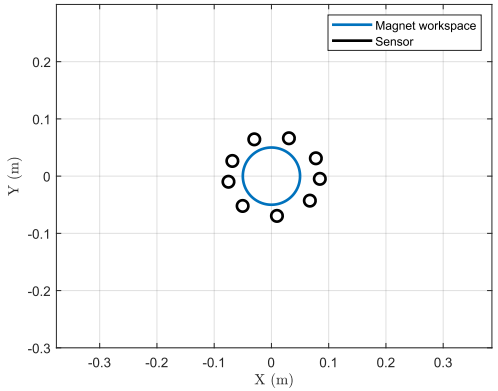


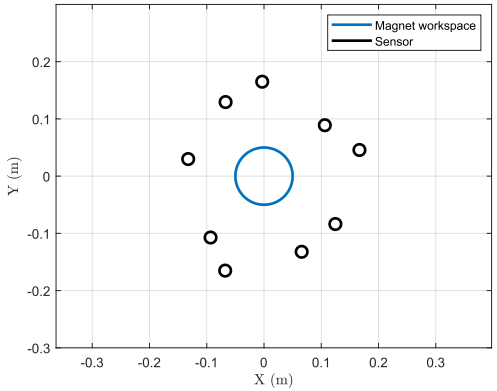


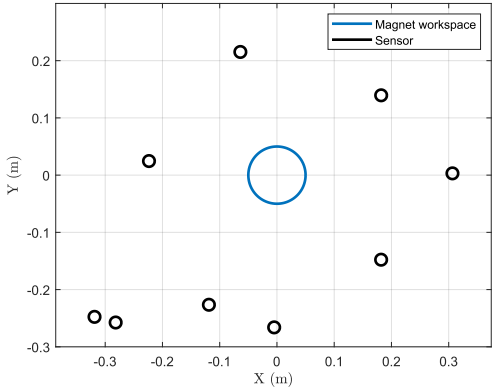


Optimization results – 9 sensors

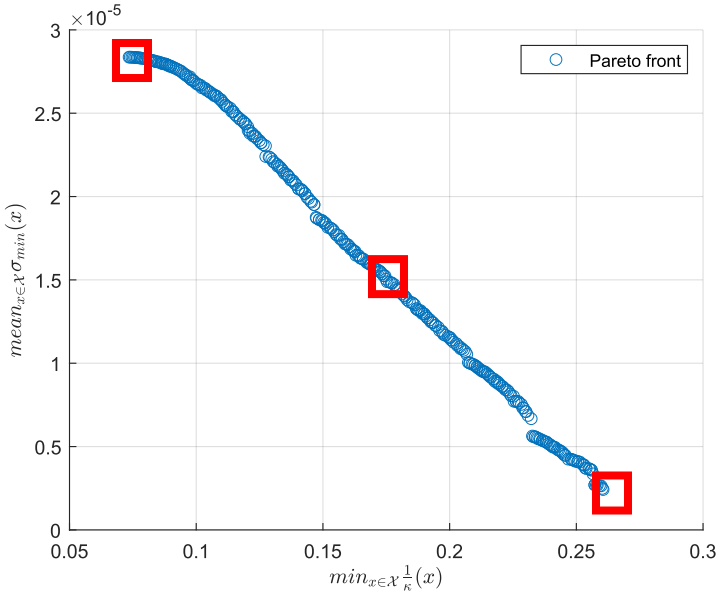


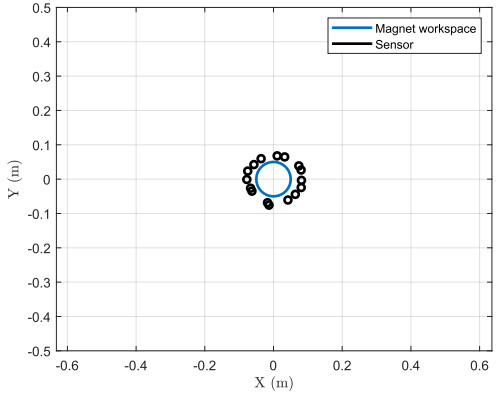


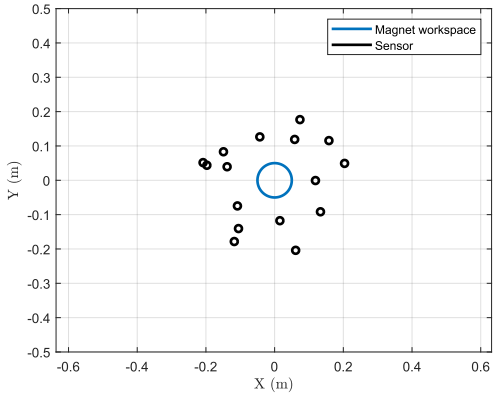


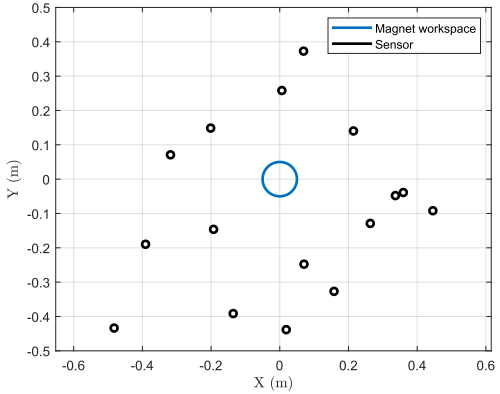


Optimization results – 16 sensors

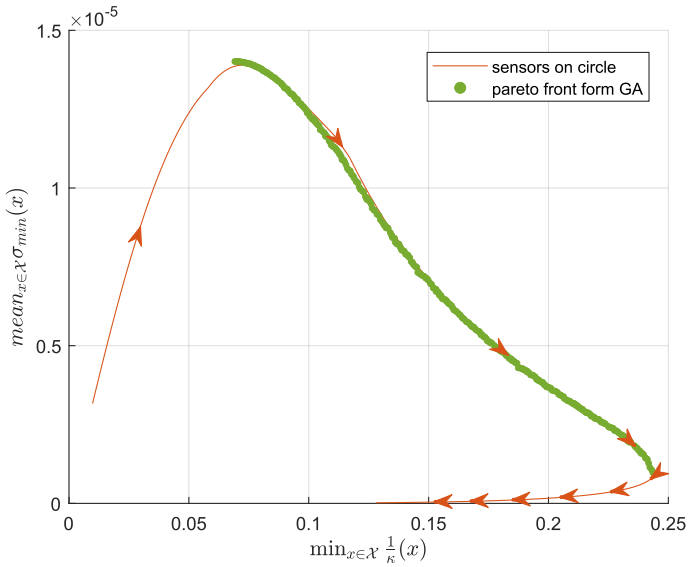




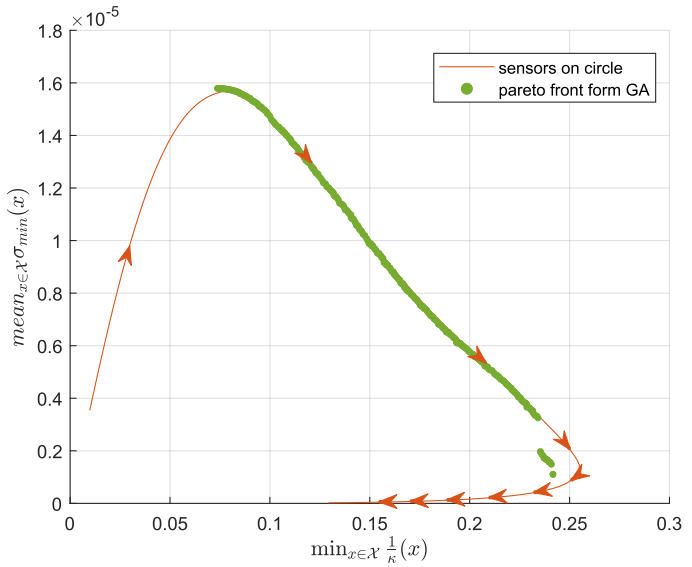




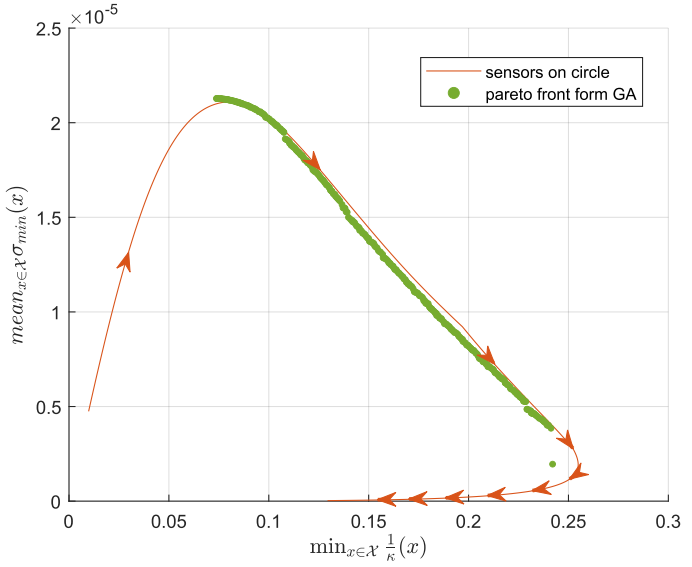
Circular configuration as the optimum



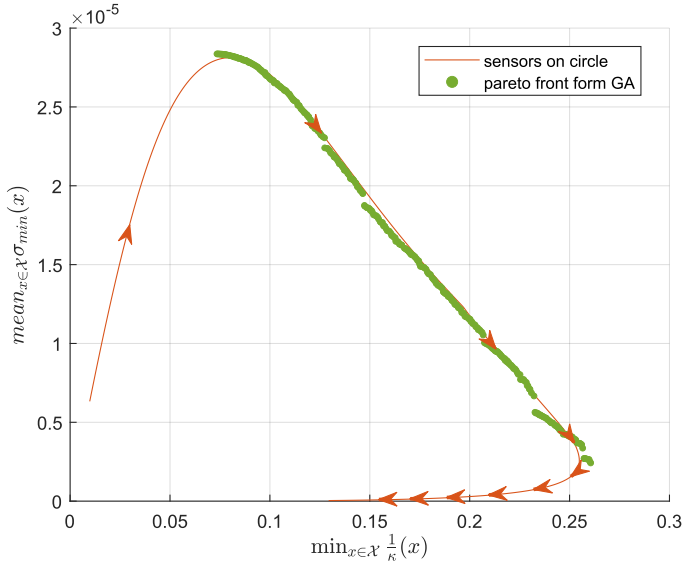
4 sensors



5 sensors



9 sensors



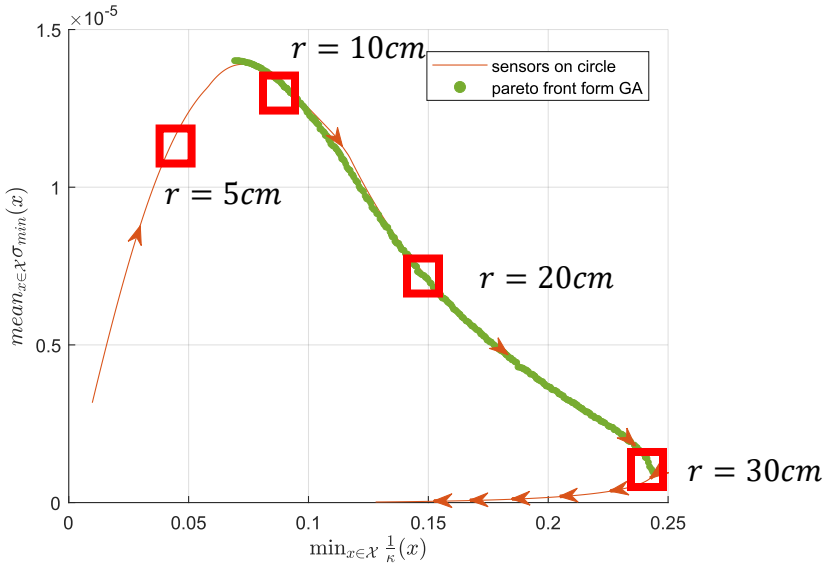
16 sensors

Summary

1. Defined the optimization criteria
   - $mean_{x \in \mathcal{X}} \sigma_{min}(x)$: accuracy
   - $\min_{x \in \mathcal{X}} \frac{1}{\kappa}(x)$: speed
2. Sensor optimization: Circular configuration is the one that optimizes both criteria.

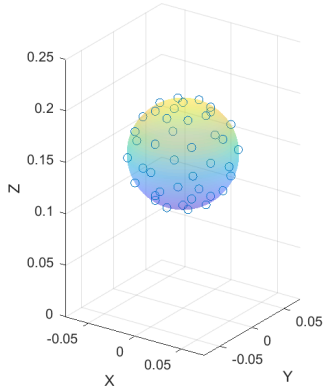
Simulated convergence test with noise for σ_{min} – settings

1. 4 sensors put on a circle of different radii, $r = \{5, 10, 20, 30\}cm$.

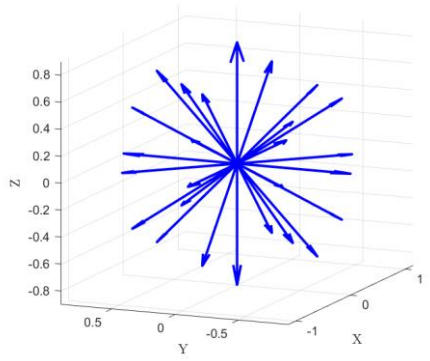


Simulated convergence test with noise for σ_{min} – settings

4. 6760 configurations are considered: 210 positions, 32 orientations at each positions.



42 positions on one sphere radius, 5 radii are considered

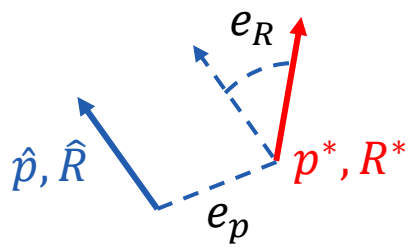


32 orientations are sampled at one position

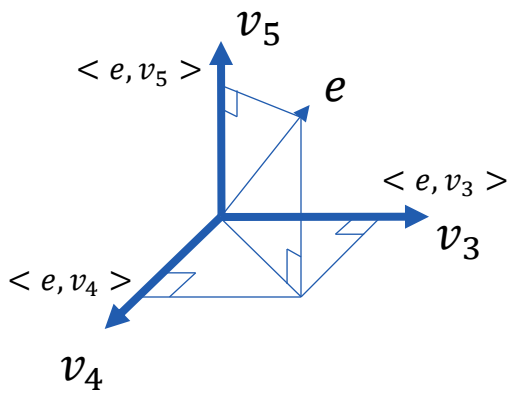
5. Simulated magnetic field measurement is computed from the dipole model.
6. Different levels of Gaussian noise are added to the measurements. $\sigma^2 = \{0.05, 0.1, 0.5\}\mu T$.
7. To ensure convergence and only compare the error after convergence, the initial guess of the algorithm is the ground truth configuration, which is close enough to the solution.

Simulated convergence test with noise for σ_{min} – settings

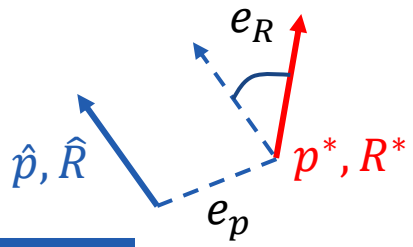
6. The position error e_p is the discrepency between the ground truth postion and the solution given by the algorithm, measured in mm. The orientation is the discrepency between the ground truth z-axis and the z-axis after convergence, measured in deg.



7. The error vector is projected on the 5 singular vectors to examine the hypothesis.



Simulated convergence test with noise for σ_{min} – results

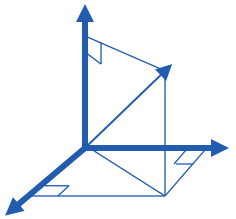


Noise:0.05μT	$r = 5cm$	$r = 10cm$	$r = 20cm$	$r = 30cm$
$mean(\sigma_{min})$	$1.03e-5$	$1.12e-5$	$4.07e-6$	$1.65e-6$
$mean(e_p)\ (mm)$	0.63	0.57	1.40	4.17
$mean(e_R)\ (deg)$	0.37	0.31	0.57	1.28

Noise:0.1μT	$r = 5cm$	$r = 10cm$	$r = 20cm$	$r = 30cm$
$mean(\sigma_{min})$	$1.03e-5$	$1.12e-5$	$4.07e-6$	$1.65e-6$
$mean(e_p)\ (mm)$	1.27	1.14	2.79	8.63
$mean(e_R)\ (deg)$	0.74	0.62	1.14	2.55

Noise:0.5μT	$r = 5cm$	$r = 10cm$	$r = 20cm$	$r = 30cm$
$mean(\sigma_{min})$	$1.03e-5$	$1.12e-5$	$4.07e-6$	$1.65e-6$
$mean(e_p)\ (mm)$	6.37	5.71	14.20	46.11
$mean(e_R)\ (deg)$	3.68	3.12	5.80	13.8

Simulated convergence test with noise for σ_{min} – results



Noise:0.05μT $r = 0.05m$	$\frac{\lvert < e, v_1 > \rvert}{\lvert e \rvert}$	$\frac{\lvert < e, v_2 > \rvert}{\lvert e \rvert}$	$\frac{\lvert < e, v_3 > \rvert}{\lvert e \rvert}$	$\frac{\lvert < e, v_4 > \rvert}{\lvert e \rvert}$	$\frac{\lvert < e, v_{min} > \rvert}{\lvert e \rvert}$
Ratio	18%	14%	30%	50%	60%

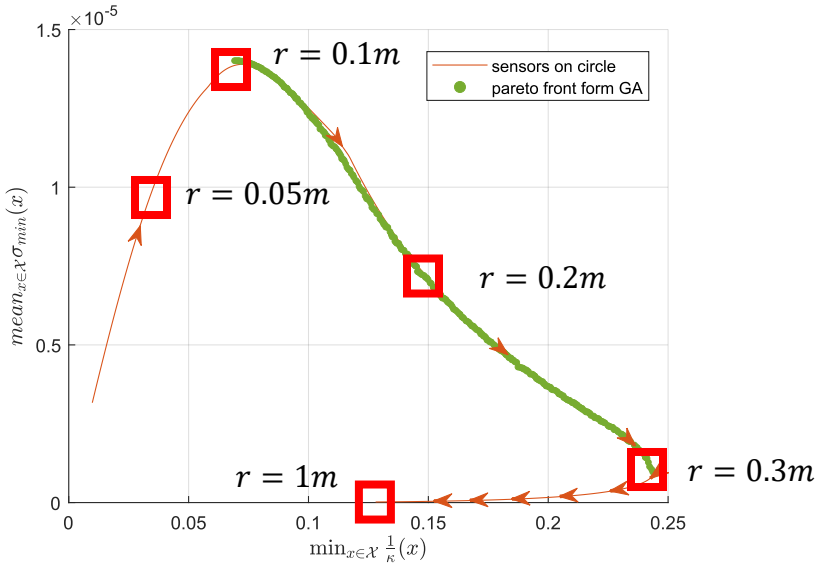
Noise:0.05μT $r = 0.10m$	$\frac{\lvert < e, v_1 > \rvert}{\lvert e \rvert}$	$\frac{\lvert < e, v_2 > \rvert}{\lvert e \rvert}$	$\frac{\lvert < e, v_3 > \rvert}{\lvert e \rvert}$	$\frac{\lvert < e, v_4 > \rvert}{\lvert e \rvert}$	$\frac{\lvert < e, v_{min} > \rvert}{\lvert e \rvert}$
Ratio	22%	19%	35%	48%	53%

Noise:0.05μT $r = 0.20m$	$\frac{\lvert < e, v_1 > \rvert}{\lvert e \rvert}$	$\frac{\lvert < e, v_2 > \rvert}{\lvert e \rvert}$	$\frac{\lvert < e, v_3 > \rvert}{\lvert e \rvert}$	$\frac{\lvert < e, v_4 > \rvert}{\lvert e \rvert}$	$\frac{\lvert < e, v_{min} > \rvert}{\lvert e \rvert}$
Ratio	30%	35%	34%	43%	45%

Noise:0.05μT $r = 0.30m$	$\frac{\lvert < e, v_1 > \rvert}{\lvert e \rvert}$	$\frac{\lvert < e, v_2 > \rvert}{\lvert e \rvert}$	$\frac{\lvert < e, v_3 > \rvert}{\lvert e \rvert}$	$\frac{\lvert < e, v_4 > \rvert}{\lvert e \rvert}$	$\frac{\lvert < e, v_{min} > \rvert}{\lvert e \rvert}$
Ratio	36%	38%	35%	38%	42%

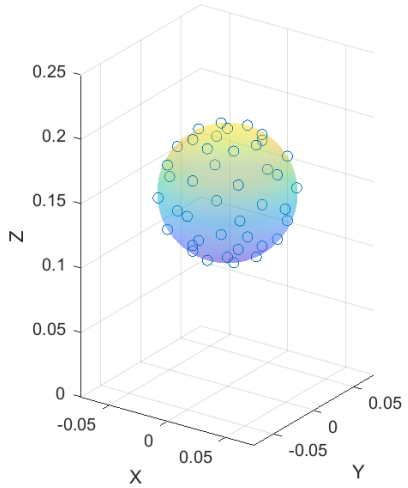
Noise-free simulated convergence test for $\frac{1}{\kappa}$ – settings

1. 4 sensors put on a circle of different radii, $r = \{0.05, 0.1, 0.2, 0.3, 1\}m$.

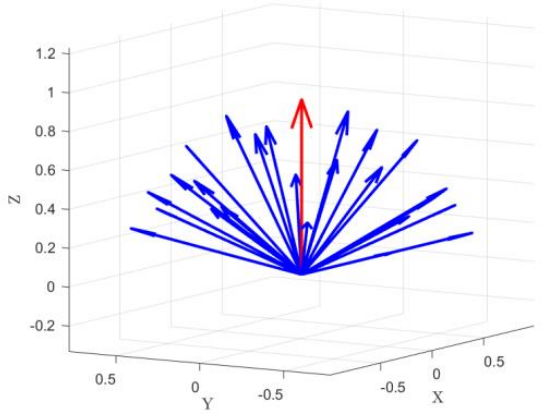


Noise-free simulated convergence test for $\frac{1}{\kappa}$ – settings

3. 5250 configurations are considered: 210 positions, 25 orientations at each positions.



42 positions on one sphere radius, 5 radii are considered



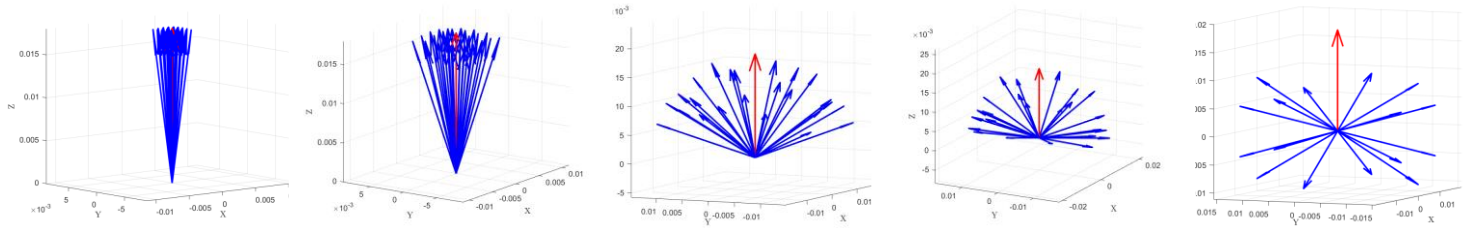
25 orientations are sampled at one position

4. Simulated magnetic field measurement is computed from the dipole model. No noise is added to the measurement.
5. The initial guess for the position is the center of the sphere; The initial guess for the orientation is the center vector of the cone.
6. We first examine if the convergence range is affected by $\min_{x\in\mathcal{X}} \frac{1}{\kappa}(x)$. Then the cone size is fixed.

   The convergence speed is tested to examine if the convergence speed is affected by $\min_{x\in\mathcal{X}} \frac{1}{\kappa}(x)$.

Noise-free simulated convergence test for $\frac{1}{\kappa}$ – results

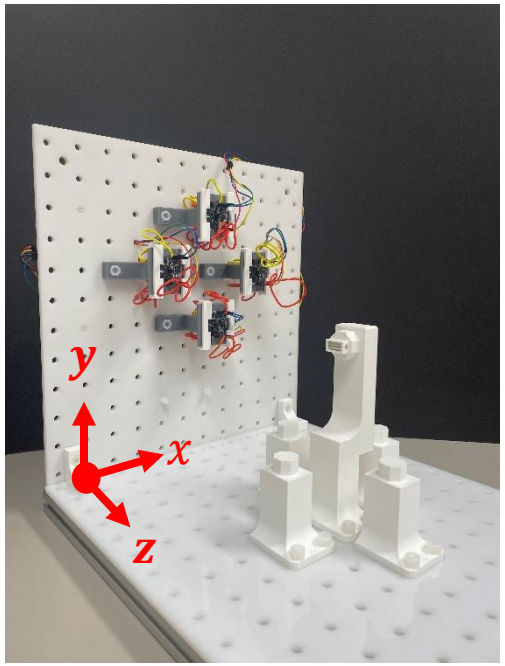
	$r = 0.05m$	$r = 0.1m$	$r = 0.2m$	$r = 0.3m$	$r = 1m$
$\min \frac{1}{\kappa}$	0.05	0.10	0.18	0.23	0.13
Cone size	5°	12.5°	55°	75°	120°

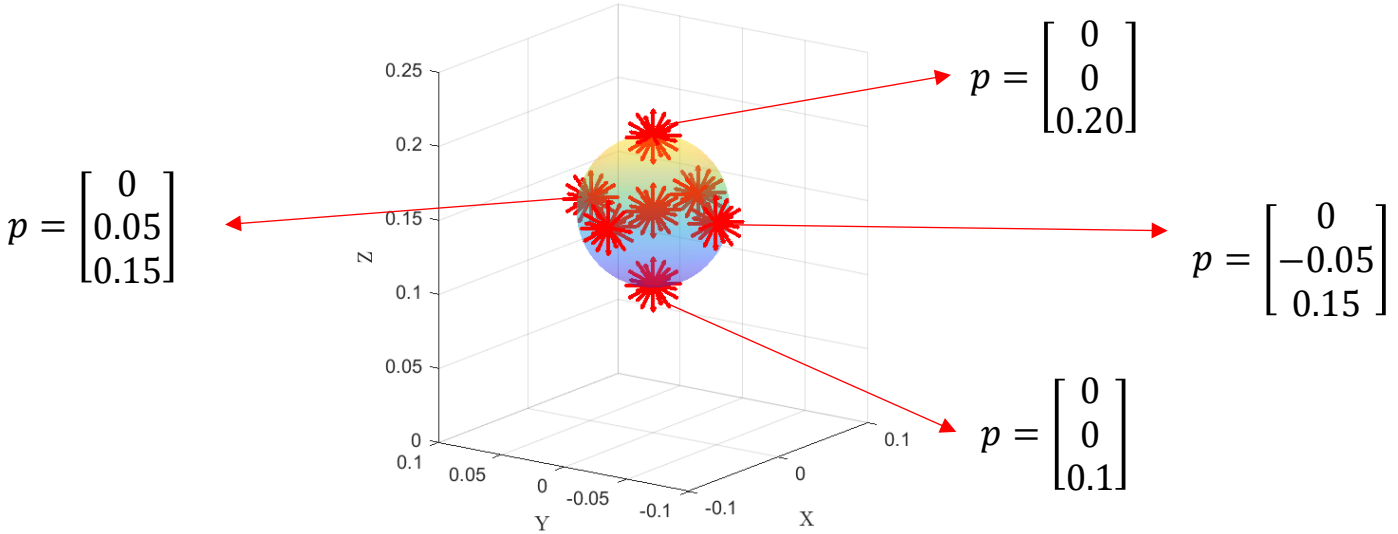


Cone size $= 12.5°$	$r = 0.1m$	$r = 0.2m$	$r = 0.3m$	$r = 1m$
$\min \frac{1}{\kappa}$	0.10	0.18	0.23	0.13
Average time for solving one test set of 5250 configurations	1.72s	1.49s	1.37s	1.22s

Physical experiments for σ_{min}– settings

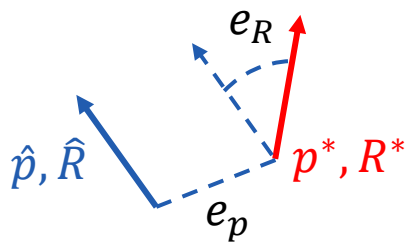
1. 4 PNI RM3100 sensors are put on a circle of two radii, $r = \{5, 10\}cm$.
2. 224 configuratioins: 7 positions are sampled in the spherical workspace, 32 orientations are sampled at each position. The magnetic field is recorded for each configuration.



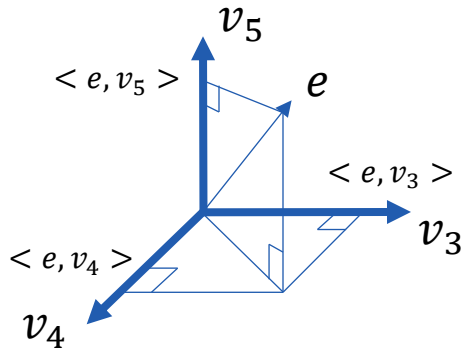


Physical experiments for σ_{min} – settings

4. The sensor positions are calibrated using the recorded field.
5. For each test configuration, the algorithm starts at the ground truth configuration to ensure convergence.
6. The position error e_p is the discrepency between the ground truth postion and the solution given by the algorithm, measured in mm. The orientation is the discrepency between the ground truth z-axis and the z-axis after convergence, measured in deg.



7. The error vector is projected on the 5 singular vectors to examine the hypothesis.

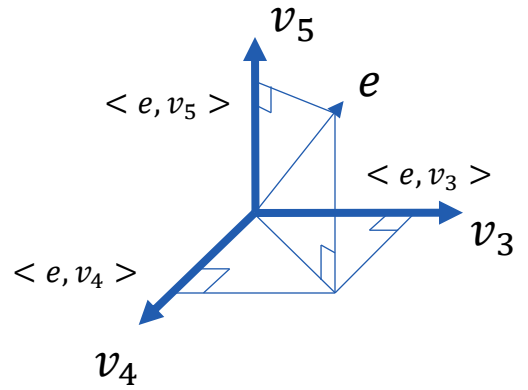


Physical experiments for σ_{min} – results

$r = 0.05m$	$mean(\sigma_{min})$	$mean(e_p)$ (mm)	$mean(e_R)$ (deg)
Overall	$1.25e-5$	3.77	2.62
$p = \begin{bmatrix} 0 \\ 0 \\ 0.1 \end{bmatrix}$	$4.43e-5$	1.90	1.39
$p = \begin{bmatrix} 0 \\ 0 \\ 0.15 \end{bmatrix}$	$9.44e-6$	2.91	2.06
$p = \begin{bmatrix} 0 \\ 0 \\ 0.20 \end{bmatrix}$	$2.78e-6$	5.61	3.23
$p = \begin{bmatrix} 0.05 \\ 0 \\ 0.15 \end{bmatrix}$	$7.49e-6$	4.32	2.90
$p = \begin{bmatrix} 0 \\ 0.05 \\ 0.15 \end{bmatrix}$	$7.60e-6$	4.49	3.45
$p = \begin{bmatrix} -0.05 \\ 0 \\ 0.15 \end{bmatrix}$	$7.95e-6$	3.50	2.66
$p = \begin{bmatrix} 0 \\ -0.05 \\ 0.15 \end{bmatrix}$	$7.61e-6$	3.70	2.69

$r = 0.10m$	$mean(\sigma_{min})$	$mean(e_p)$ (mm)	$mean(e_R)$ (deg)
Overall	$1.18e-5$	2.74	1.68
$p = \begin{bmatrix} 0 \\ 0 \\ 0.1 \end{bmatrix}$	$3.09e-5$	1.27	1.33
$p = \begin{bmatrix} 0 \\ 0 \\ 0.15 \end{bmatrix}$	$1.05e-5$	2.01	1.47
$p = \begin{bmatrix} 0 \\ 0 \\ 0.20 \end{bmatrix}$	$3.79e-6$	4.22	2.57
$p = \begin{bmatrix} 0.05 \\ 0 \\ 0.15 \end{bmatrix}$	$9.00e-6$	2.66	1.55
$p = \begin{bmatrix} 0 \\ 0.05 \\ 0.15 \end{bmatrix}$	$9.35e-6$	3.44	1.72
$p = \begin{bmatrix} -0.05 \\ 0 \\ 0.15 \end{bmatrix}$	$9.35e-6$	2.55	1.30
$p = \begin{bmatrix} 0 \\ -0.05 \\ 0.15 \end{bmatrix}$	$9.31e-6$	3.00	1.81

Physical experiments for σ_{min} – results



$r = 0.05m$	$\frac{\lvert < e, v_1 > \rvert}{\lvert e \rvert}$	$\frac{\lvert < e, v_2 > \rvert}{\lvert e \rvert}$	$\frac{\lvert < e, v_3 > \rvert}{\lvert e \rvert}$	$\frac{\lvert < e, v_4 > \rvert}{\lvert e \rvert}$	$\frac{\lvert < e, v_{min} > \rvert}{\lvert e \rvert}$
Ratio	22%	25%	21%	57%	52%

$r = 0.10m$	$\frac{\lvert < e, v_1 > \rvert}{\lvert e \rvert}$	$\frac{\lvert < e, v_2 > \rvert}{\lvert e \rvert}$	$\frac{\lvert < e, v_3 > \rvert}{\lvert e \rvert}$	$\frac{\lvert < e, v_4 > \rvert}{\lvert e \rvert}$	$\frac{\lvert < e, v_{min} > \rvert}{\lvert e \rvert}$
Ratio	24%	27%	21%	51%	54%

Conclusion and outlooks

1. Properly linearized magnetic dipole model
2. Algorithm that is more than 100 times faster than the classic one
3. Criteria that affects the convergence accuracy of the algorithm
4. How globally or how fast the algorithm converges?

Xiaowei Lin
xiaoweilin@student.ethz.ch

Multi-Scale Robotics Lab
ETH Zurich
Tannenstrasse 3
8092 Zurich
https://msrl.ethz.ch/

Dupont Lab
Boston Children's Hospital, Harvard Medical School
300 Longwood Ave
Boston, MA 02115
https://robotics.tch.harvard.edu/