Tobias Wulf
tobias.wulf@haw-hamburg.de

HAW HAMBURG

Bachelor thesis

Angular Measurement by Magnetic Sensor Arrays and Tolerance Compensation by Gaussian Process

Fakultät Technik und Informatik
Department Informations- und Elektrotechnik

Faculty of Computer Science and Engineering
Department of Information and Electrical Engineering

02.07.2021

Overview

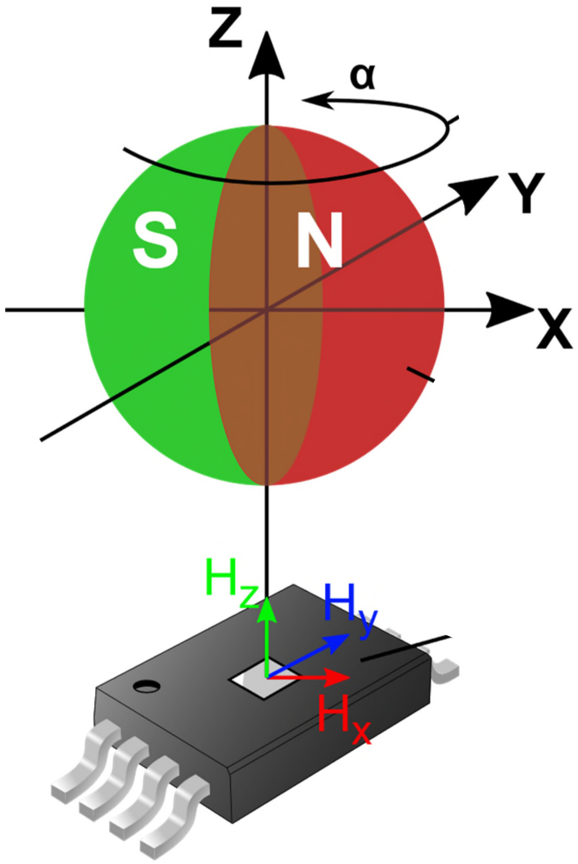
- Application
- Characteristics
- Data Adaption
- Gaussian Processes

Overview

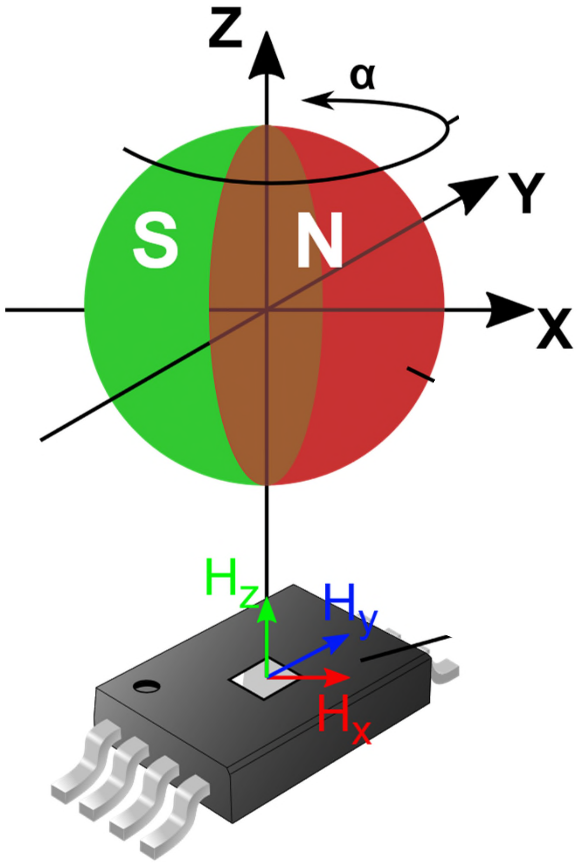
HAW HAMBURG

- Application
- Characteristics
- Data Adaption
- Gaussian Processes
- Software Development
- Simulation – Sensor Array
- Simulation – Gaussian Processes
- Testing Experiments
- Summary

Application

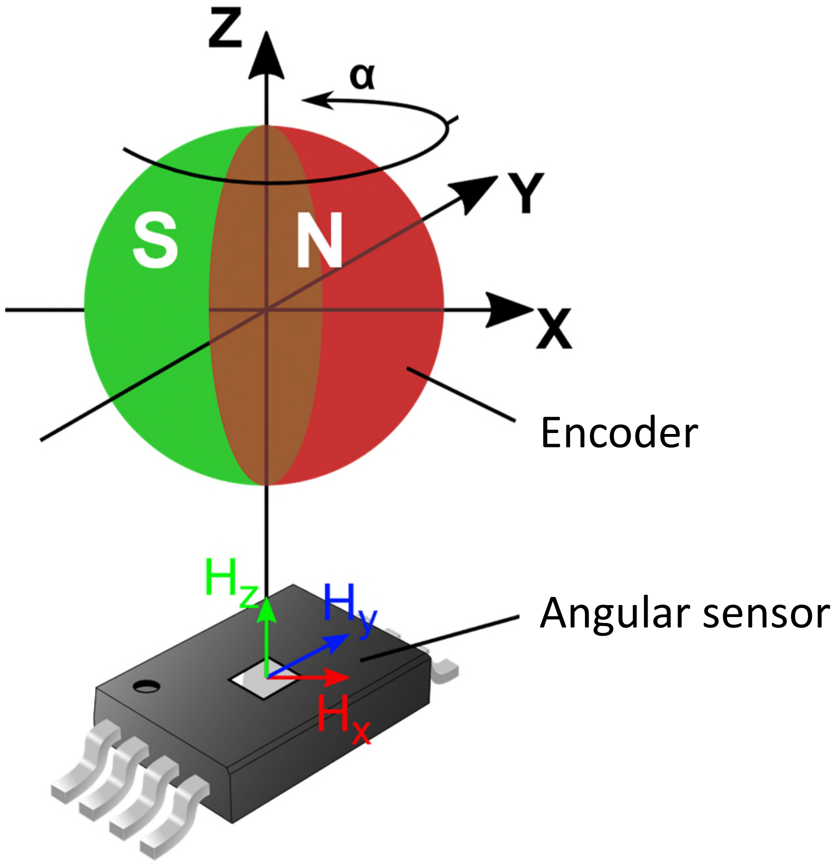


Application



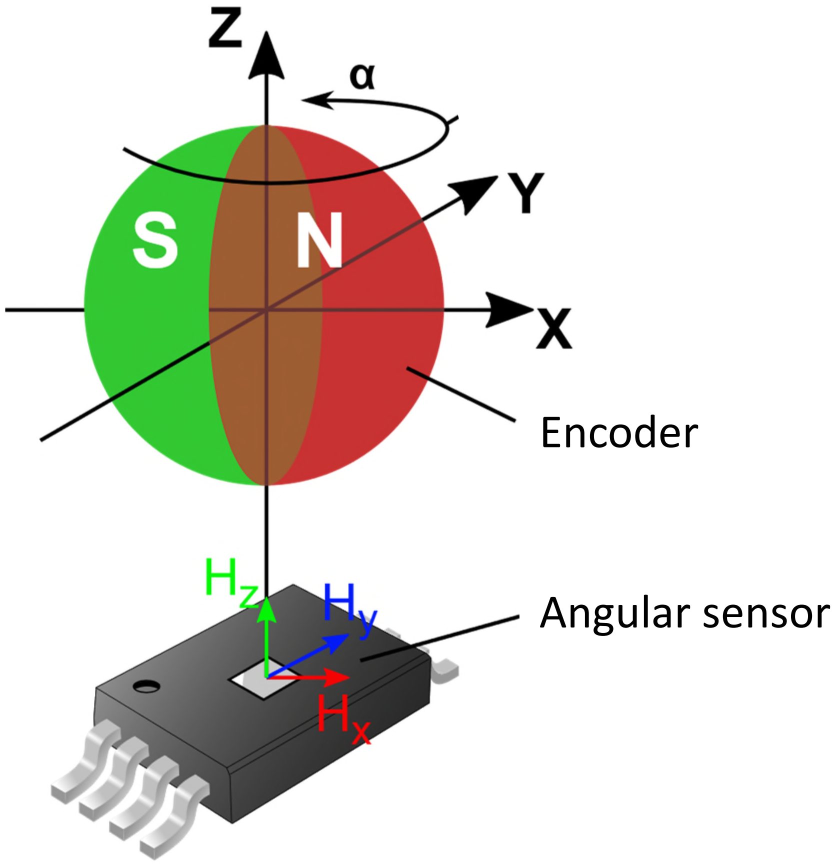
- Contactless angle measurement

Application



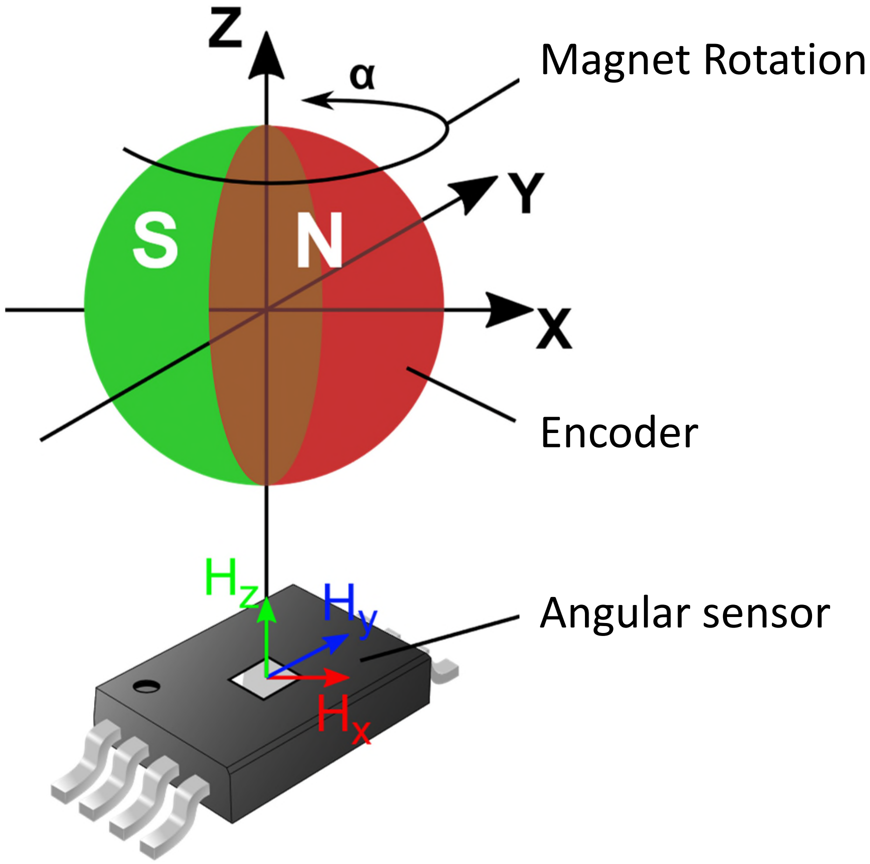
- Contactless angle measurement
- Angle encoded by magnetic field

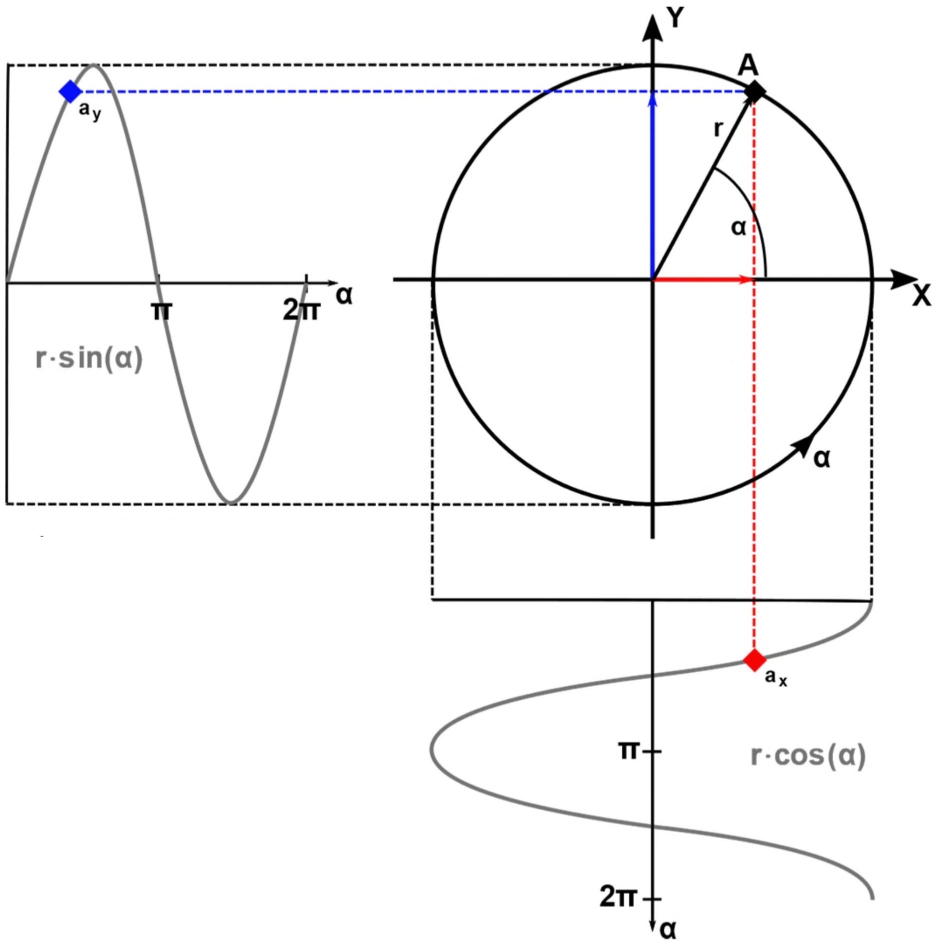
Application



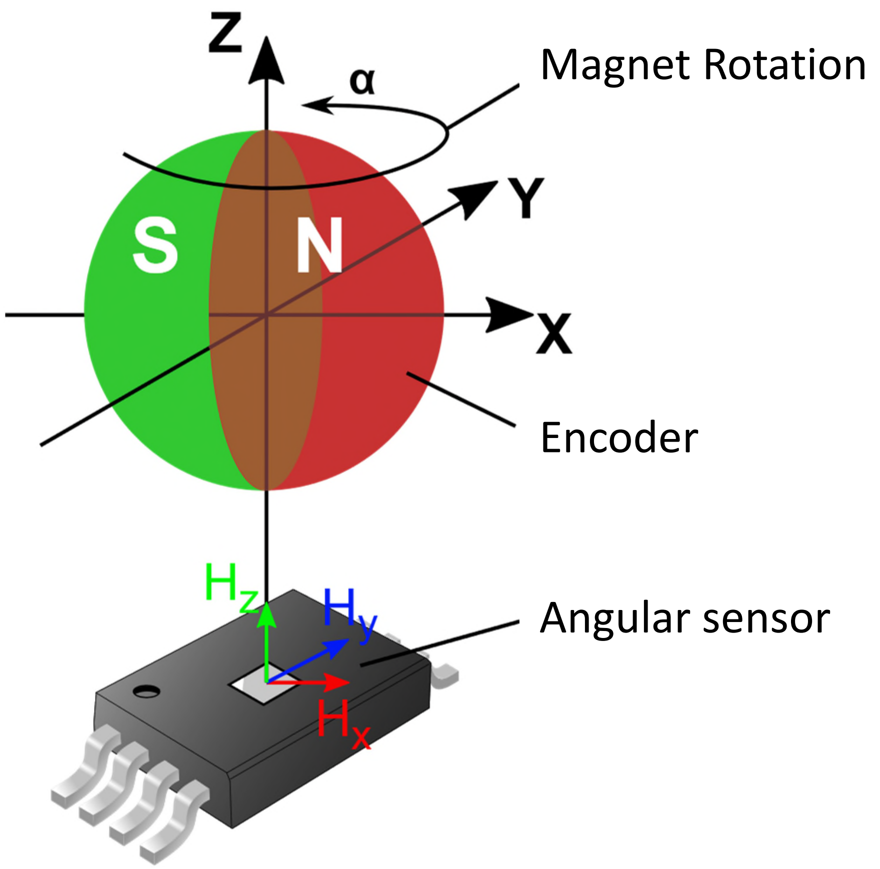
- Contactless angle measurement
- Angle encoded by magnetic field
- Field strength measurement in X-/ Y-direction

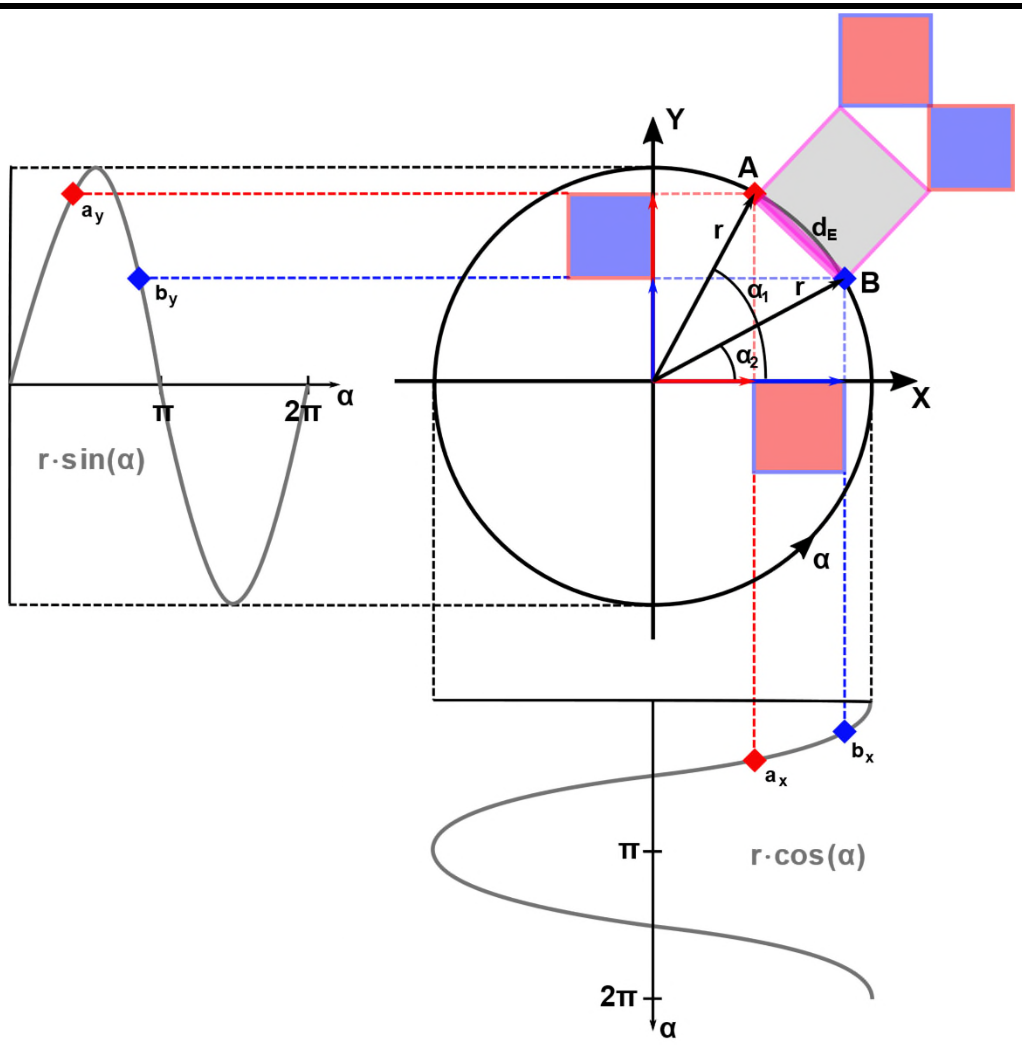
Application



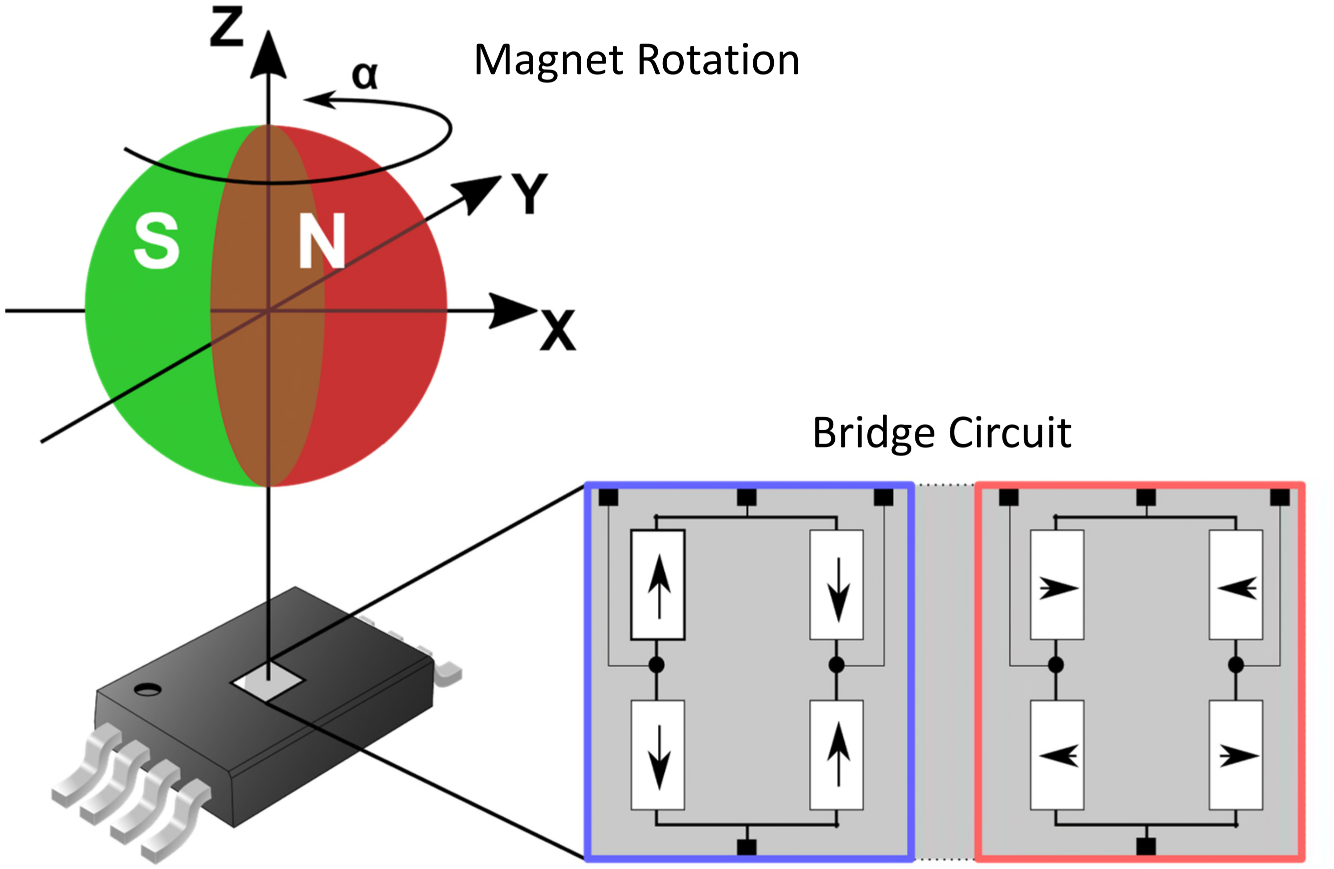


Application

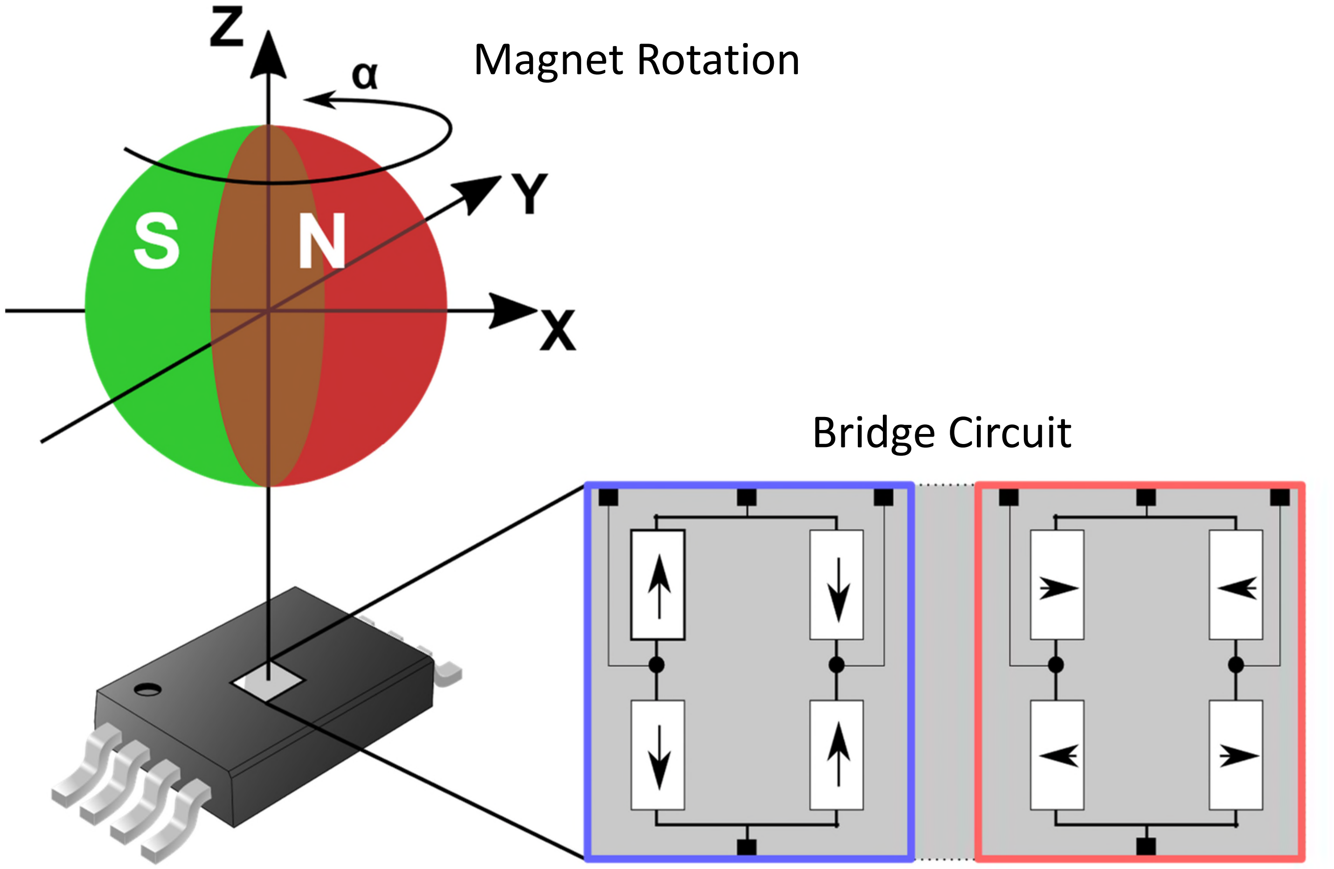




Application

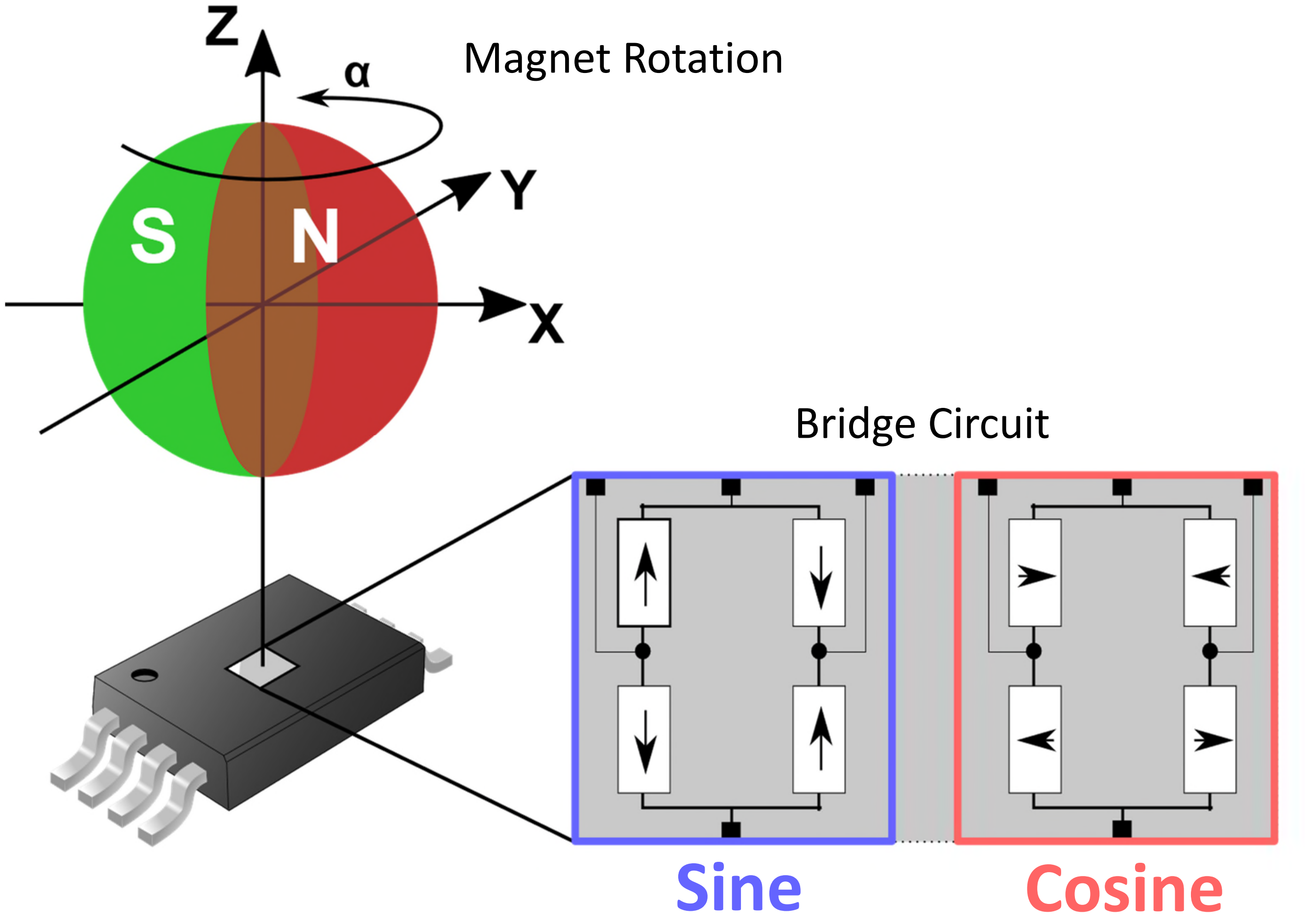


Application



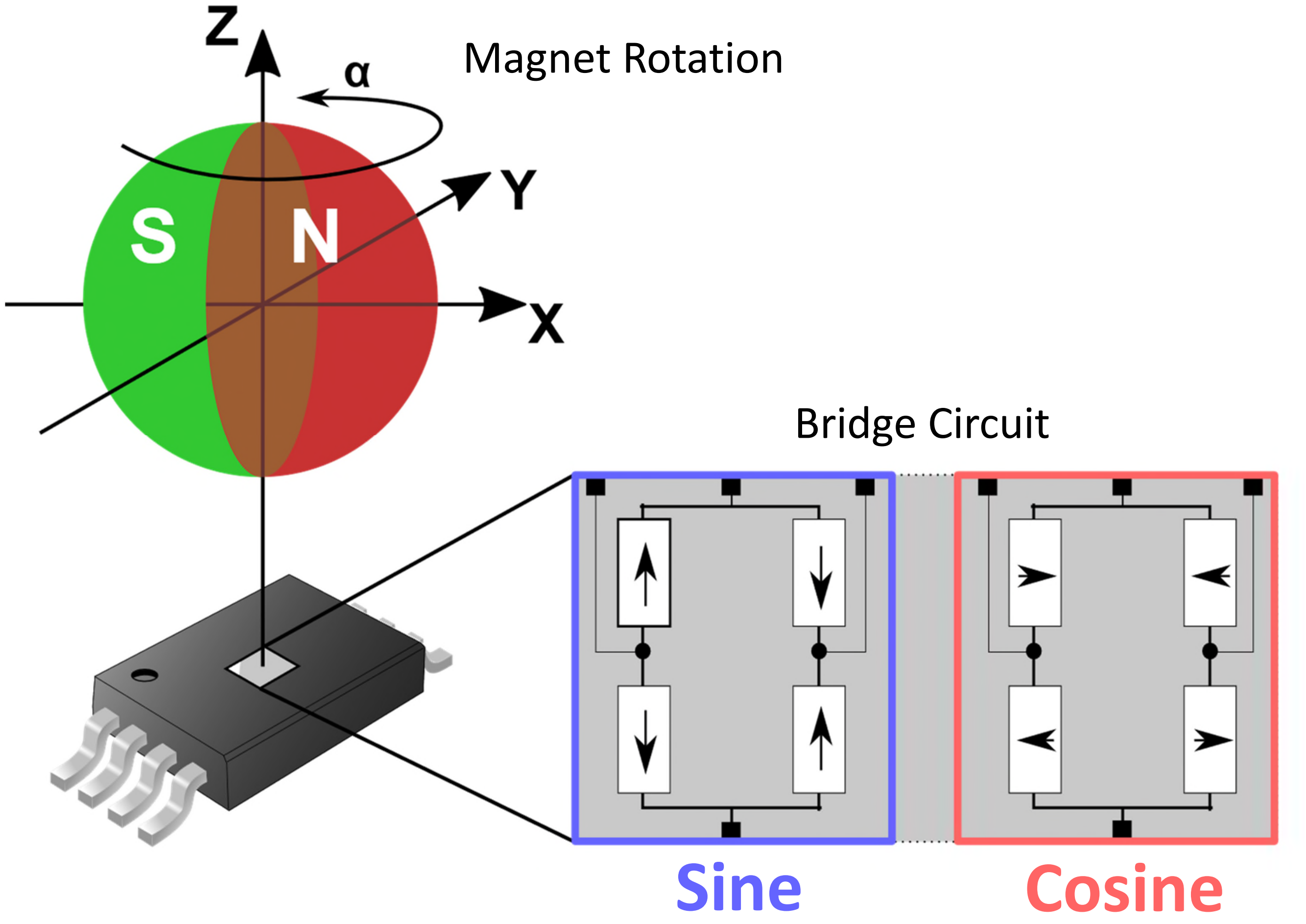
- TMR Sensor

Application



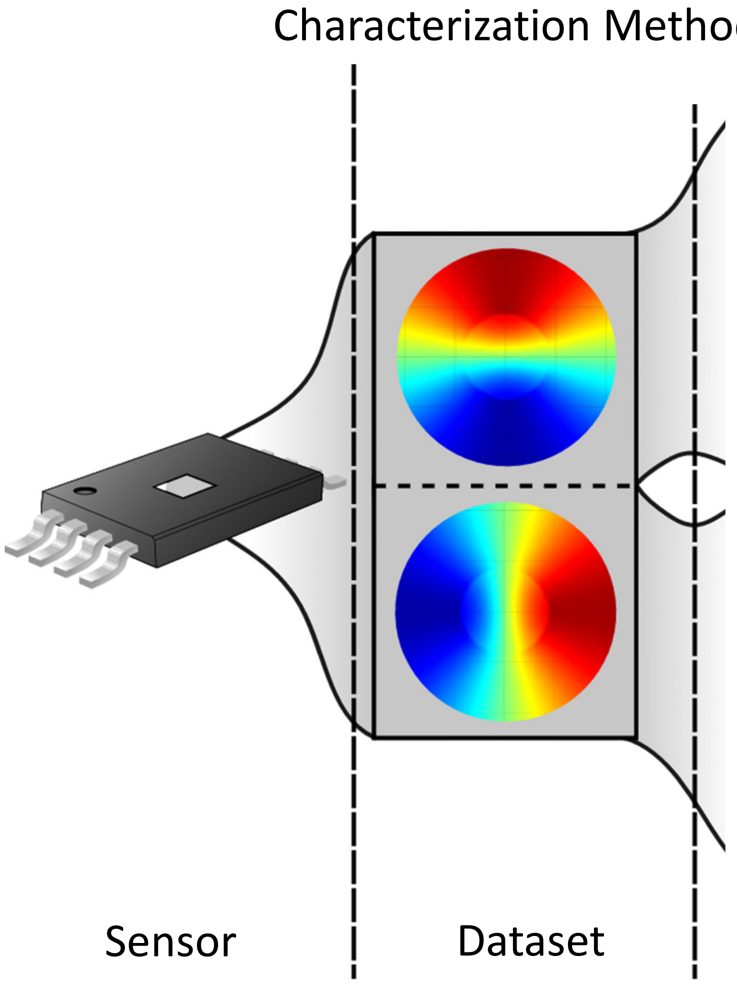
- TMR Sensor
- Wheatstone Bridge

Application



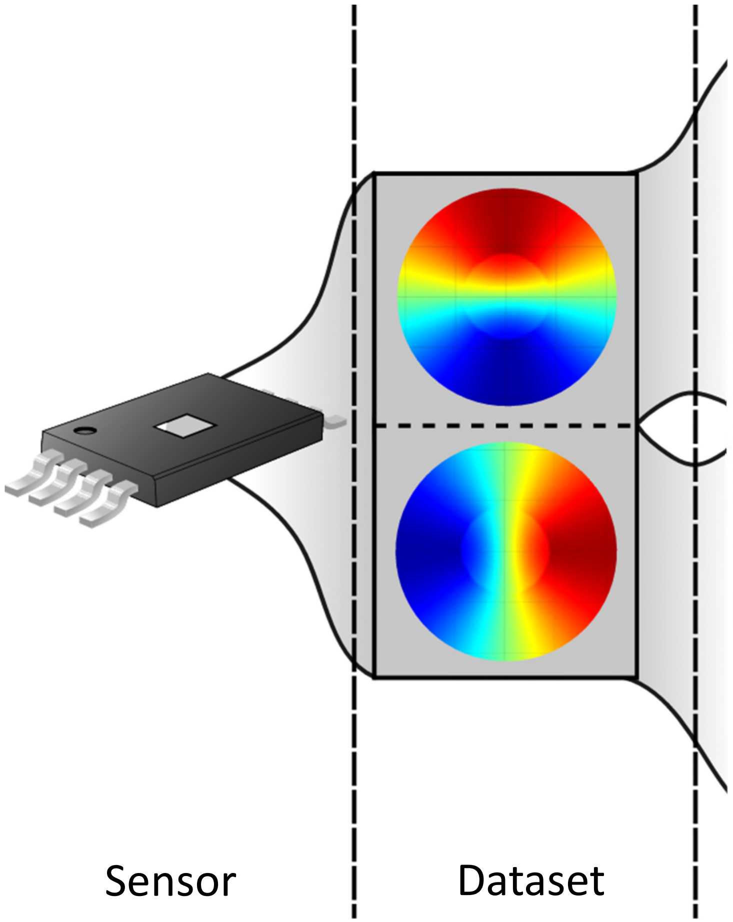
- TMR Sensor
- Wheatstone Bridge
- 90° rotated to each other
- 360° periodicity

Characteristics



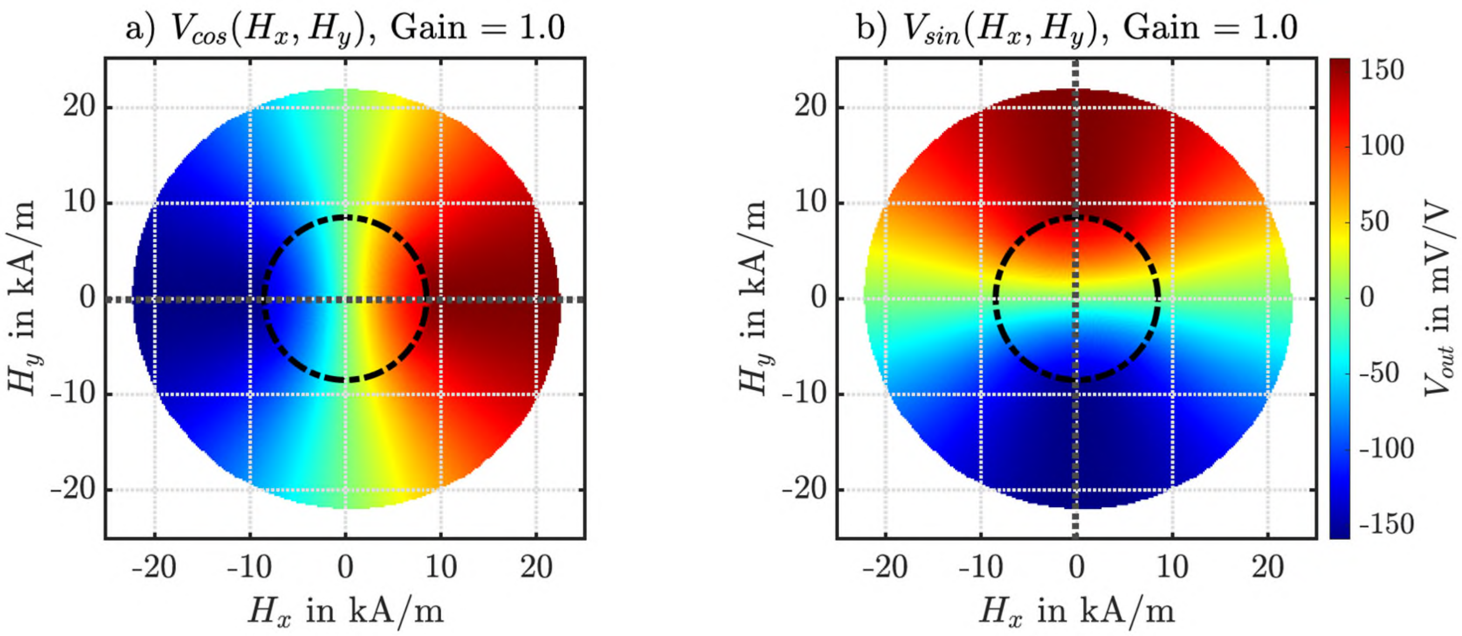
Characteristics

Characterization Method

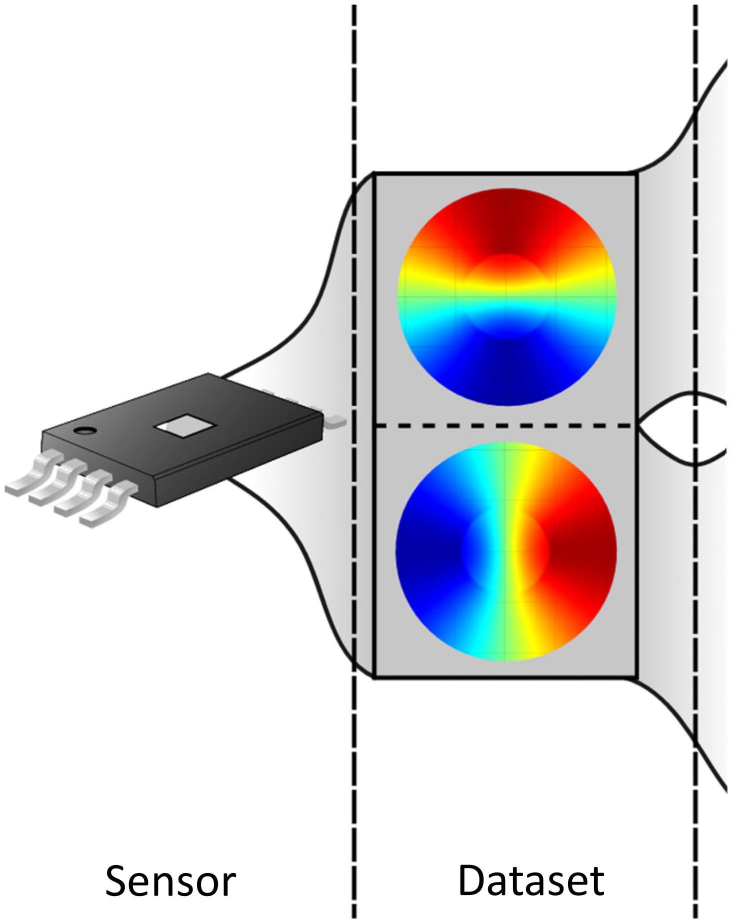


360° periodicity

90° rotated

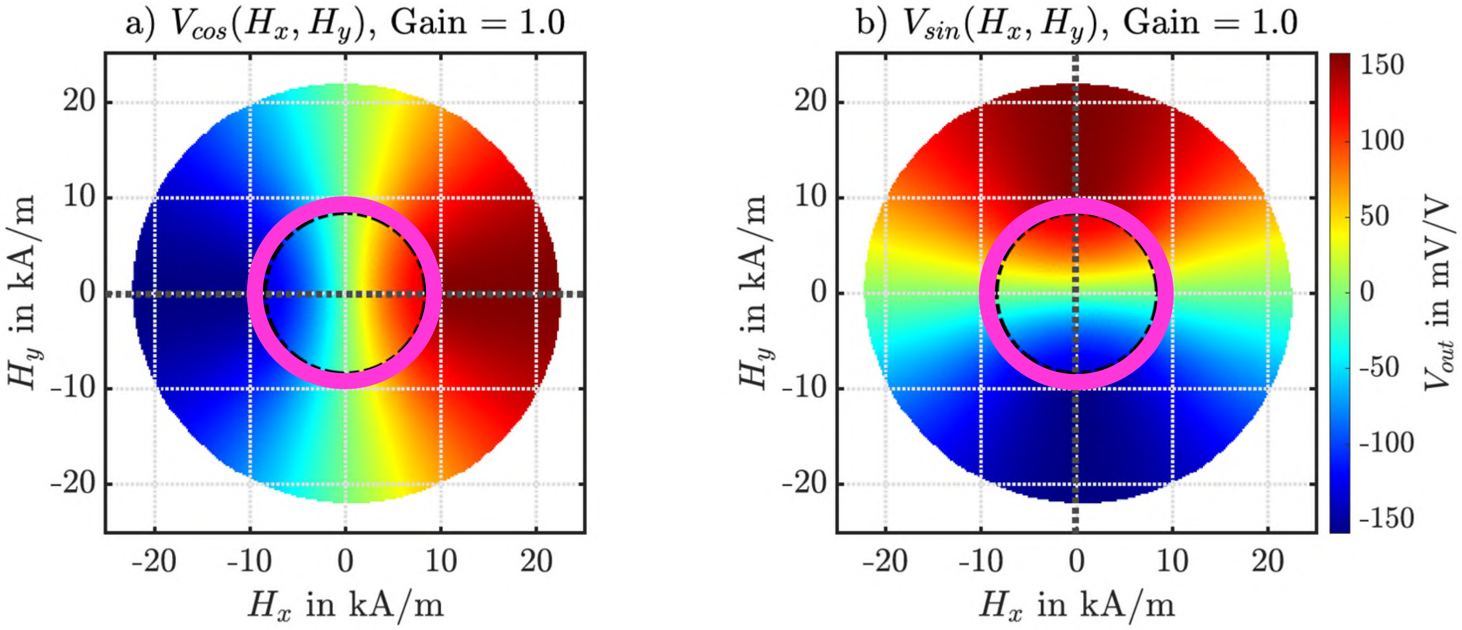


Characteristics



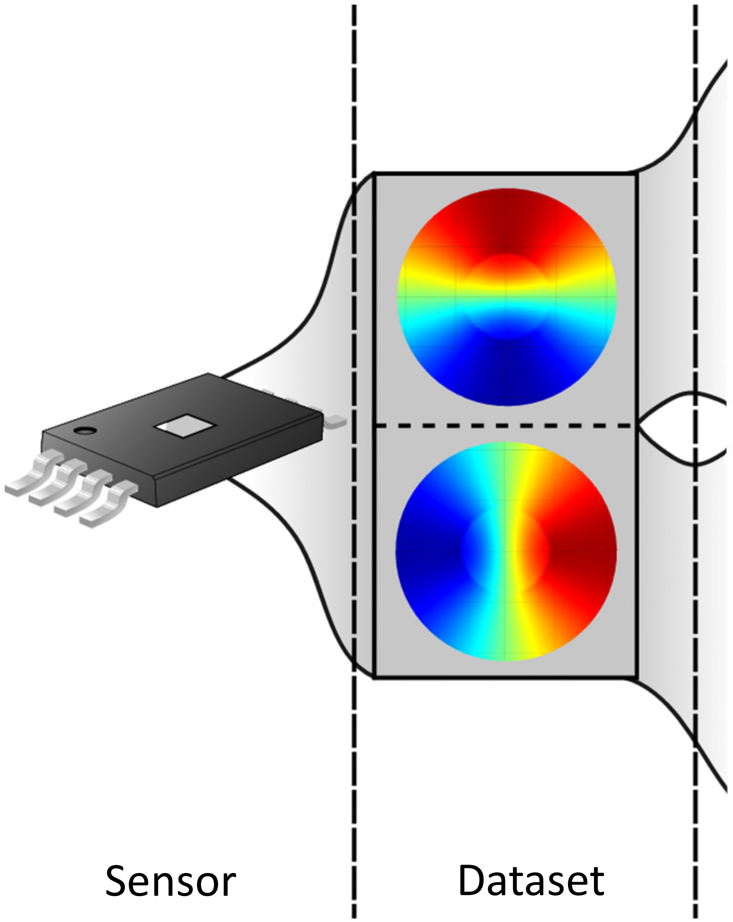
360° periodicity

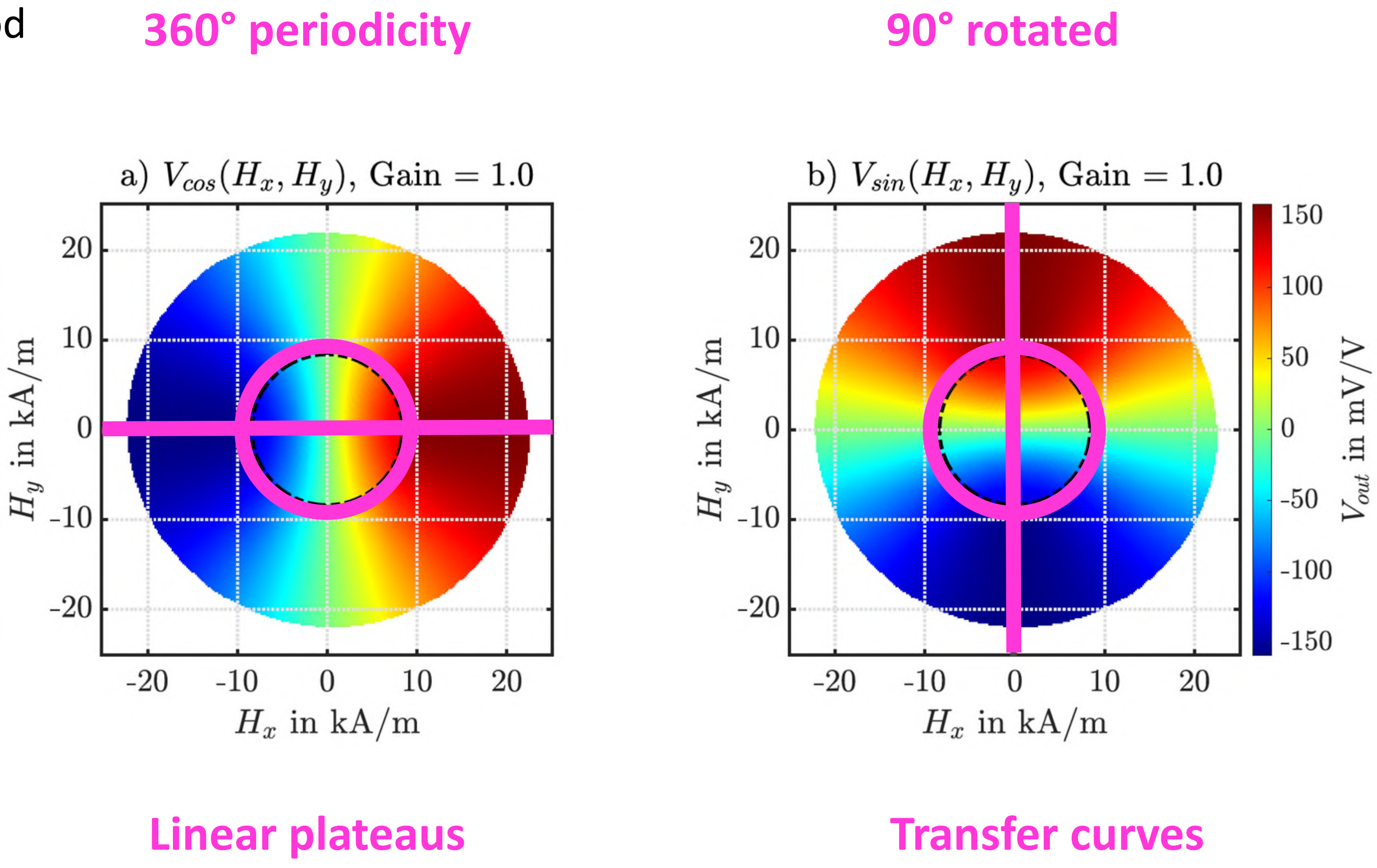
90° rotated



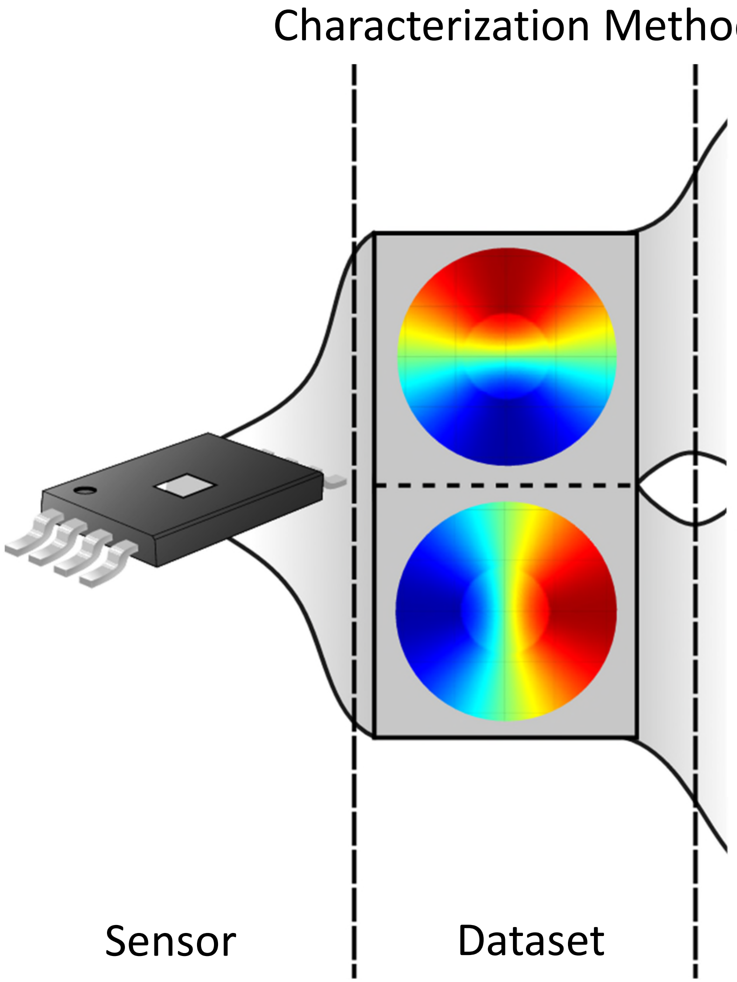
Linear plateaus

Characteristics

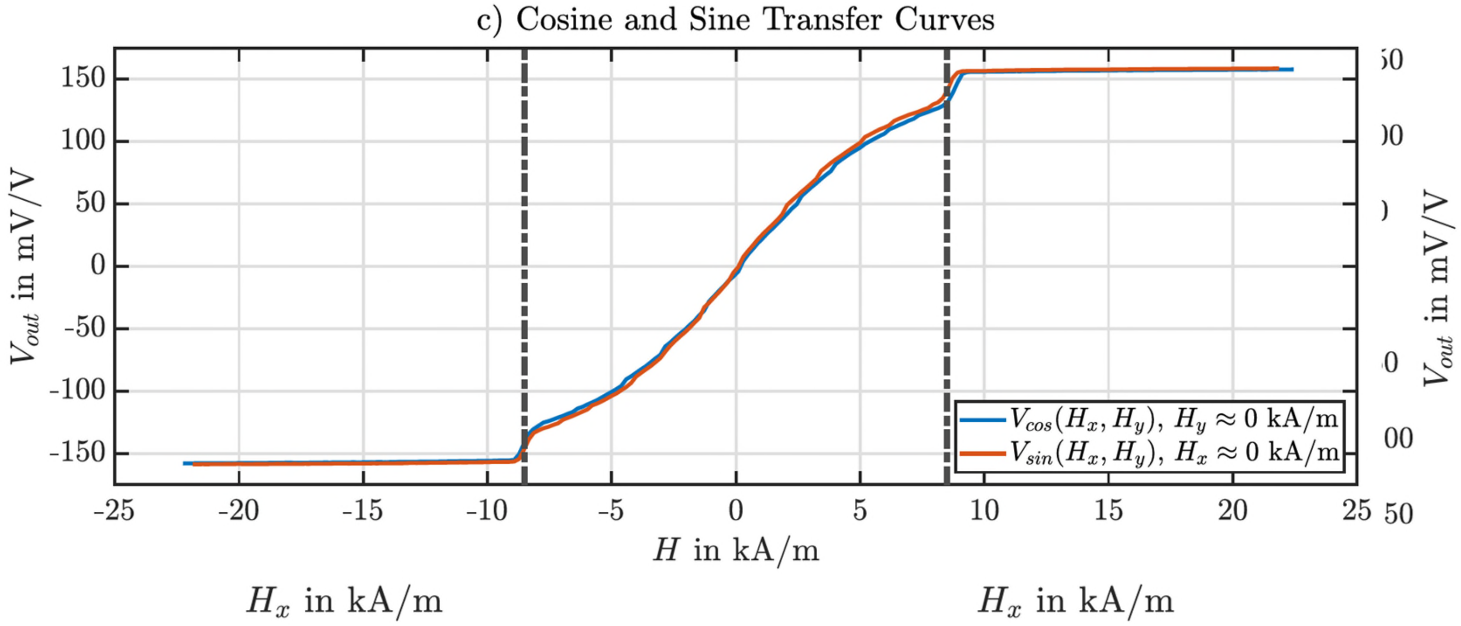




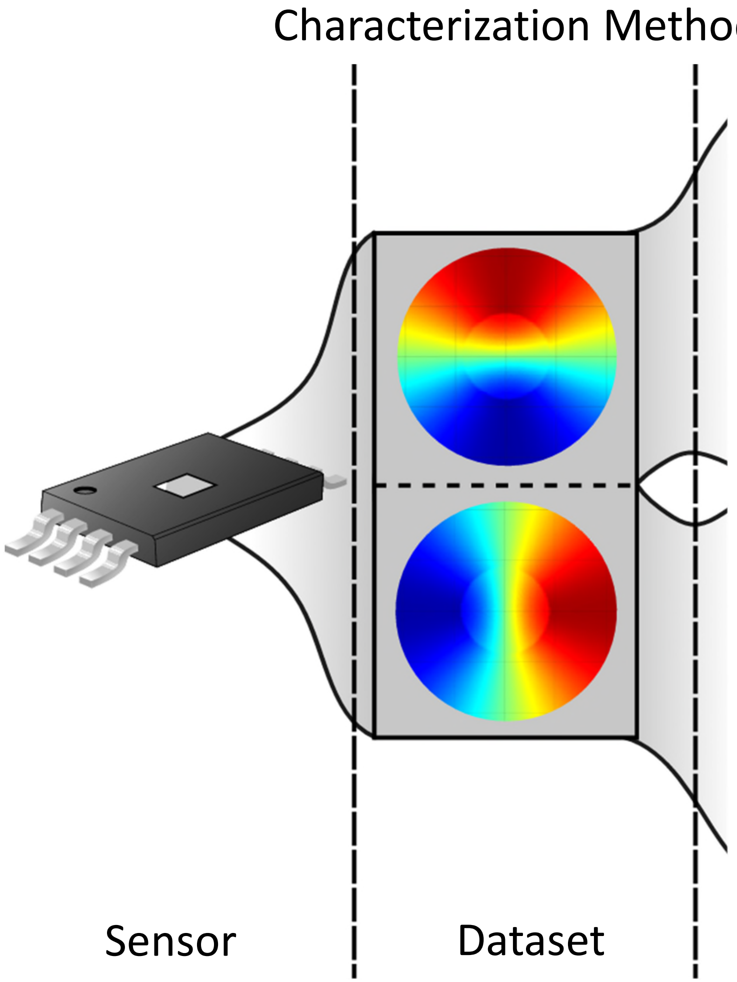
Characterization



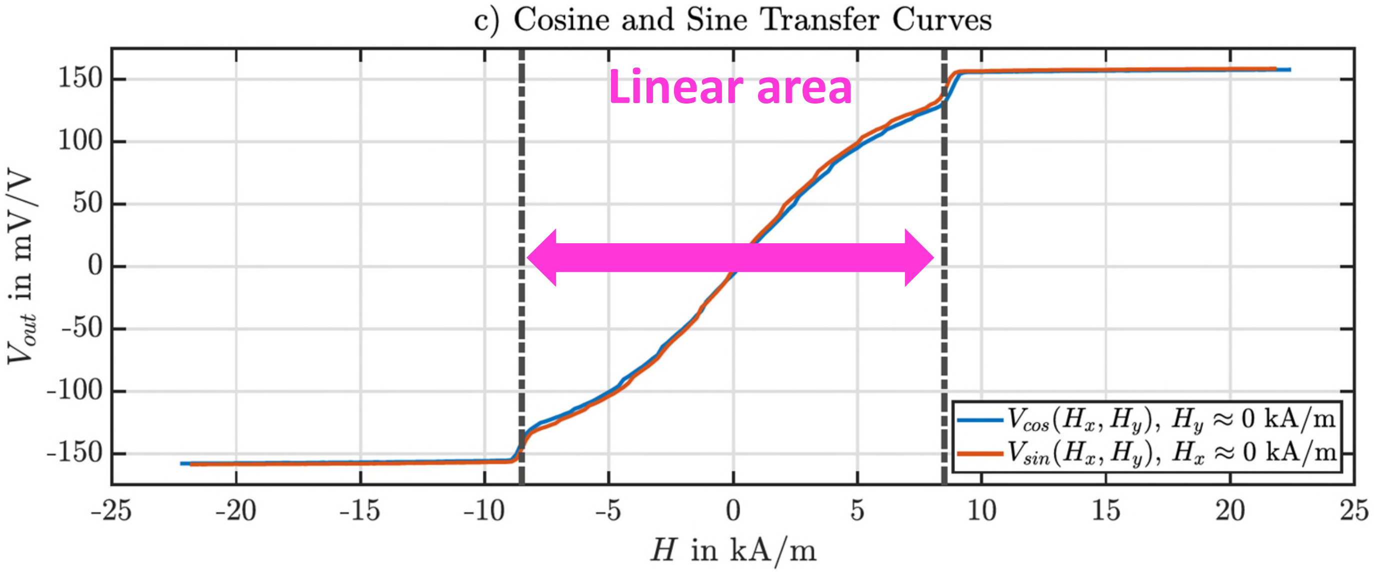
Transfer curves as slices



Characteristics

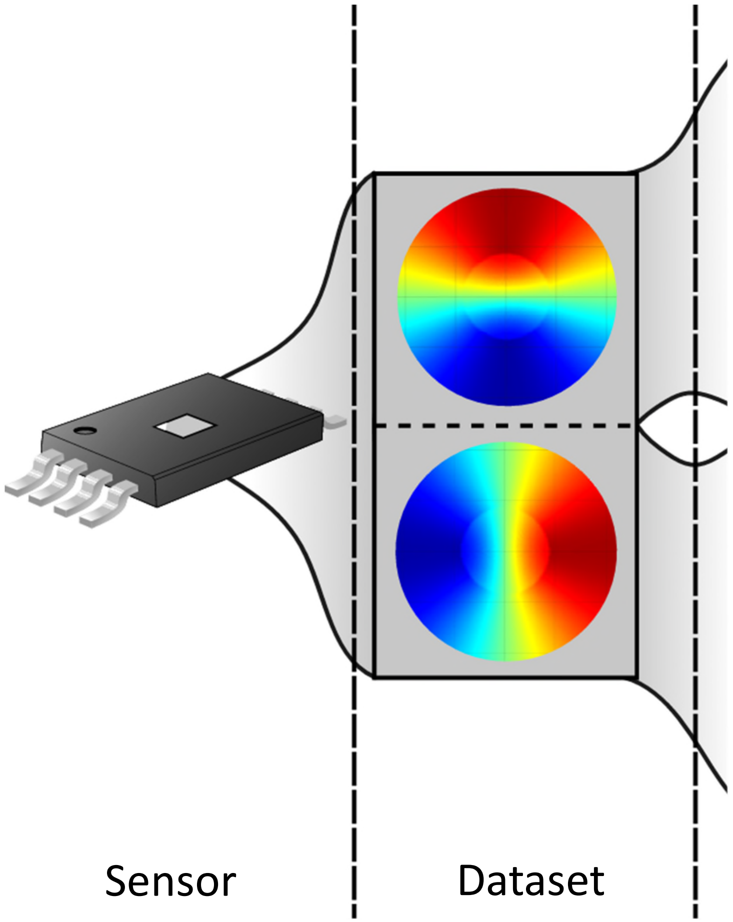


Transfer curves as slices

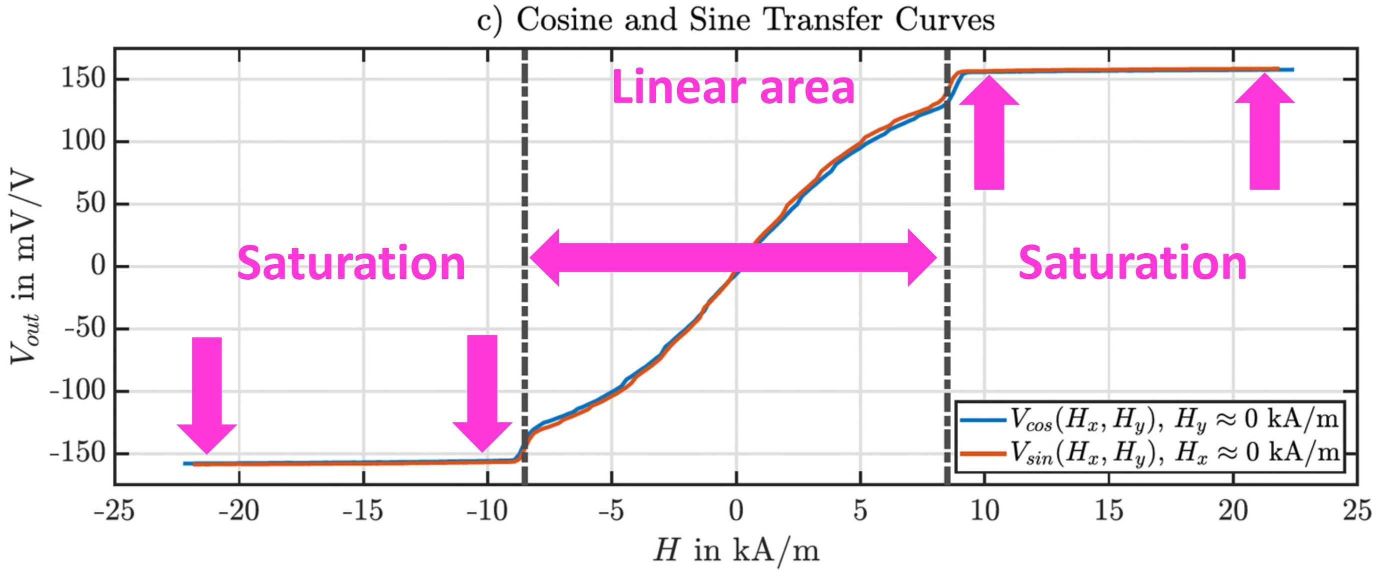


Characteristics

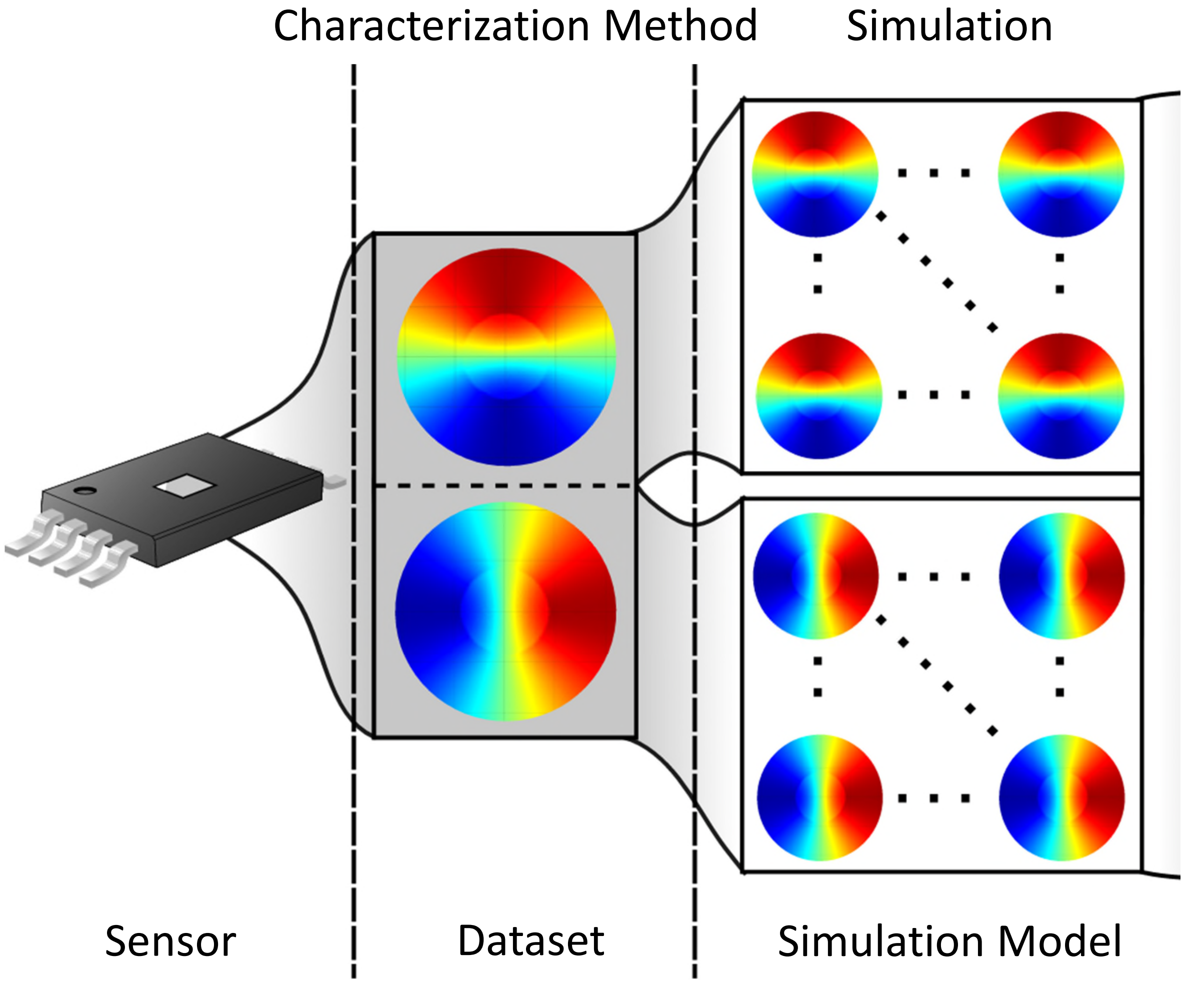
HAW HAMBURG



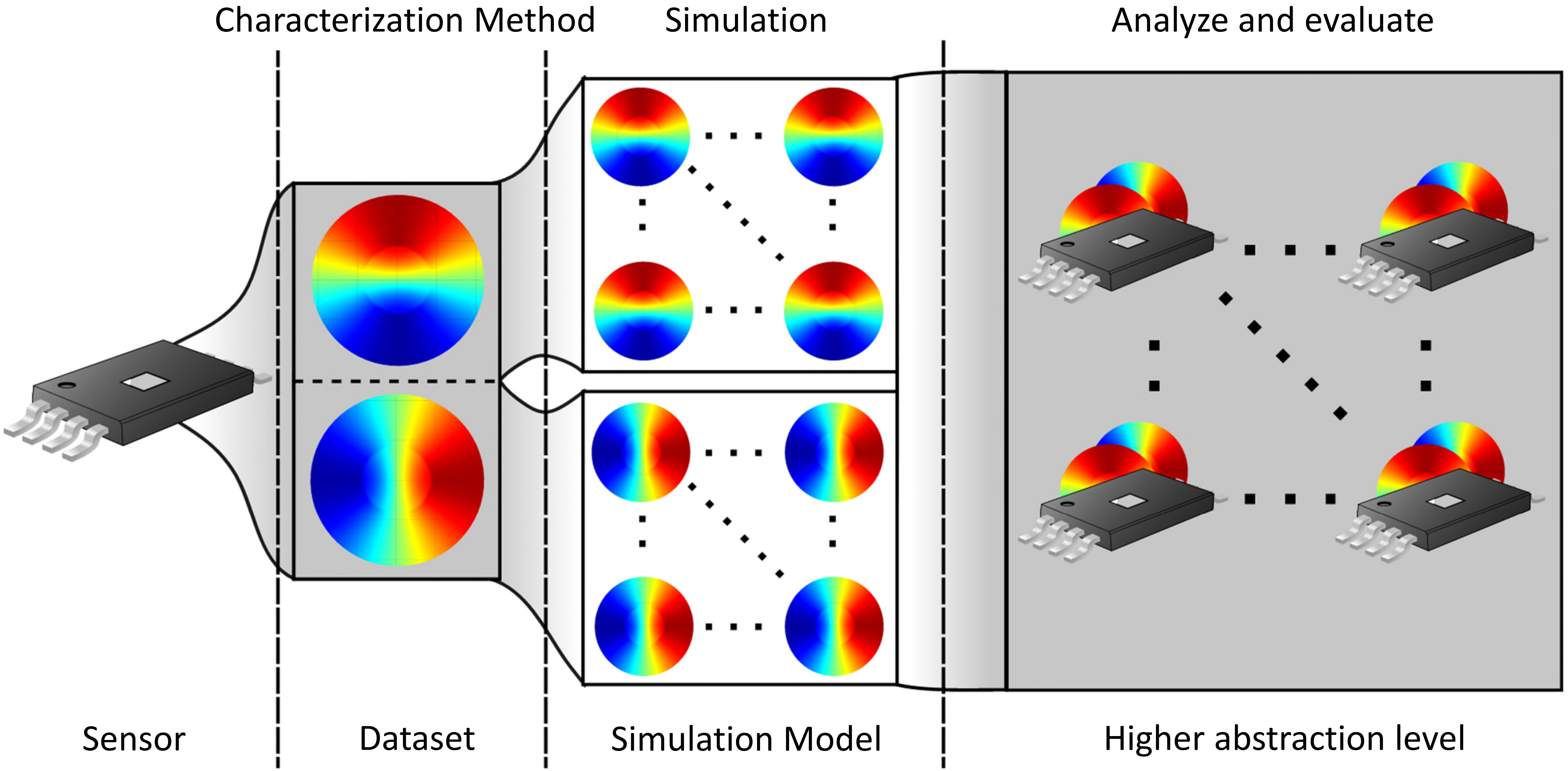
Transfer curves as slices



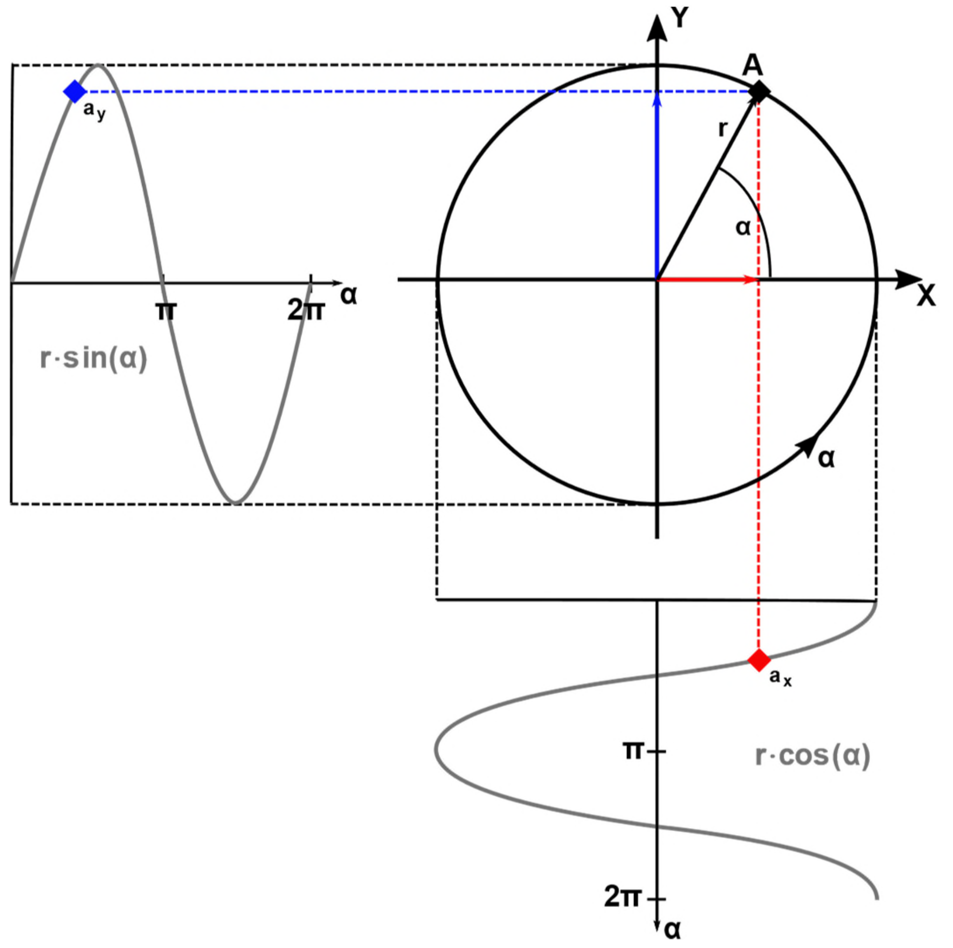
Characteristics



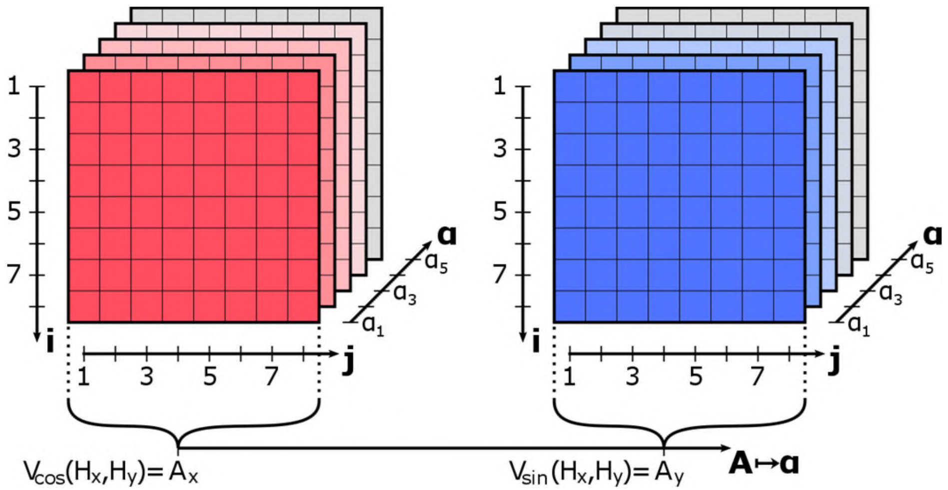
Characteristics

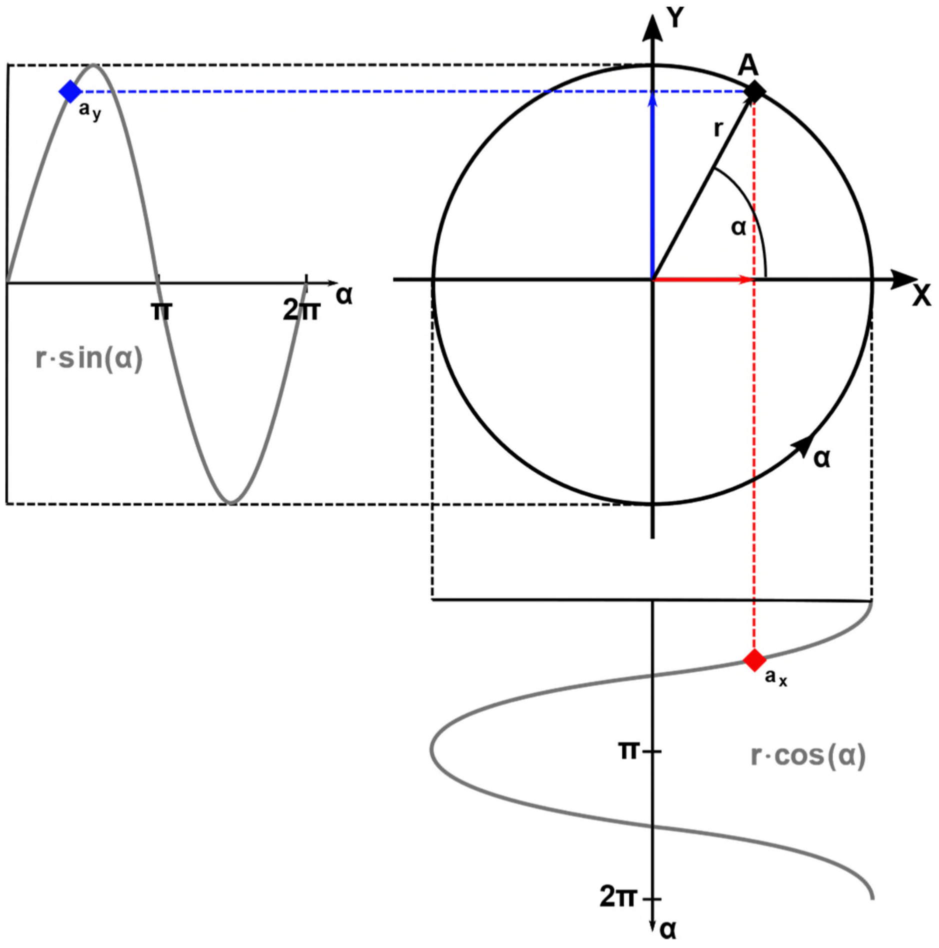


Data Adaption

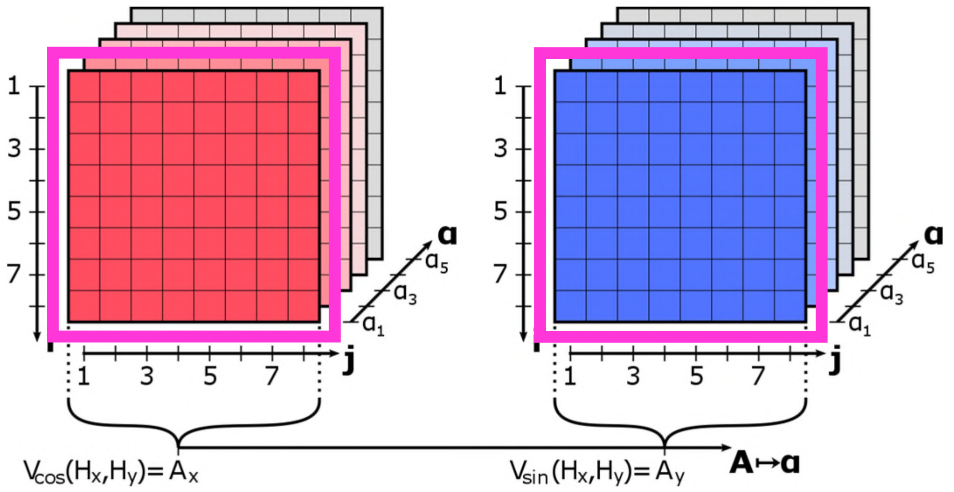


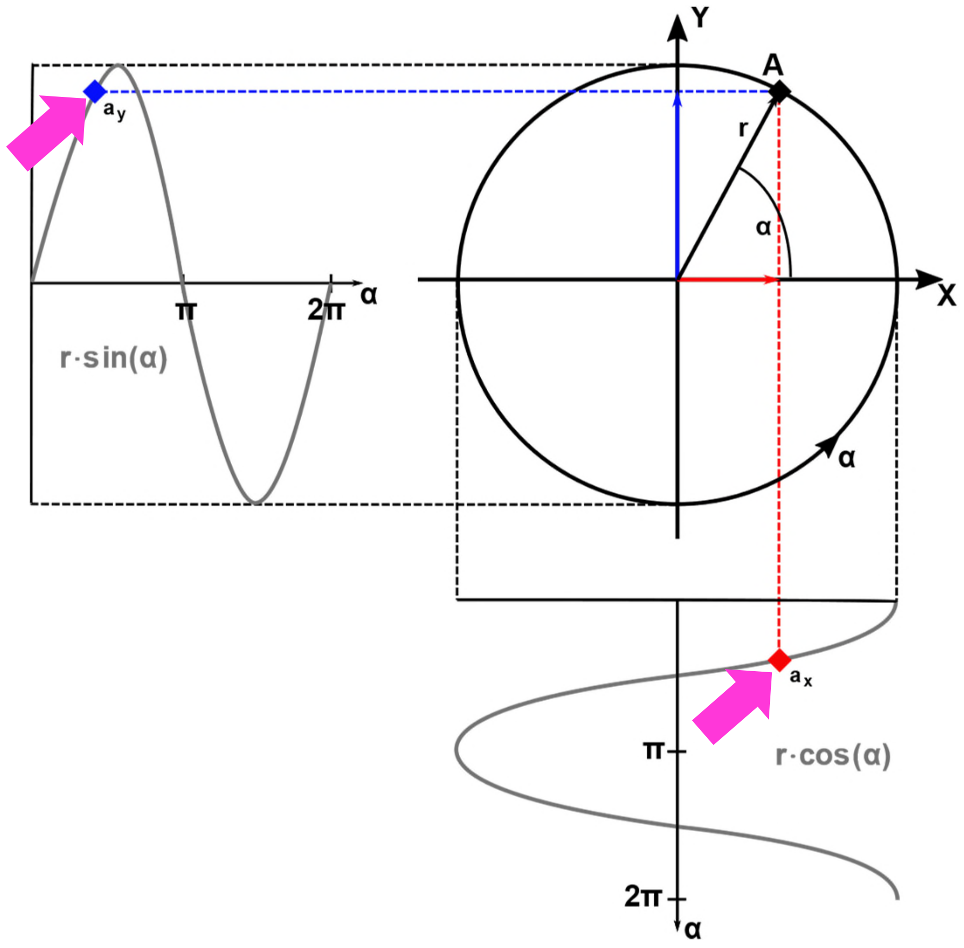
Data Adaption



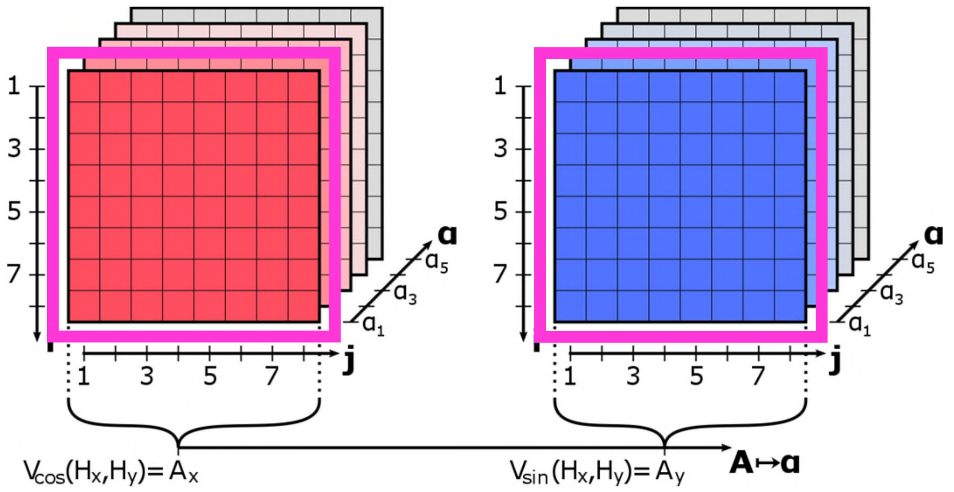


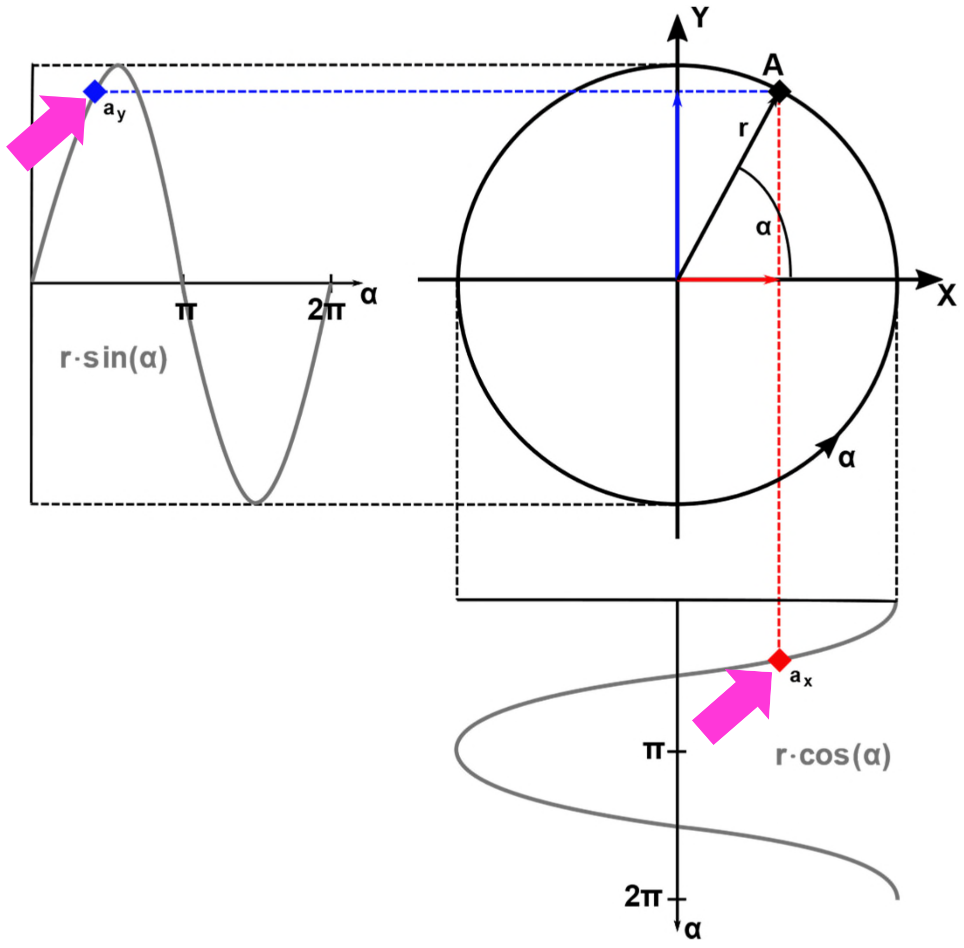
Data Adaption





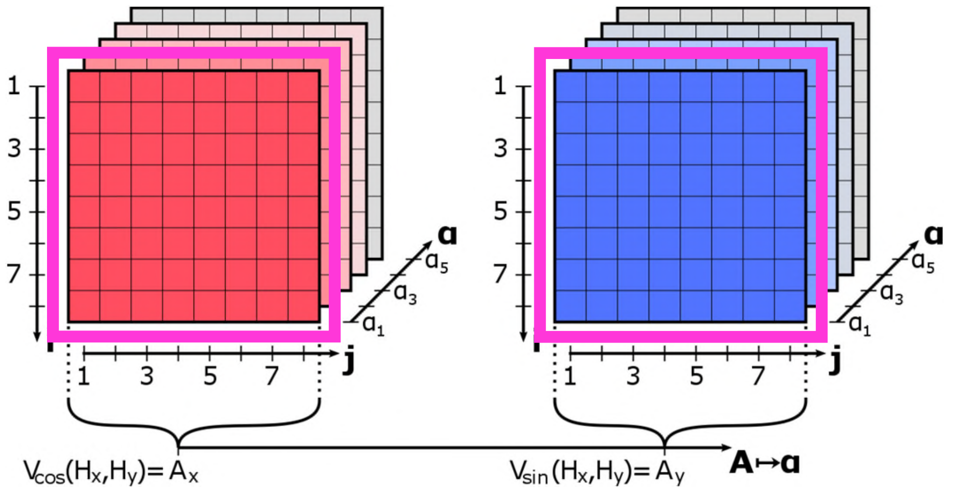
Data Adaption

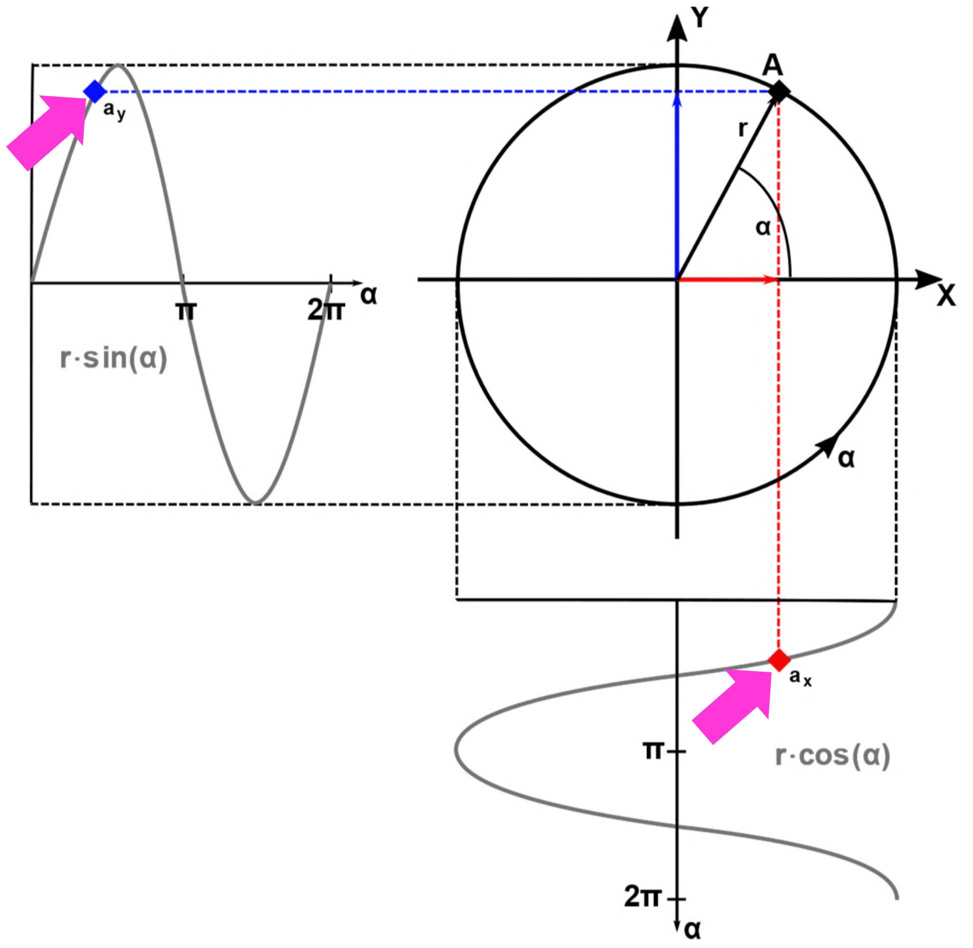




- Arrays as long vectors

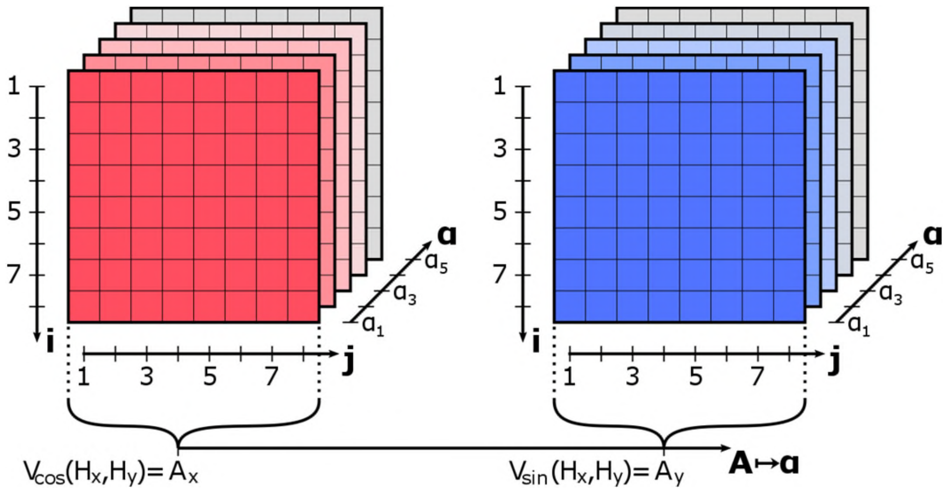
Data Adaption



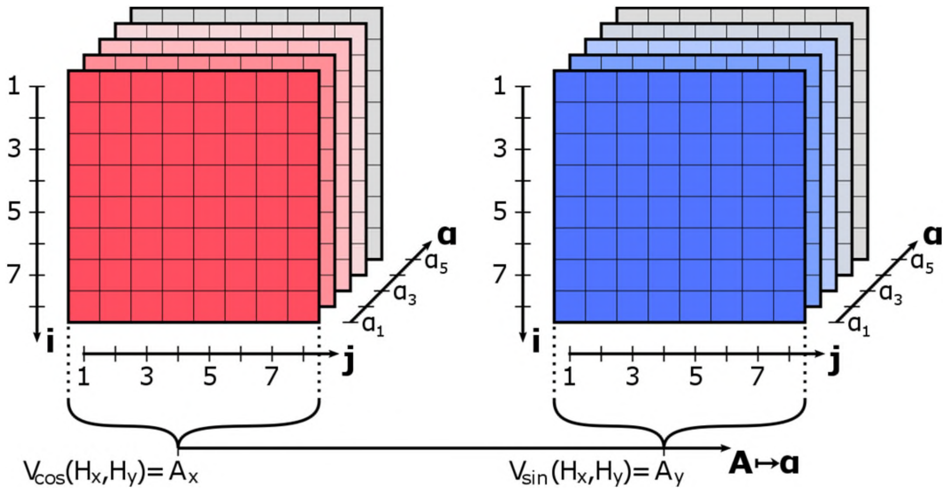


- Arrays as long vectors
- Frobenius-Norm

Gaussian Processes

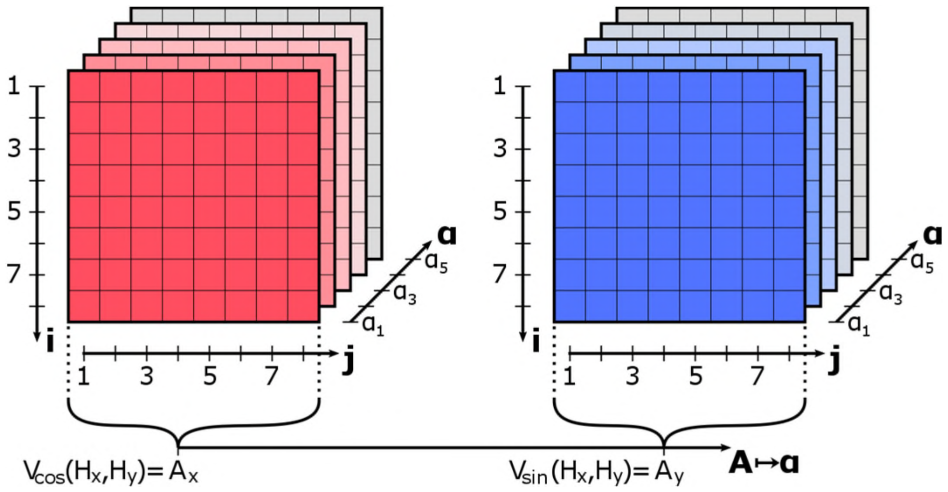


Gaussian Processes



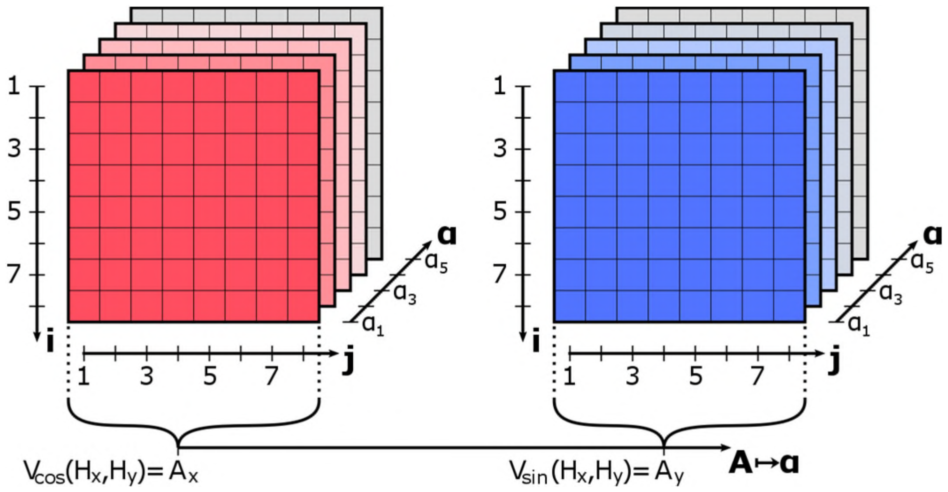
- Learning by training data

Gaussian Processes



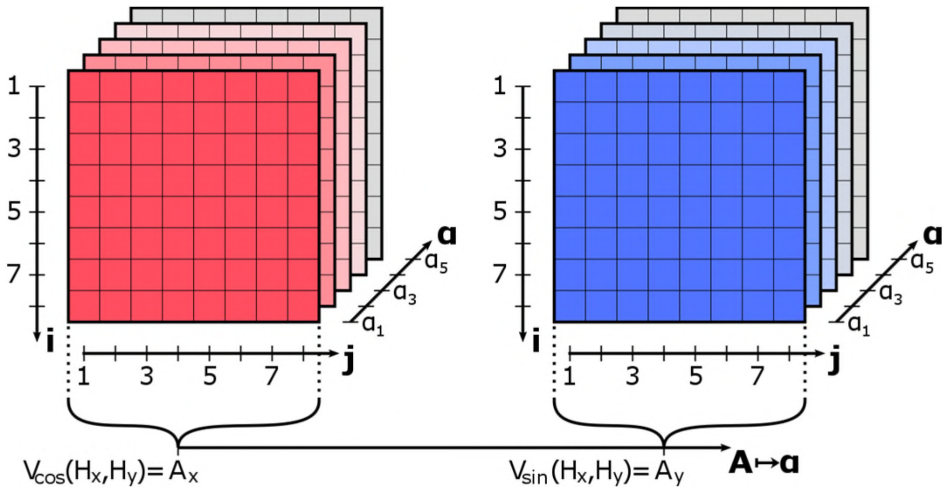
- Learning by training data
- Adjust by test data
- Optimize by model parameter

Gaussian Processes



- Learning by training data
- Adjust by test data
- Optimize by model parameter
- Prediction by regression weights

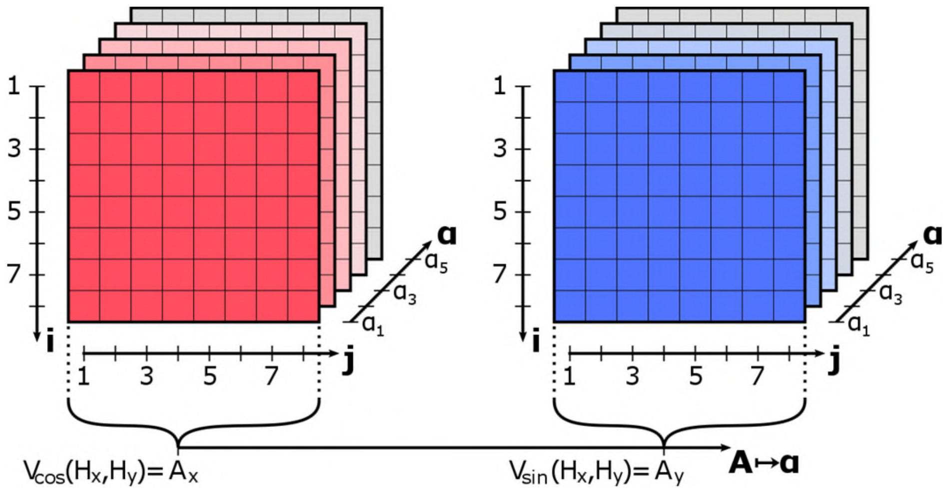
Gaussian Processes

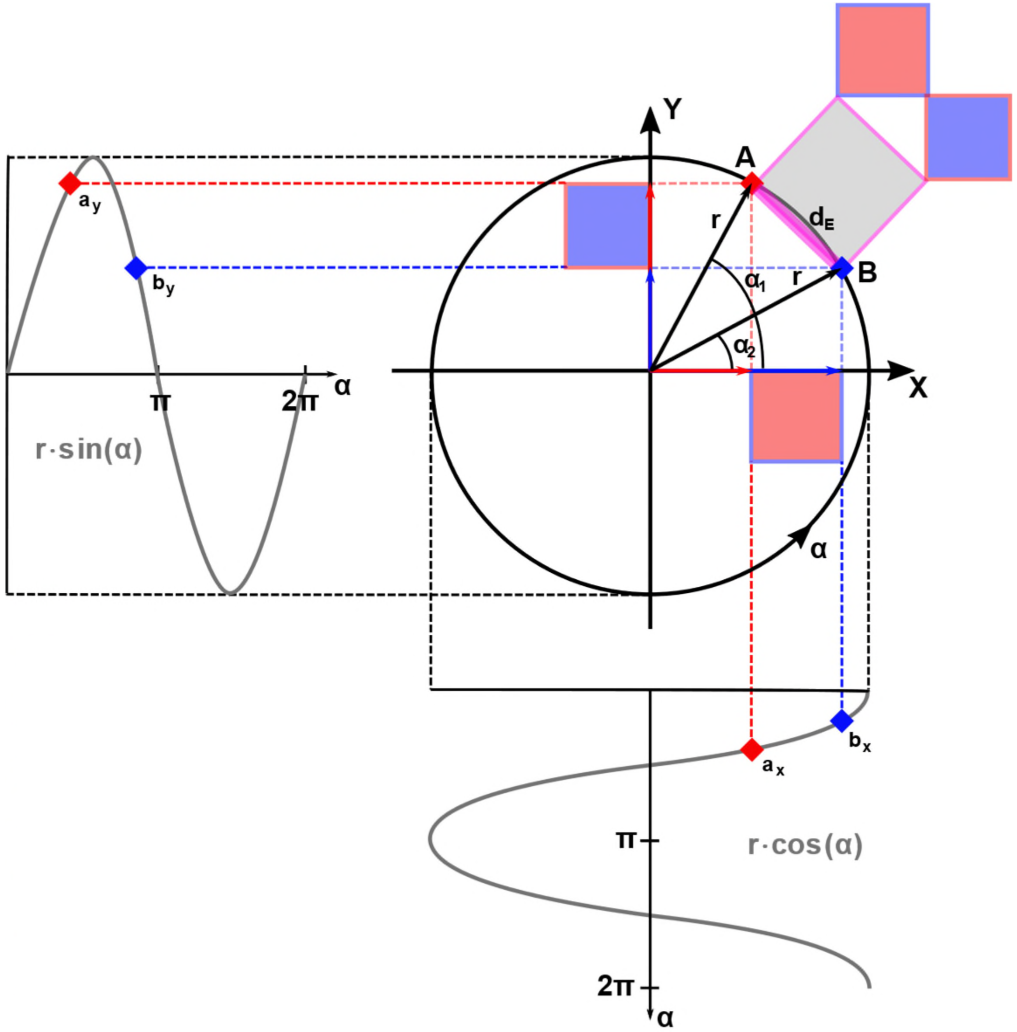


- Learning by training data
- Adjust by test data
- Optimize by model parameter
- Prediction by regression weights

Covariance function (Kernel) drives the model behavior!

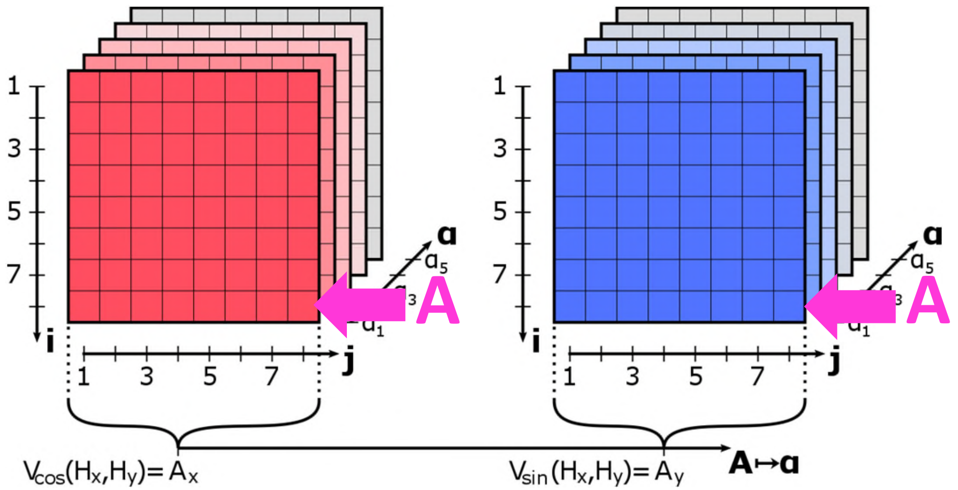
Gaussian Processes

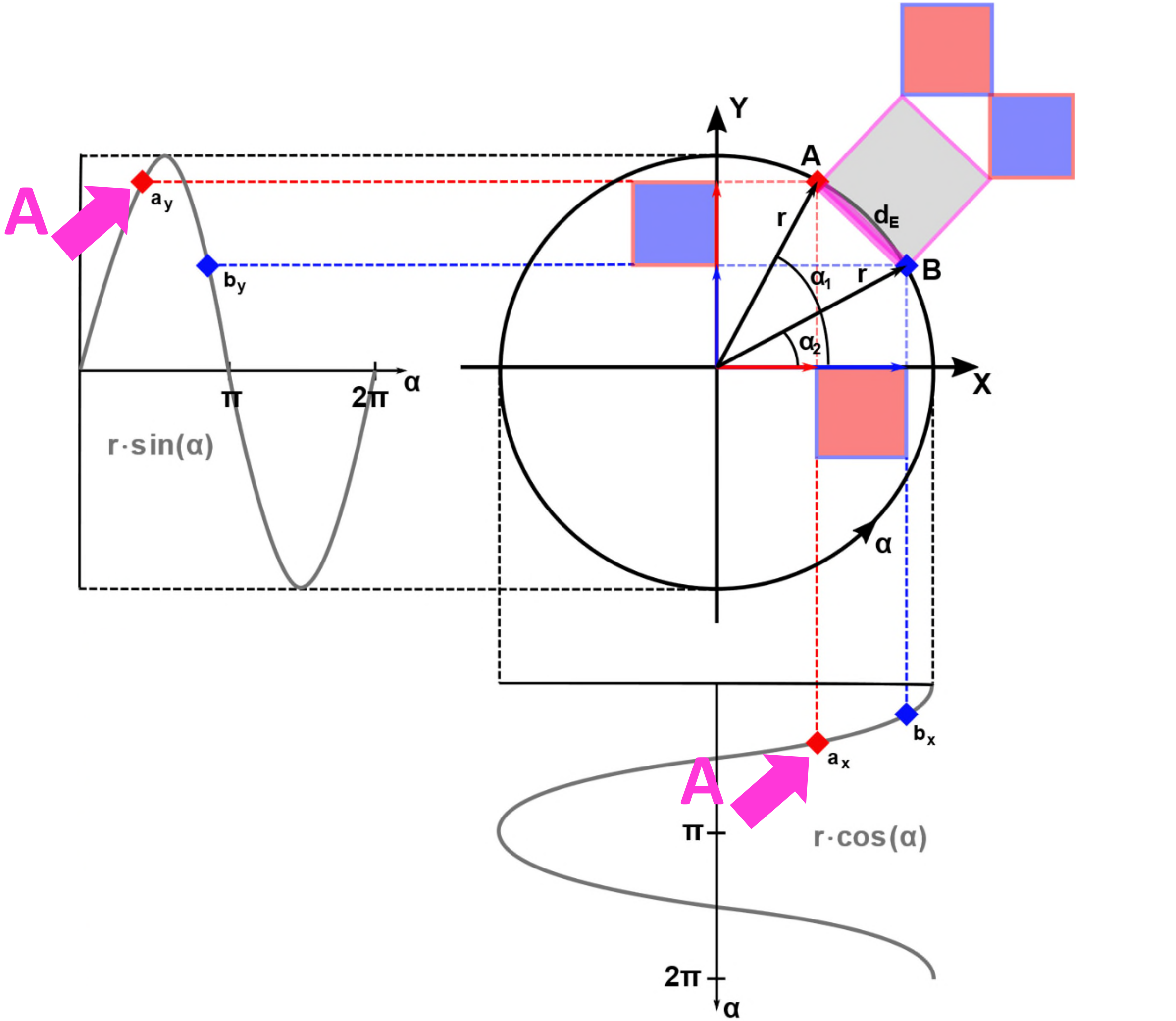




$$d_x^2\langle \mathbf{A},\mathbf{B}\rangle$$

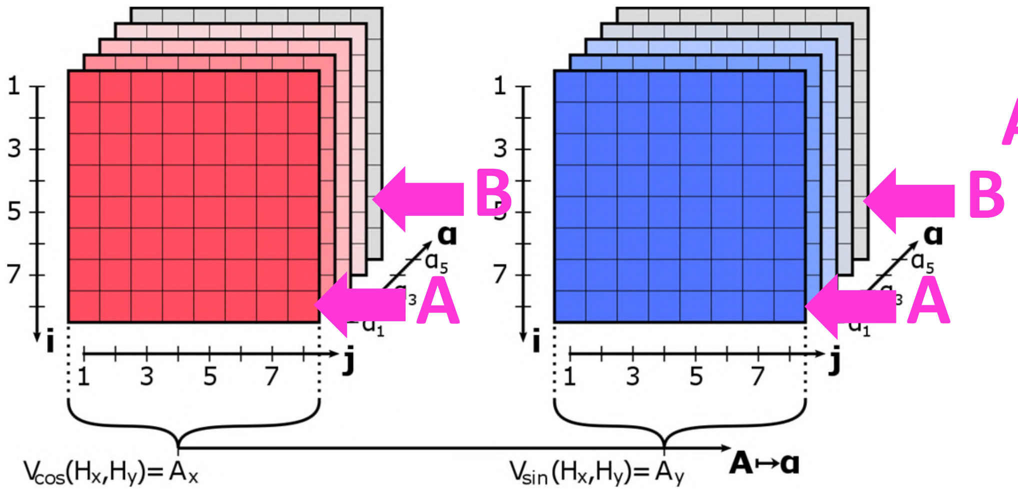
Gaussian Processes

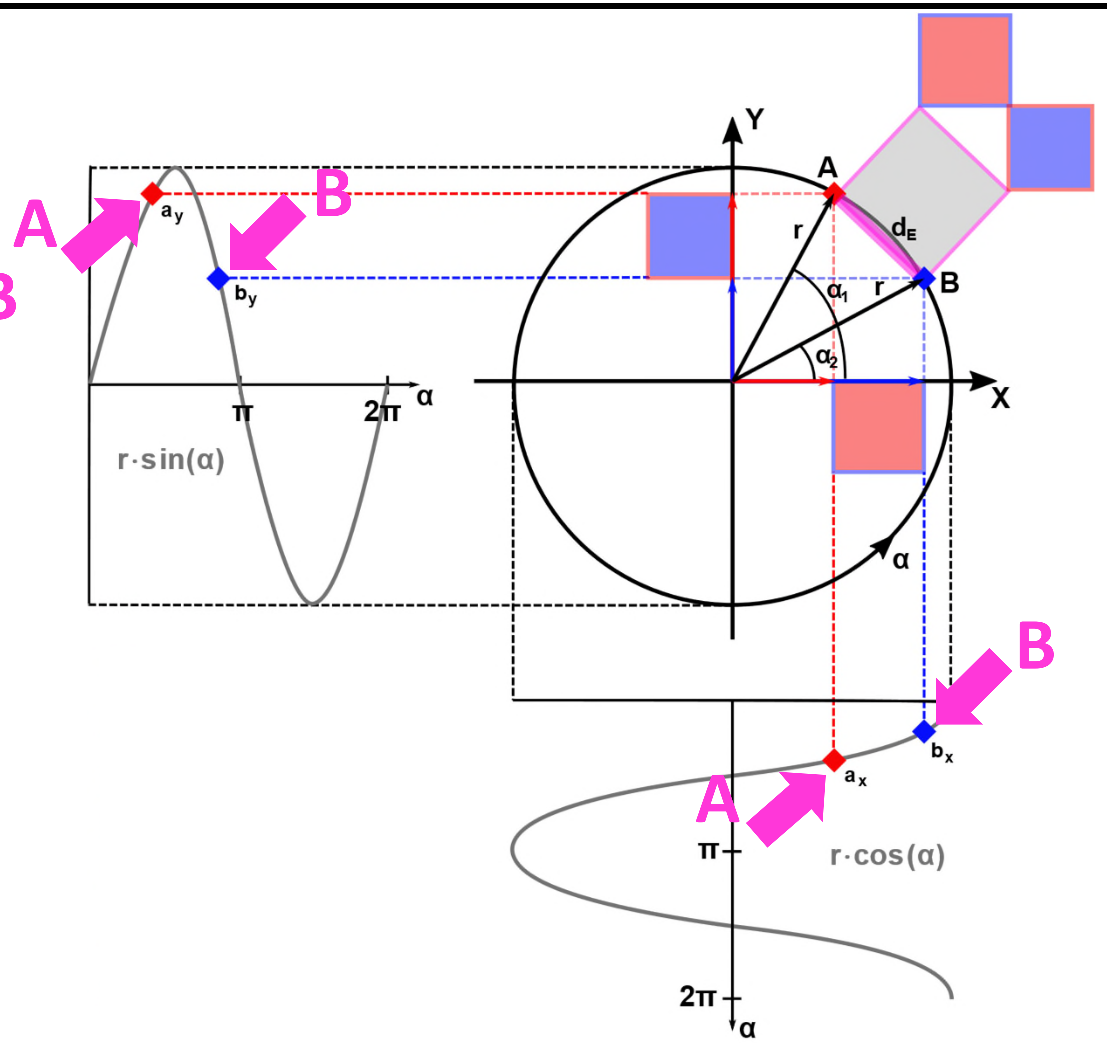




$d_x^2\langle \mathbf{A},\mathbf{B}\rangle$

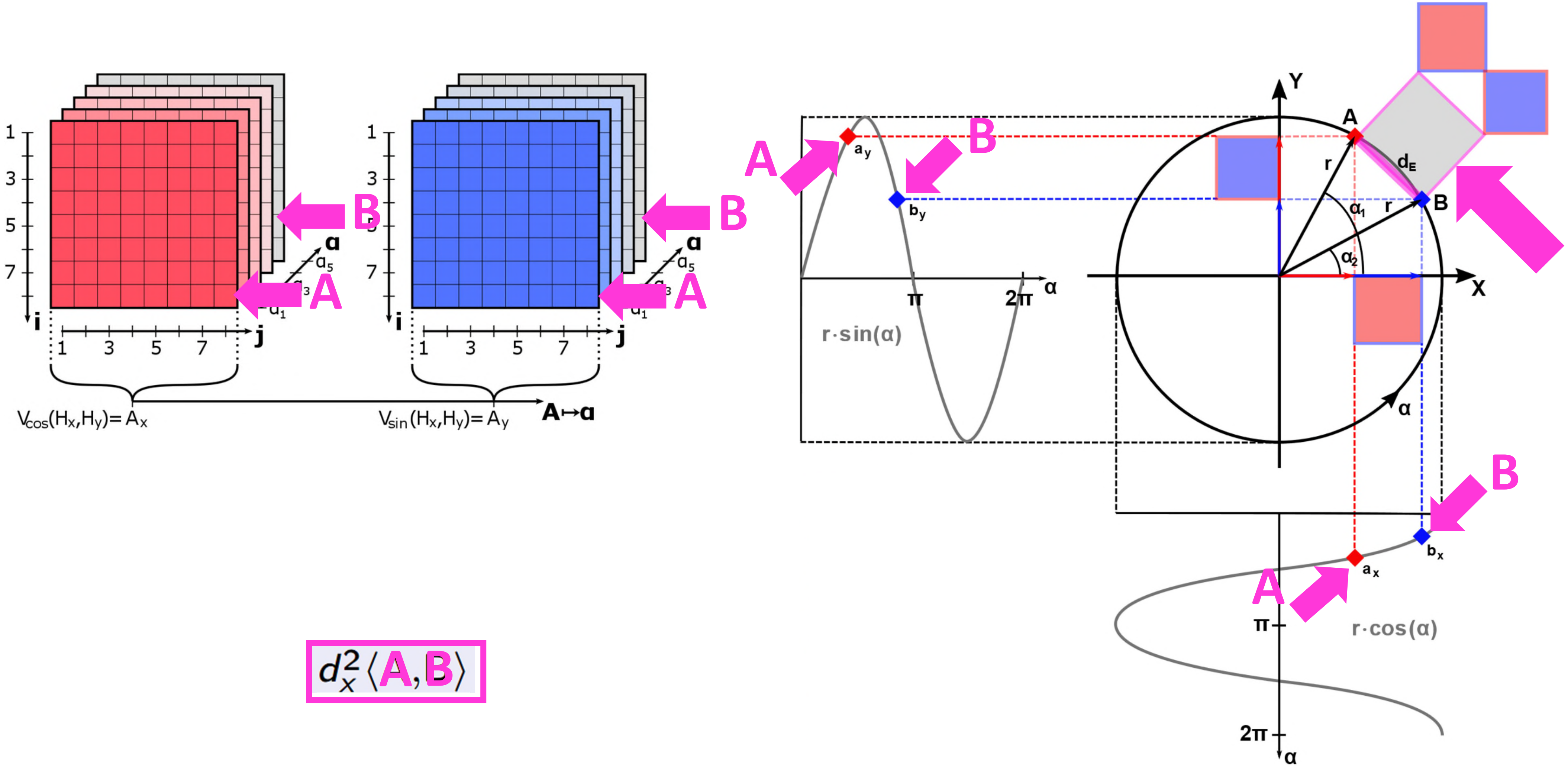
Gaussian Processes





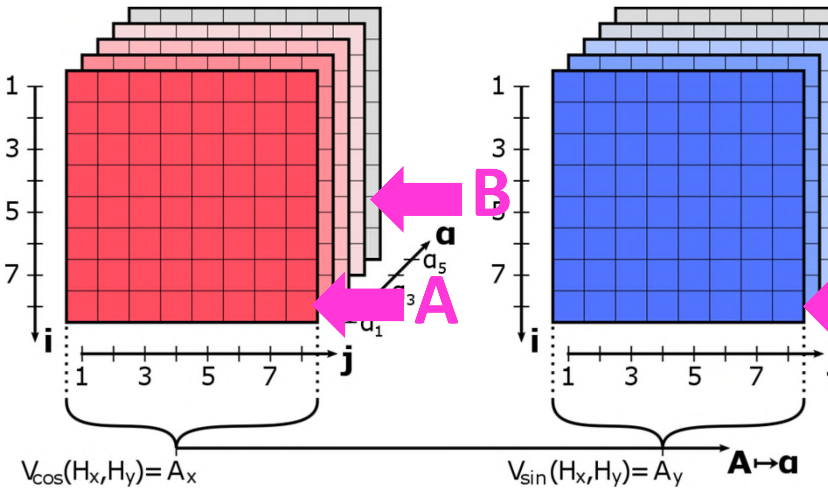
$d_x^2\langle A,B\rangle$

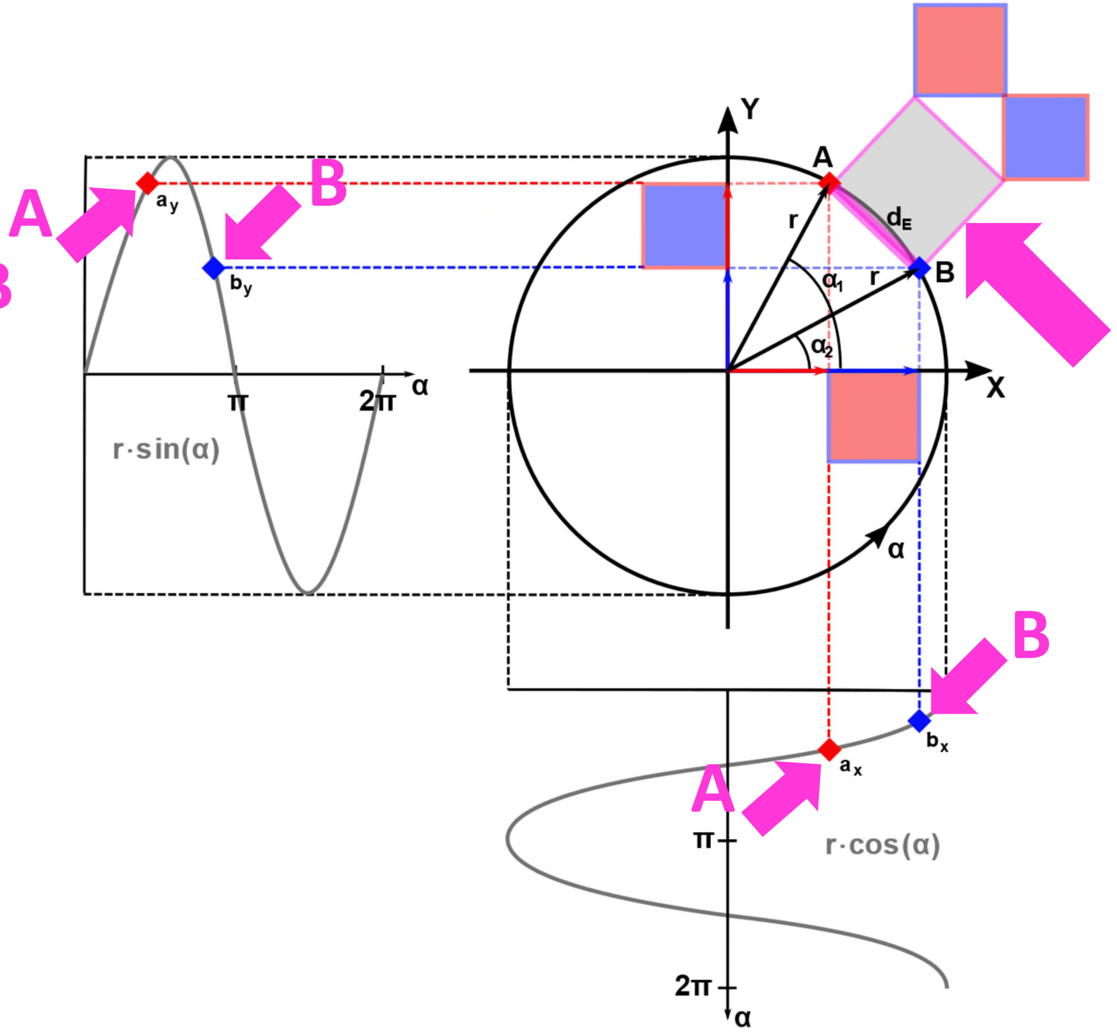
Gaussian Processes



$d_x^2\langle A,B\rangle$

Gaussian Processes



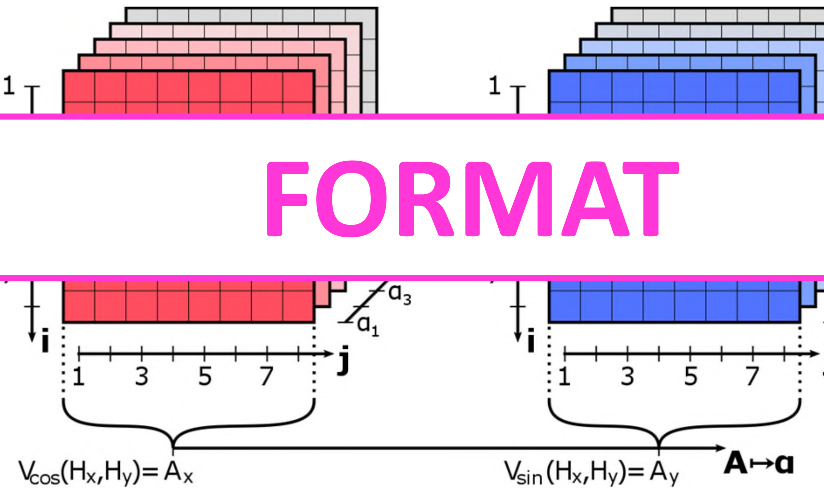


Fractional Kernel

$$\frac{\sigma_f^2}{1 + \boxed{\frac{d_x^2\langle A,B\rangle}{2\sigma_l^2}}}$$

Gaussian Processes

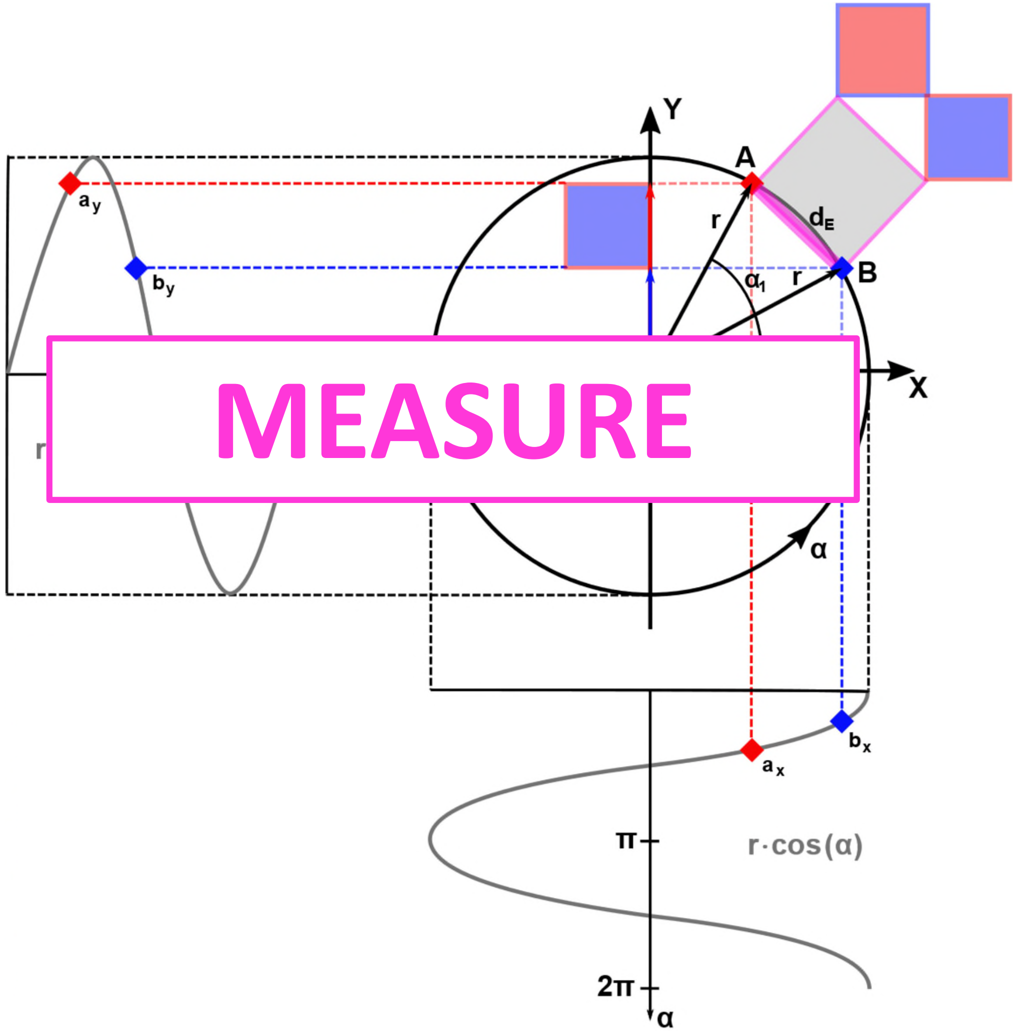
FORMAT



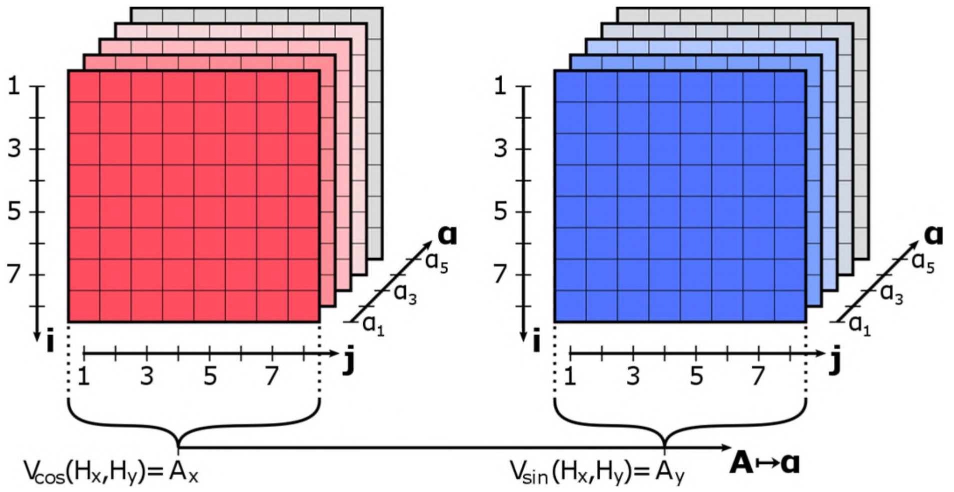
VALUE

$$\frac{\sigma_f^2}{1 + \frac{d_x^2\langle \mathbf{A},\mathbf{B}\rangle}{2\sigma_l^2}}$$

MEASURE

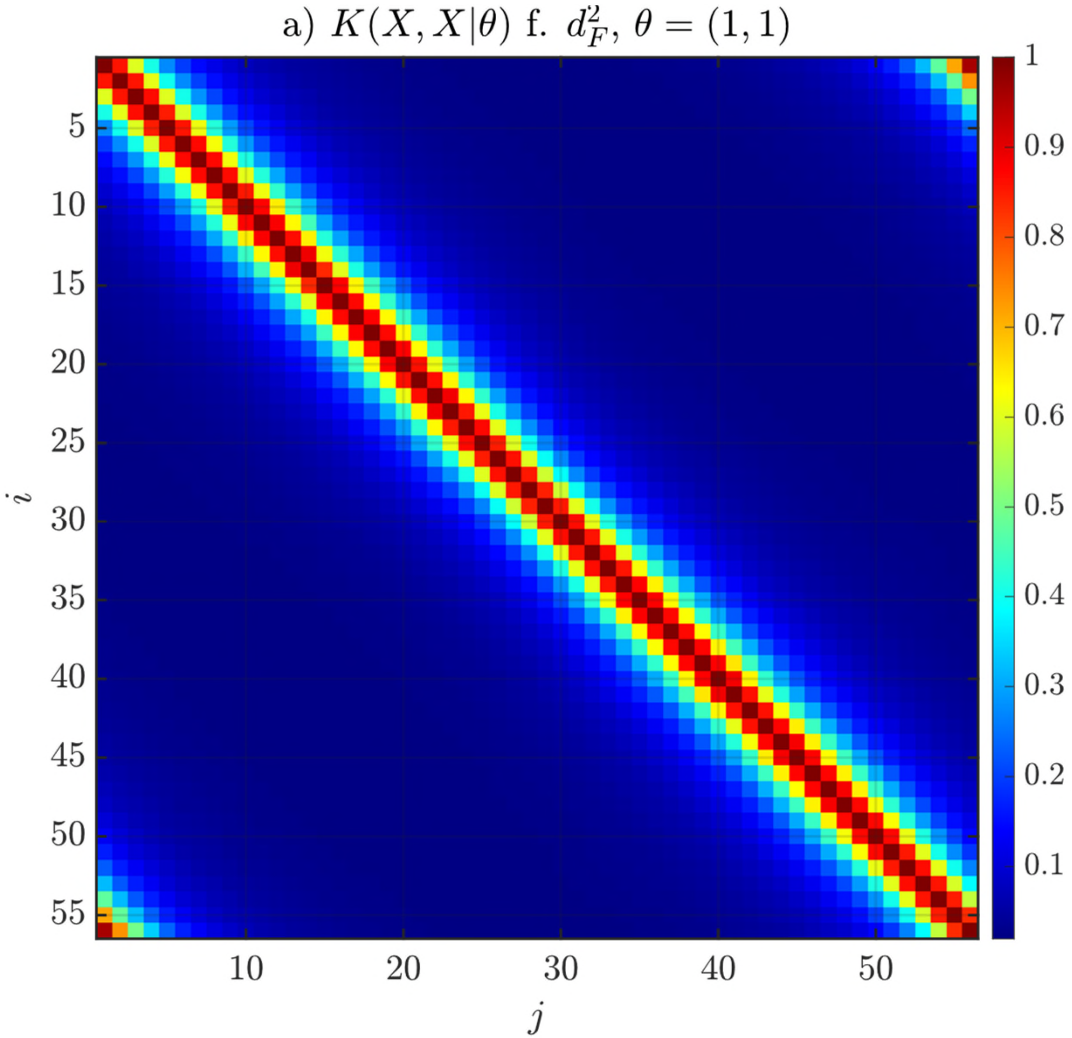


Gaussian Processes

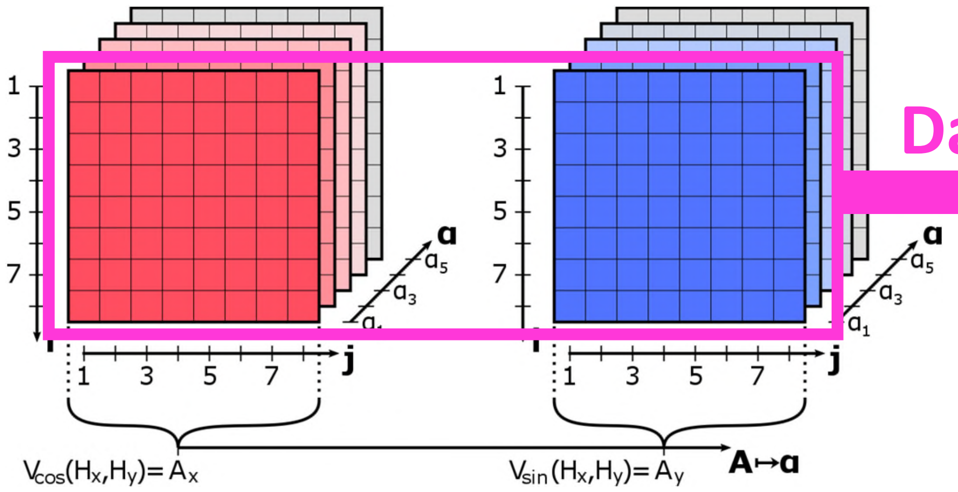


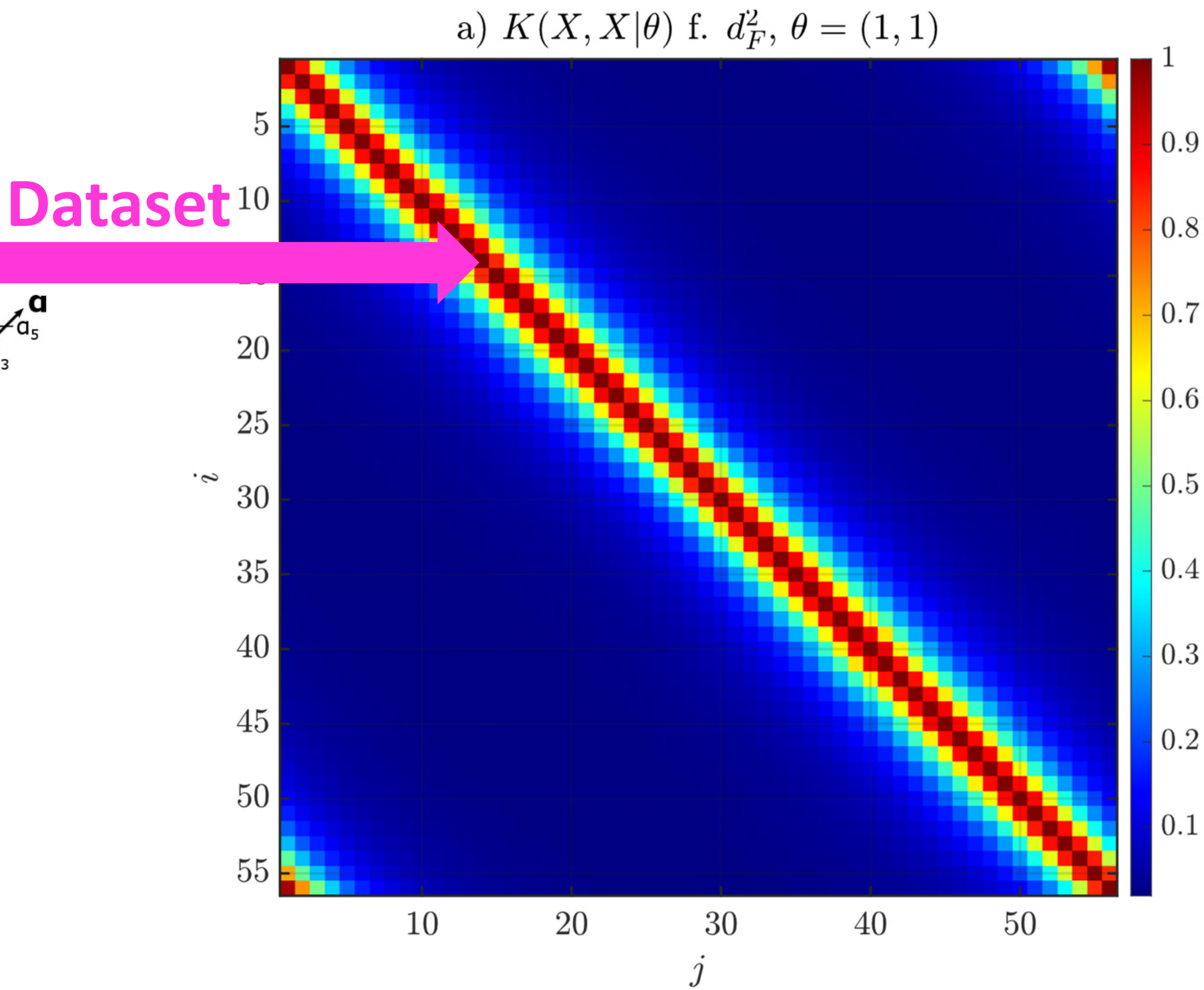
Fractional Kernel

$$\frac{\sigma_f^2}{1 + \frac{d_x^2\langle \mathbf{A},\mathbf{B}\rangle}{2\sigma_l^2}}$$



Gaussian Processes

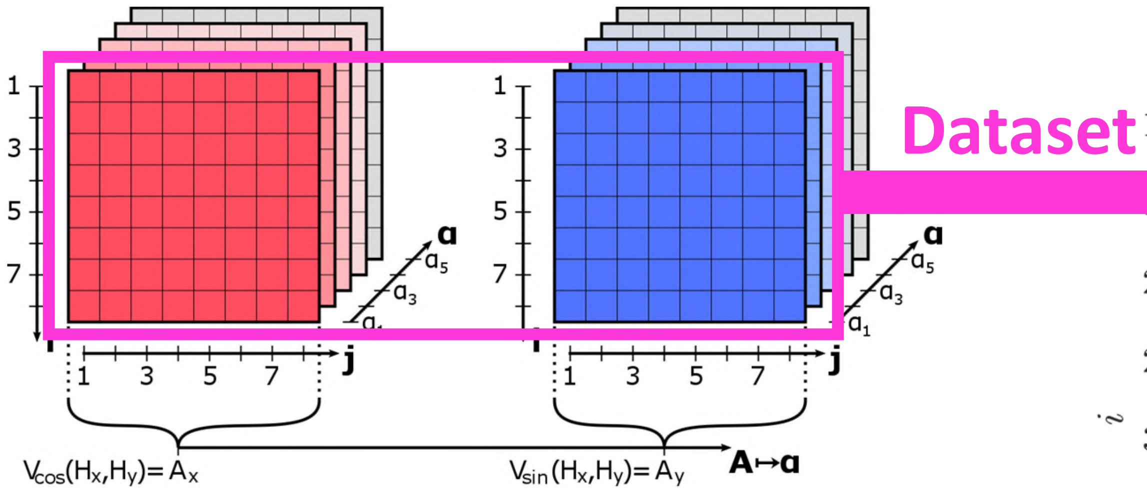




Fractional Kernel

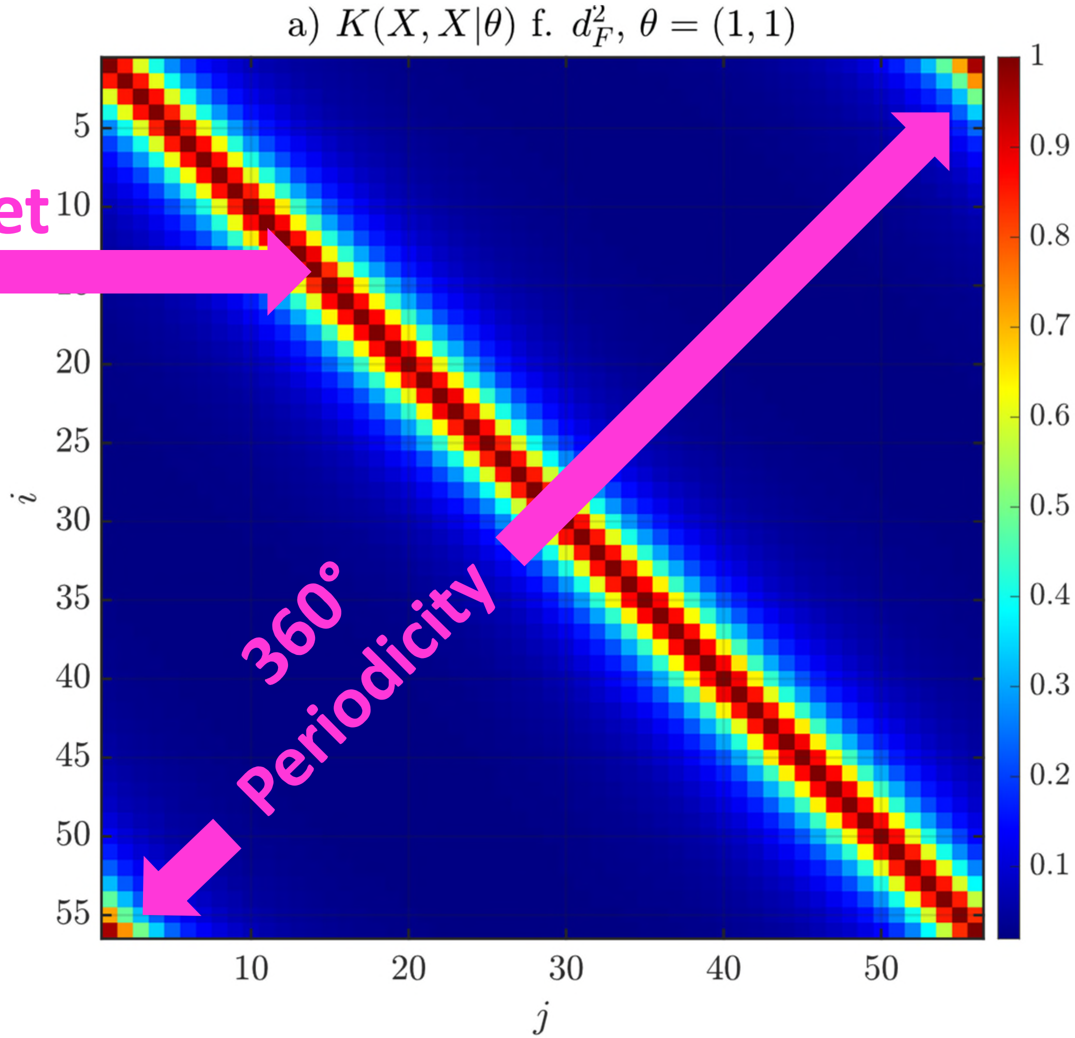
$$\frac{\sigma_f^2}{1 + \frac{d_x^2\langle \mathbf{A},\mathbf{B}\rangle}{2\sigma_l^2}}$$

Gaussian Processes

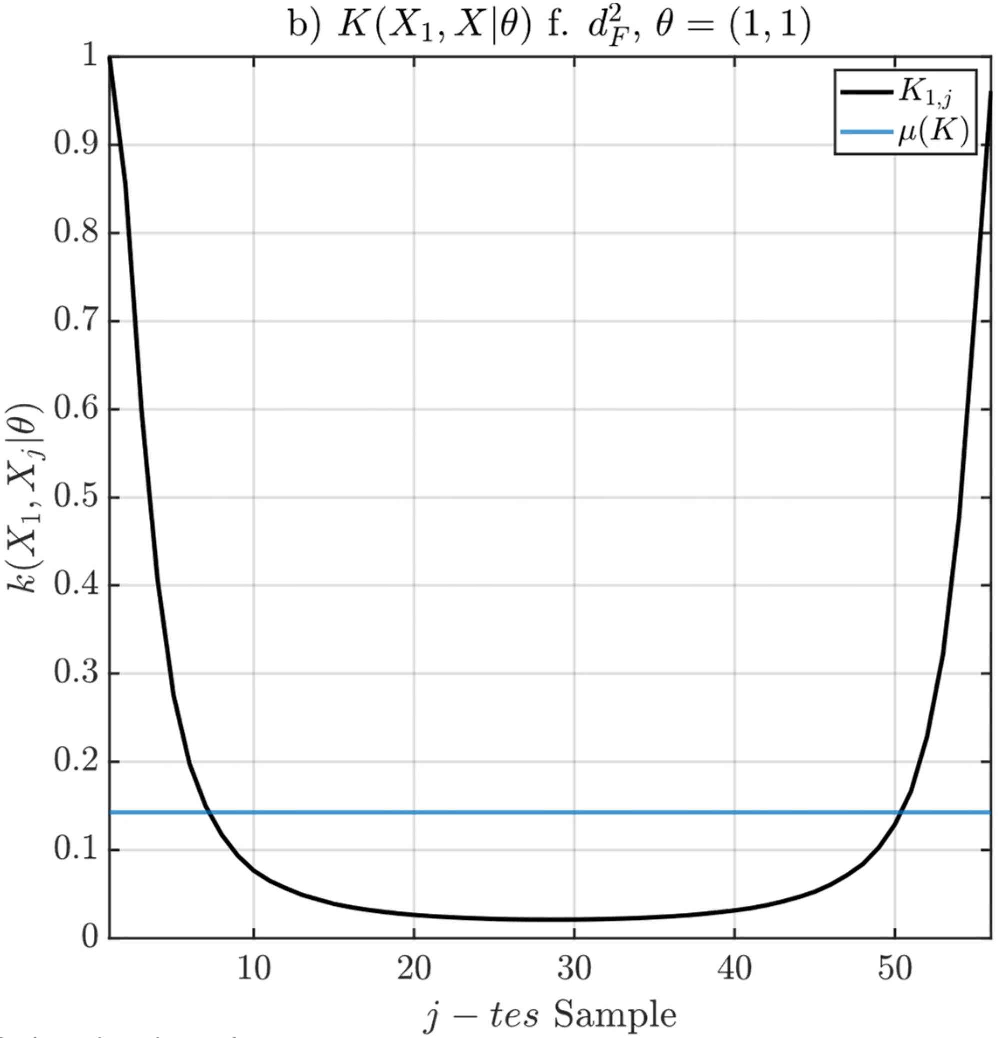


Fractional Kernel

$$\frac{\sigma_f^2}{1 + \frac{d_x^2\langle \mathbf{A},\mathbf{B}\rangle}{2\sigma_l^2}}$$



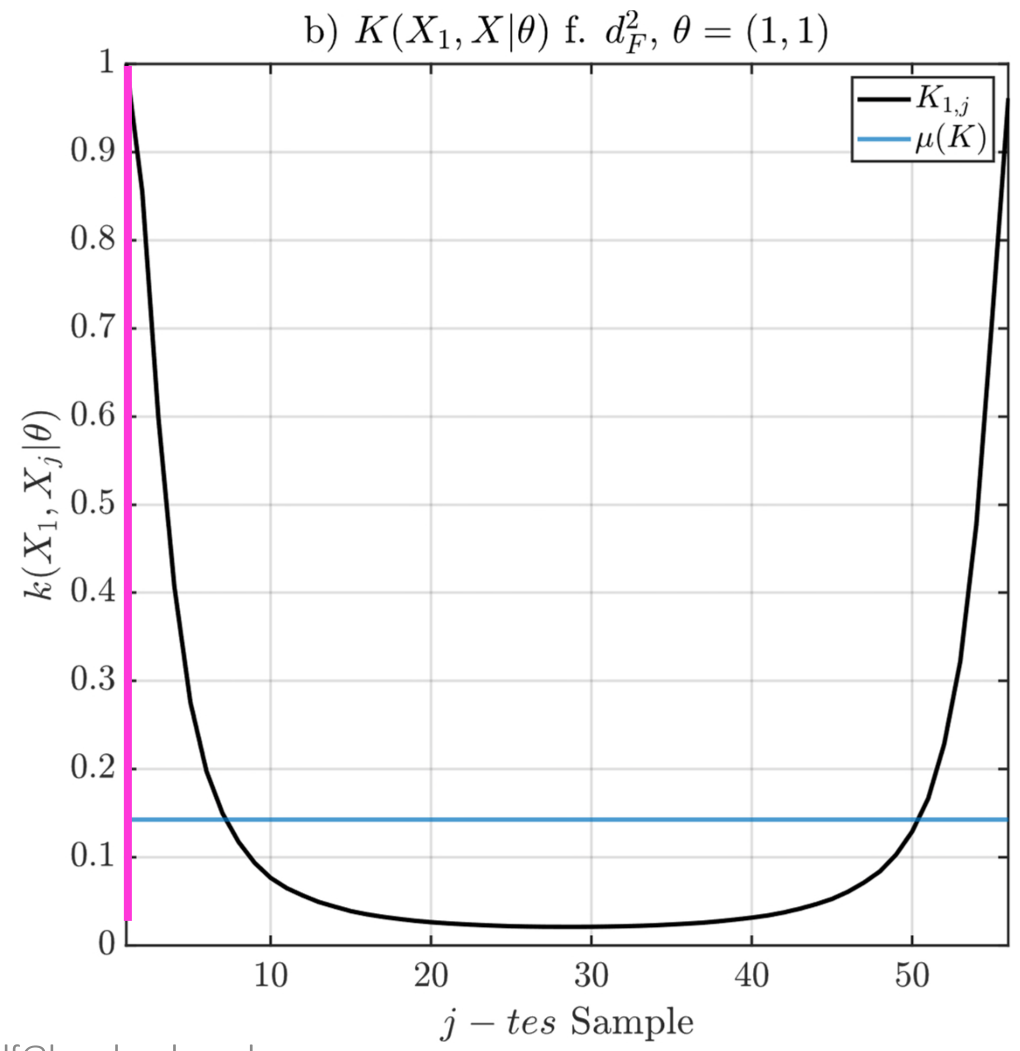
Gaussian Processes



Fractional Kernel

$$\frac{\sigma_f^2}{1 + \frac{d_x^2\langle \mathbf{A},\mathbf{B}\rangle}{2\sigma_l^2}}$$

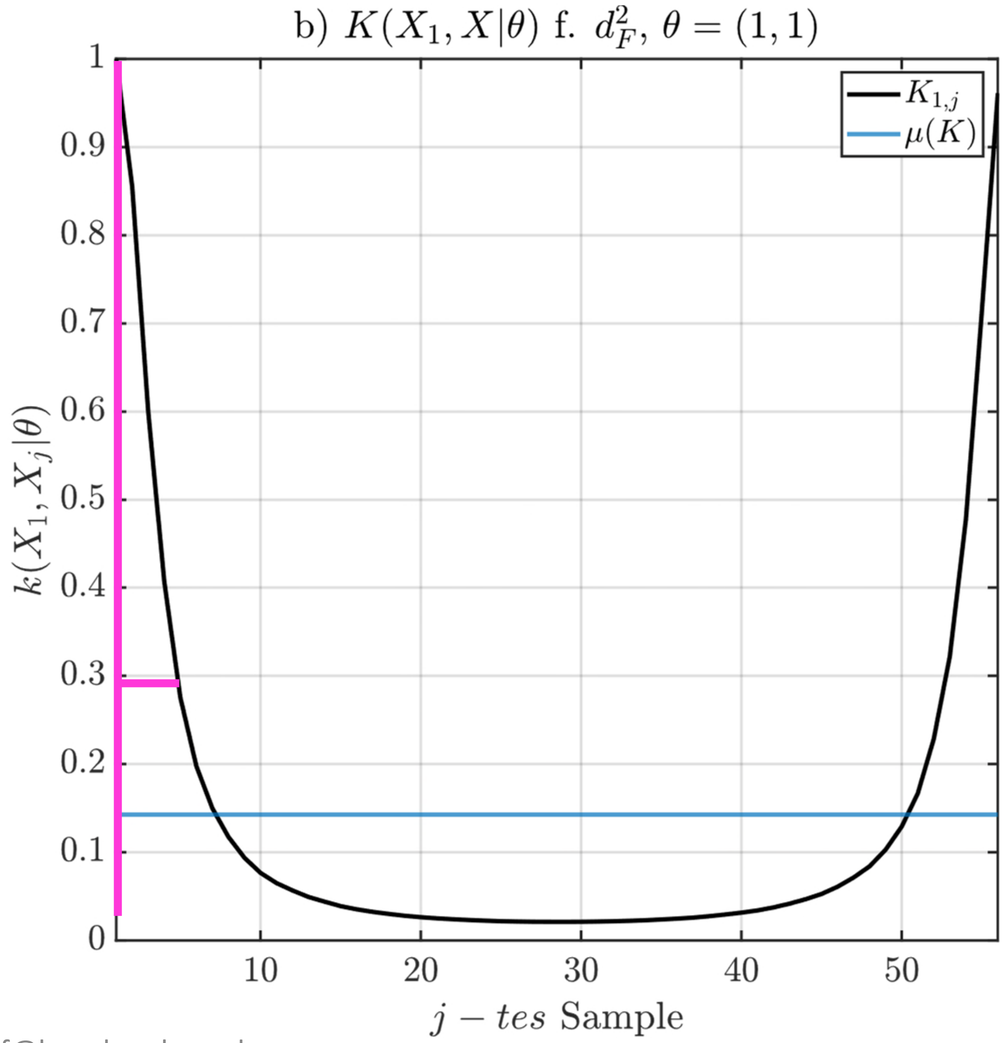
Gaussian Processes



Fractional Kernel

$$\frac{\boxed{\sigma_f^2}}{1 + \frac{d_x^2\langle \mathbf{A},\mathbf{B}\rangle}{2\sigma_l^2}}$$

Gaussian Processes

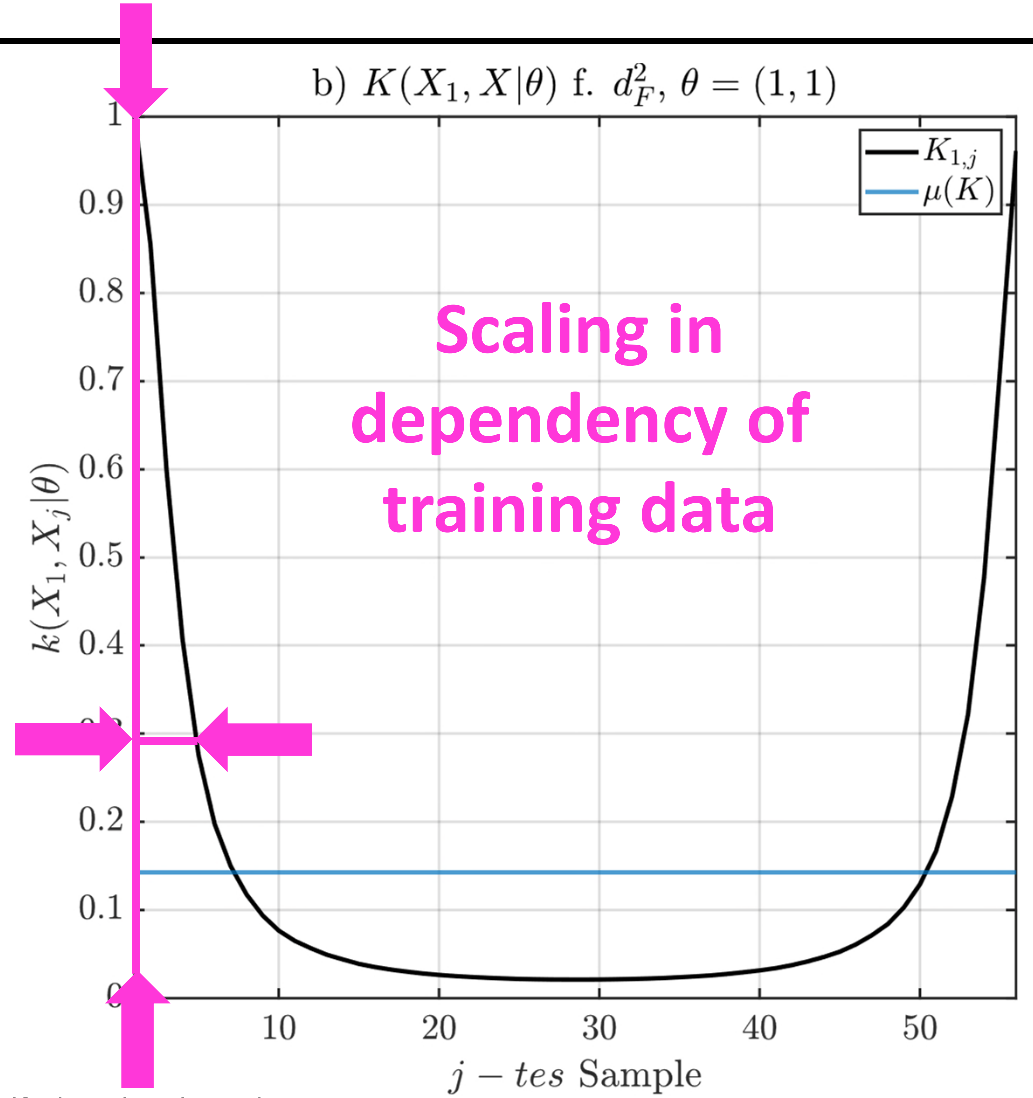


Fractional Kernel

$$\frac{\boxed{\sigma_f^2}}{1 + \frac{d_x^2\langle \mathbf{A},\mathbf{B}\rangle}{2\boxed{\sigma_l^2}}}$$

Gaussian Processes

HAW HAMBURG



Fractional Kernel

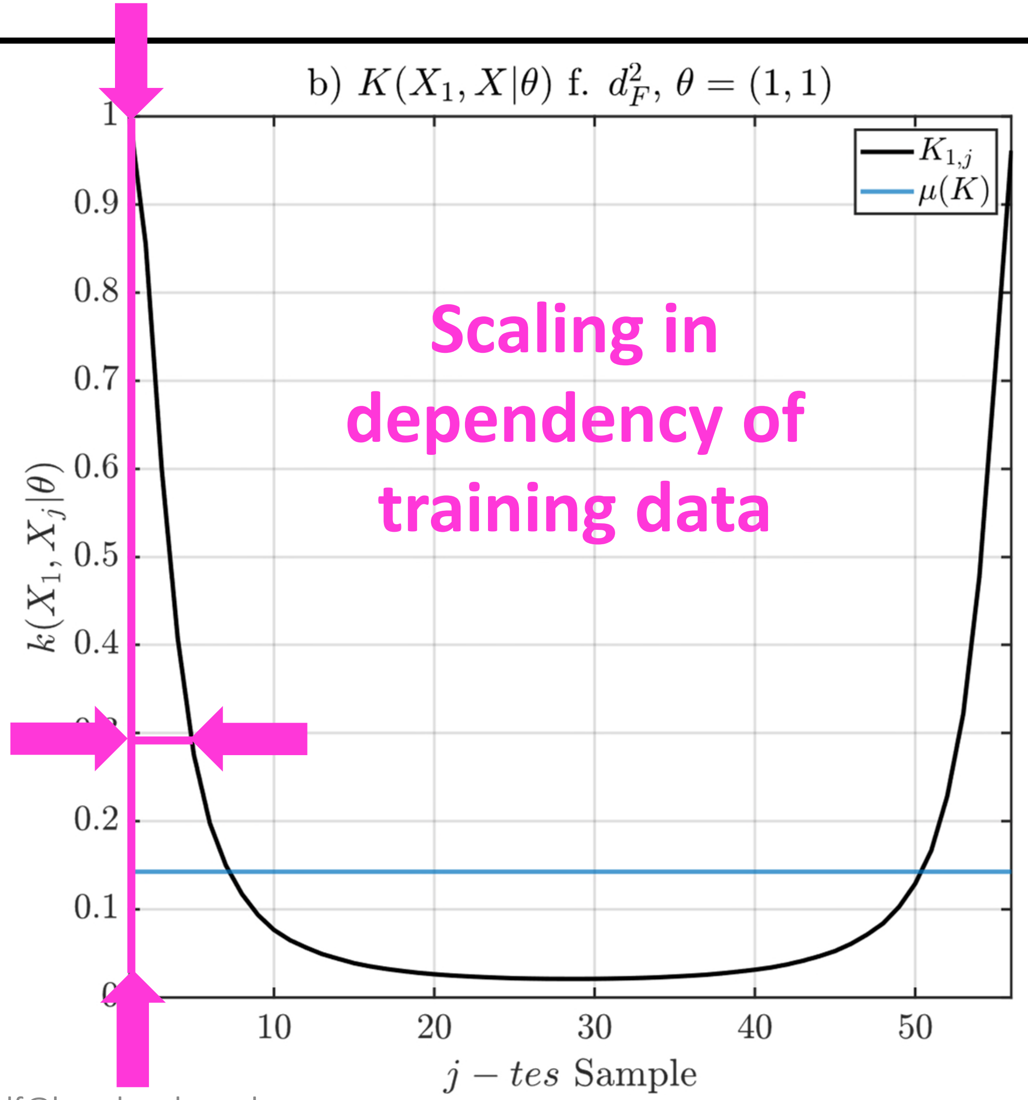
$$\frac{\boxed{\sigma_f^2}}{1 + \frac{d_x^2\langle \mathbf{A},\mathbf{B}\rangle}{2\boxed{\sigma_l^2}}}$$

Gaussian Processes

- Control of influence

Fractional Kernel

$$\frac{\boxed{\sigma_f^2}}{1+\frac{d_x^2\langle \mathbf{A},\mathbf{B}\rangle}{2\boxed{\sigma_l^2}}}$$



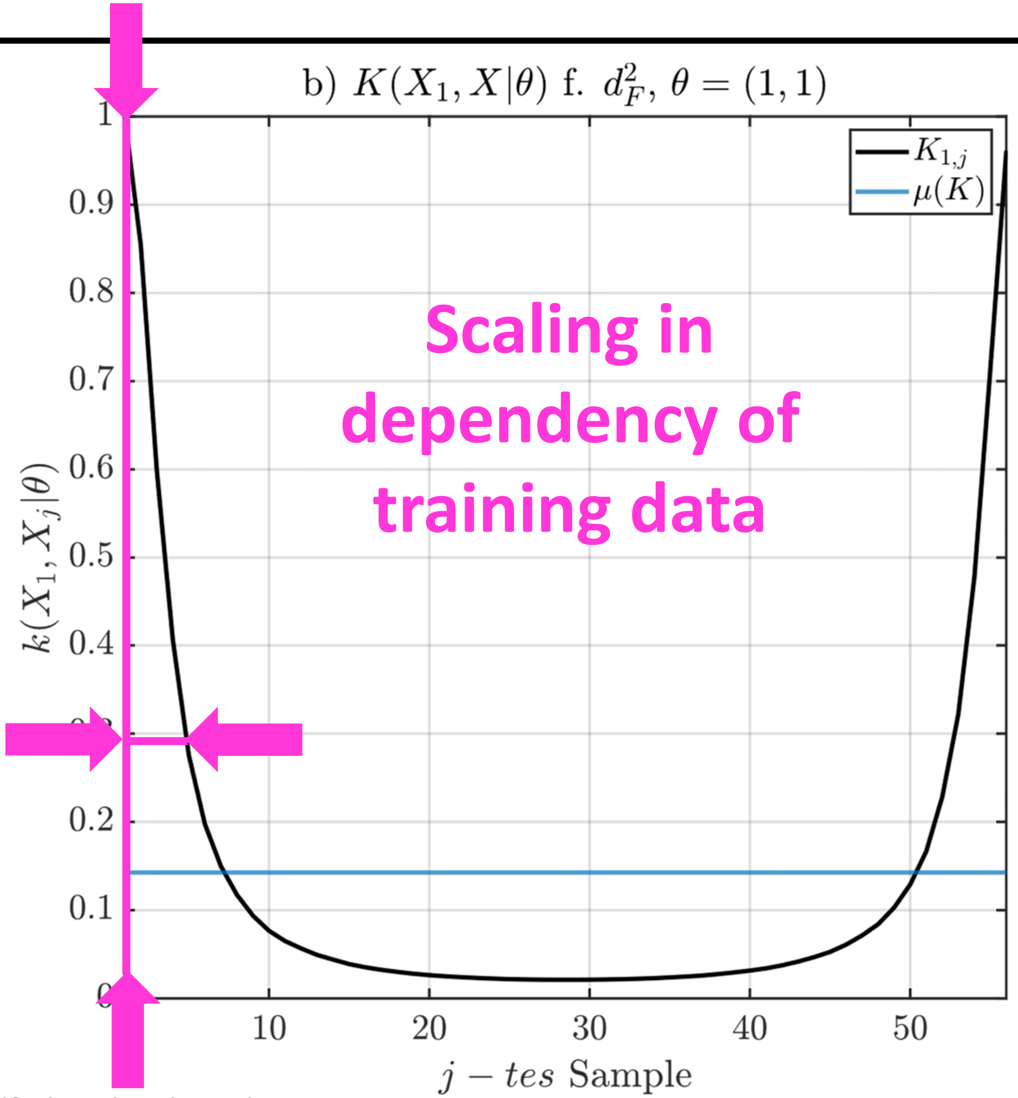
Gaussian Processes

HAW HAMBURG

- Control of influence
- Data fit by model likelihoods

Fractional Kernel

$$\frac{\boxed{\sigma_f^2}}{1 + \frac{d_x^2\langle \mathbf{A},\mathbf{B}\rangle}{2\boxed{\sigma_l^2}}}$$

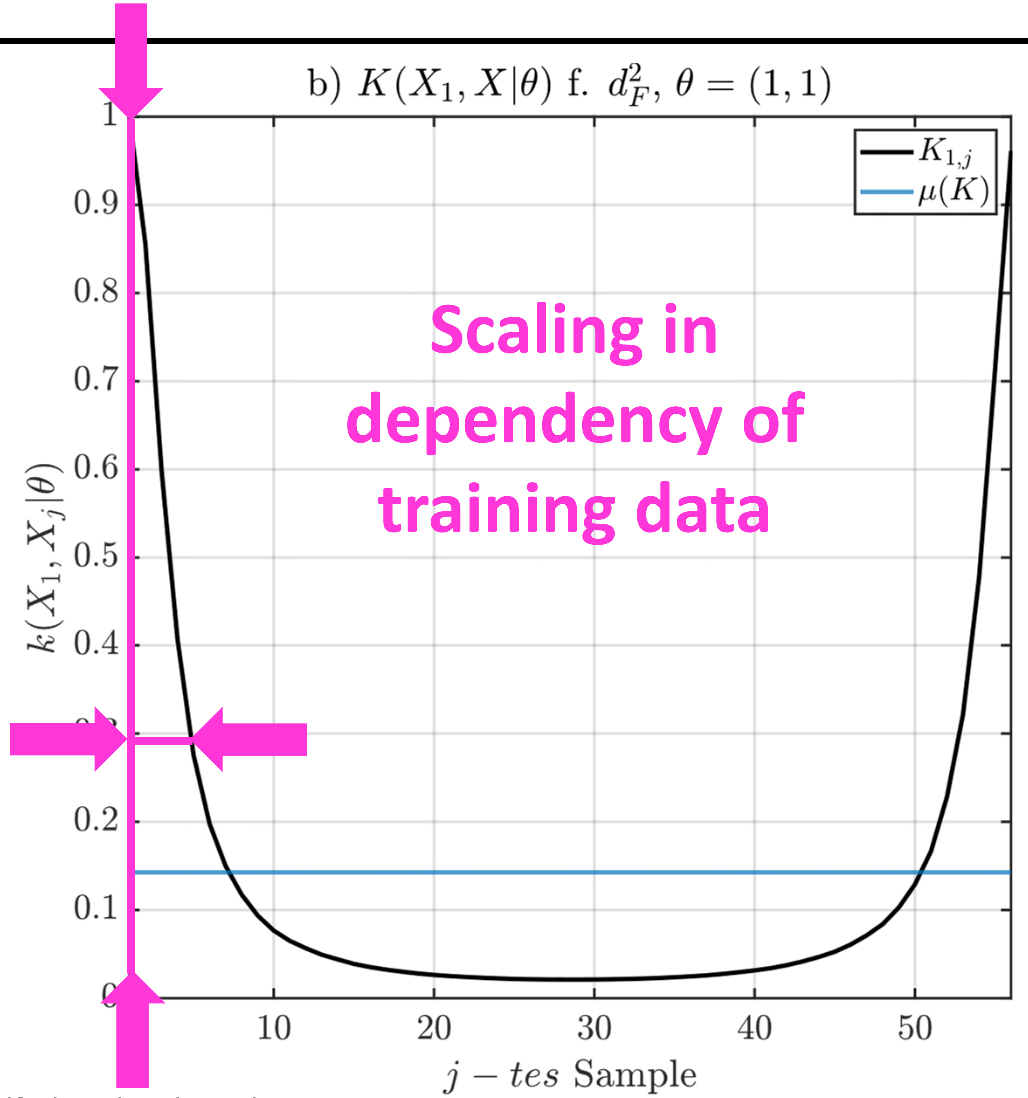


Gaussian Processes

- Control of influence
- Data fit by model likelihoods
- Model generalization by losses

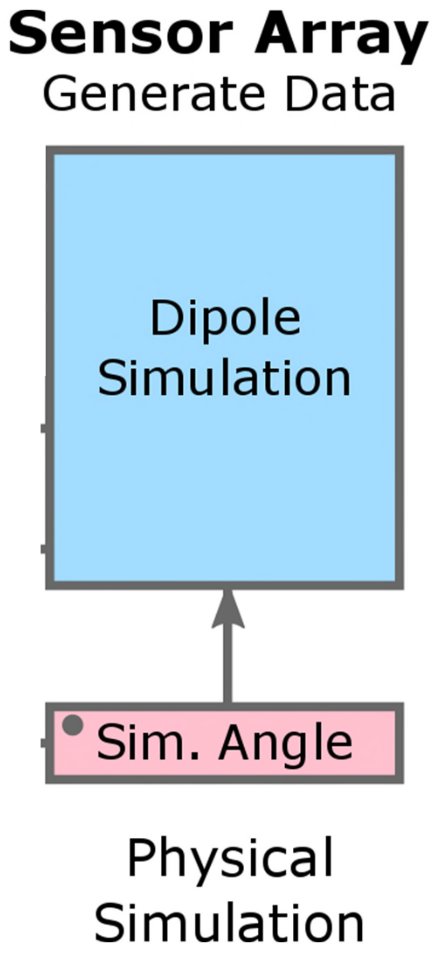
Fractional Kernel

$$\frac{\boxed{\sigma_f^2}}{1 + \frac{d_x^2\langle \mathbf{A},\mathbf{B}\rangle}{2\boxed{\sigma_l^2}}}$$



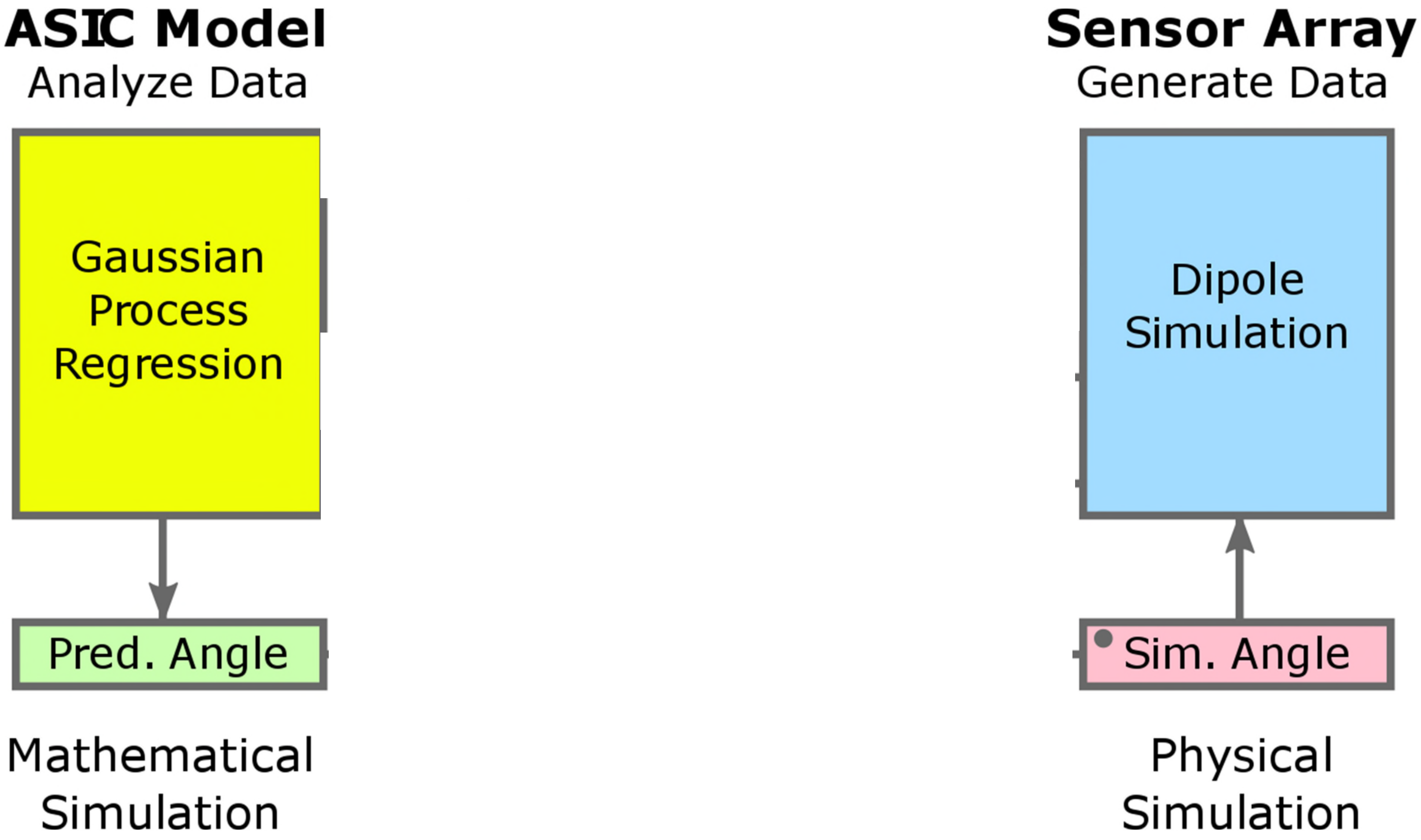
Software Development

Software Development

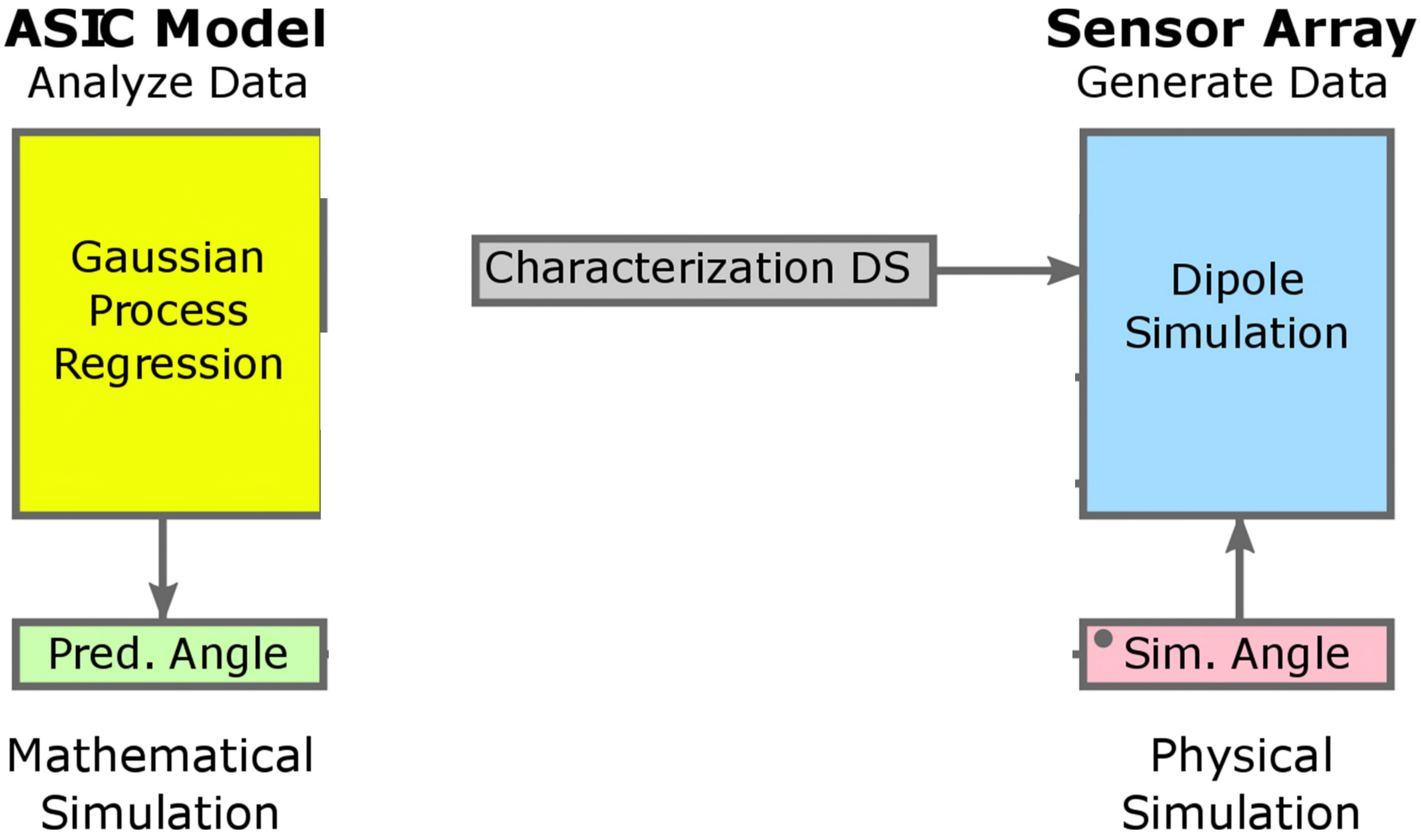


Software Development

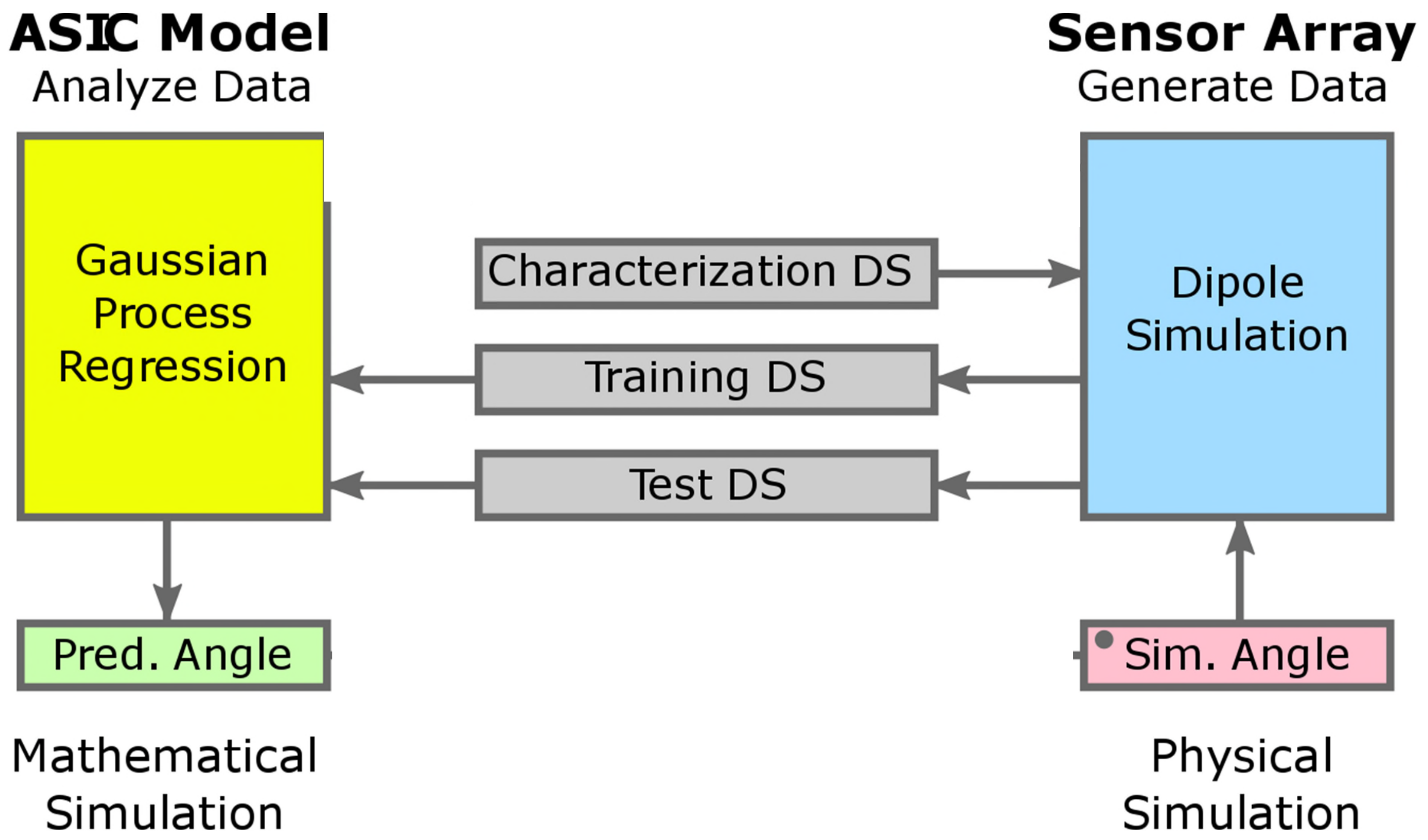
HAW HAMBURG



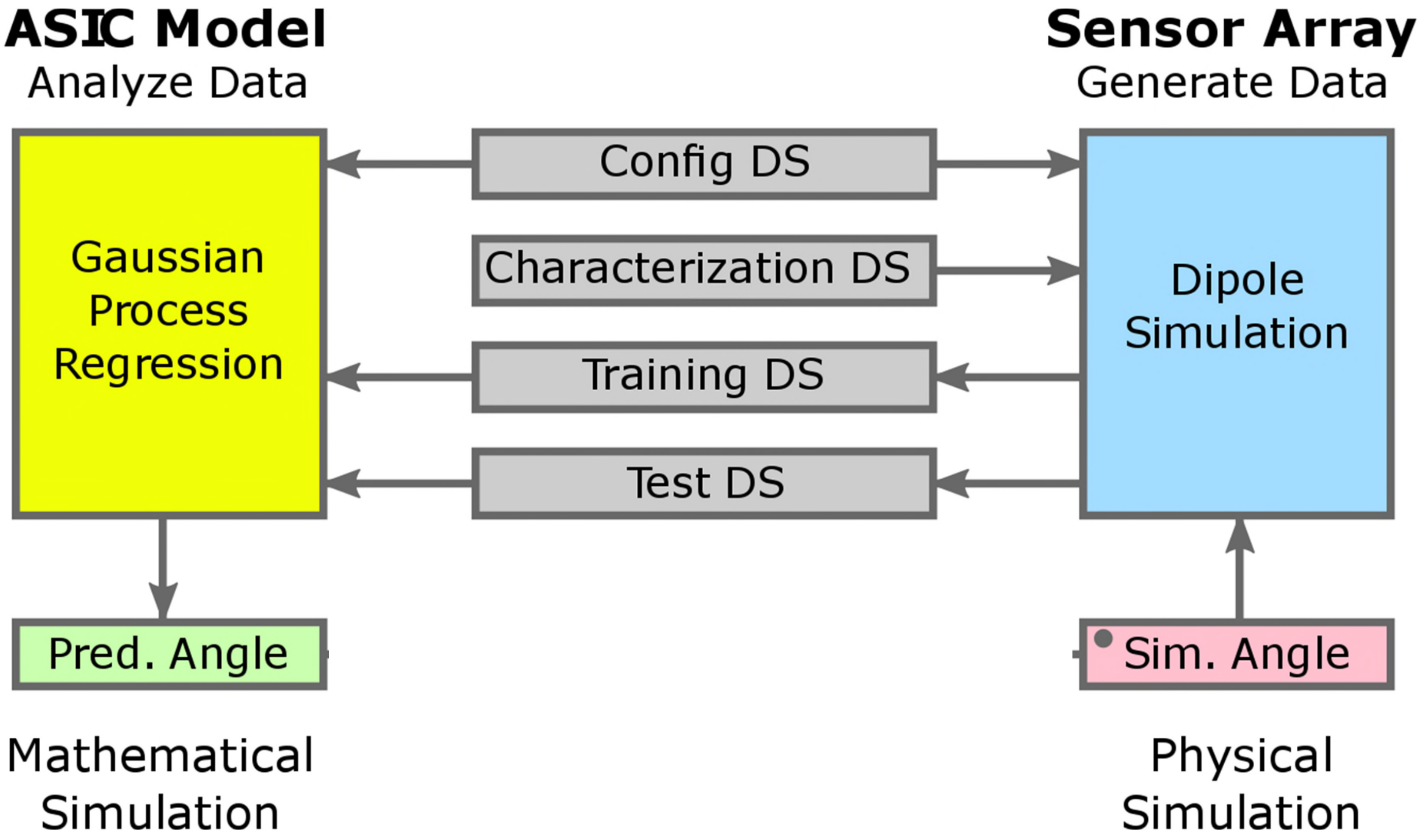
Software Development



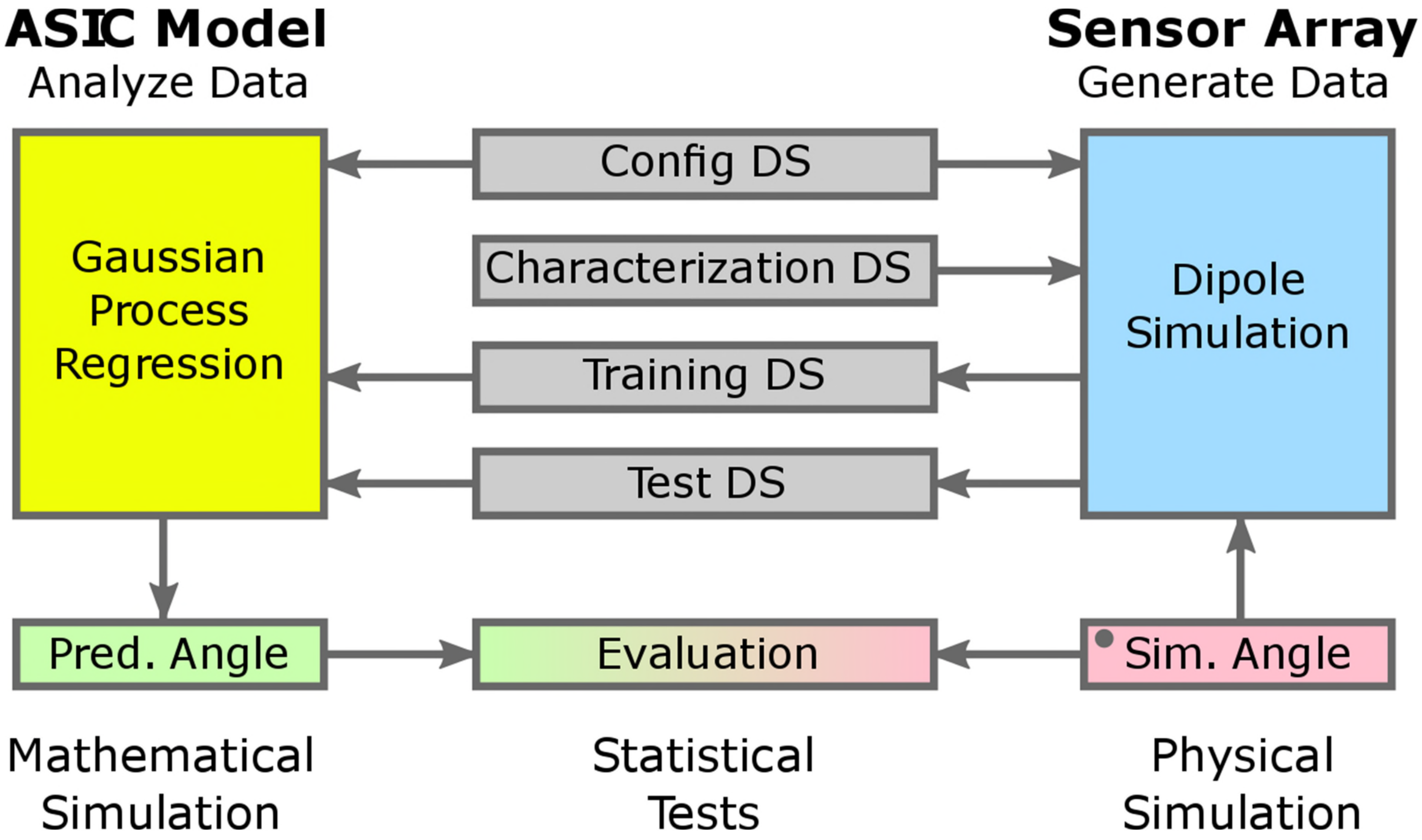
Software Development



Software Development

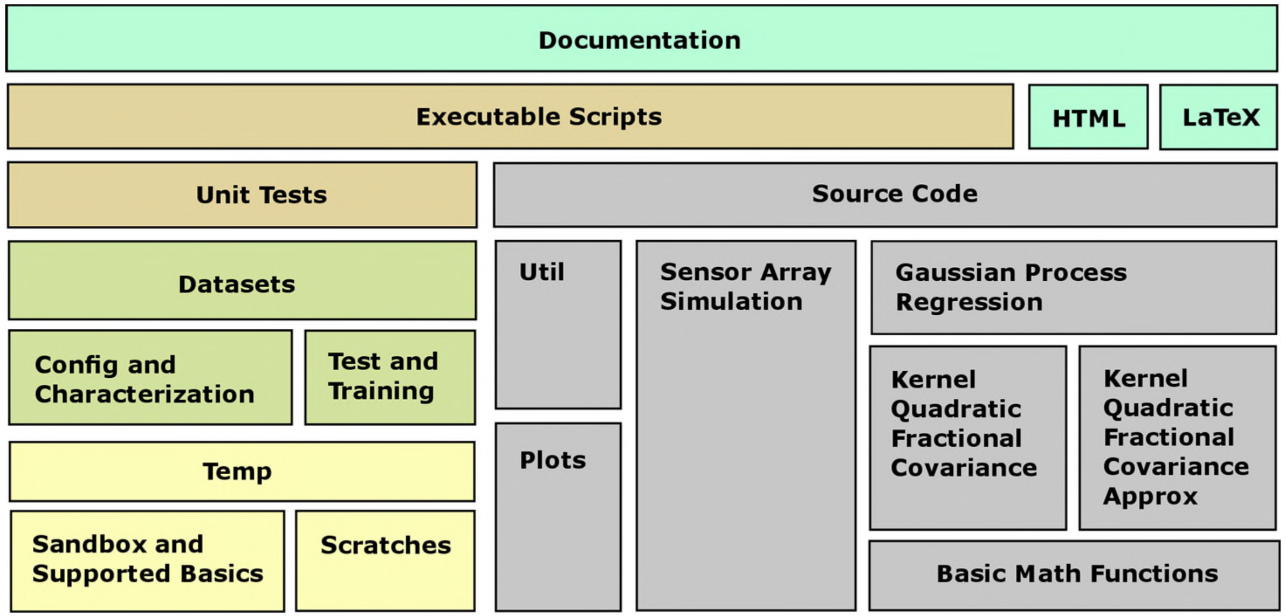


Software Development



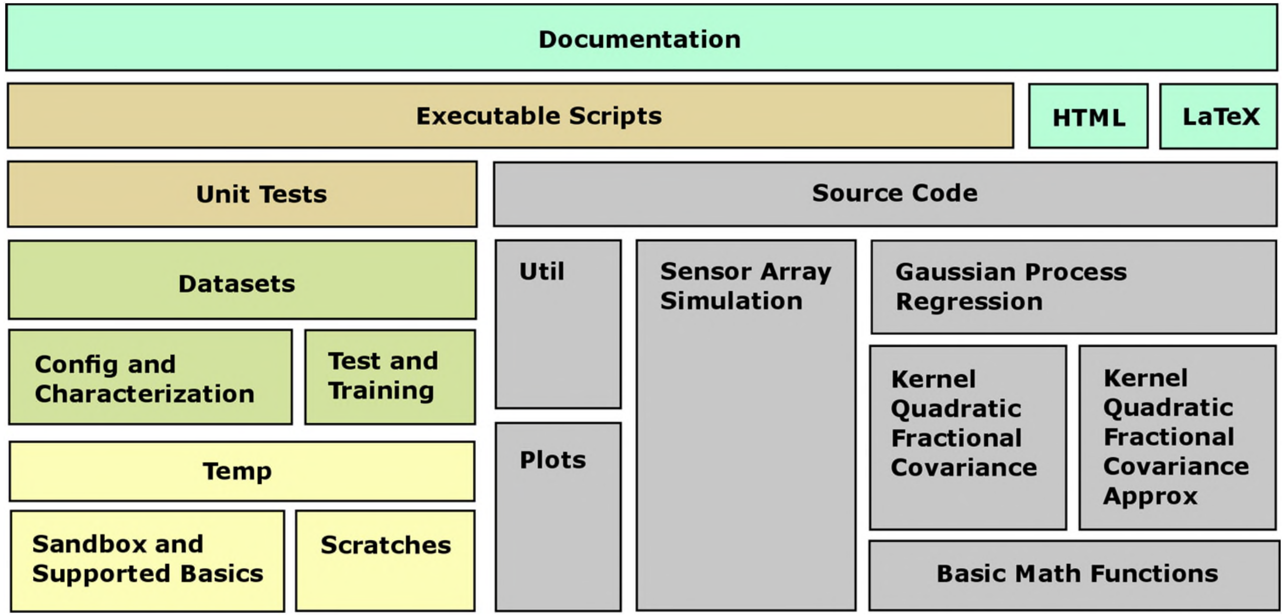
Software Development

- Modul driven



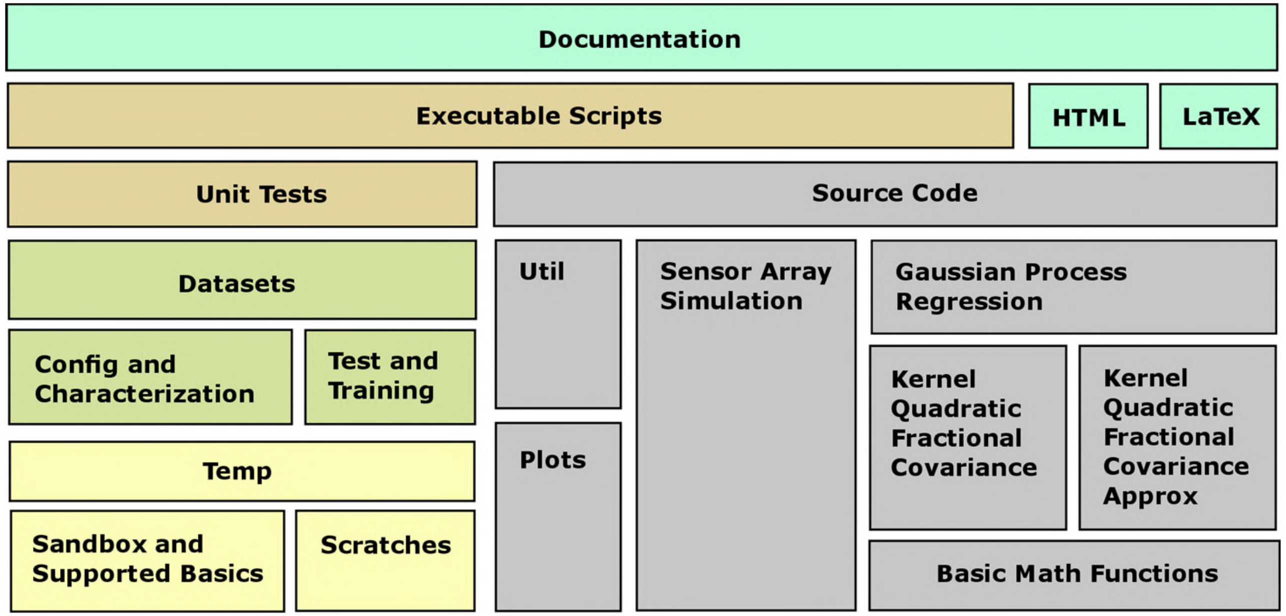
Software Development

- Modul driven
- Reusability
- Expandability

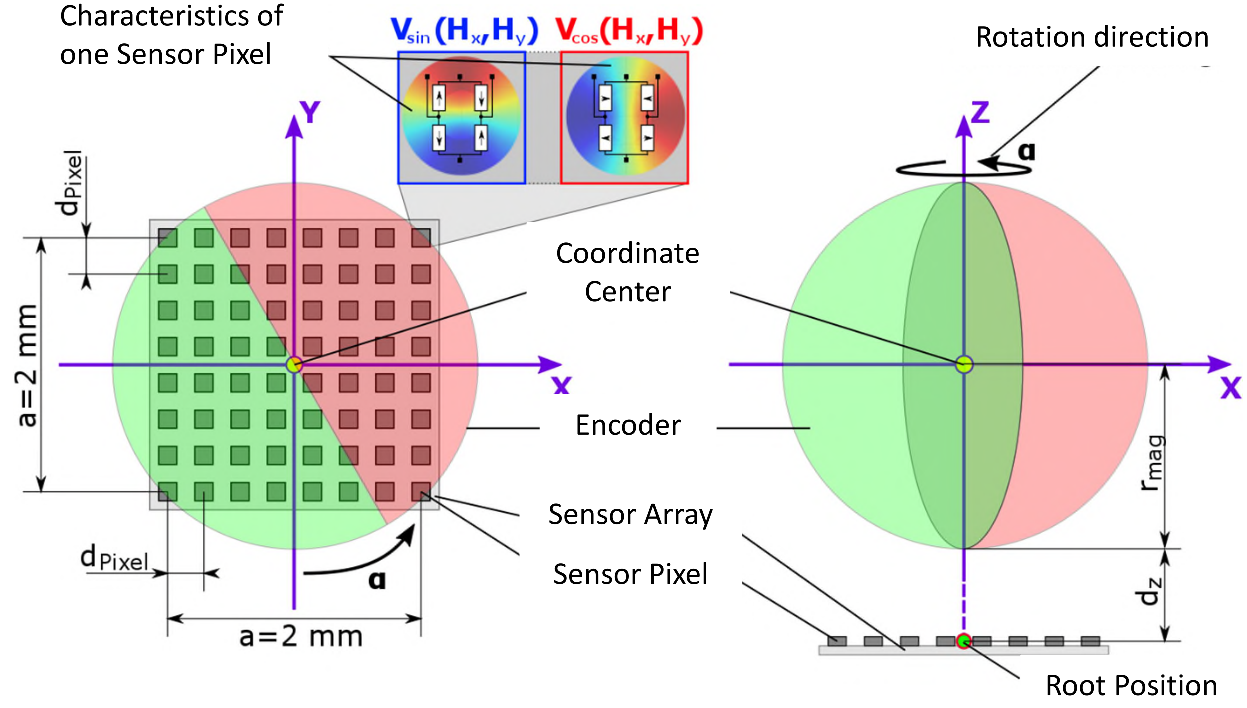


Software Development

- Modul driven
- Reusability
- Expandability
- Integration
- Documentation

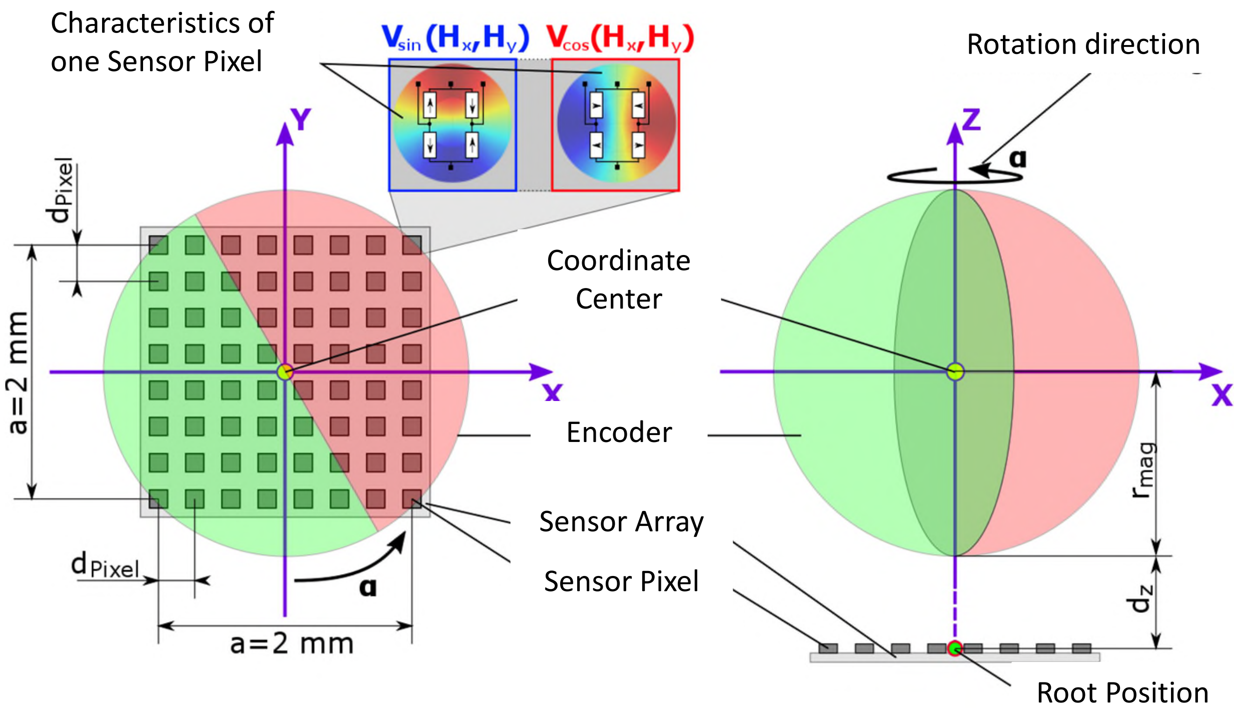


Simulation – Sensor Array



Simulation – Sensor Array

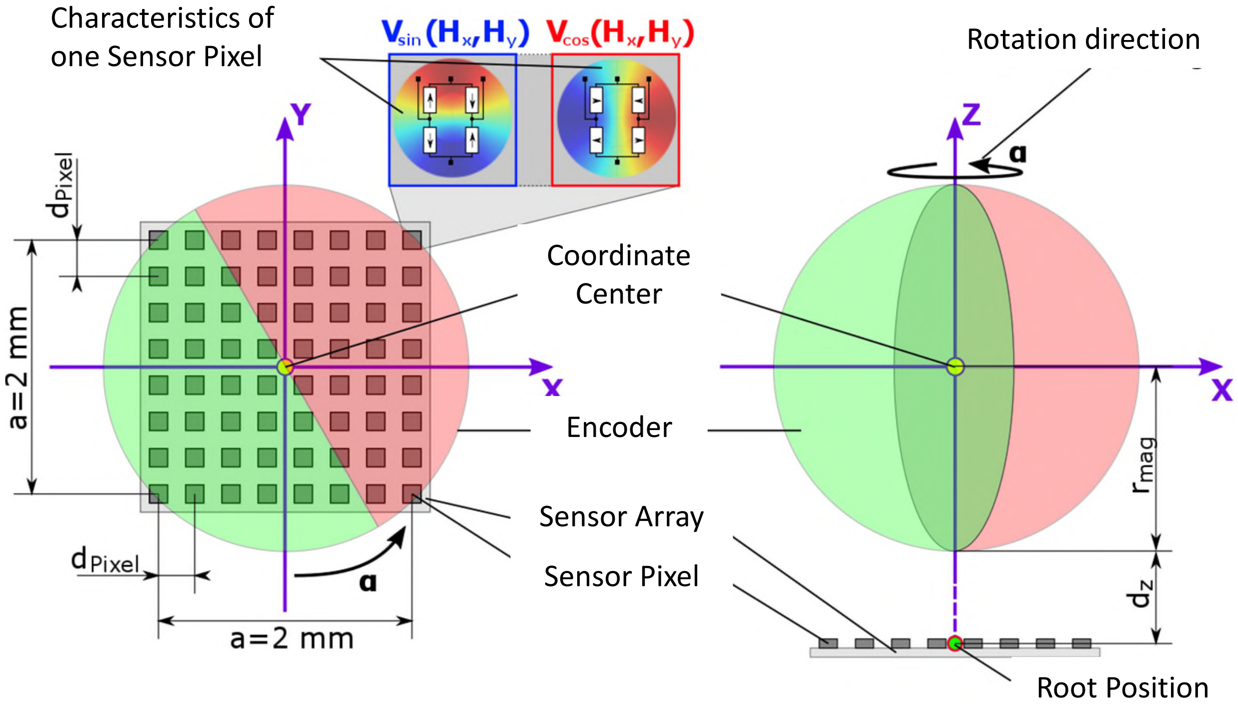
- Configuration of sizes, position, shape and voltage supply



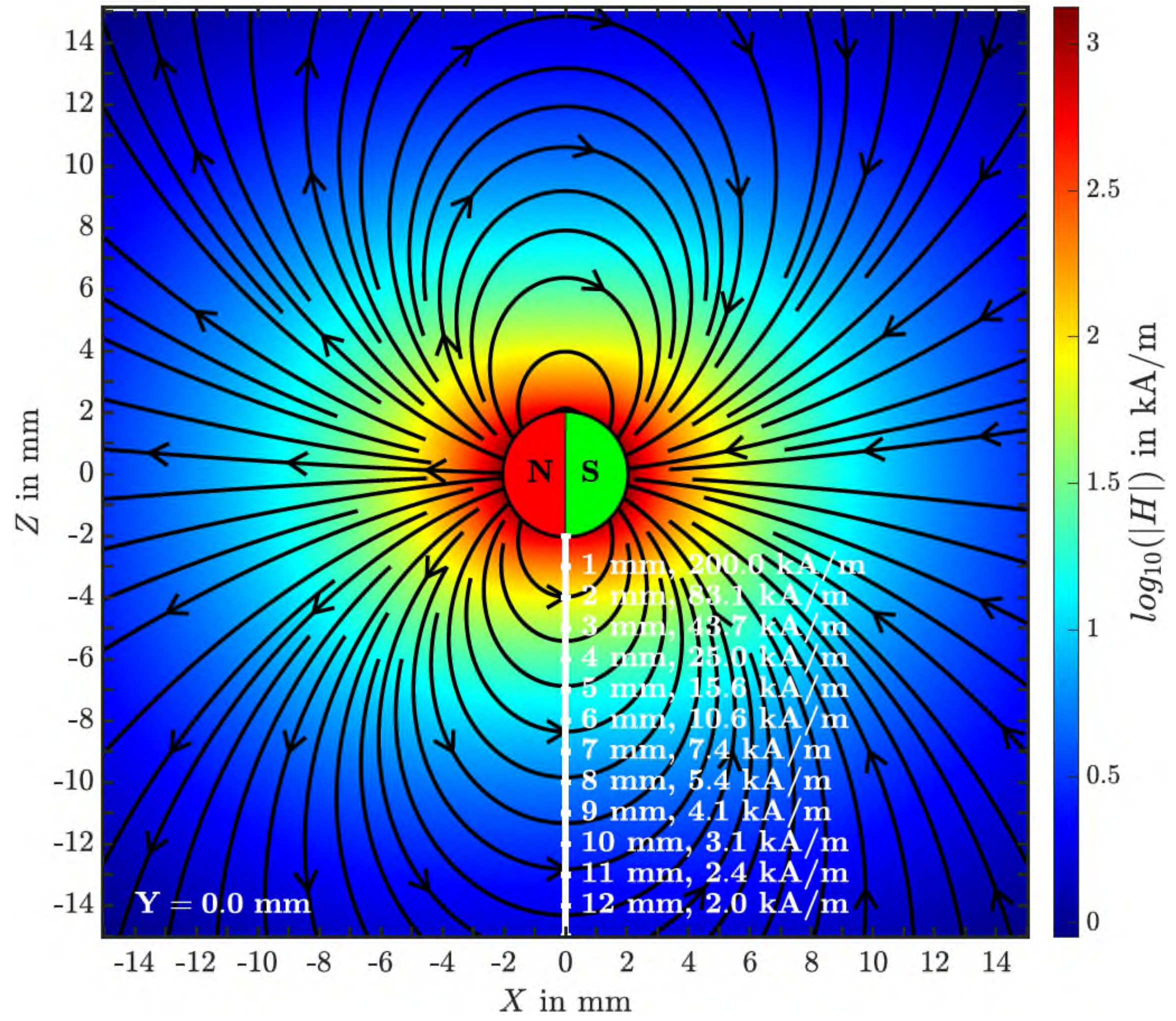
Simulation – Sensor Array

HAW HAMBURG

- Configuration of sizes, position, shape and voltage supply
- Determine start and rest position of the system

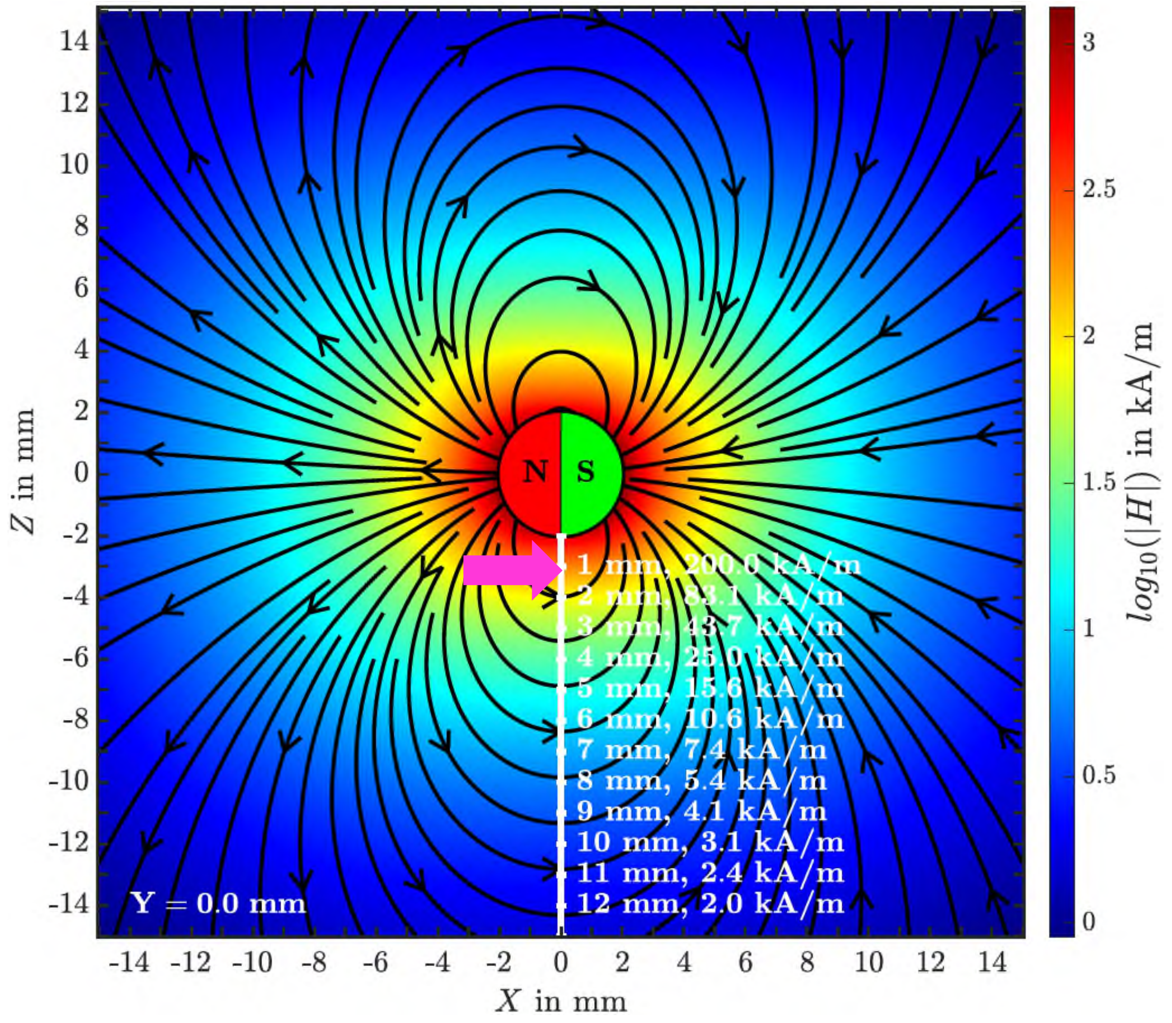


Simulation – Sensor Array



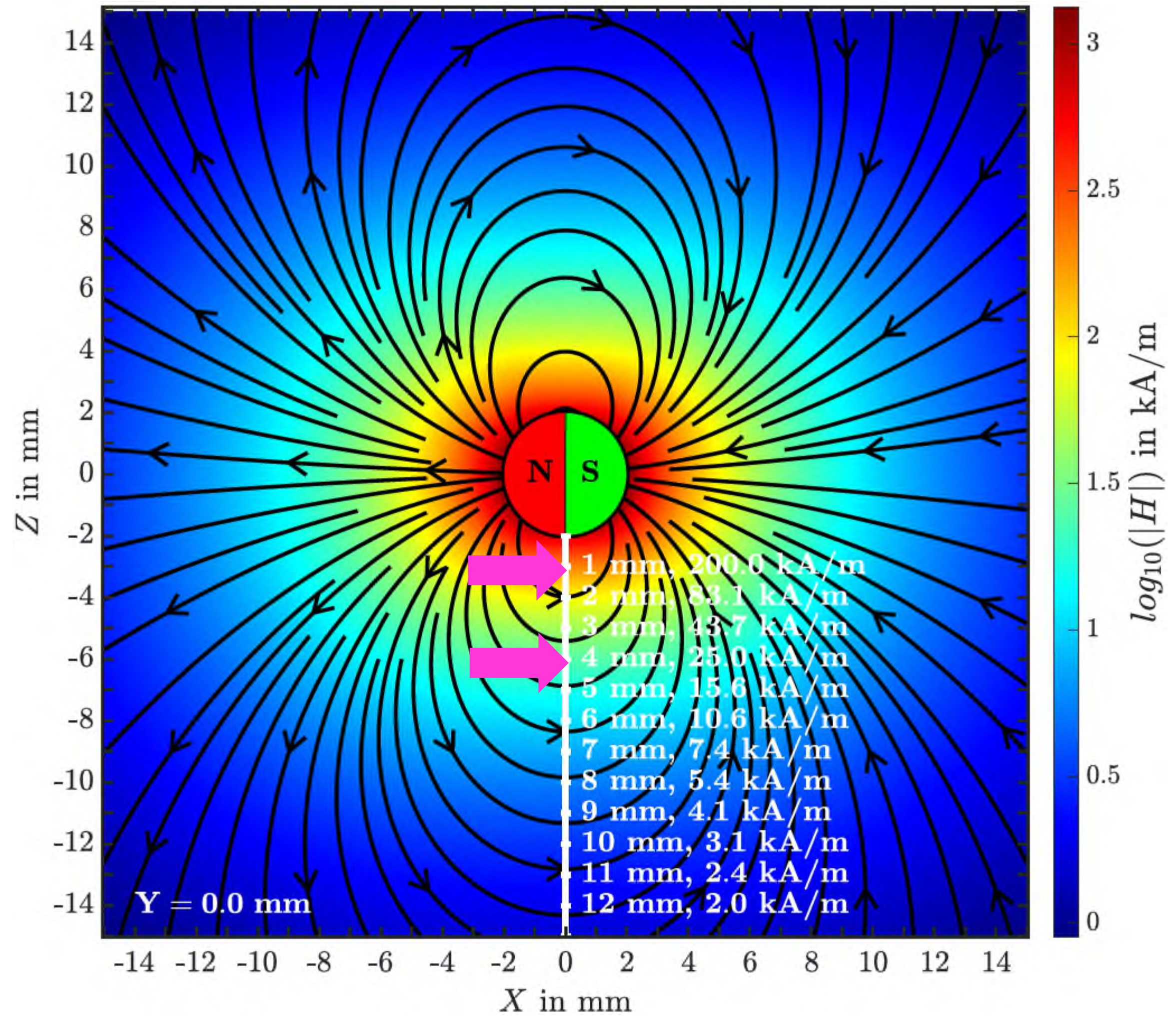
Simulation – Sensor Array

- Imprinting configuration at 1 mm and 200 kA/m



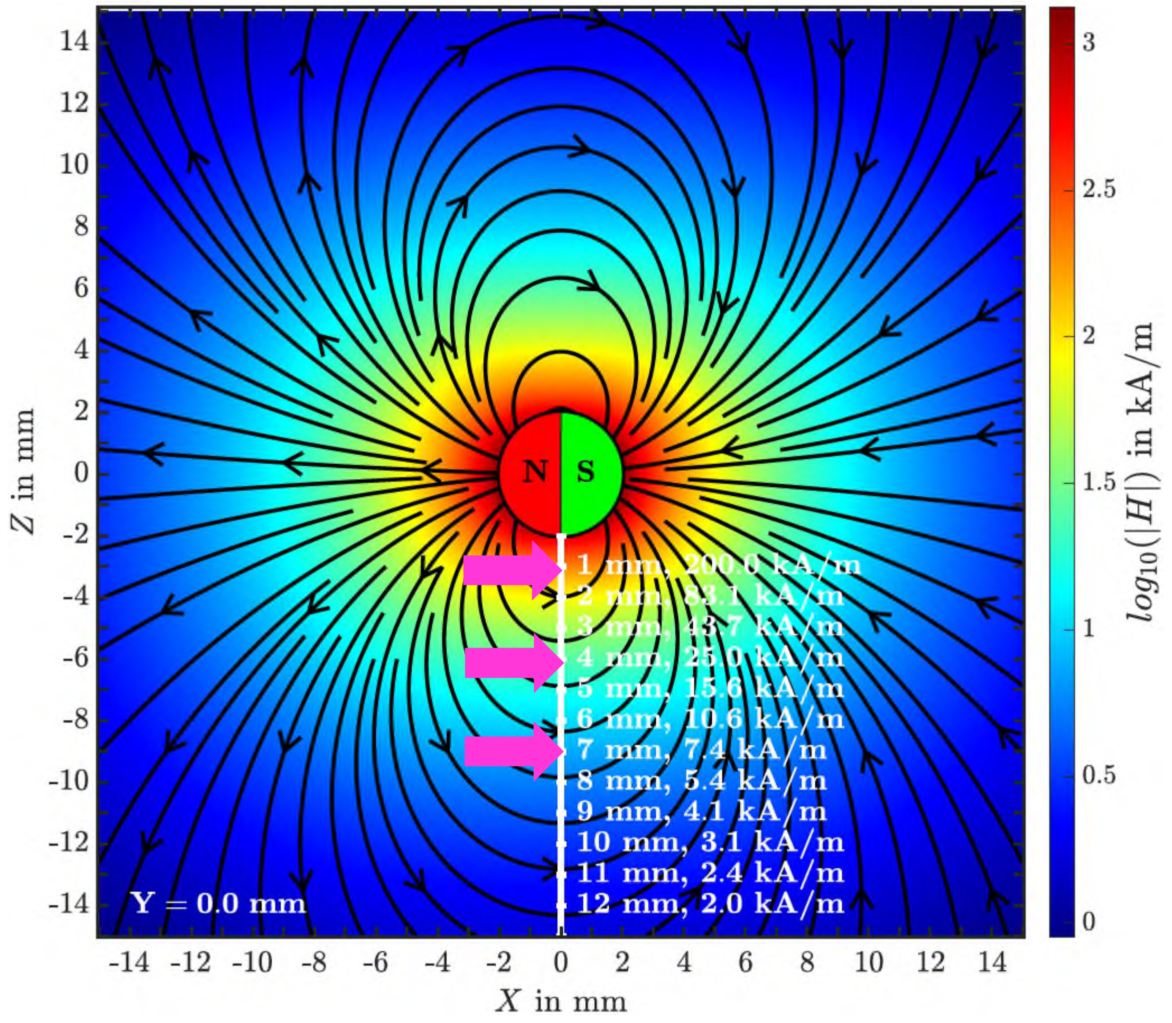
Simulation – Sensor Array

- Imprinting configuration at 1 mm and 200 kA/m
- Minimal distance at 4 mm and 25 kA/m

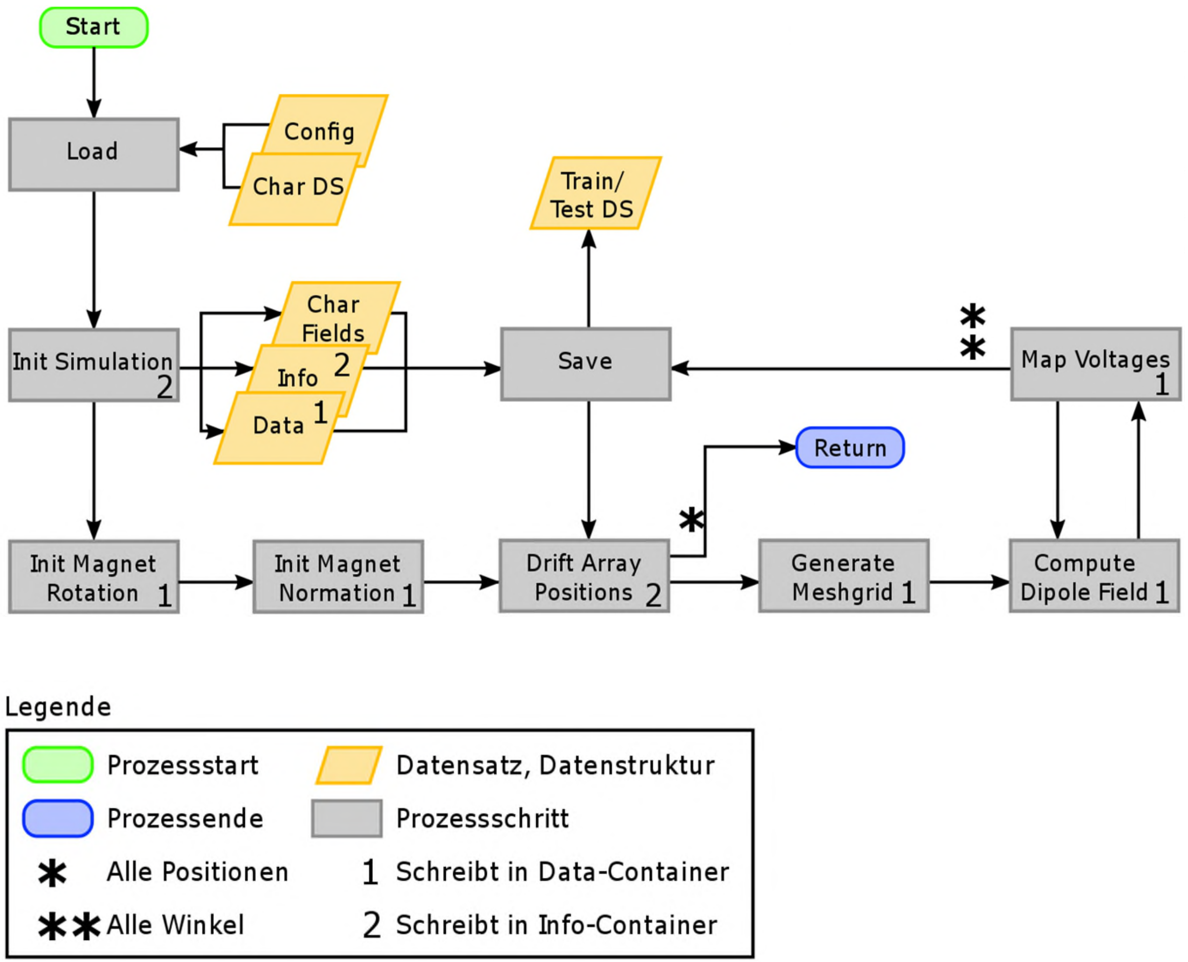


Simulation – Sensor Array

- Imprinting configuration at 1 mm and 200 kA/m
- Minimal distance at 4 mm and 25 kA/m
- Linear area at 7 mm and 7,4 kA/m

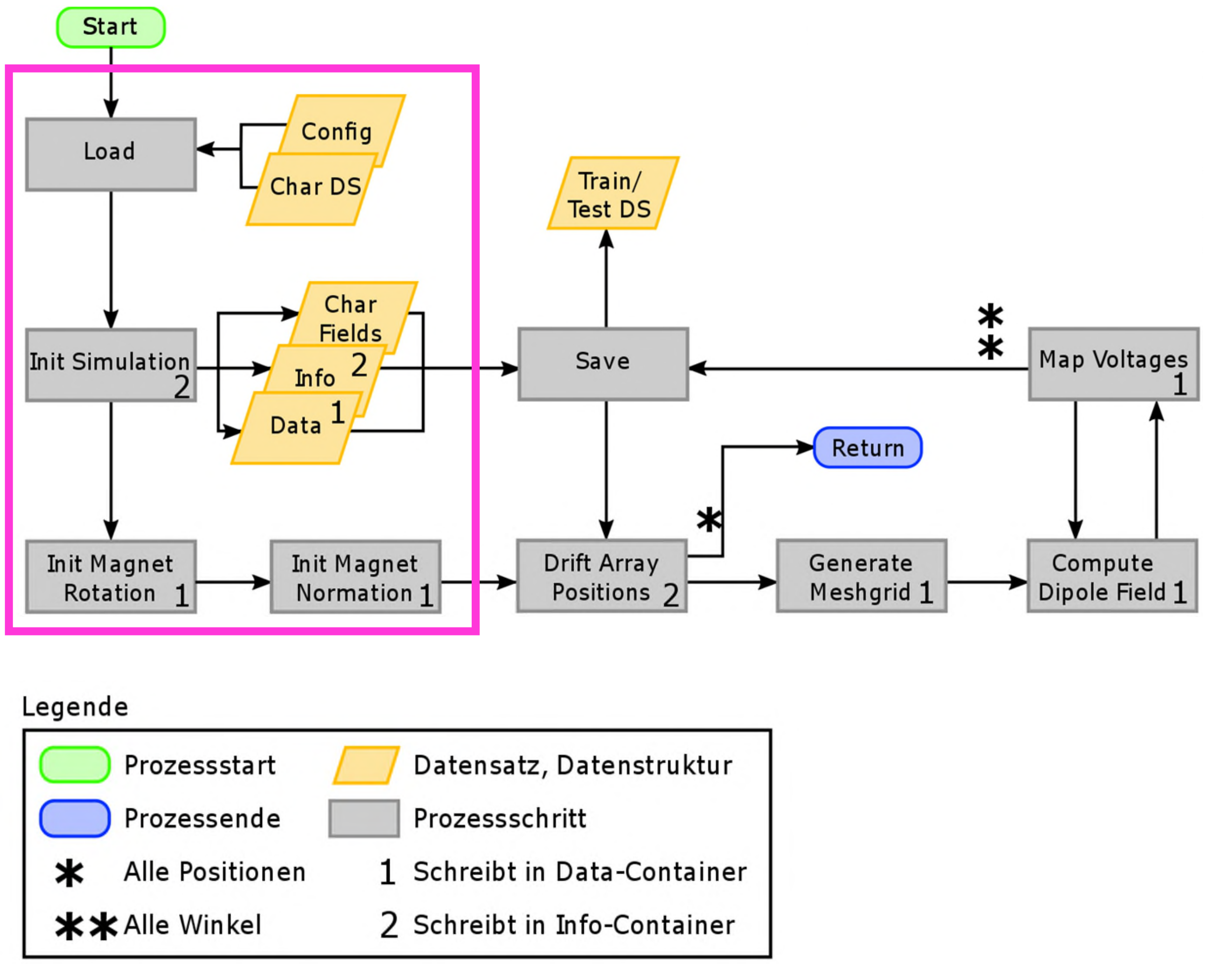


Simulation – Sensor Array



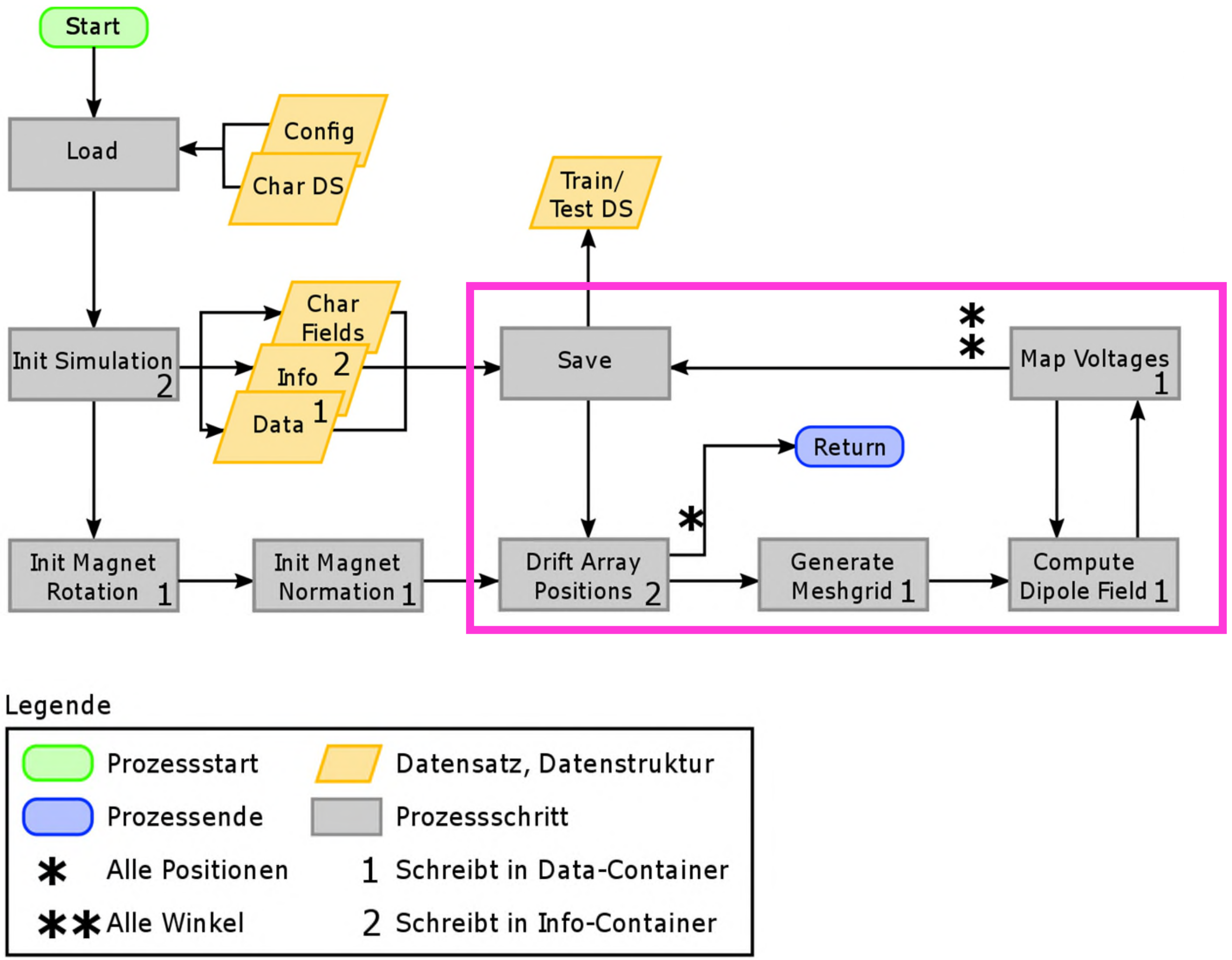
Simulation – Sensor Array

- Load und initiate simulation parameters



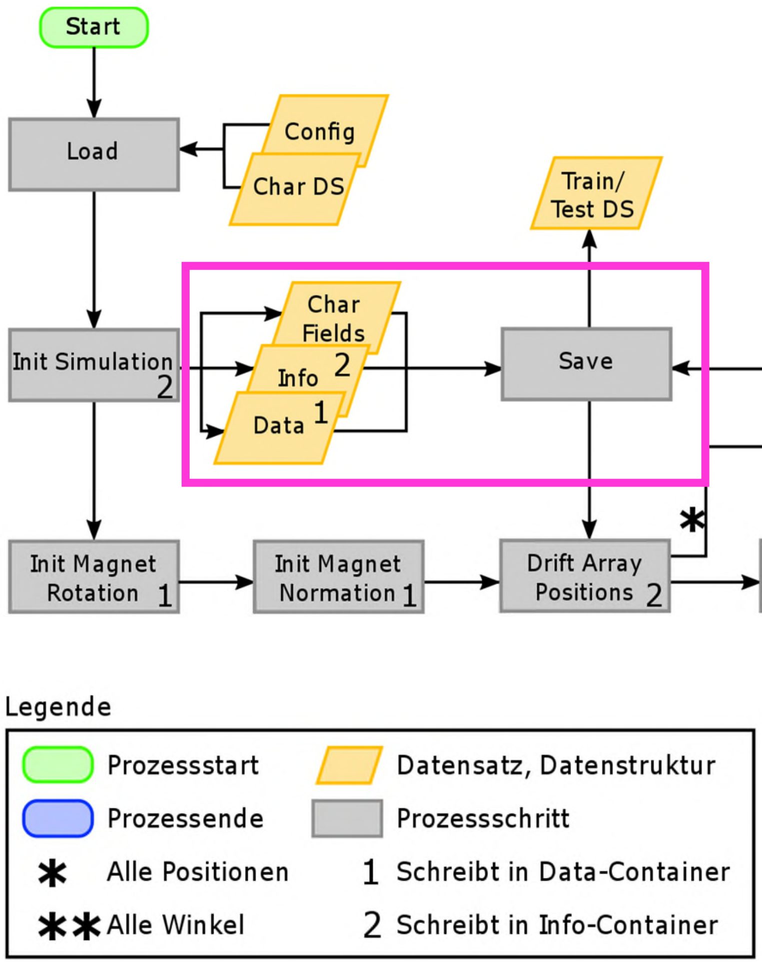
Simulation – Sensor Array

- Load und initiate simulation parameters
- Execute in nested for loops

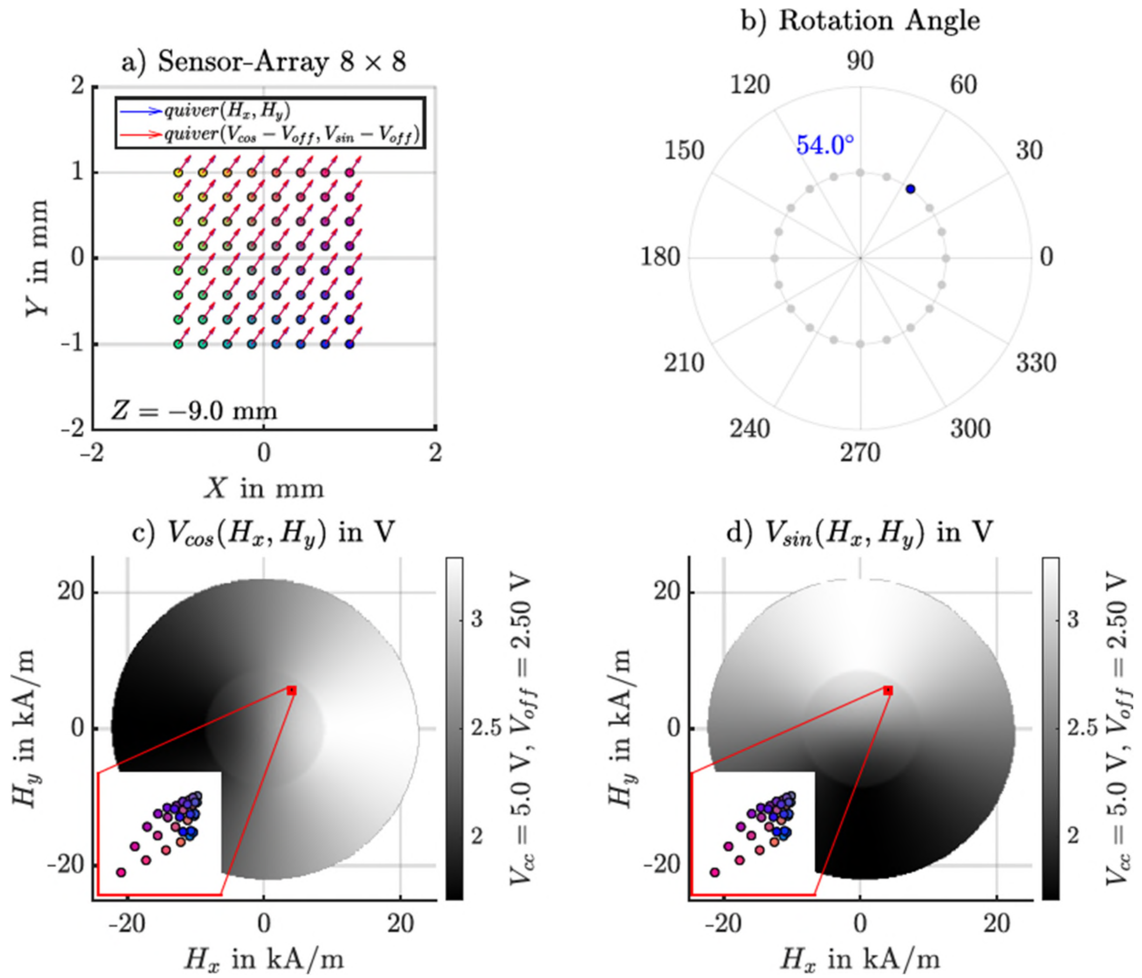


Simulation – Sensor Array

- Load und initiate simulation parameters
- Execute in nested for loops
- Save data containers

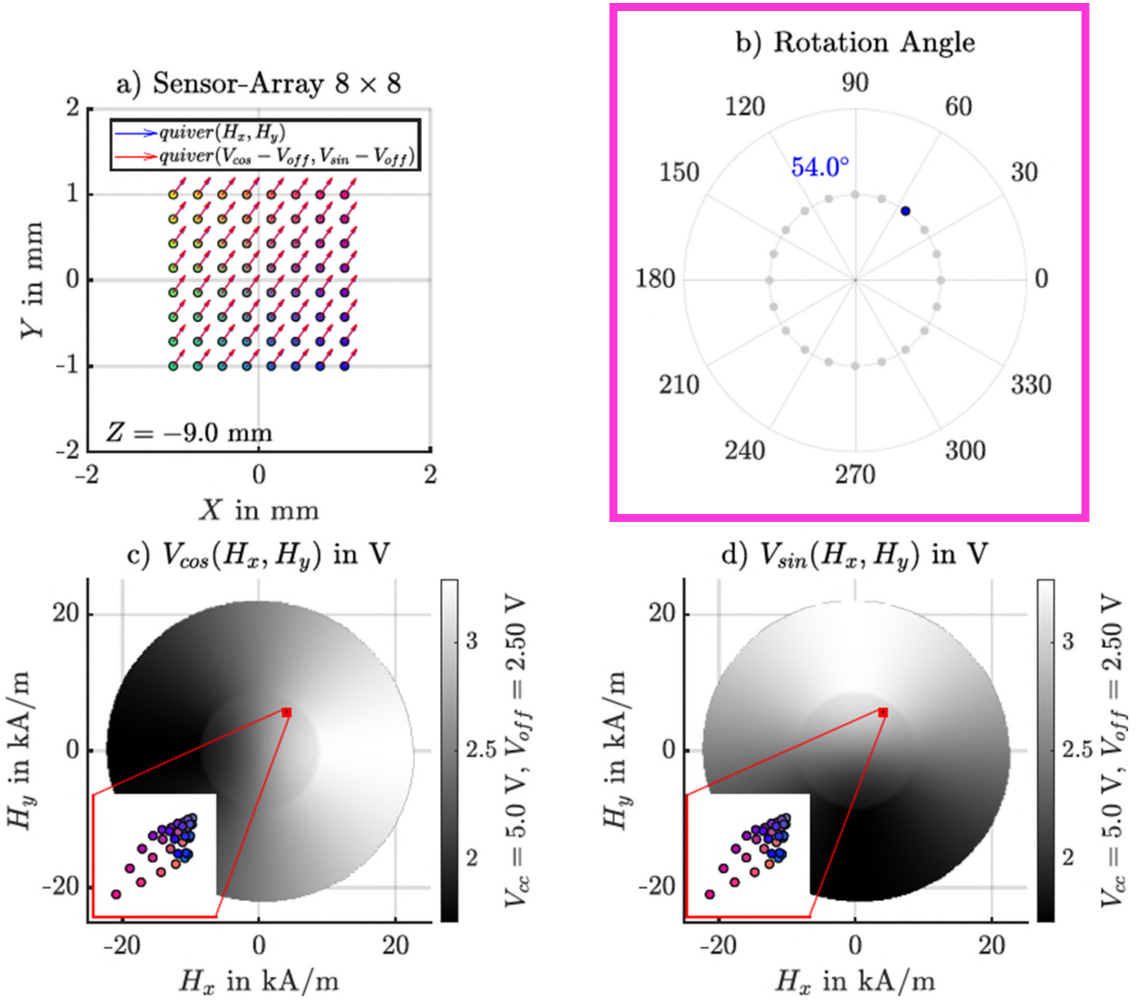


Simulation – Sensor Array



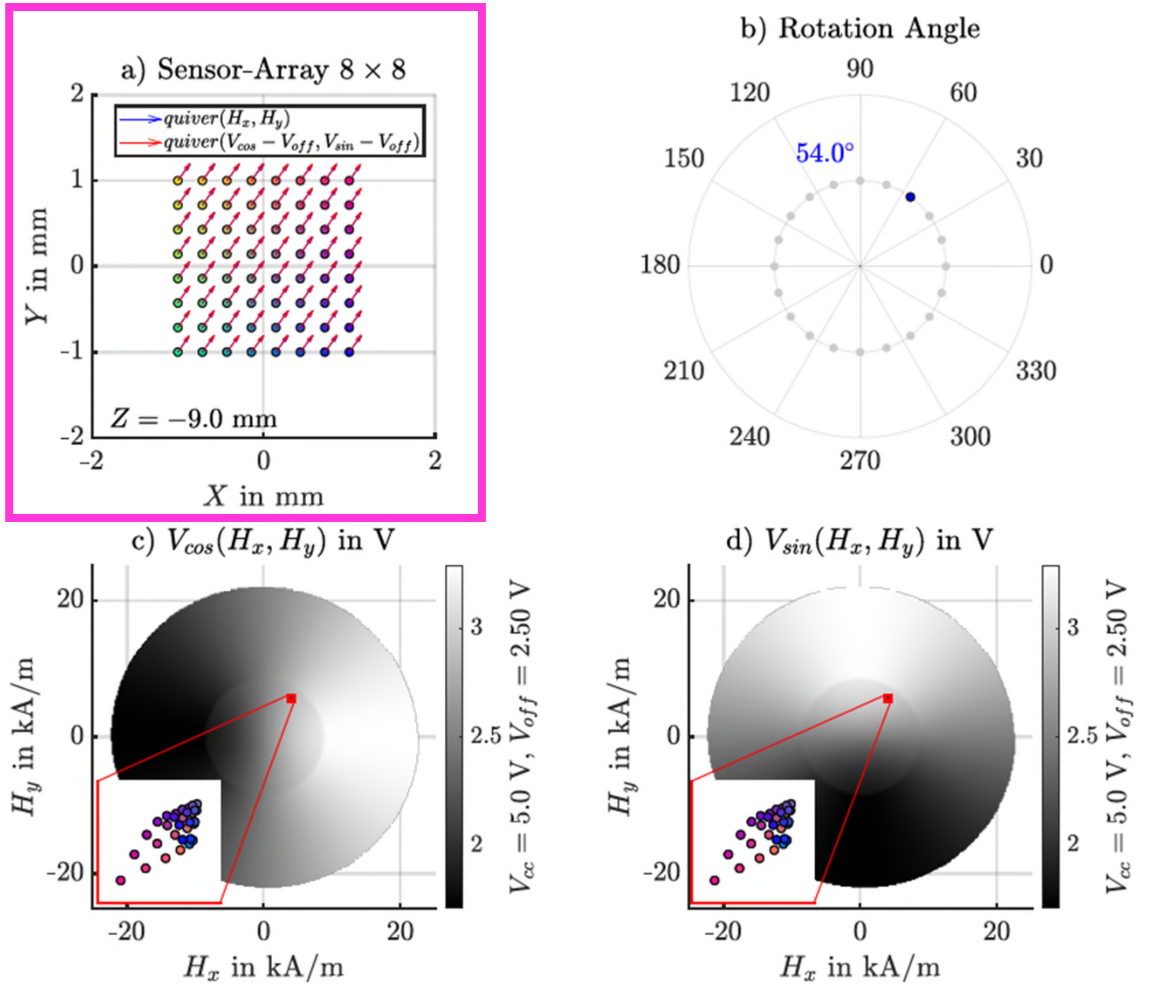
Simulation – Sensor Array

- Input – simulation angle



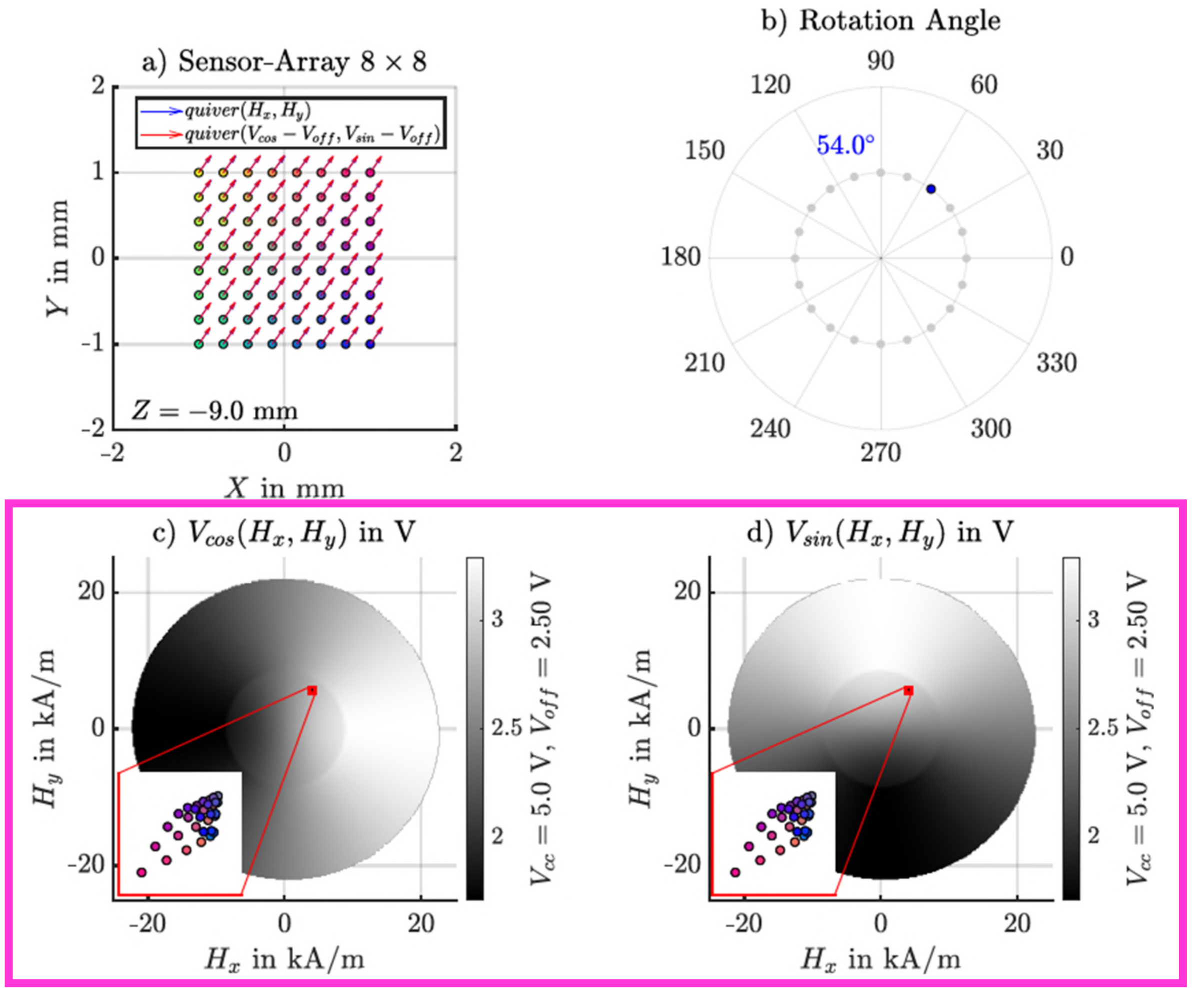
Simulation – Sensor Array

- Input – simulation angle
- Mesh grid – calculation at each Sensor Pixel



Simulation – Sensor Array

- Input – simulation angle
- Mesh grid – calculation at each Sensor Pixel
- Mapping – Take reference voltages



Sim. – Gaussian Processes

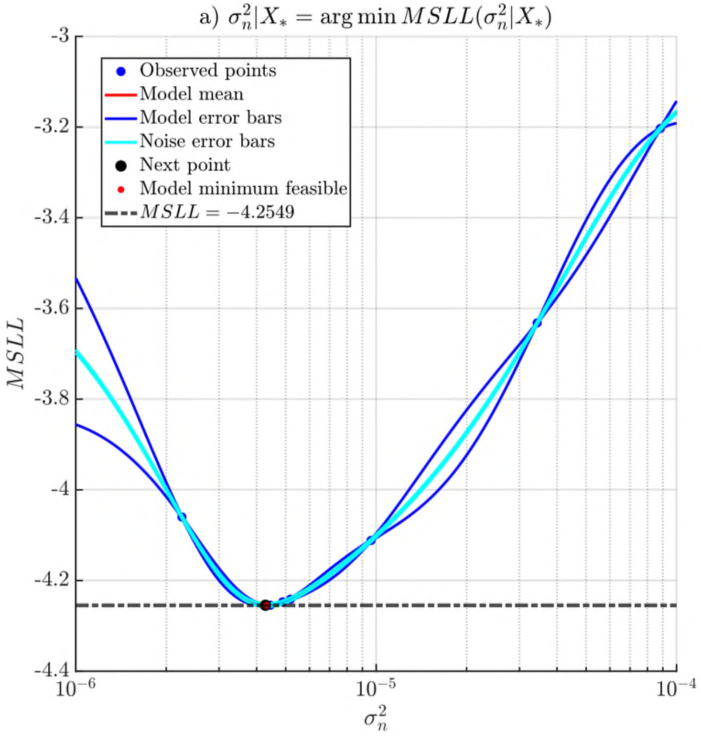
- Configuration

MSLL – Mean Standardized Logarithmic Loss

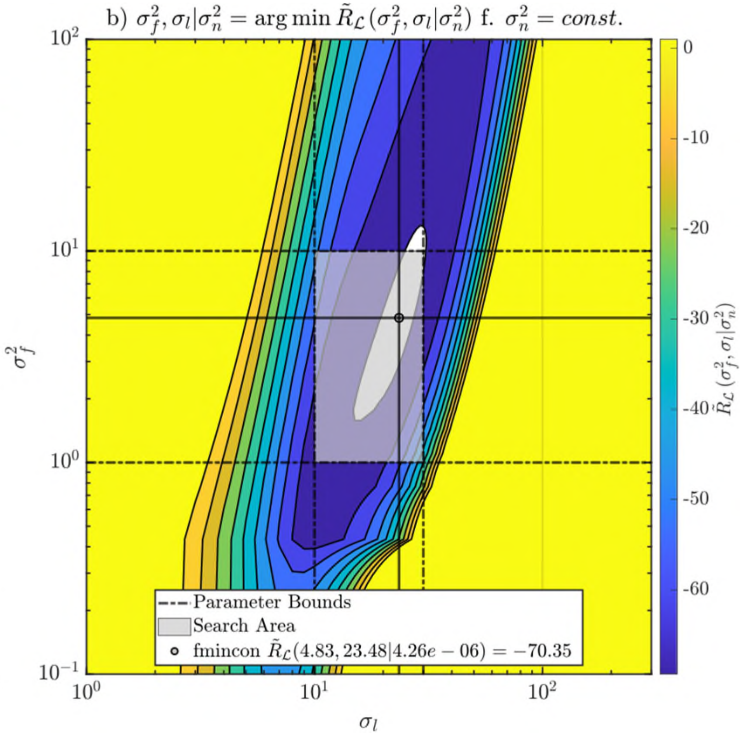
Sim. – Gaussian Processes

- Configuration
  - Kernel Modul
  - Mean computation

Generalization



Kernel Scaling

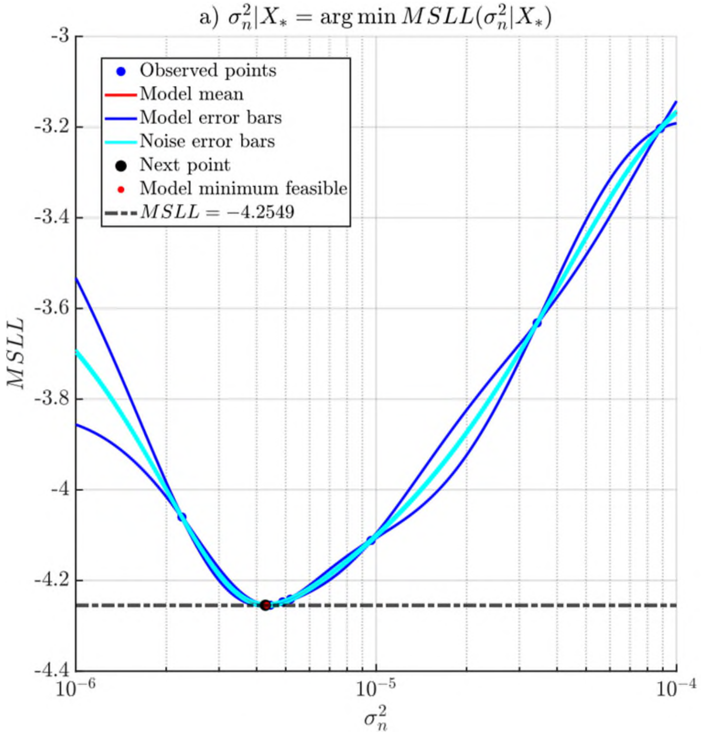


MSLL – Mean Standardized Logarithmic Loss

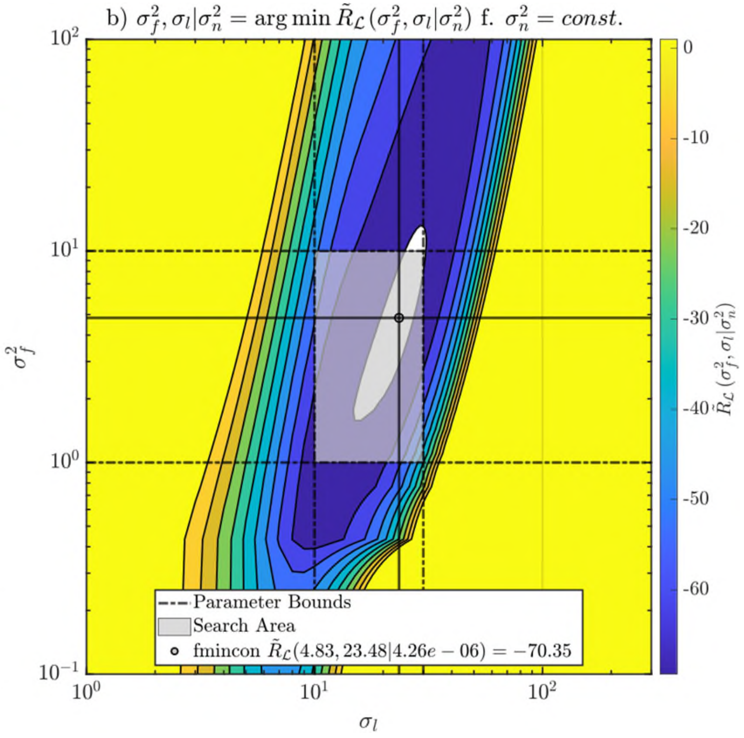
Sim. – Gaussian Processes

- Configuration
  - Kernel Modul
  - Mean computation
  - Loss computation

Generalization



Kernel Scaling

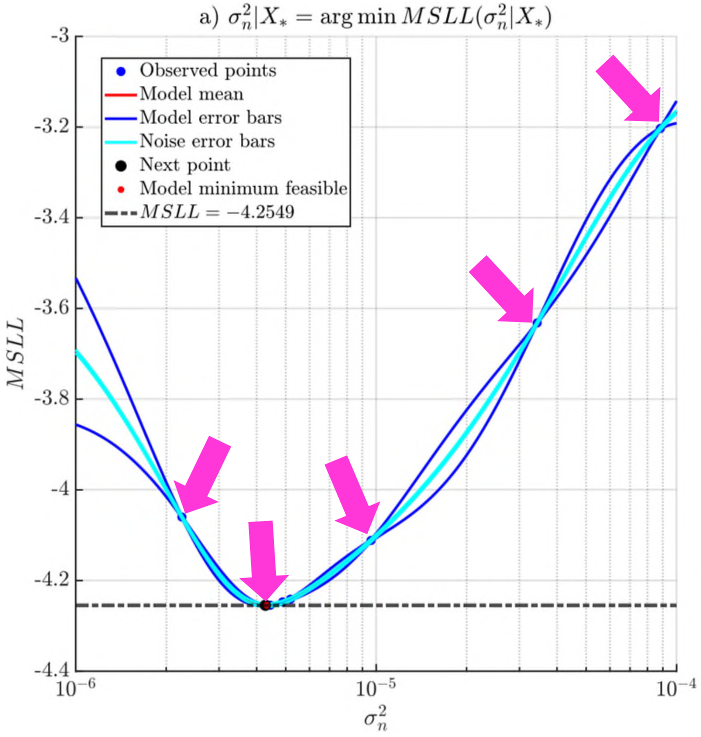


MSLL – Mean Standardized Logarithmic Loss

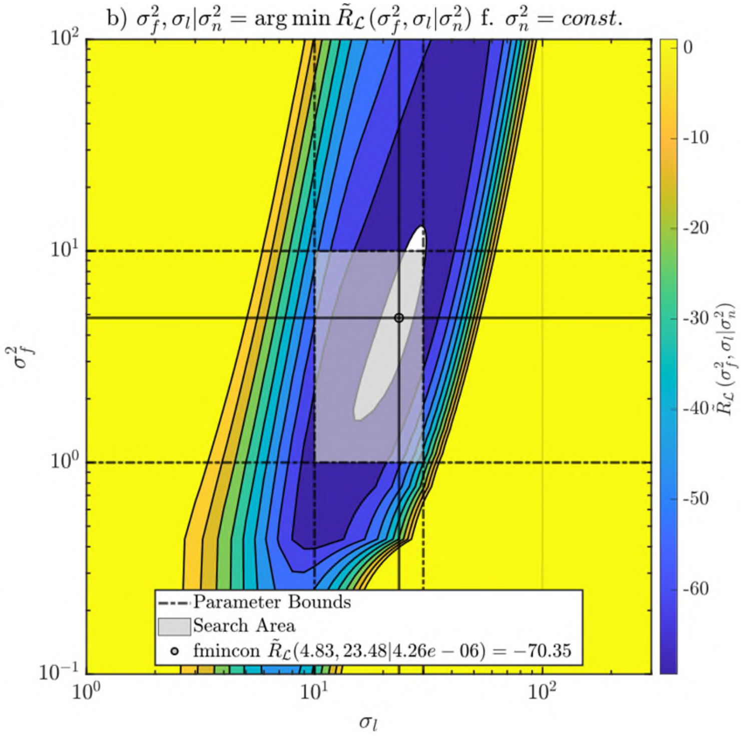
Sim. – Gaussian Processes

- Configuration
  - Kernel Modul
  - Mean computation
  - Loss computation
  - Number of runs

Generalization



Kernel Scaling

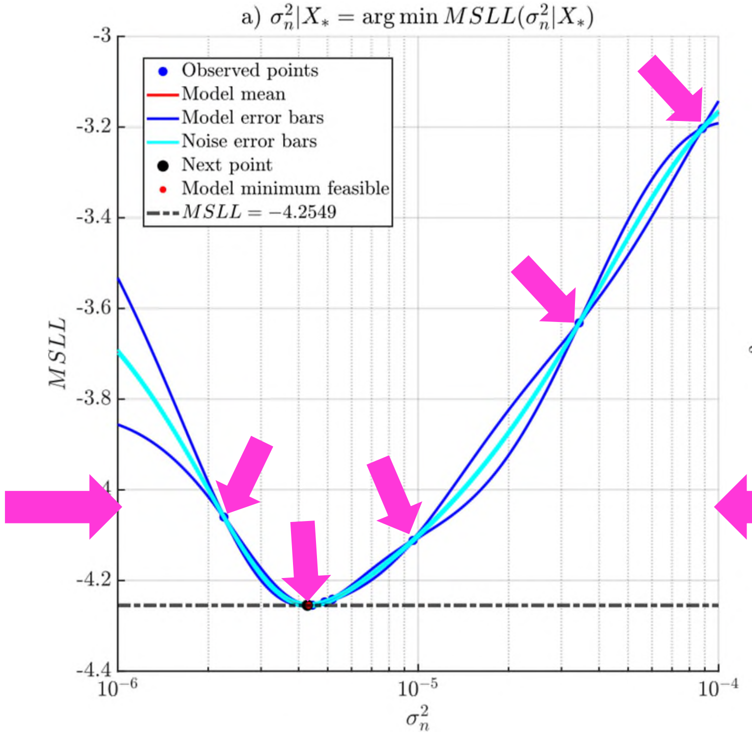


MSLL – Mean Standardized Logarithmic Loss

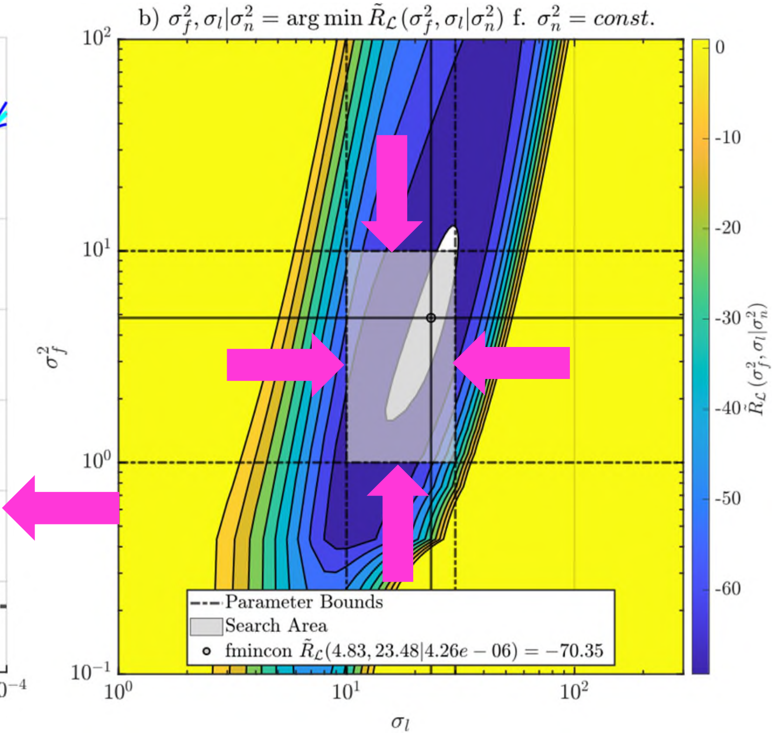
Sim. – Gaussian Processes

- Configuration
  - Kernel Modul
  - Mean computation
  - Loss computation
  - Number of runs
  - Parameter bounds

Generalization



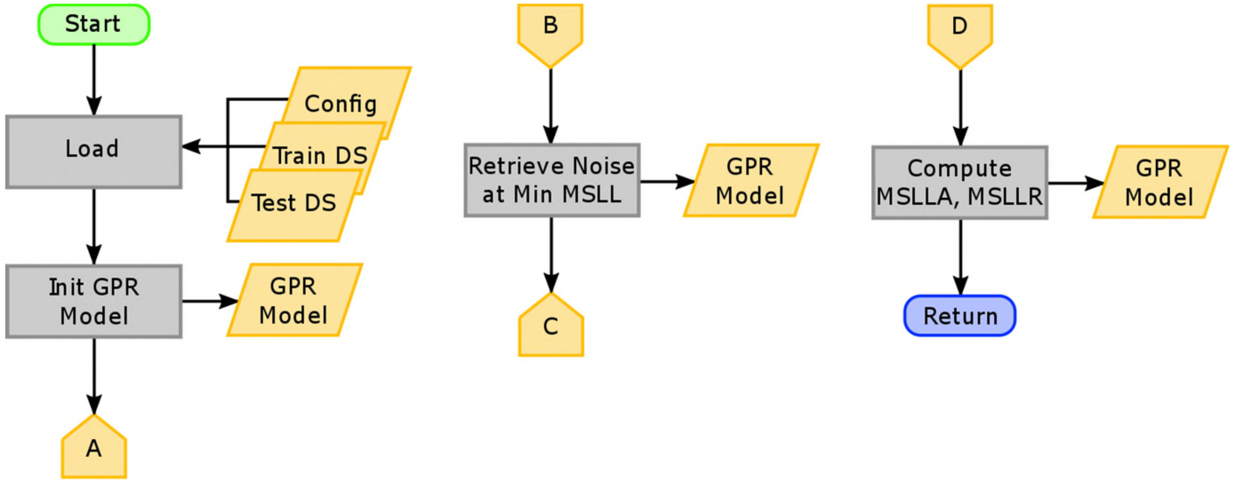
Kernel Scaling



MSLL – Mean Standardized Logarithmic Loss

Sim. – Gaussian Processes

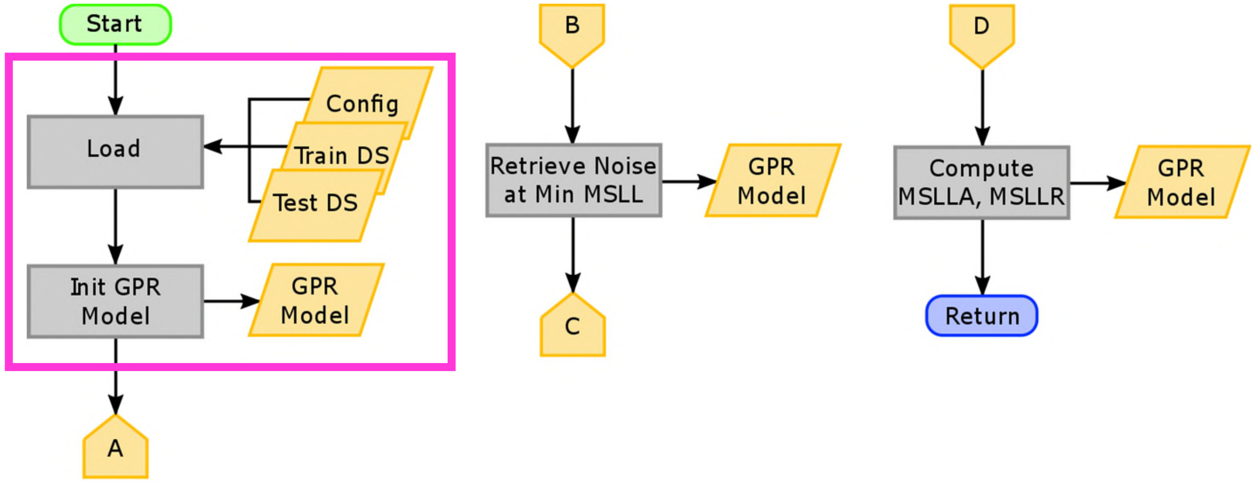
- Training mode



GPR – Gaussian Processes for Regression
MSLL – Mean Standardized Logarithmic Loss
MSLLA – Mean Standardized Logarithmic Loss Angle
MSLLR – Mean Standardized Logarithmic Loss Radius

Sim. – Gaussian Processes

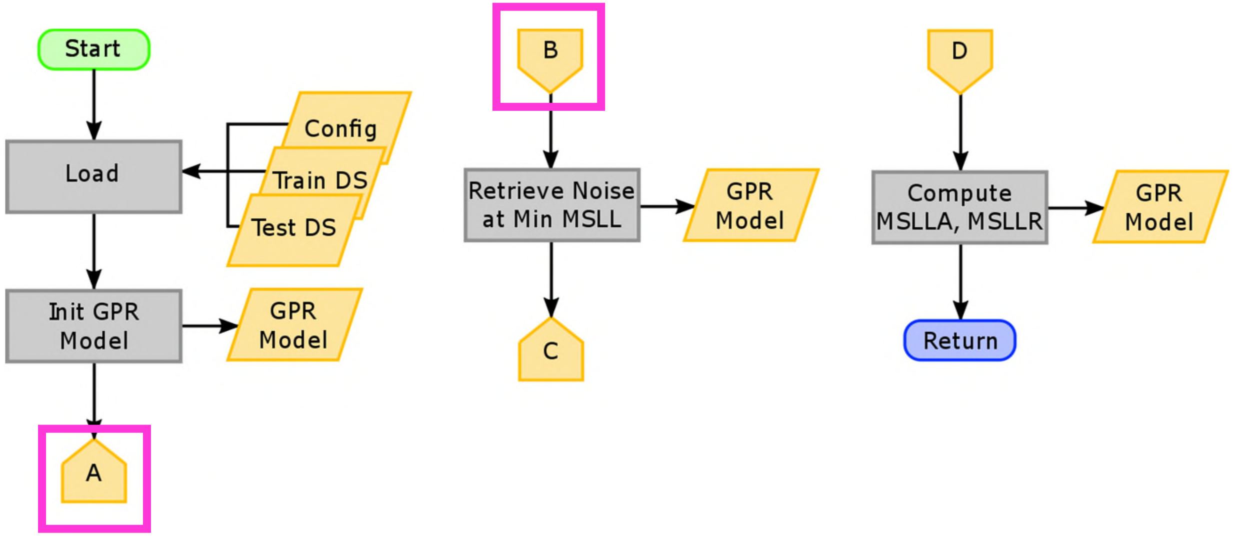
- Training mode
  - Load and initiate



GPR – Gaussian Processes for Regression
MSLL – Mean Standardized Logarithmic Loss
MSLLA – Mean Standardized Logarithmic Loss Angle
MSLLR – Mean Standardized Logarithmic Loss Radius

Sim. – Gaussian Processes

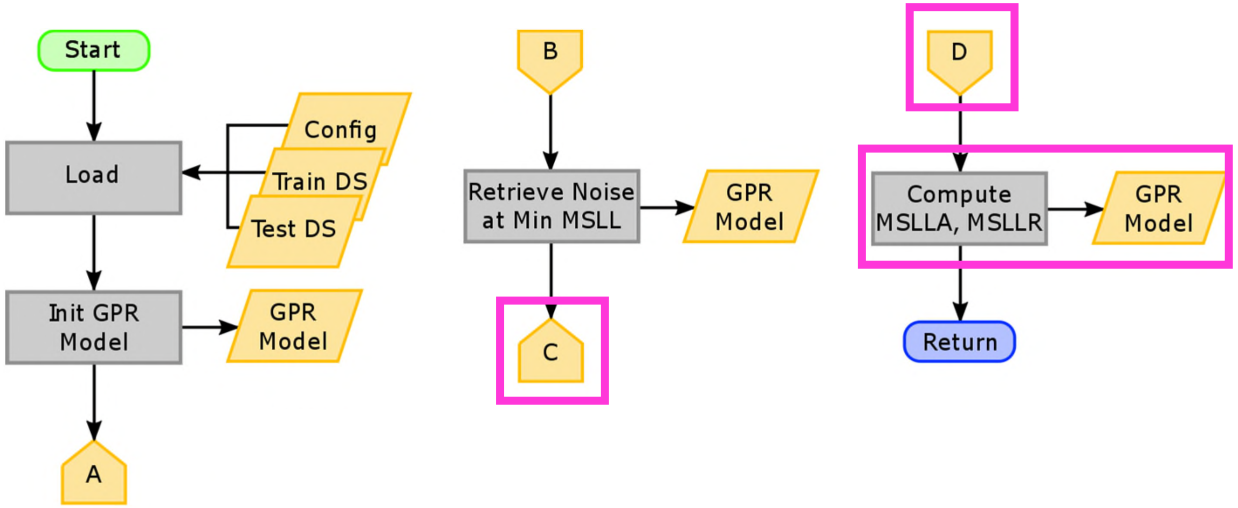
- Training mode
  - Load and initiate
  - Generalization with embedded scaling



GPR – Gaussian Processes for Regression
MSLL – Mean Standardized Logarithmic Loss
MSLLA – Mean Standardized Logarithmic Loss Angle
MSLLR – Mean Standardized Logarithmic Loss Radius

Sim. – Gaussian Processes

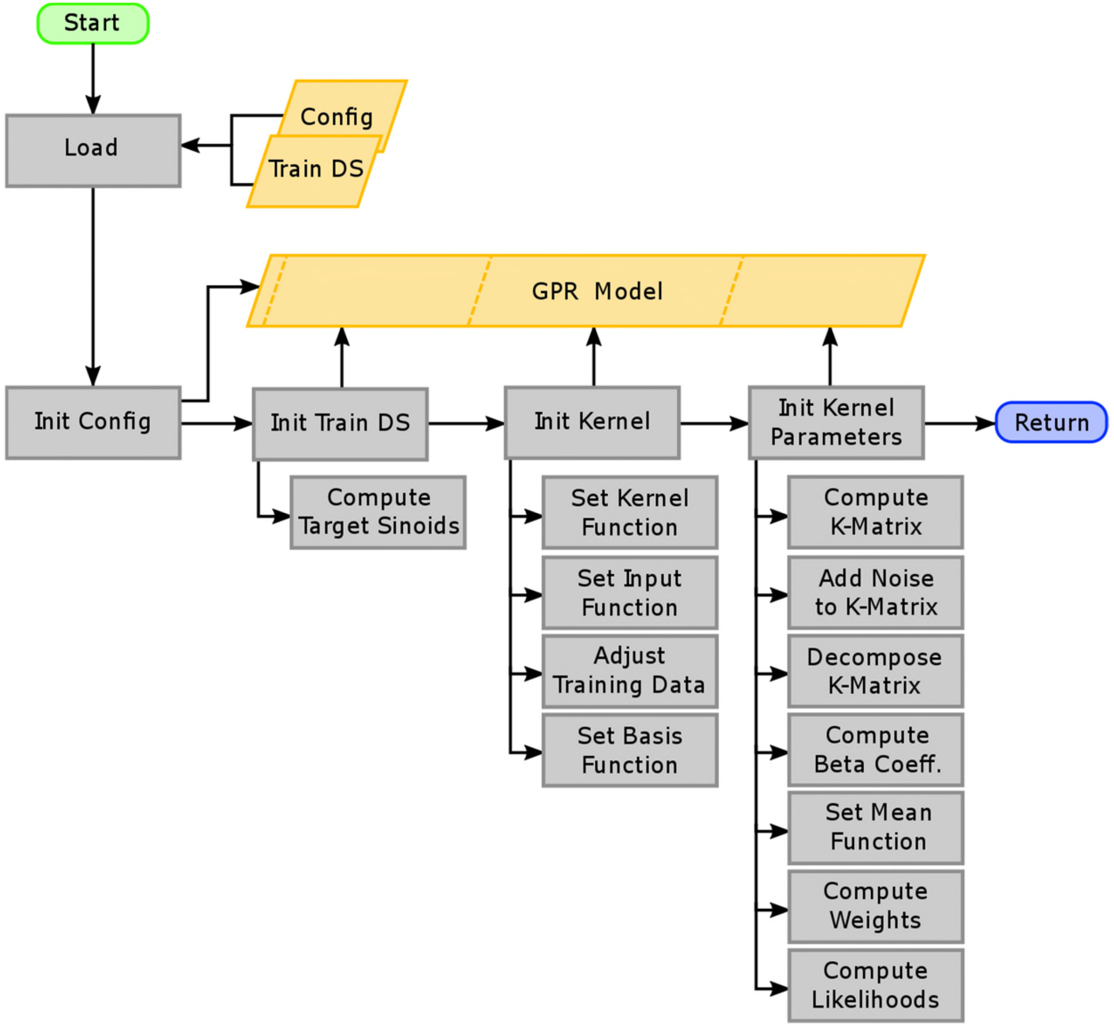
- Training mode
  - Load and initiate
  - Generalization with embedded scaling
  - Final scaling and valuing



GPR – Gaussian Processes for Regression
MSLL – Mean Standardized Logarithmic Loss
MSLLA – Mean Standardized Logarithmic Loss Angle
MSLLR – Mean Standardized Logarithmic Loss Radius

Sim. – Gaussian Processes

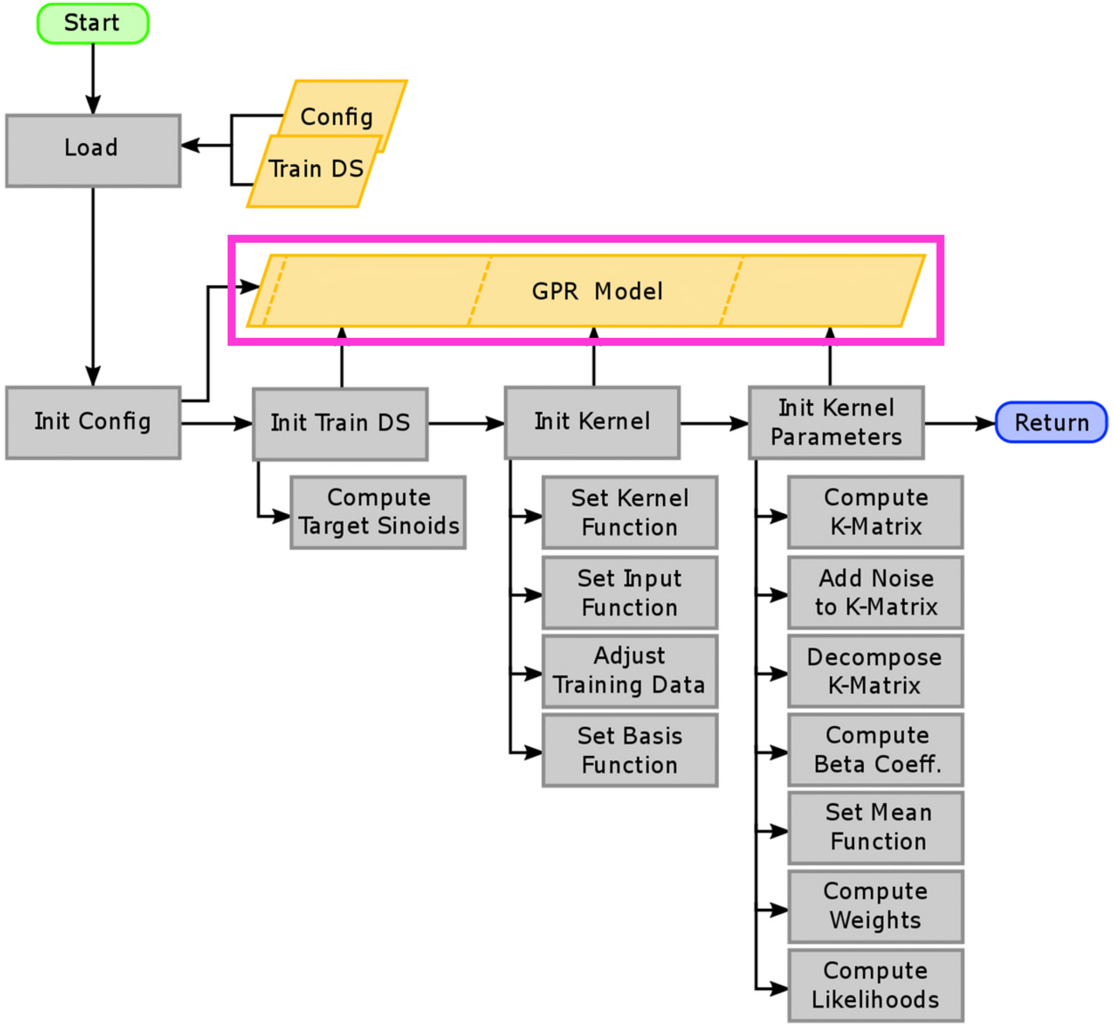
- Sequential initialization



GPR – Gaussian Processes for Regression
DS – Dataset

Sim. – Gaussian Processes

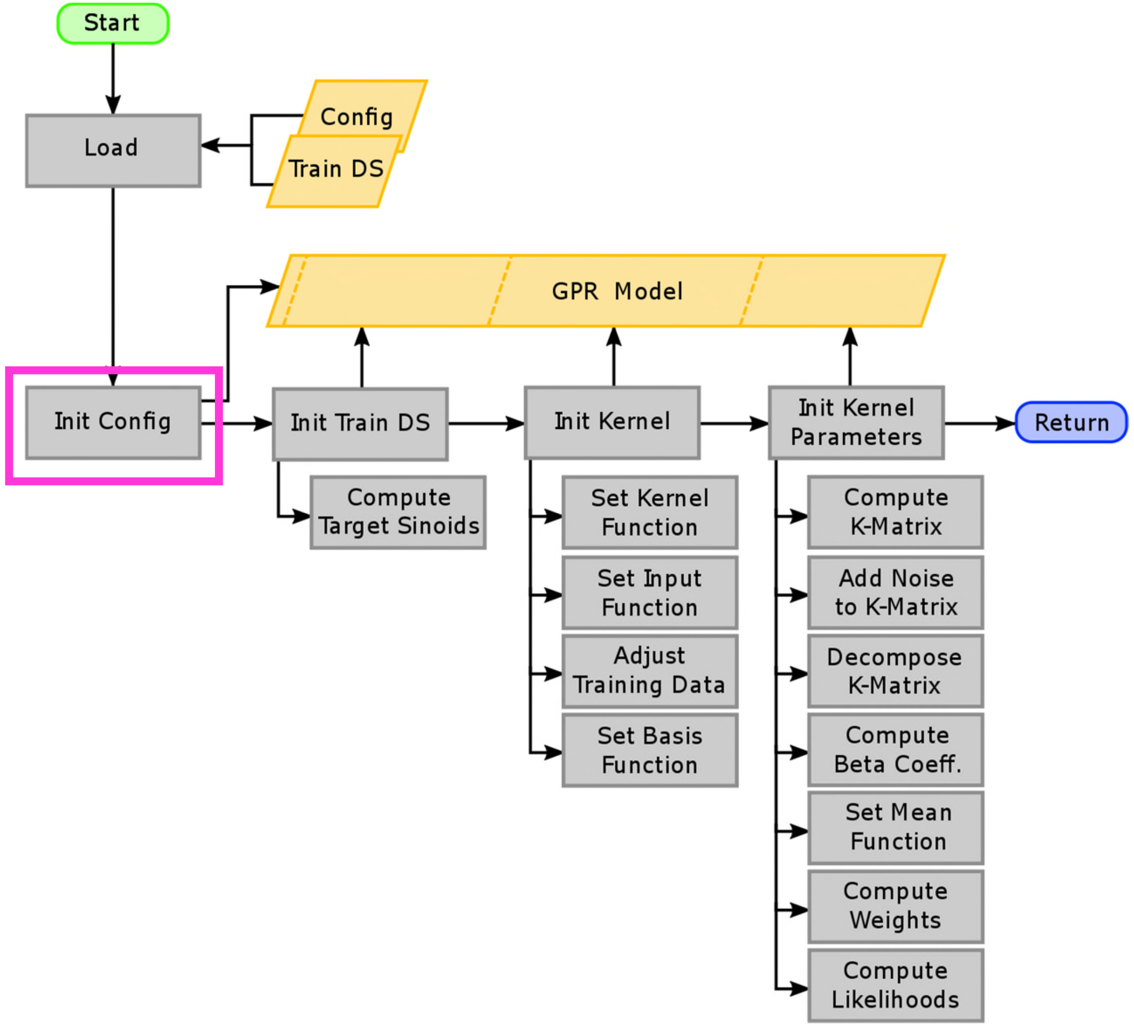
- Sequential initialization
  - Struct based model



GPR – Gaussian Processes for Regression
DS – Dataset

Sim. – Gaussian Processes

- Sequential initialization
  - Struct based model
  - Framework configuration

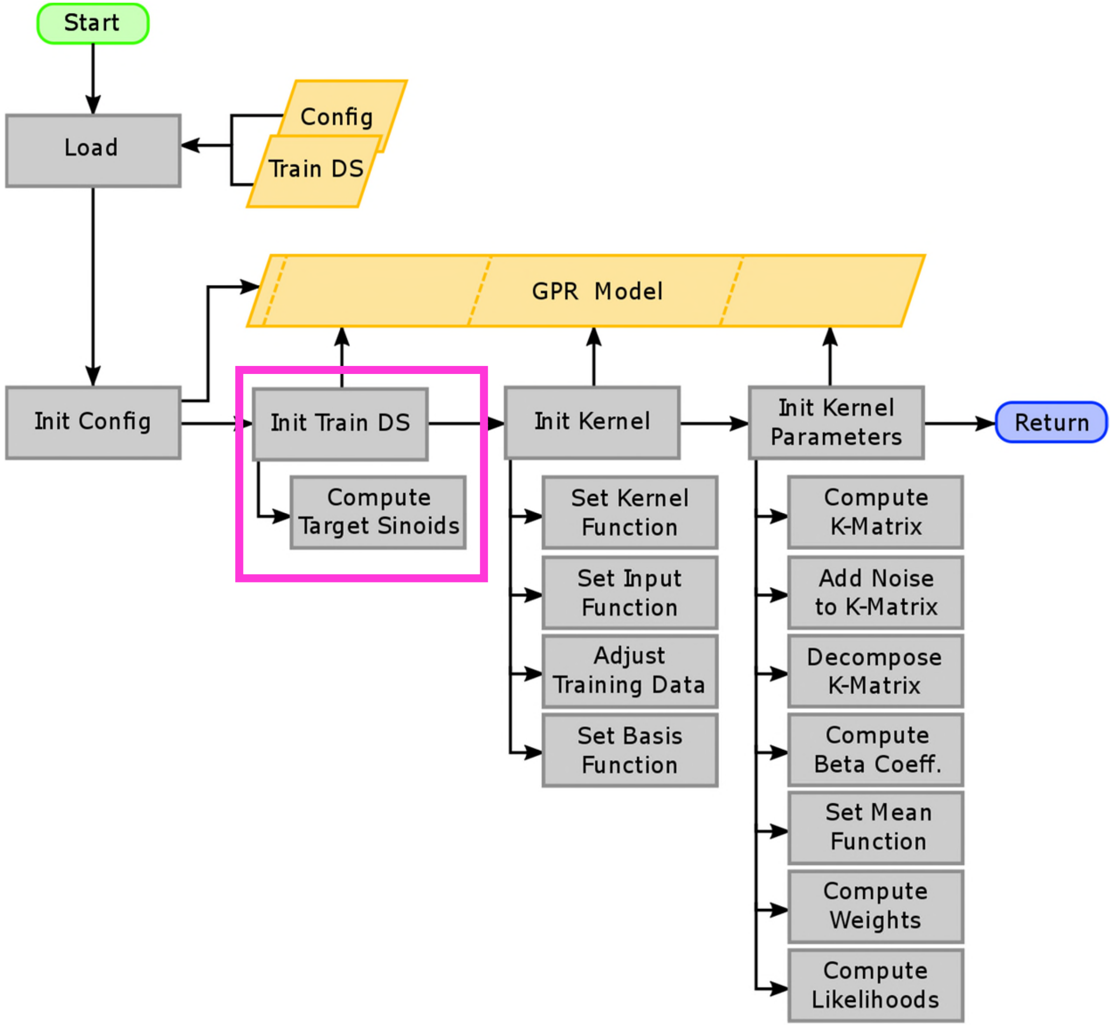


GPR – Gaussian Processes for Regression
DS – Dataset

Sim. – Gaussian Processes

HAW HAMBURG

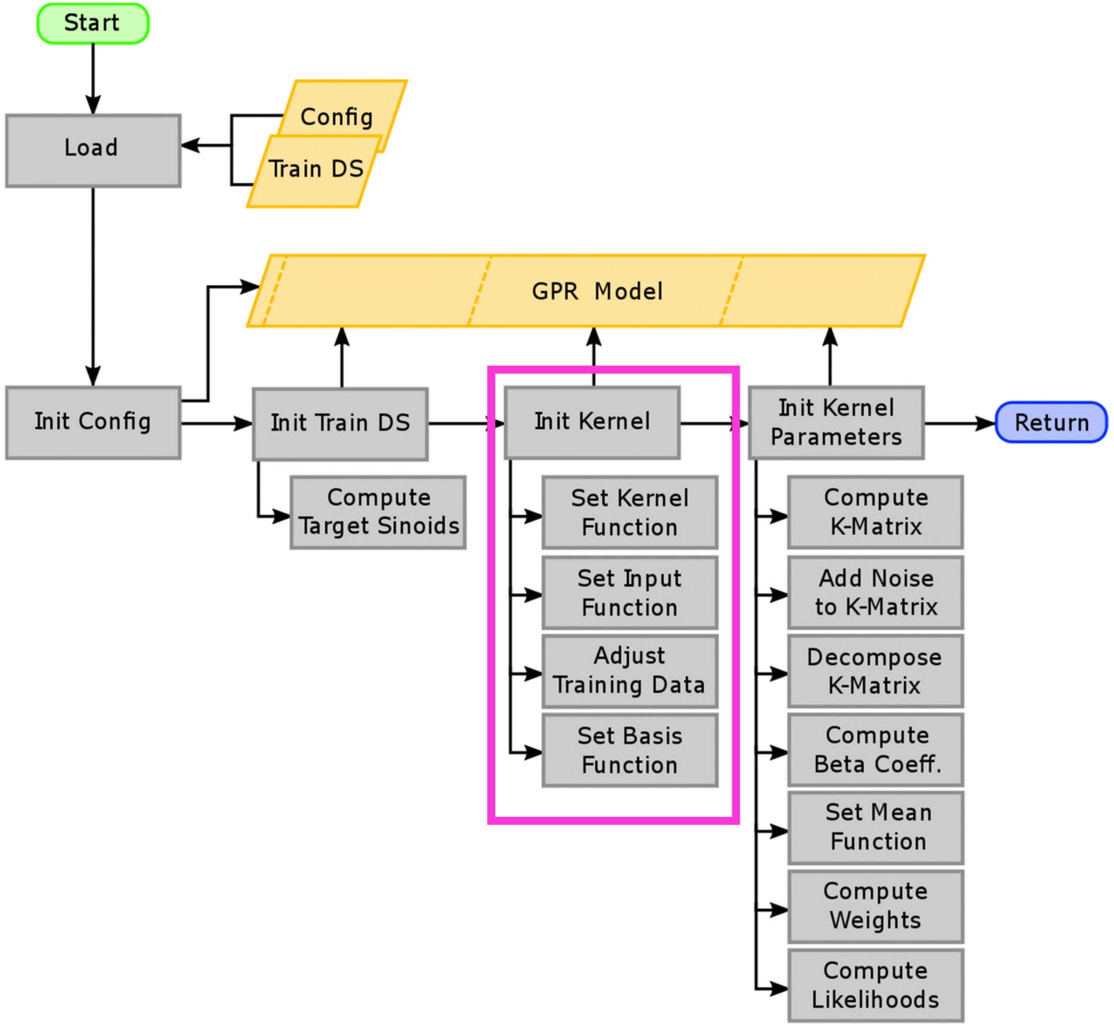
- Sequential initialization
  - Struct based model
  - Framework configuration
  - Build up references



GPR – Gaussian Processes for Regression
DS – Dataset

Sim. – Gaussian Processes

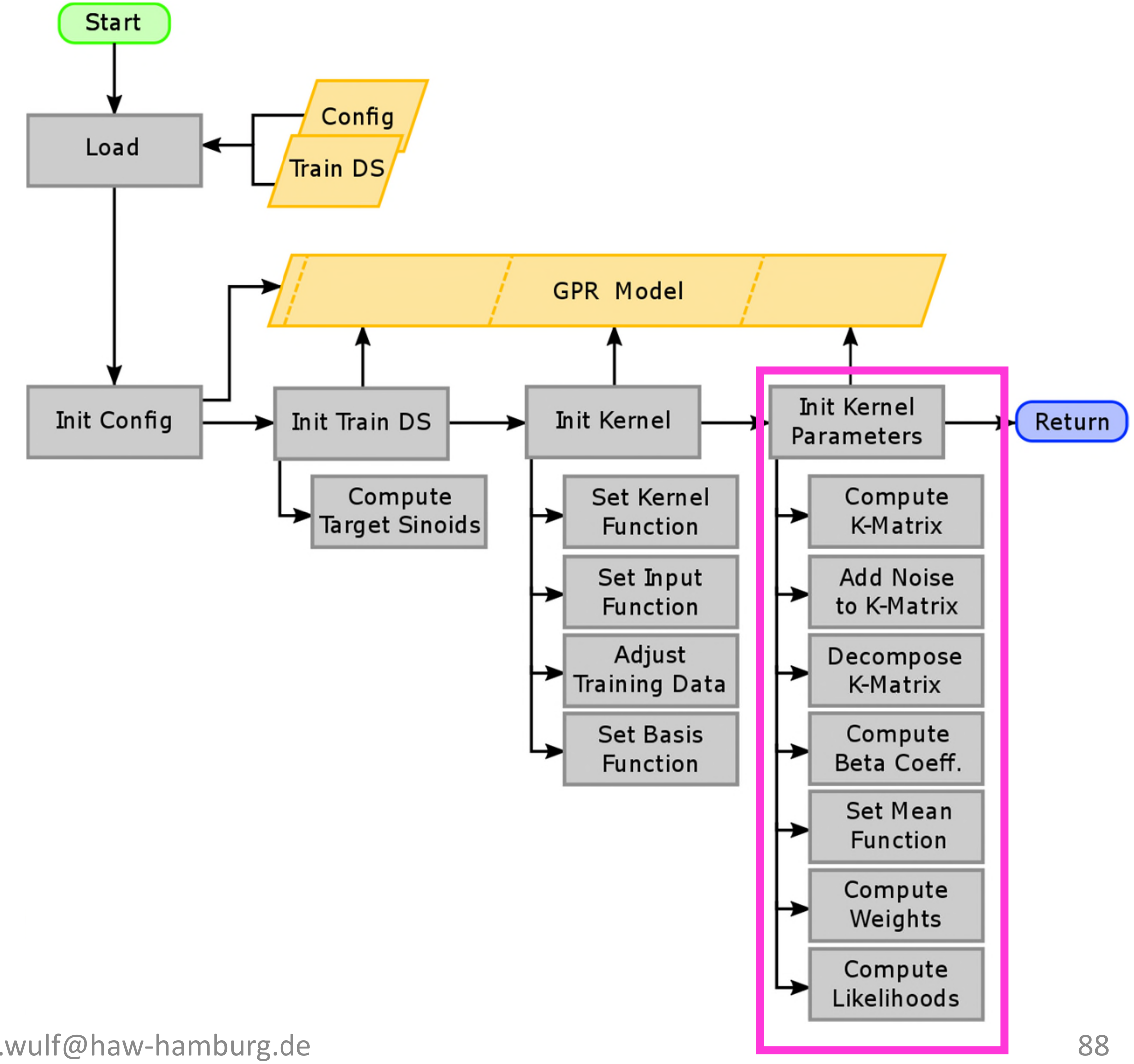
- Sequential initialization
  - Struct based model
  - Framework configuration
  - Build up references
  - Load functionality



GPR – Gaussian Processes for Regression
DS – Dataset

Sim. – Gaussian Processes

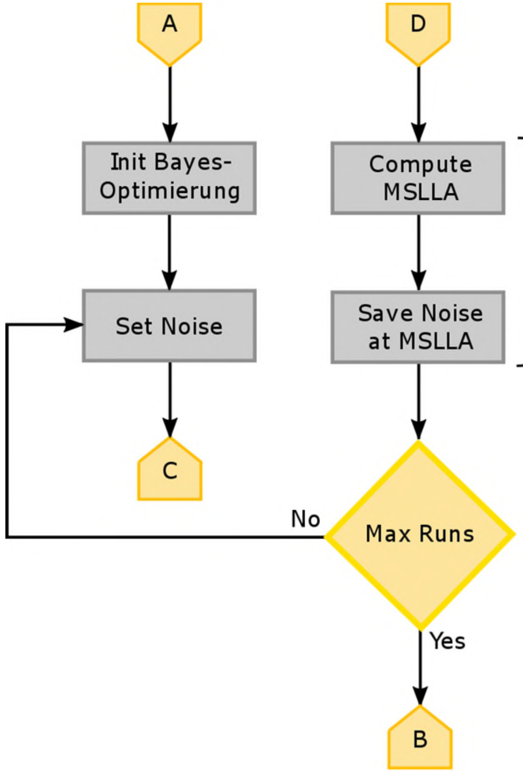
- Sequential initialization
  - Struct based model
  - Framework configuration
  - Build up references
  - Load functionality
  - Parametrize model

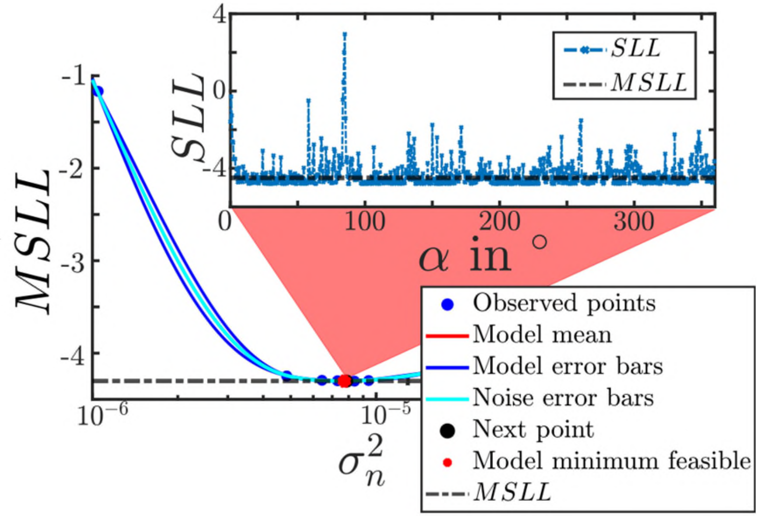


GPR – Gaussian Processes for Regression
DS – Dataset

Sim. – Gaussian Processes

- Generalization

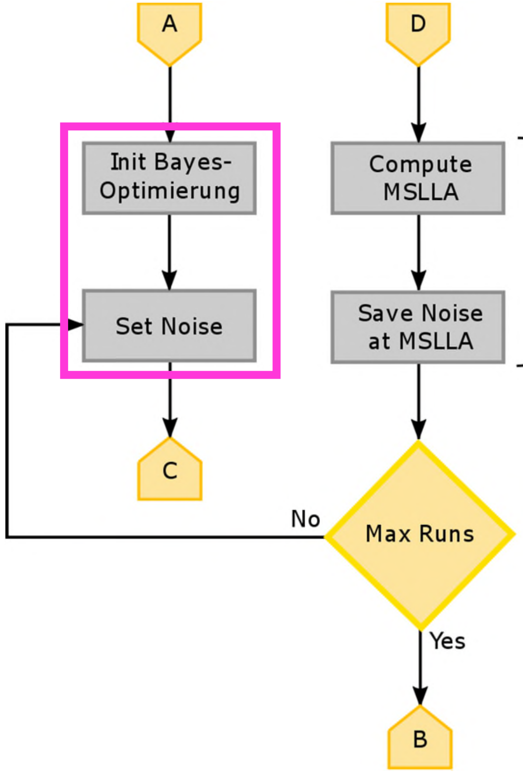


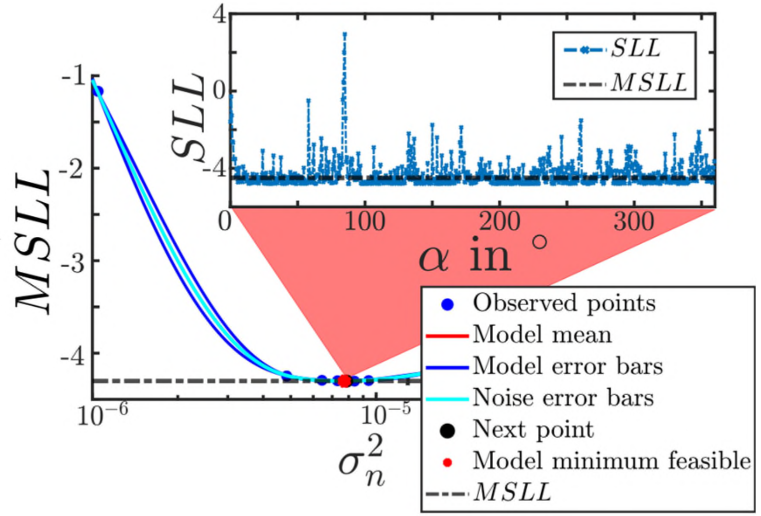


SLL – Standardized Logarithmic Loss
MSLL – Mean Standardized Logarithmic Loss
MSLLA – Mean Standardized Logarithmic Loss Angle

Sim. – Gaussian Processes

- Generalization
  - Set Noise Variance

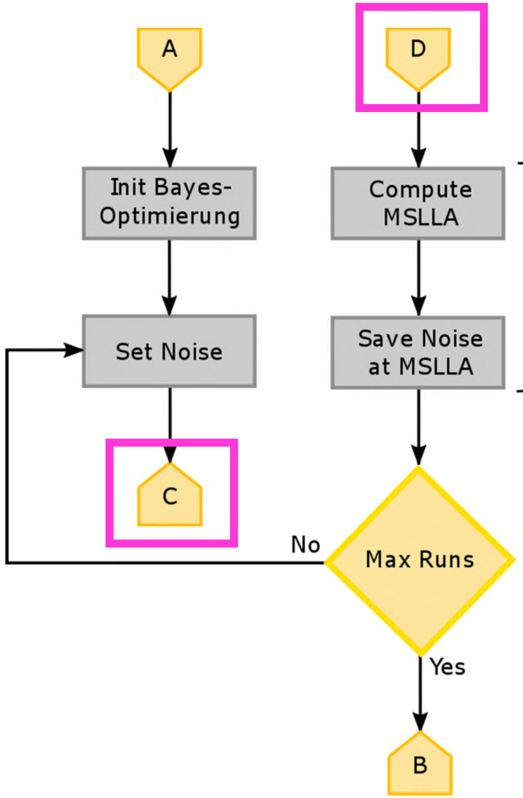


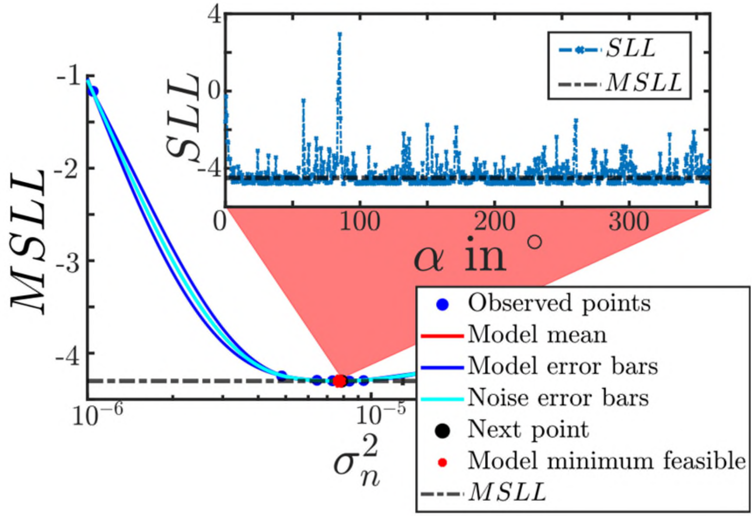


SLL – Standardized Logarithmic Loss
MSLL – Mean Standardized Logarithmic Loss
MSLLA – Mean Standardized Logarithmic Loss Angle

Sim. – Gaussian Processes

- Generalization
  - Set Noise Variance
  - Kernel scaling



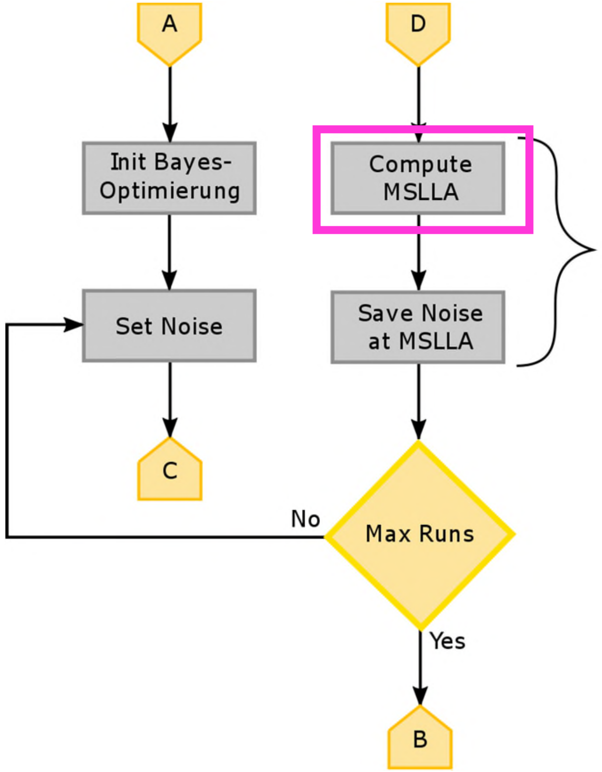


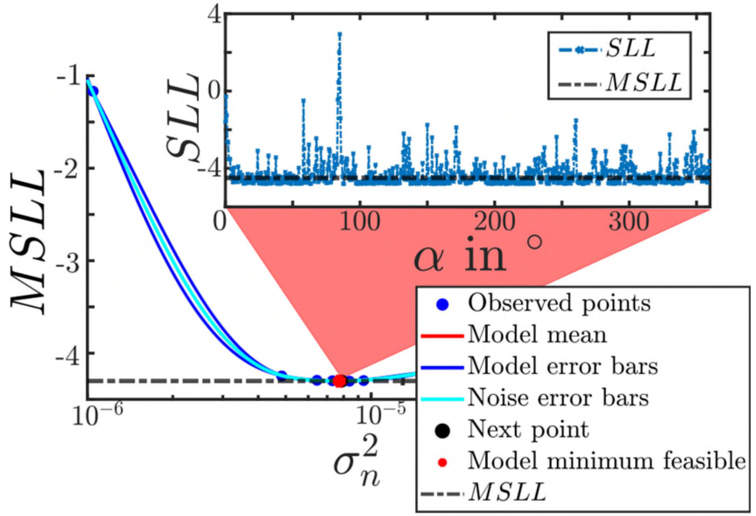
SLL – Standardized Logarithmic Loss
MSLL – Mean Standardized Logarithmic Loss
MSLLA – Mean Standardized Logarithmic Loss Angle

Sim. – Gaussian Processes

HAW HAMBURG

- Generalization
  - Set Noise Variance
  - Kernel scaling
  - Loss computation

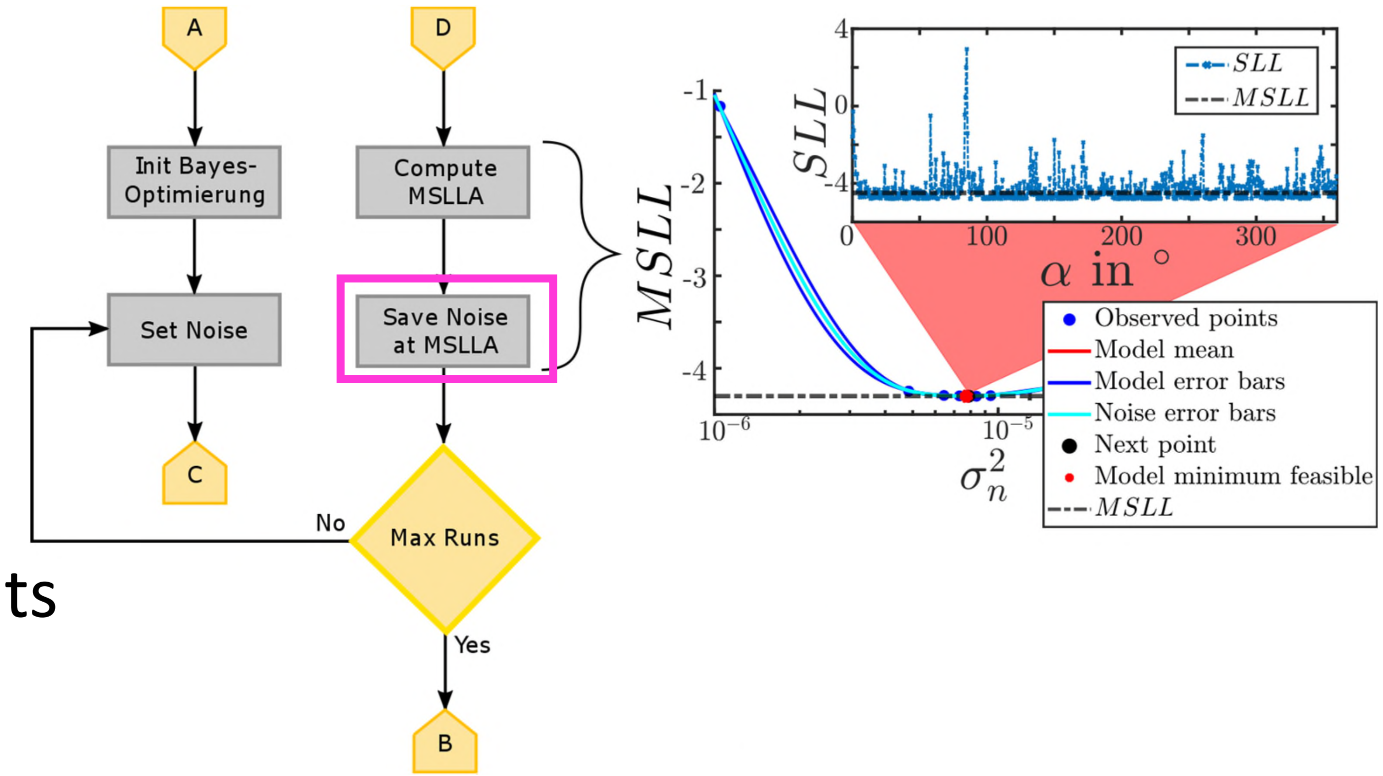




SLL – Standardized Logarithmic Loss
MSLL – Mean Standardized Logarithmic Loss
MSLLA – Mean Standardized Logarithmic Loss Angle

Sim. – Gaussian Processes

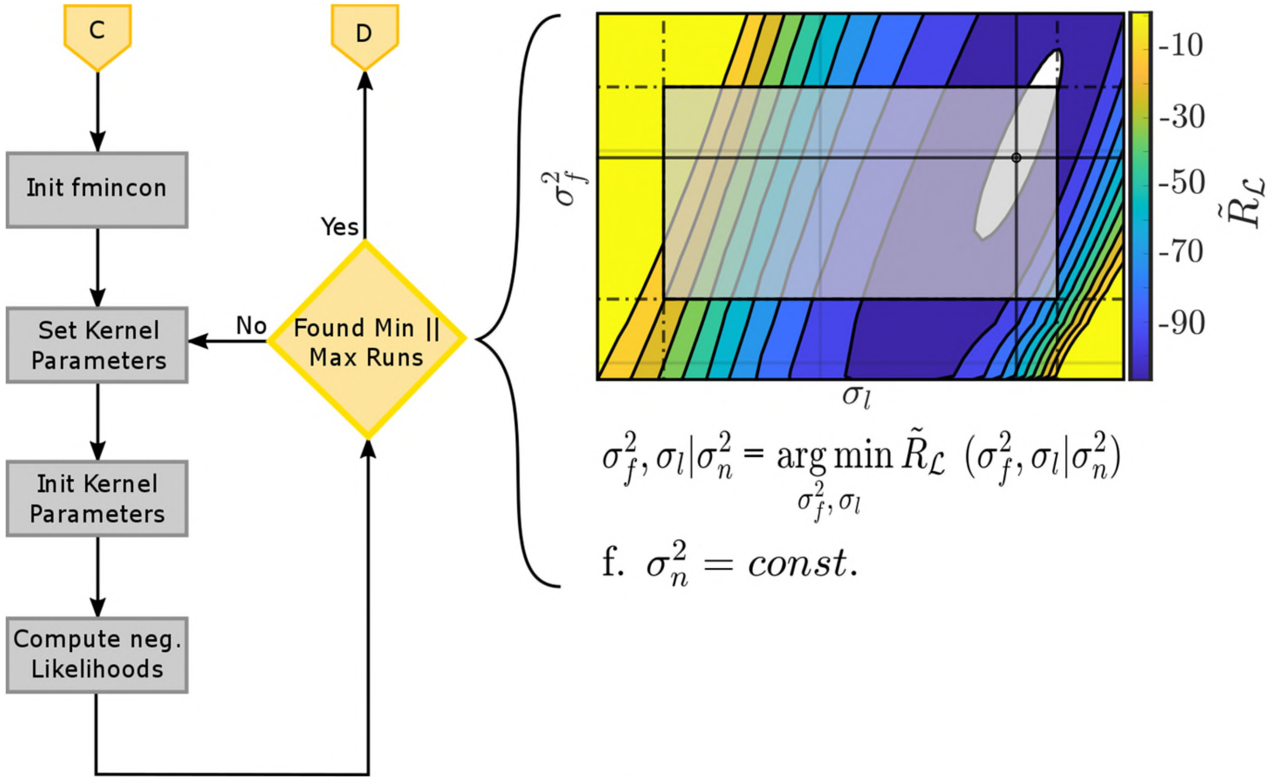
- Generalization
  - Set Noise Variance
  - Kernel scaling
  - Loss computation
  - Save intermediate results



SLL – Standardized Logarithmic Loss
MSLL – Mean Standardized Logarithmic Loss
MSLLA – Mean Standardized Logarithmic Loss Angle

Sim. – Gaussian Processes

- Kernel Scaling

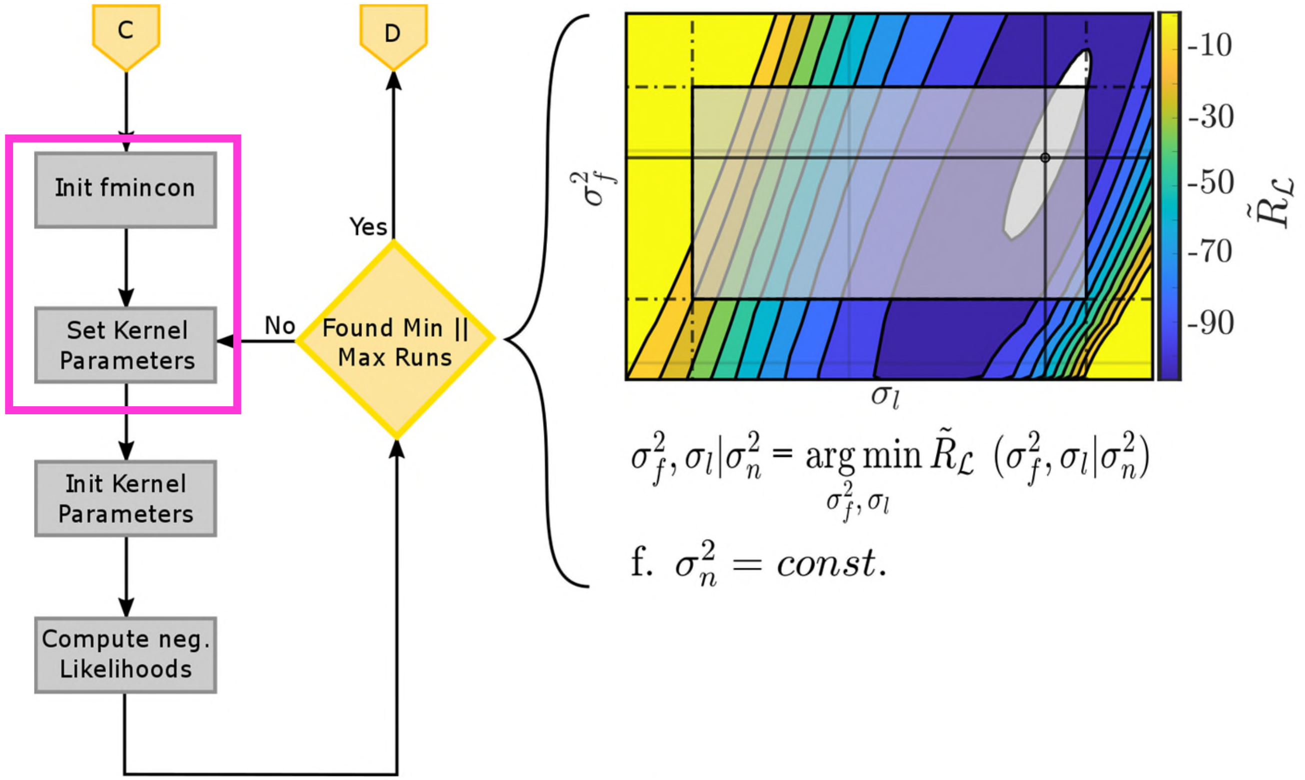


$$\sigma_f^2, \sigma_l | \sigma_n^2 = \arg\min_{\sigma_f^2, \sigma_l} \tilde{R}_{\mathcal{L}} \left(\sigma_f^2, \sigma_l | \sigma_n^2\right)$$

$$\text{f. } \sigma_n^2 = const.$$

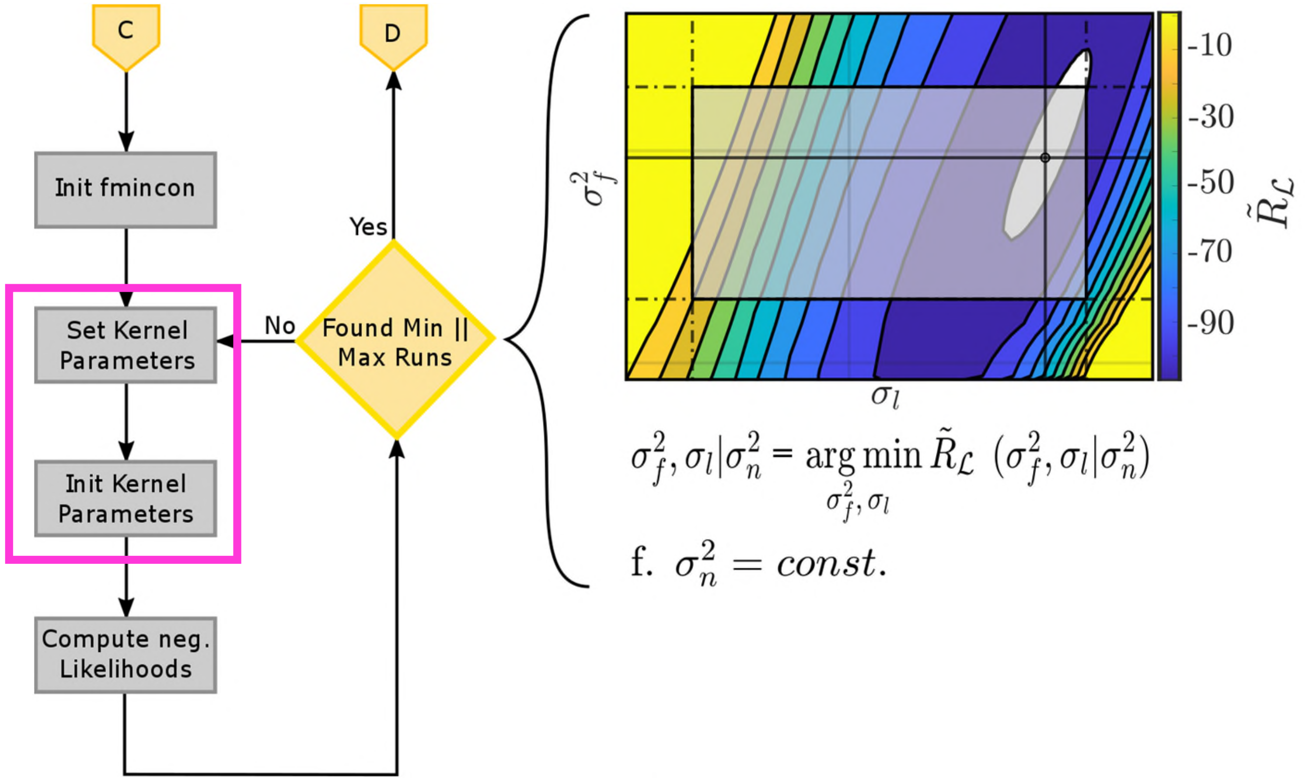
Sim. – Gaussian Processes

- Kernel Scaling
  - Set kernel parameter



Sim. – Gaussian Processes

- Kernel Scaling
  - Set kernel parameter
  - Partial reinitialization

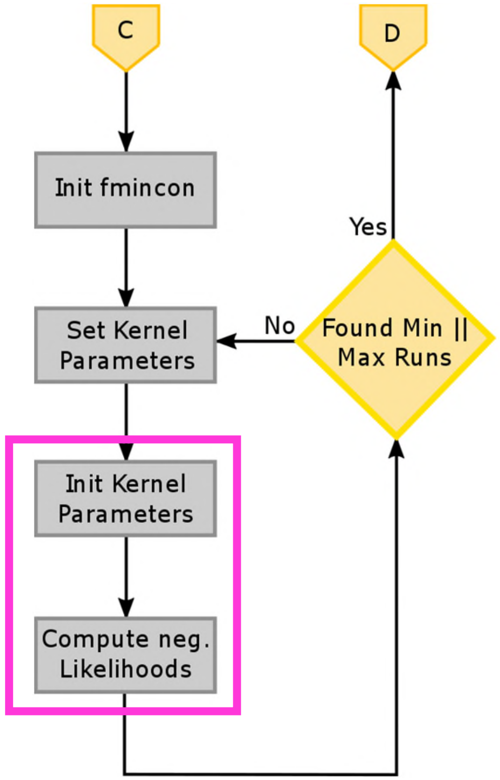


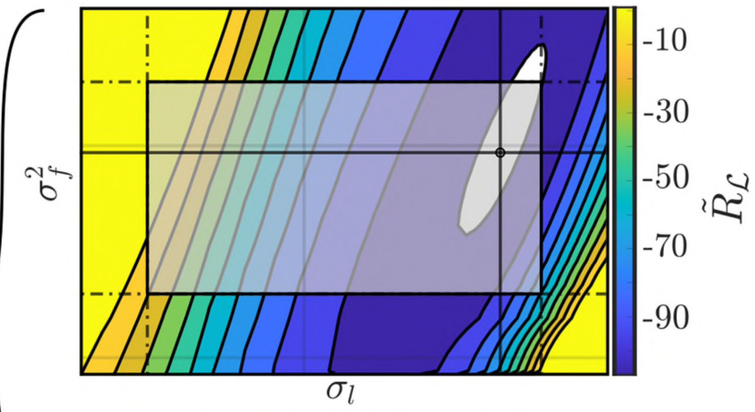
$$\sigma_f^2, \sigma_l | \sigma_n^2 = \underset{\sigma_f^2, \sigma_l}{\arg\min}\, \tilde{R}_{\mathcal{L}}\ (\sigma_f^2, \sigma_l | \sigma_n^2)$$

f. $\sigma_n^2 = const.$

Sim. – Gaussian Processes

- Kernel Scaling
  - Set kernel parameter
  - Partial reinitialization
  - Likelihood computation





$$\sigma_f^2, \sigma_l | \sigma_n^2 = \underset{\sigma_f^2, \sigma_l}{\arg\min}\, \tilde{R}_{\mathcal{L}} \left(\sigma_f^2, \sigma_l | \sigma_n^2\right)$$

f. $\sigma_n^2 = const.$

Sim. – Gaussian Processes

- Working mode

Sim. – Gaussian Processes

- Working mode
  - Switch model direction from fitting to prediction

Sim. – Gaussian Processes

- Working mode
  - Switch model direction from fitting to prediction
  - Minimized parameter set and functional driver for prediction

Sim. – Gaussian Processes

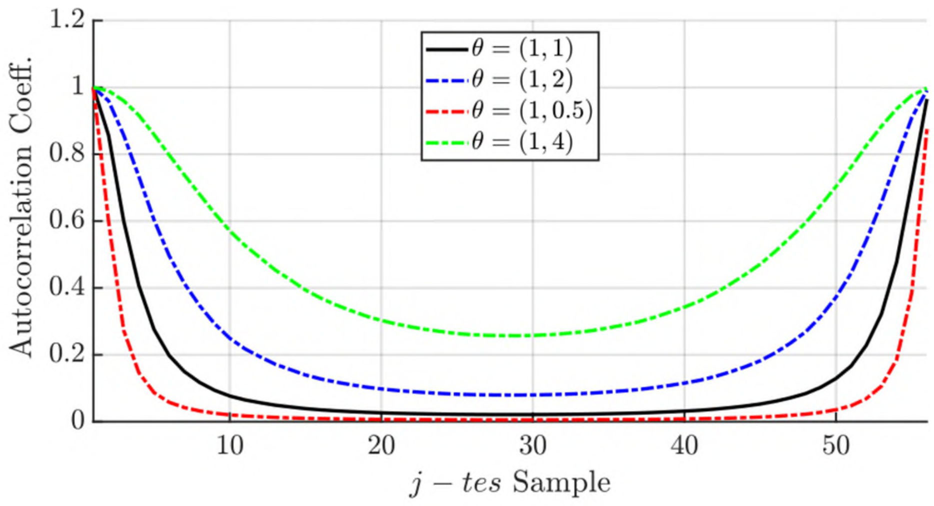
- Working mode
  - Switch model direction from fitting to prediction
  - Minimized parameter set and functional driver for prediction
  - Framewise or block wise prediction

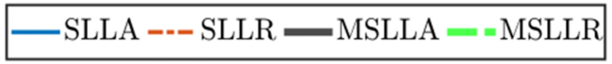
Sim. – Gaussian Processes

- Working mode
  - Switch model direction from fitting to prediction
  - Minimized parameter set and functional driver for prediction
  - Framewise or block wise prediction
  - Results, derivates and quality measures return as vectors

Testing Experiments

- Scaling of covariance function



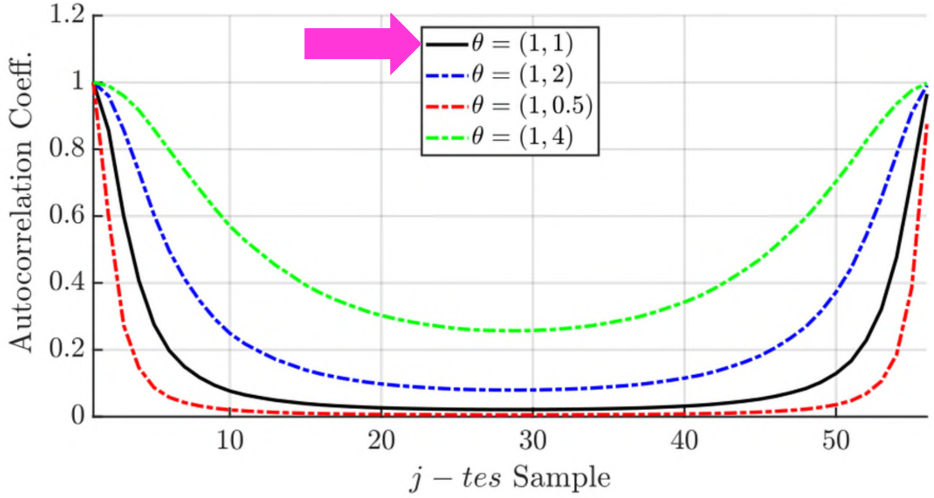


SLLA – Standardized Logarithmic Loss Angle
SLLR – Standardized Logarithmic Loss Radius
MSLLA – Mean Standardized Logarithmic Loss Angle
MSLLR – Mean Standardized Logarithmic Loss Radius

Testing Experiments

HAW HAMBURG

- Scaling of covariance function
  - Empirical without optimization



SLLA SLLR MSLLA MSLLR

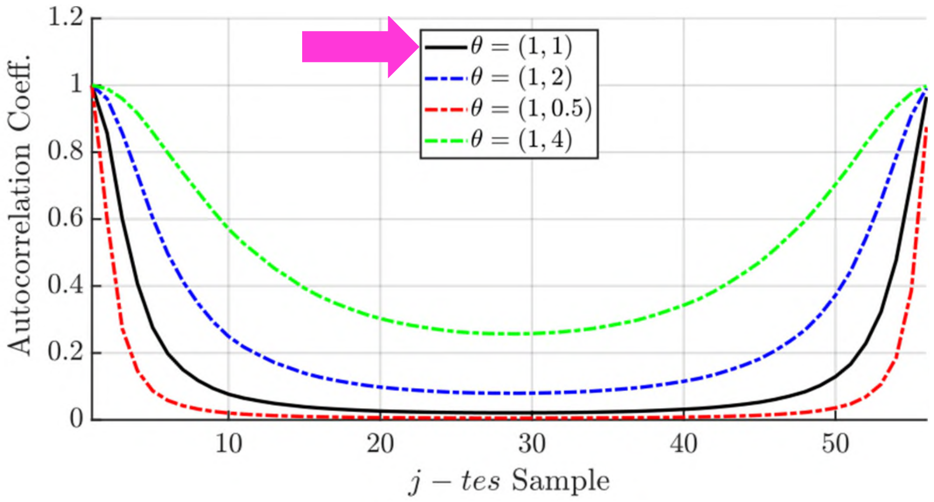
SLLA – Standardized Logarithmic Loss Angle
SLLR – Standardized Logarithmic Loss Radius
MSLLA – Mean Standardized Logarithmic Loss Angle
MSLLR – Mean Standardized Logarithmic Loss Radius

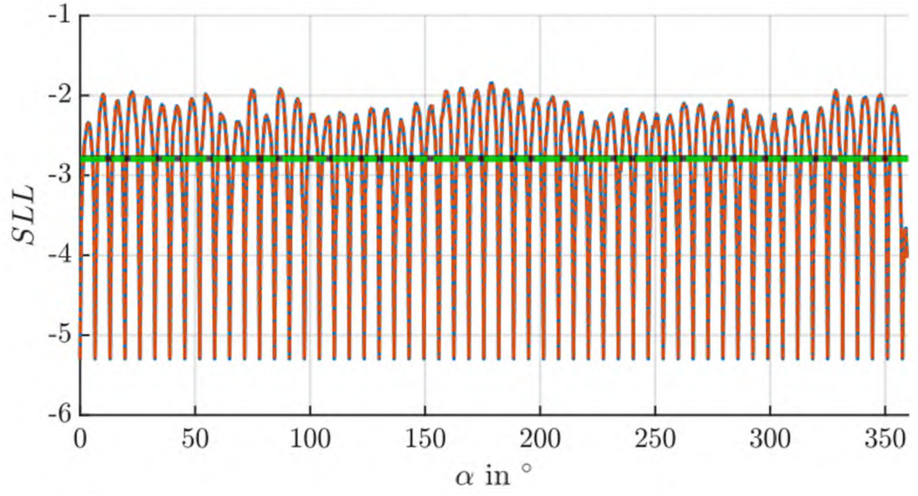
Testing Experiments

HAW HAMBURG

- Scaling of covariance function
  - Empirical without optimization
  - Value the generalization

Disabled





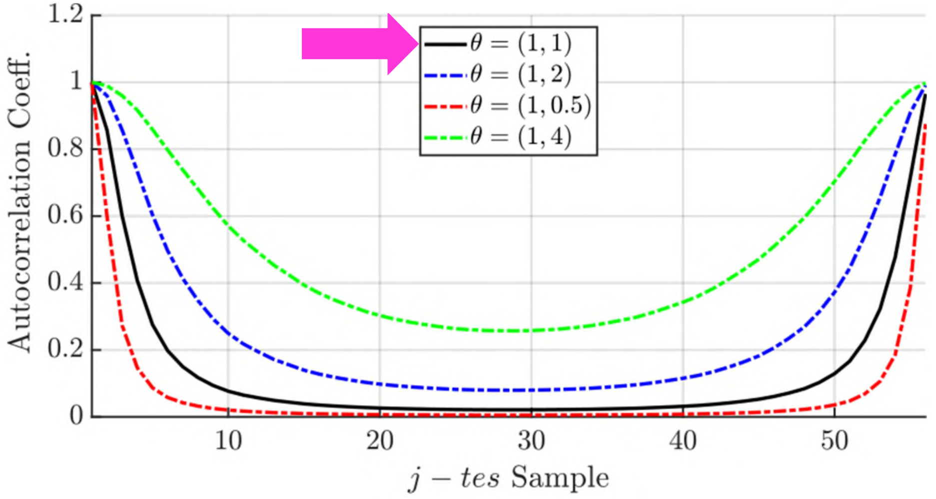
SLLA SLLR MSLLA MSLLR

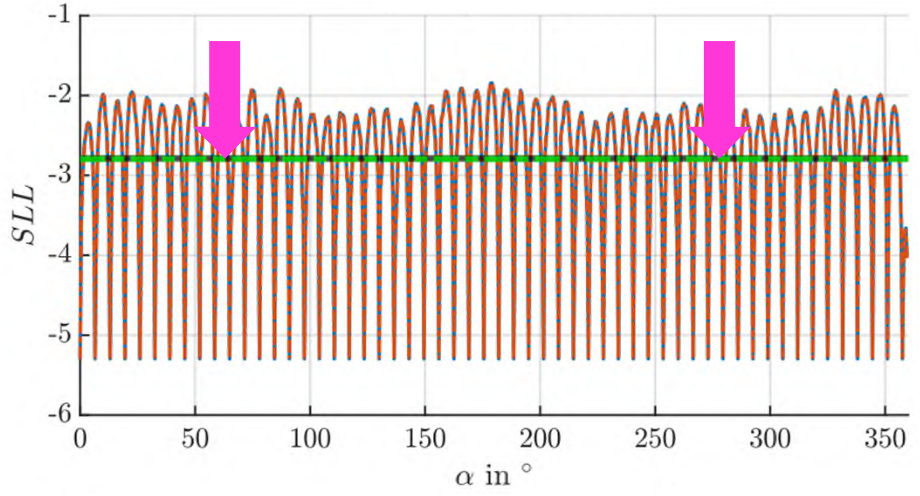
SLLA – Standardized Logarithmic Loss Angle
SLLR – Standardized Logarithmic Loss Radius
MSLLA – Mean Standardized Logarithmic Loss Angle
MSLLR – Mean Standardized Logarithmic Loss Radius

Testing Experiments

- Scaling of covariance function
  - Empirical without optimization
  - Value the generalization

Disabled





SLLA - - - SLLR — MSLLA - - MSLLR

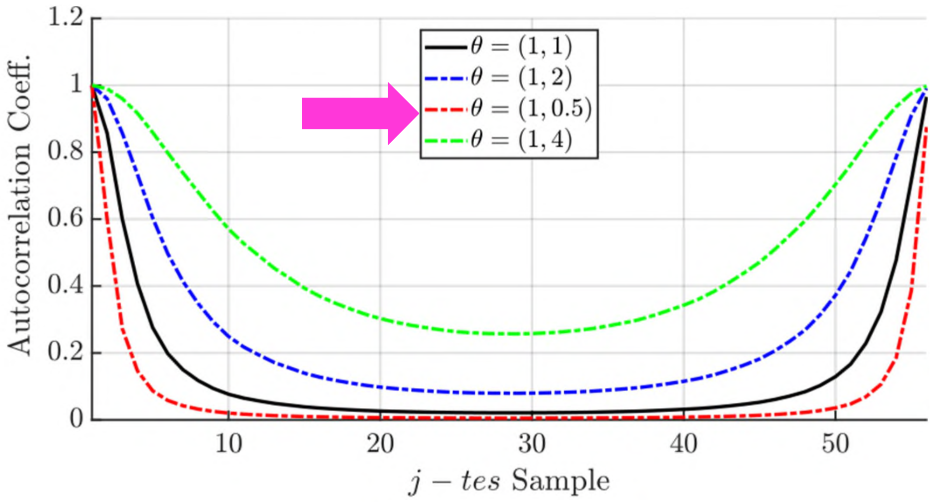
SLLA – Standardized Logarithmic Loss Angle
SLLR – Standardized Logarithmic Loss Radius
MSLLA – Mean Standardized Logarithmic Loss Angle
MSLLR – Mean Standardized Logarithmic Loss Radius

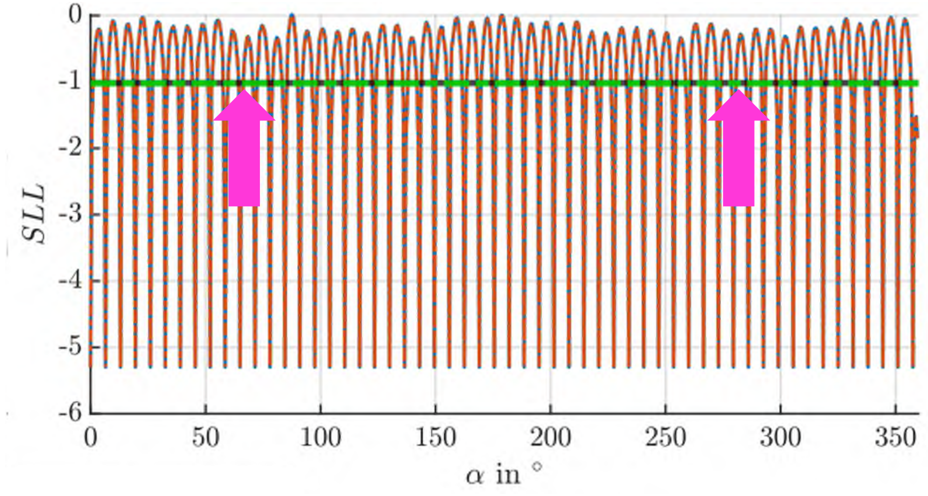
Testing Experiments

HAW HAMBURG

- Scaling of covariance function
  - Empirical without optimization
  - Value the generalization

Overfitting





SLLA --- SLLR --- MSLLA --- MSLLR

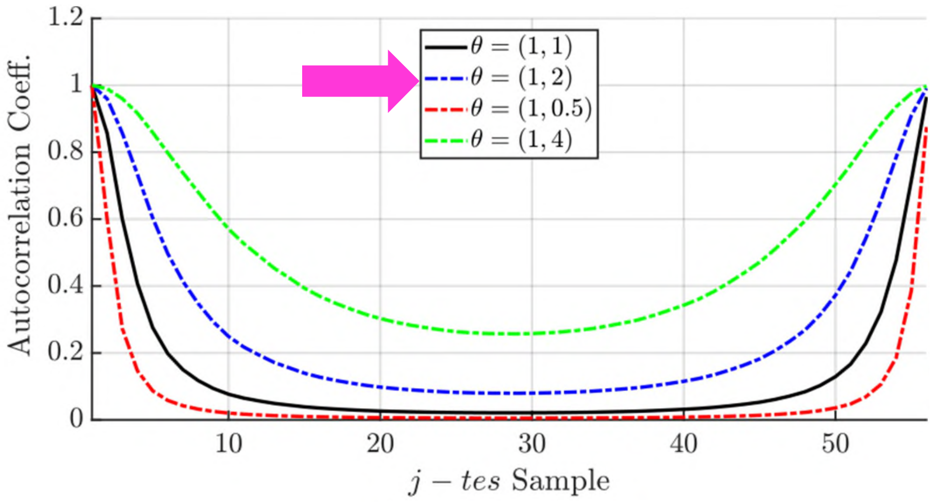
SLLA – Standardized Logarithmic Loss Angle
SLLR – Standardized Logarithmic Loss Radius
MSLLA – Mean Standardized Logarithmic Loss Angle
MSLLR – Mean Standardized Logarithmic Loss Radius

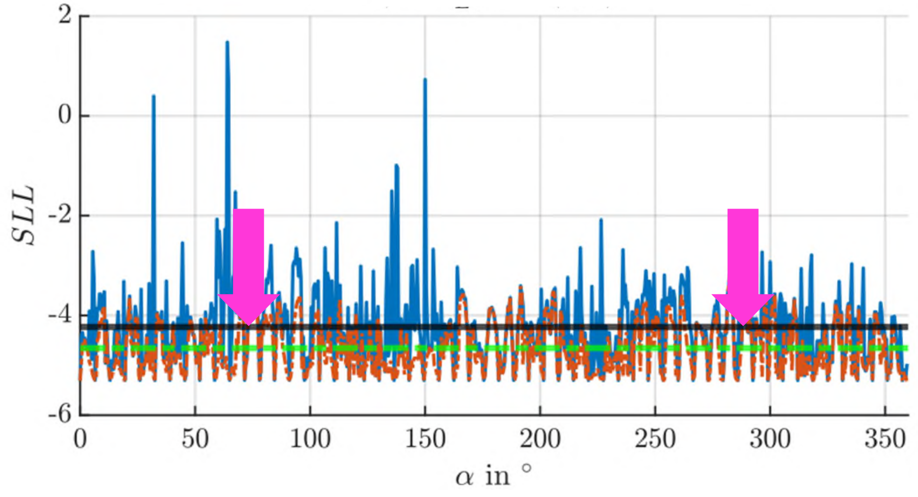
Testing Experiments

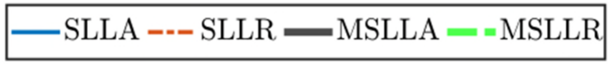
HAW HAMBURG

- Scaling of covariance function
  - Empirical without optimization
  - Value the generalization

Improved







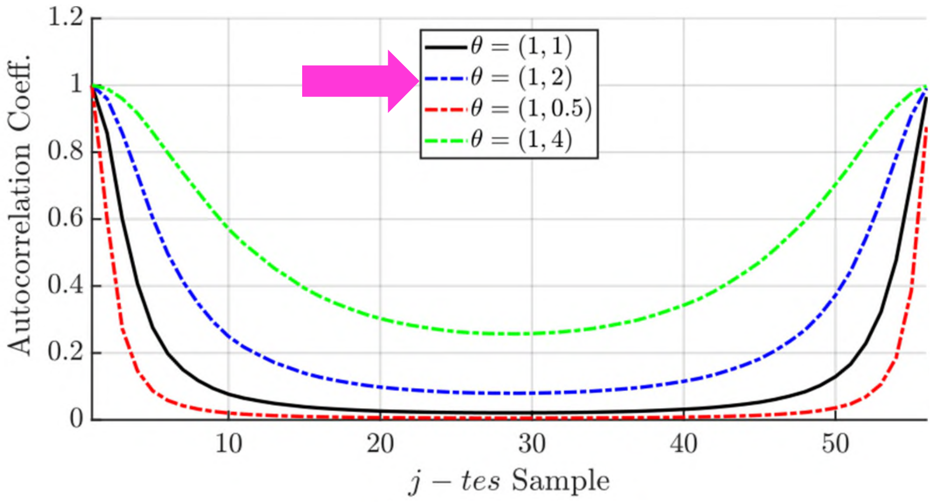
SLLA – Standardized Logarithmic Loss Angle
SLLR – Standardized Logarithmic Loss Radius
MSLLA – Mean Standardized Logarithmic Loss Angle
MSLLR – Mean Standardized Logarithmic Loss Radius

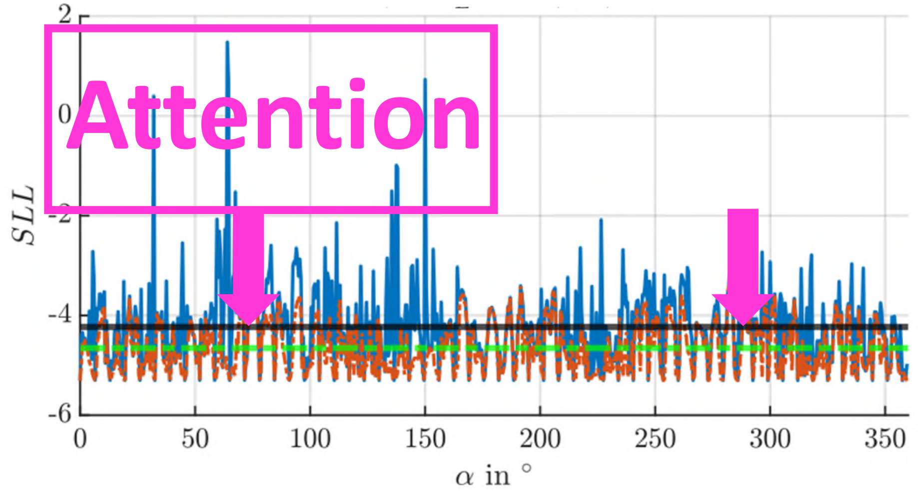
Testing Experiments

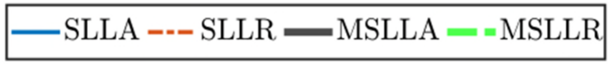
HAW HAMBURG

- Scaling of covariance function
  - Empirical without optimization
  - Value the generalization

Improved





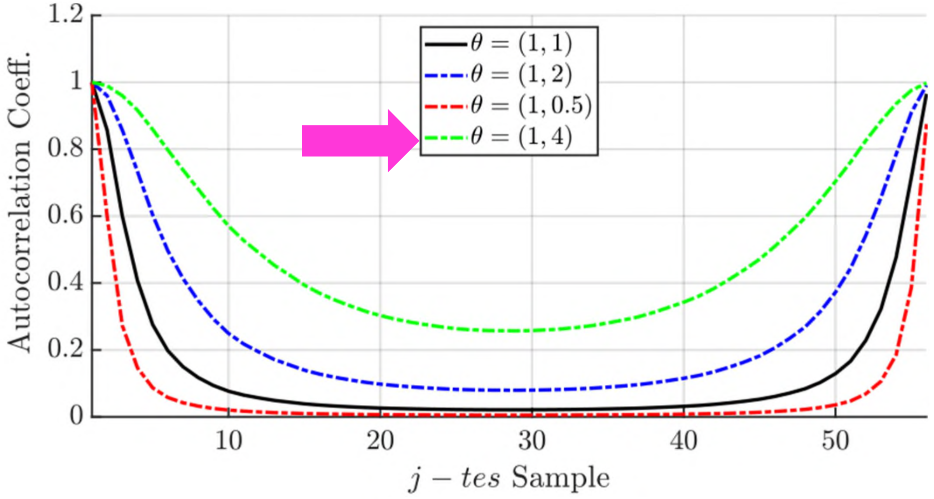


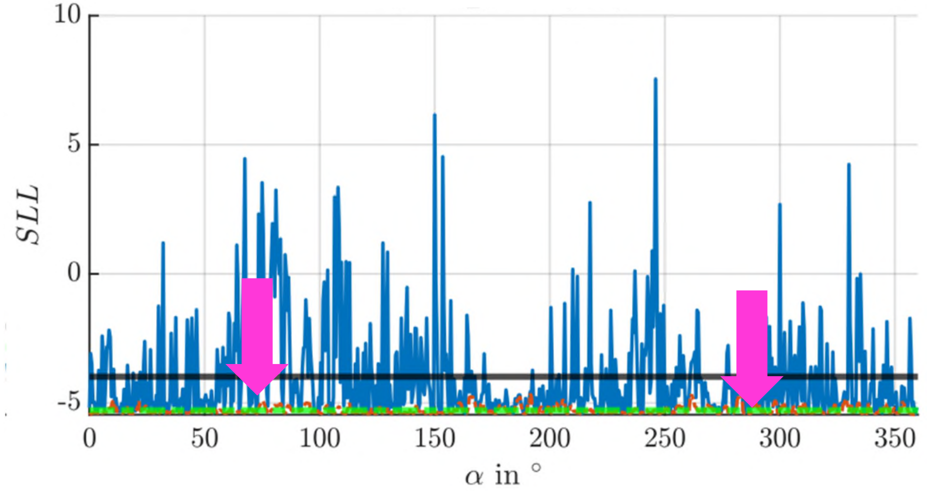
SLLA – Standardized Logarithmic Loss Angle
SLLR – Standardized Logarithmic Loss Radius
MSLLA – Mean Standardized Logarithmic Loss Angle
MSLLR – Mean Standardized Logarithmic Loss Radius

Testing Experiments

- Scaling of covariance function
    - Empirical without optimization
    - Value the generalization

Optimizable





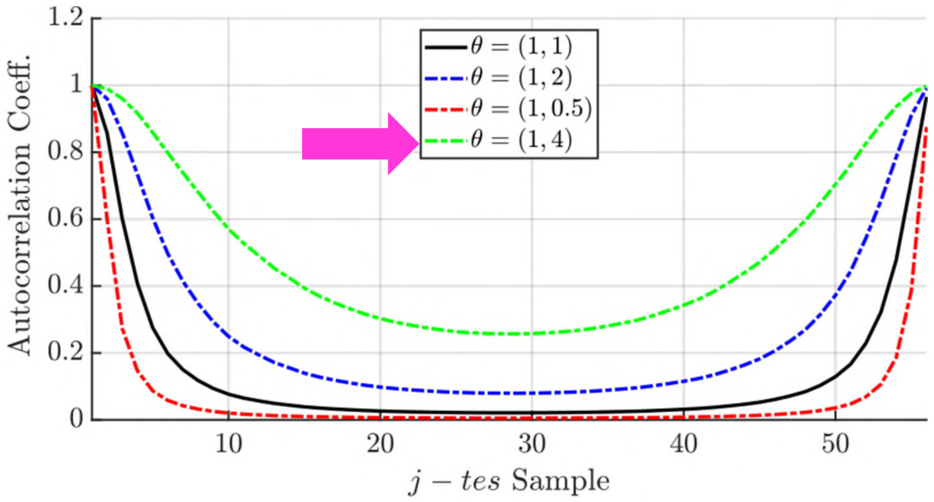
SLLA SLLR MSLLA MSLLR

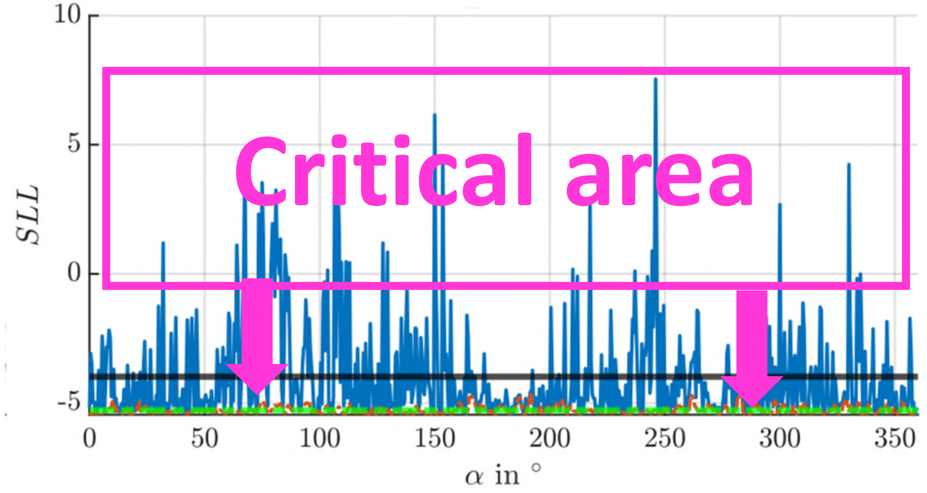
SLLA – Standardized Logarithmic Loss Angle
SLLR – Standardized Logarithmic Loss Radius
MSLLA – Mean Standardized Logarithmic Loss Angle
MSLLR – Mean Standardized Logarithmic Loss Radius

Testing Experiments

- Scaling of covariance function
  - Empirical without optimization
  - Value the generalization

Optimizable



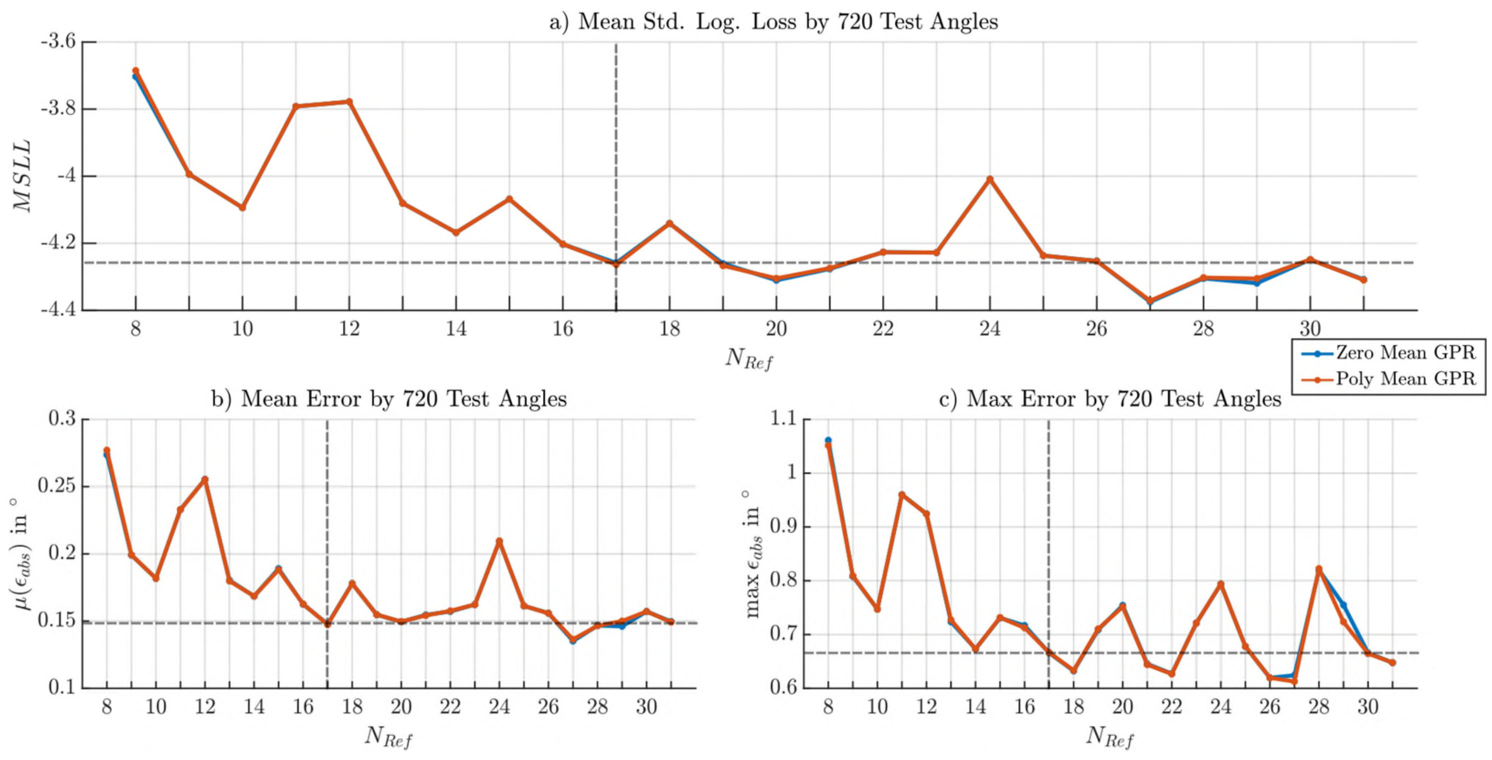


SLLA --- SLLR --- MSLLA --- MSLLR

SLLA – Standardized Logarithmic Loss Angle
SLLR – Standardized Logarithmic Loss Radius
MSLLA – Mean Standardized Logarithmic Loss Angle
MSLLR – Mean Standardized Logarithmic Loss Radius

Testing Experiments

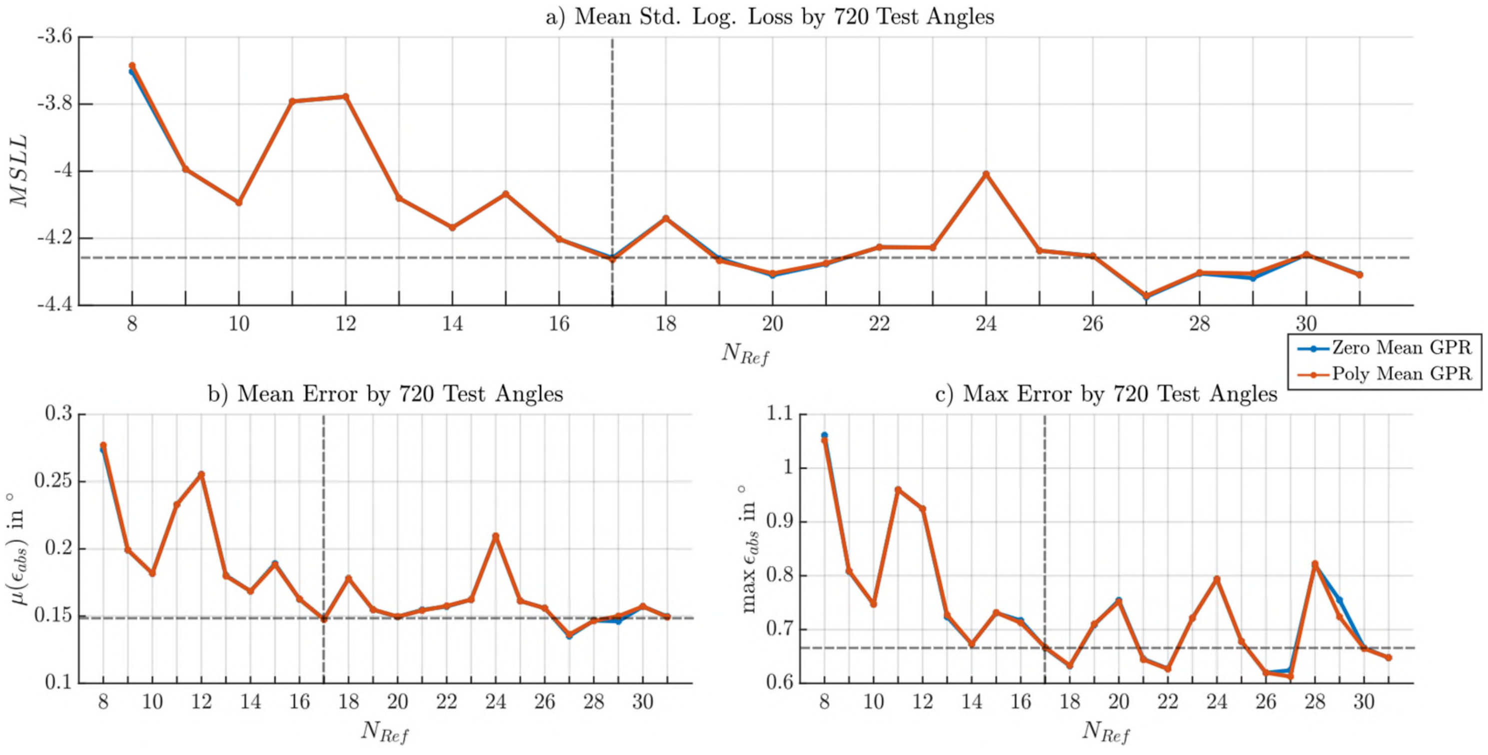
- Trade-off



MSLL – Mean-Standardized-Logarithmic-Loss

Testing Experiments

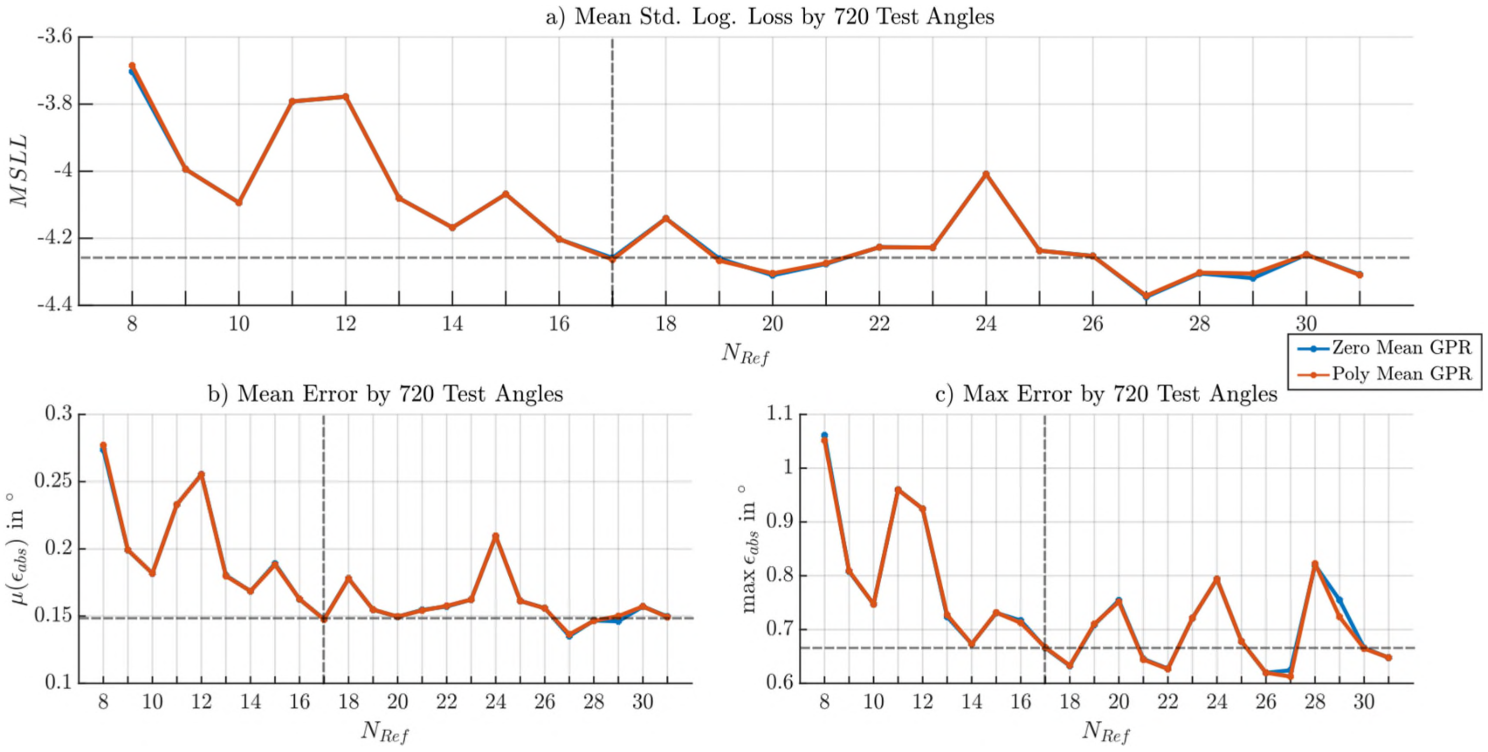
- Trade-off
  - Switch on optimization



MSLL – Mean-Standardized-Logarithmic-Loss

Testing Experiments

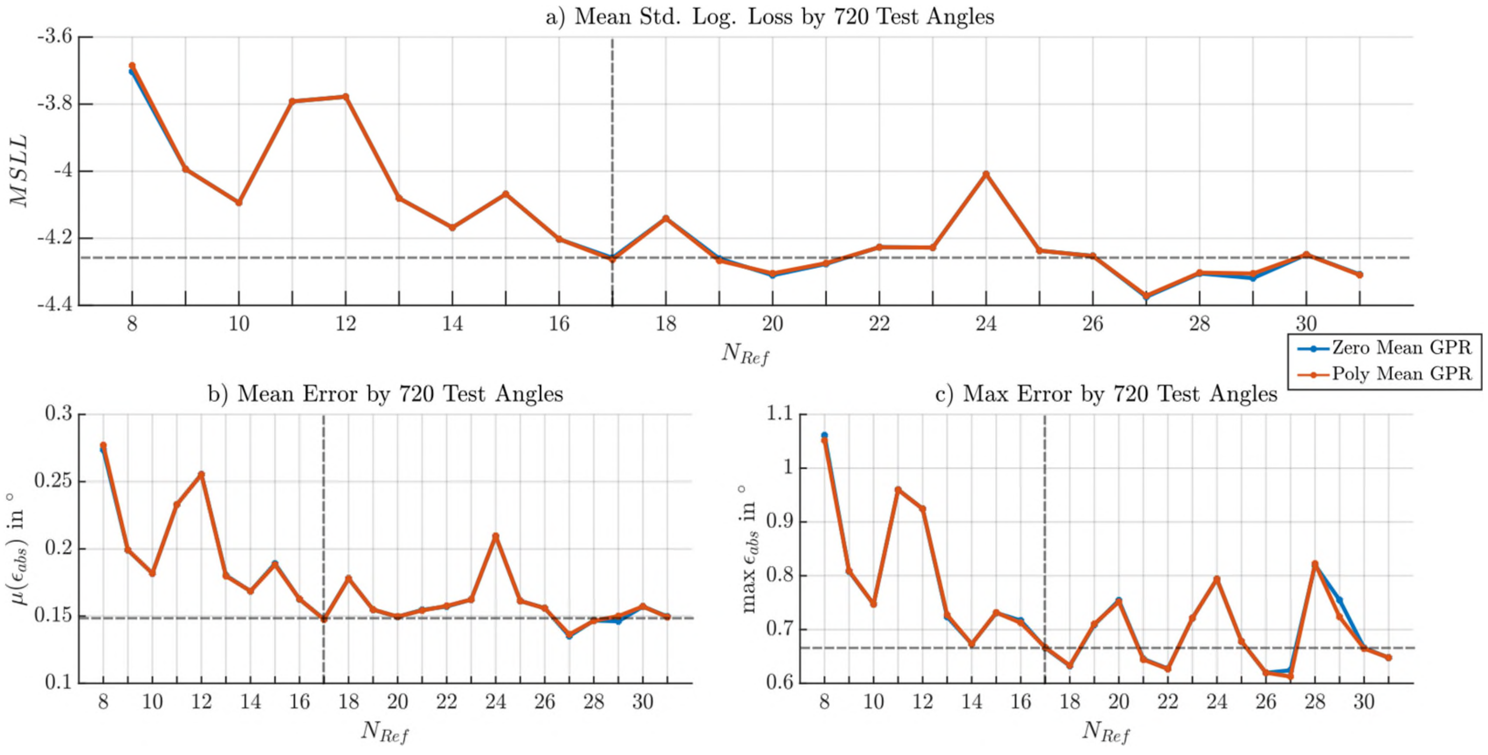
- Trade-off
  - Switch on optimization
  - Resources
  - Effort



MSLL – Mean-Standardized-Logarithmic-Loss

Testing Experiments

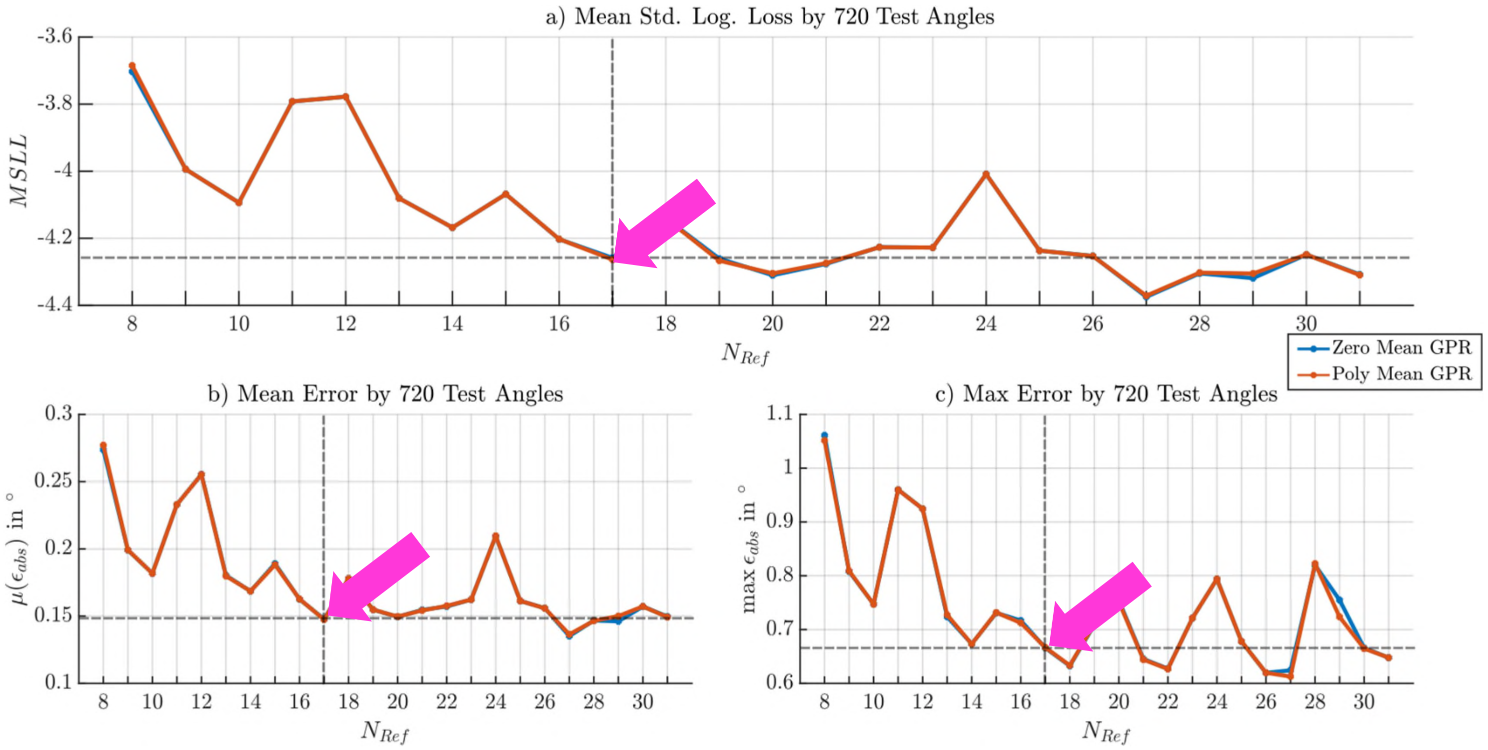
- Trade-off
  - Switch on optimization
  - Resources
  - Effort



MSLL – Mean-Standardized-Logarithmic-Loss

Testing Experiments

- Trade-off
  - Switch on optimization
  - Resources
  - Effort
  - Balancing error and loss

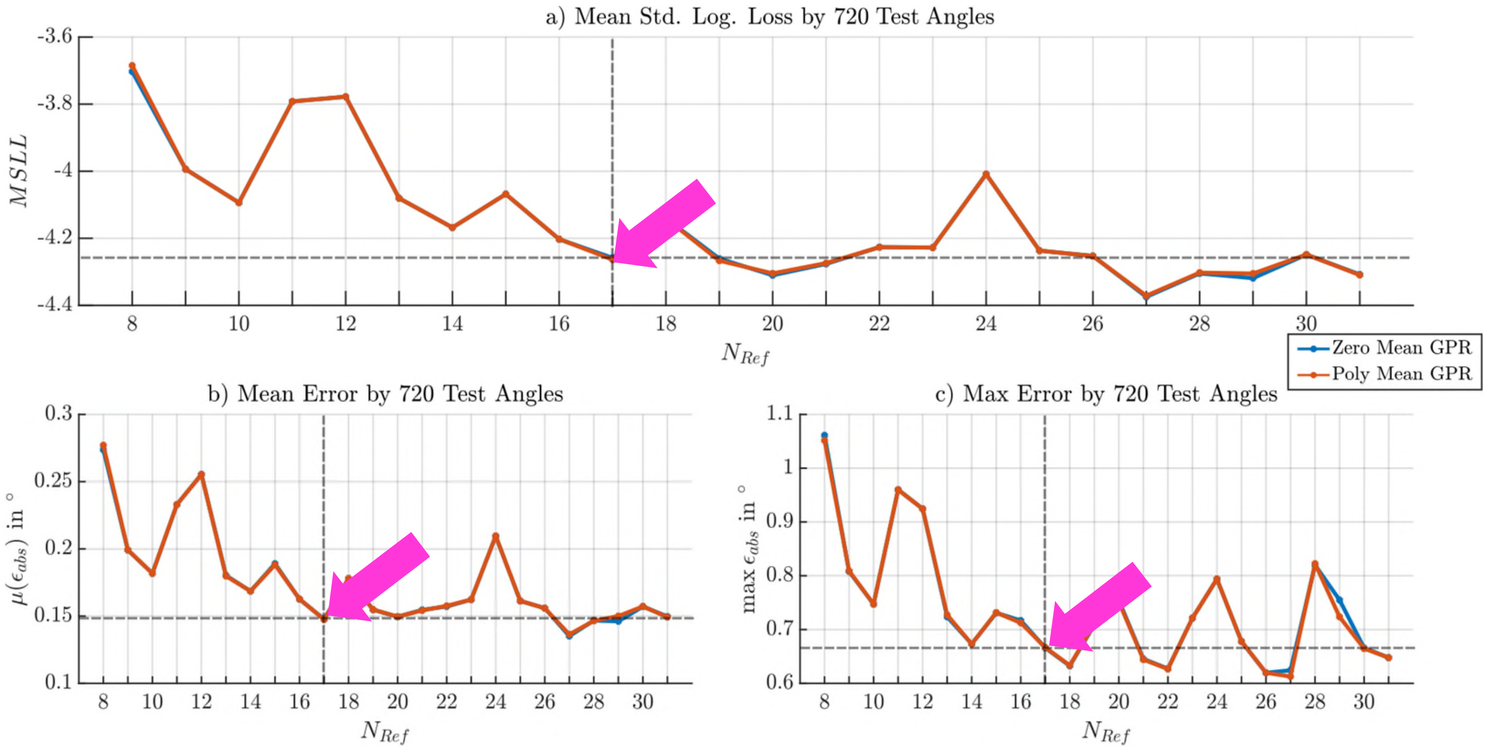


MSLL – Mean-Standardized-Logarithmic-Loss

Testing Experiments

17 Reference angles

- Trade-off
  - Switch on optimization
  - Resources
  - Effort
  - Balancing error and loss



MSLL – Mean-Standardized-Logarithmic-Loss

Testing Experiments

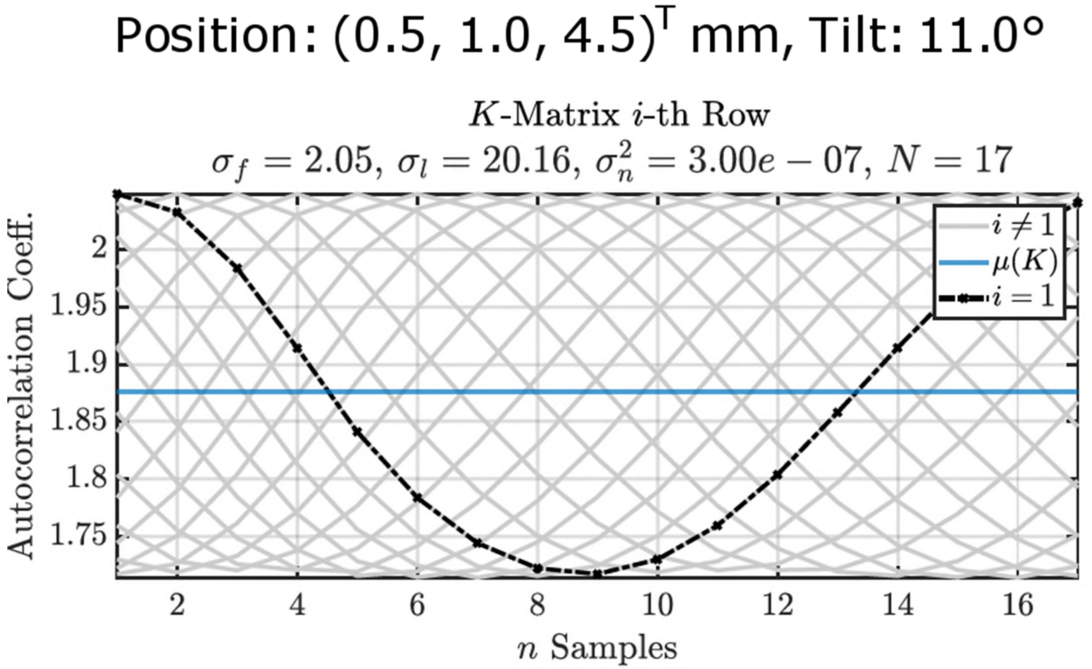
Position: $(0.5, 1.0, 4.5)^T$ mm, Tilt: 11.0°

Testing Experiments

Position: $(0.5, 1.0, 4.5)^T$ mm, Tilt: 11.0°

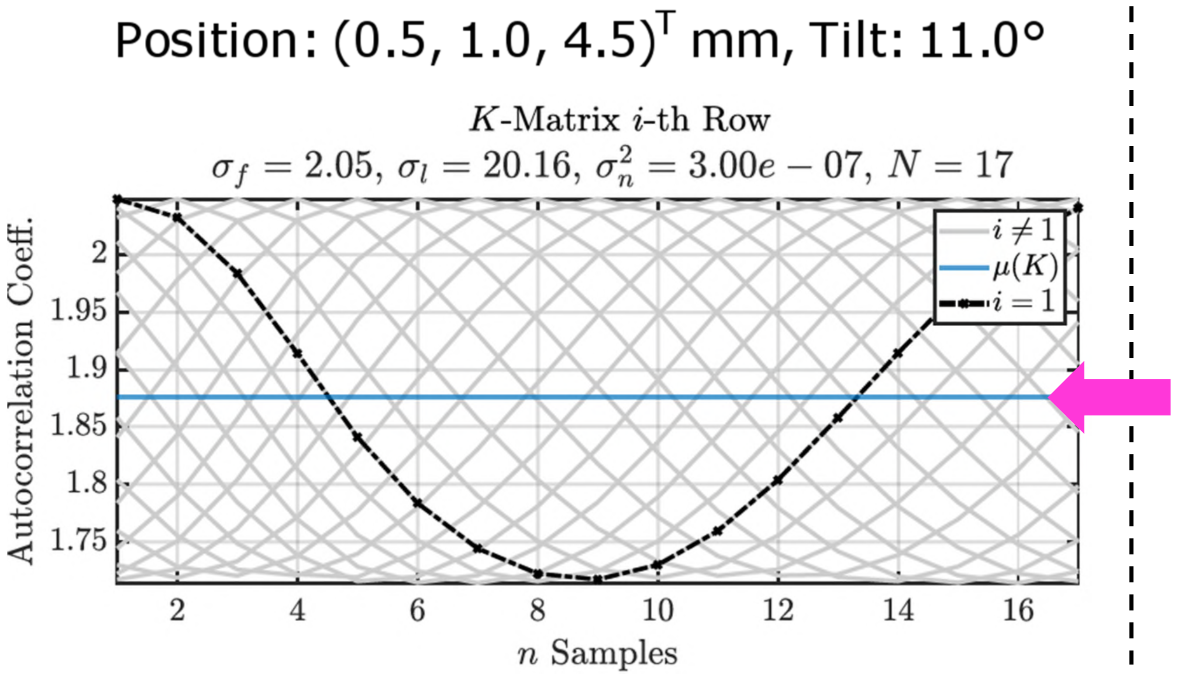
- Drifted in X-/ Y-direction
- Small distance
- Tilt magnet

Testing Experiments



- Drifted in X-/ Y-direction
- Small distance
- Tilt magnet

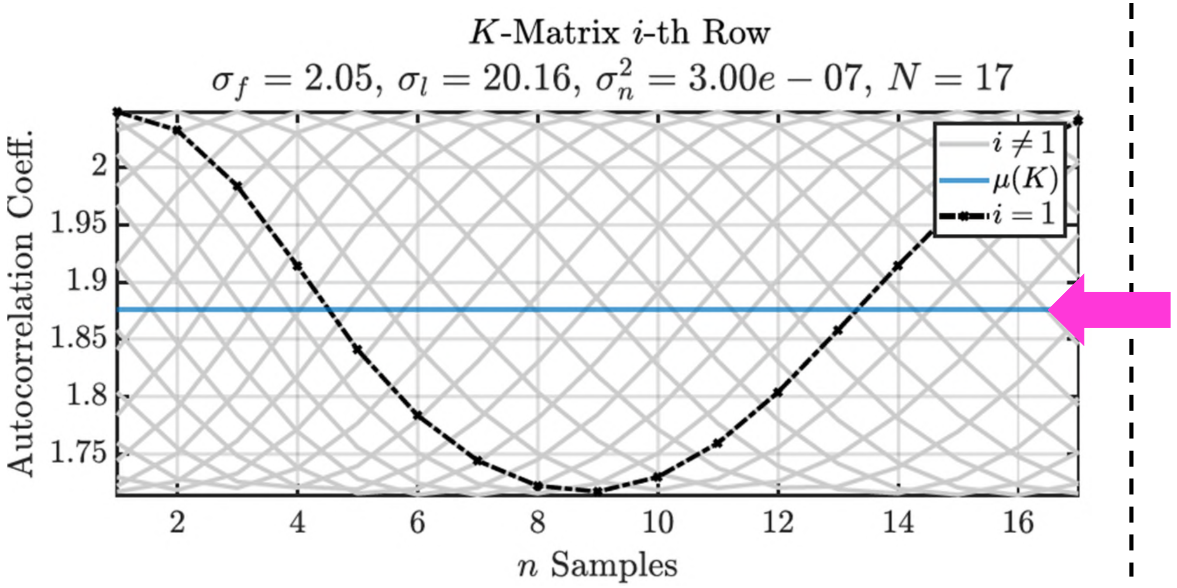
Testing Experiments



- Drifted in X-/ Y-direction
- Small distance
- Tilt magnet
- Balancing references

Testing Experiments

Position: $(0.5, 1.0, 4.5)^T$ mm, Tilt: 11.0°

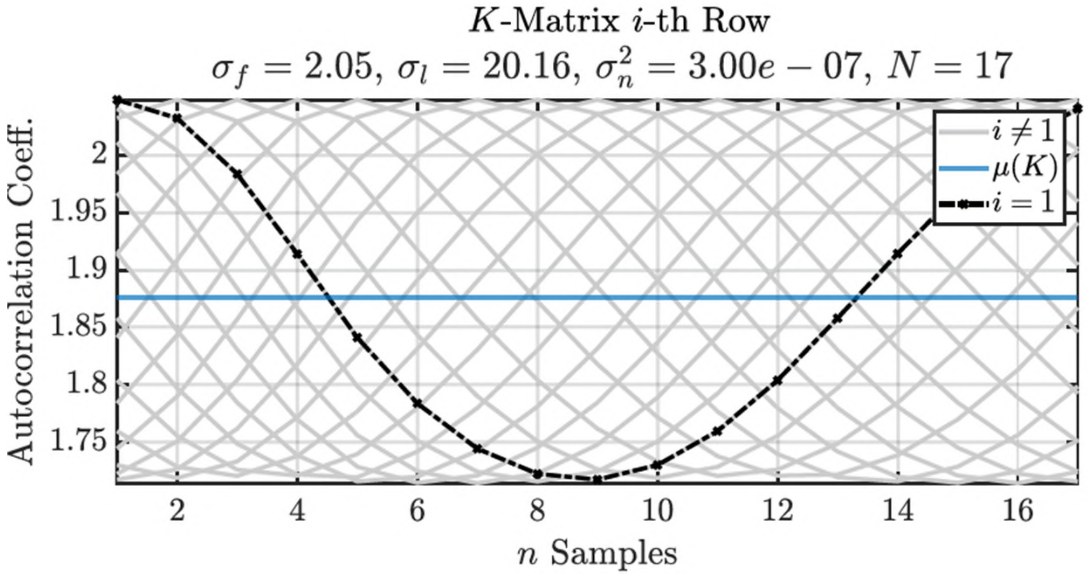


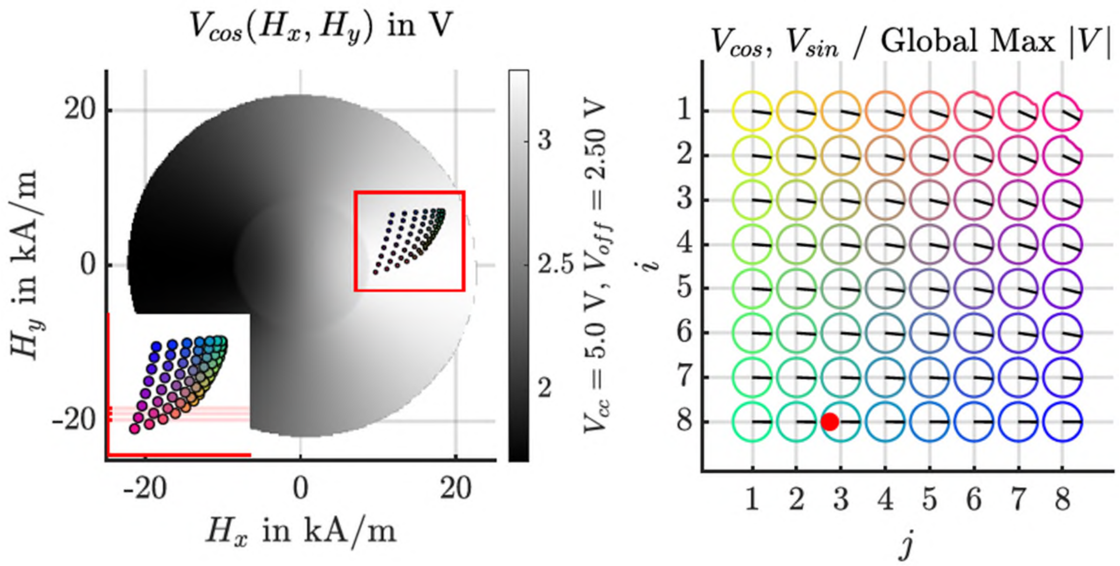
Good preconditions

- Drifted in X-/ Y-direction
- Small distance
- Tilt magnet
- Balancing references
- Scaling ration nearby 1:10

Testing Experiments

Position: $(0.5, 1.0, 4.5)^T$ mm, Tilt: 11.0°

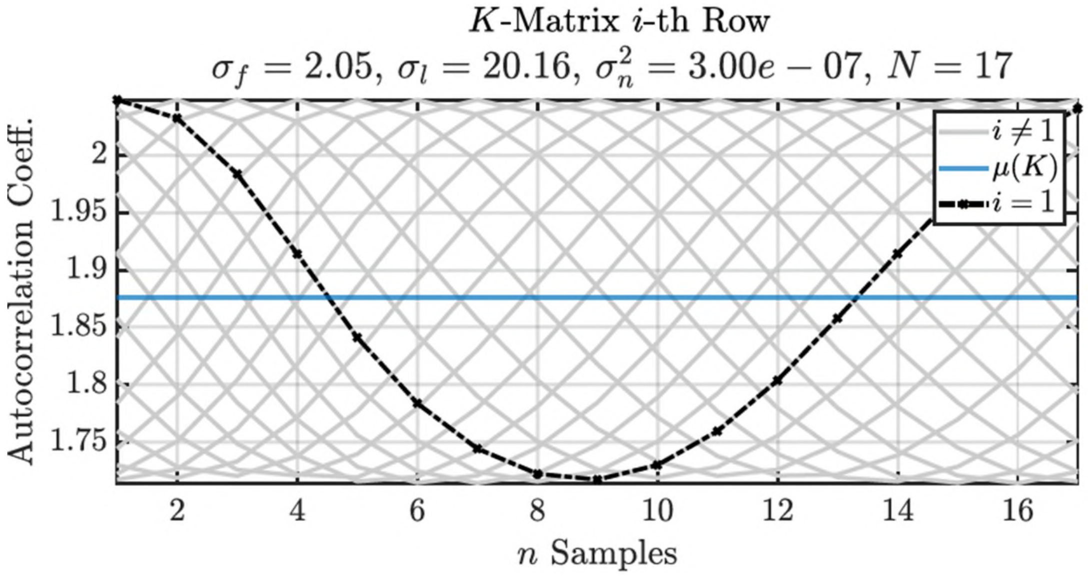


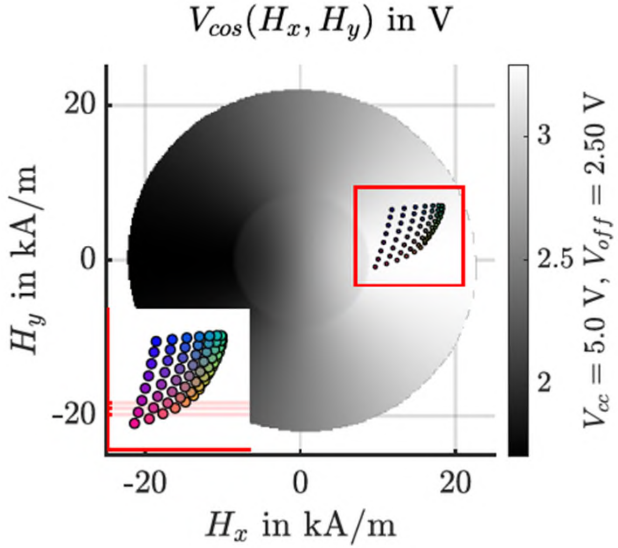


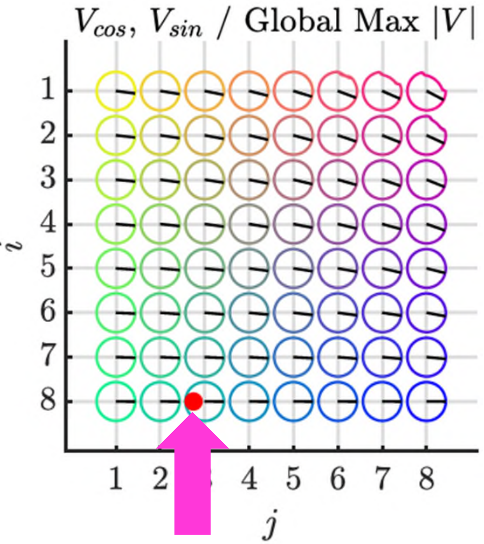
Testing Experiments

HAW HAMBURG

Position: $(0.5, 1.0, 4.5)^T$ mm, Tilt: 11.0°



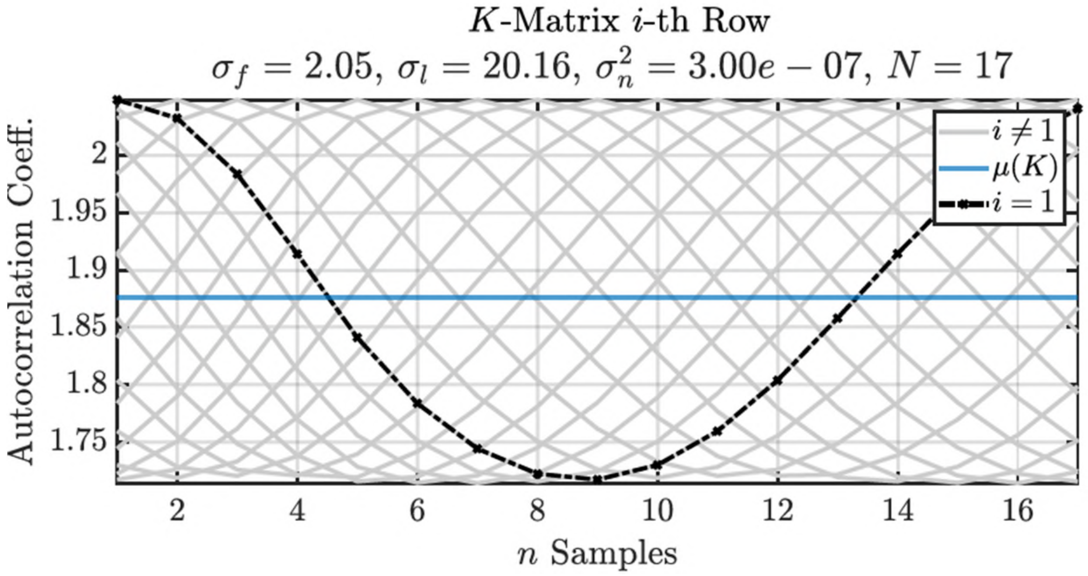




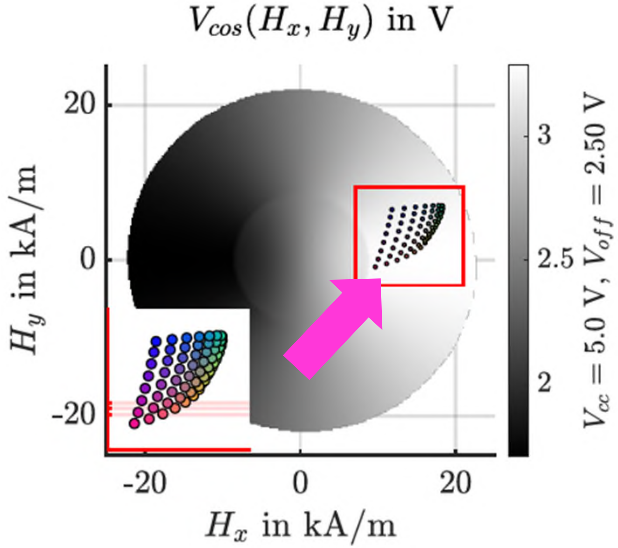
Magnet at array edge

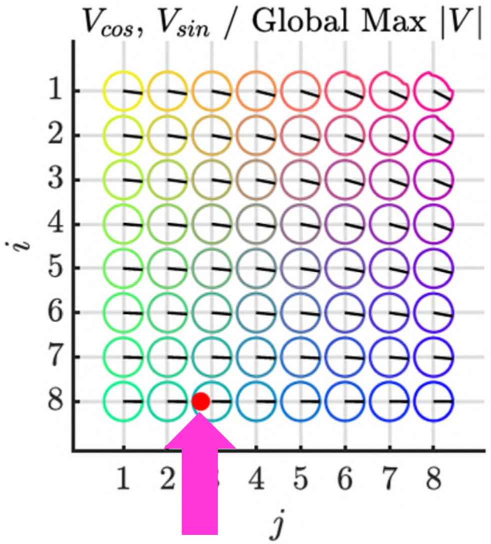
Testing Experiments

Position: $(0.5, 1.0, 4.5)^T$ mm, Tilt: 11.0°



Saturation + Dispersion

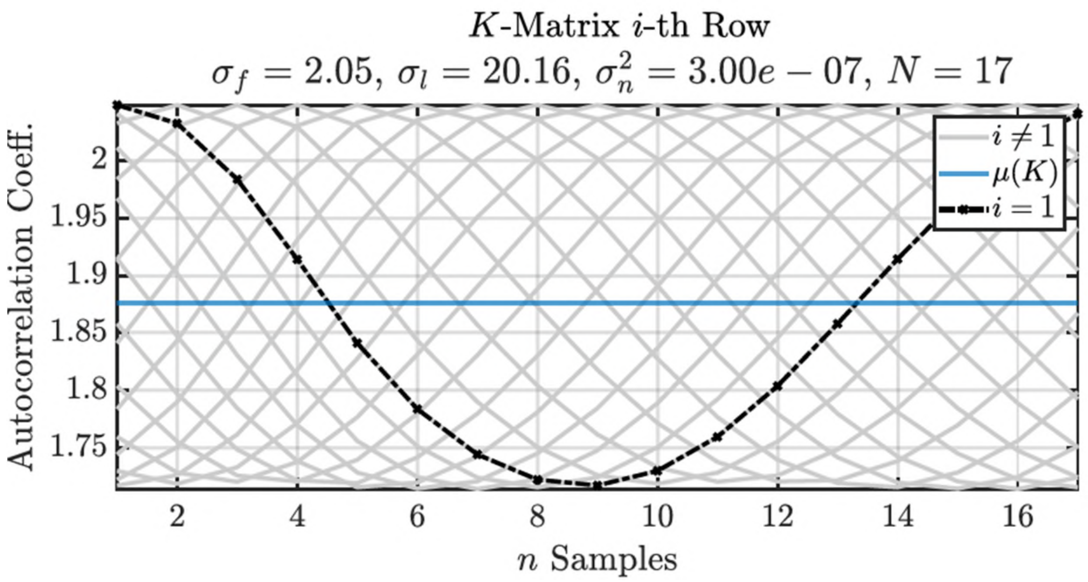




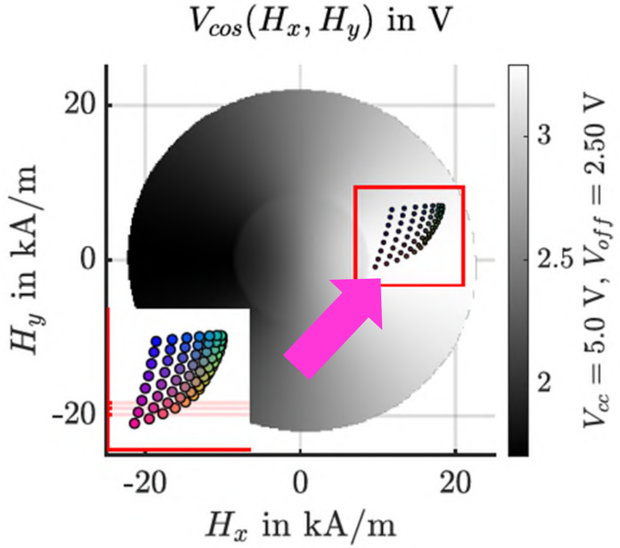
Magnet at array edge

Testing Experiments

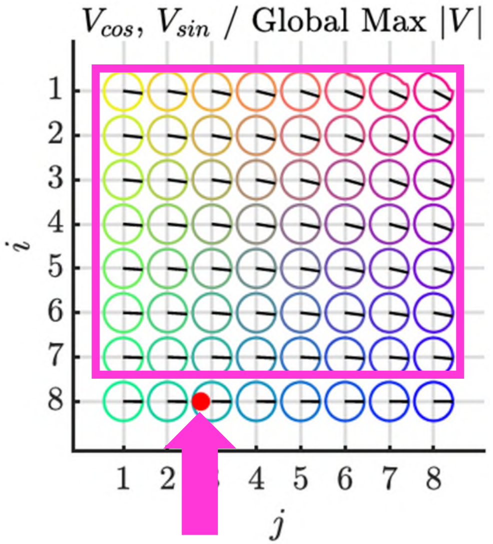
Position: $(0.5, 1.0, 4.5)^{T}$ mm, Tilt: 11.0°



Saturation + Dispersion



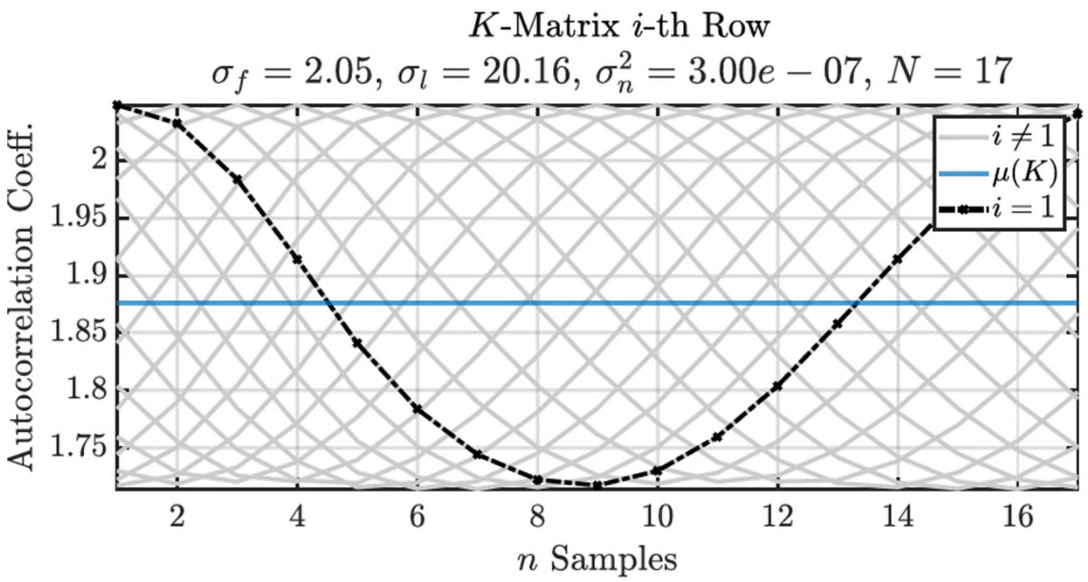
Misaligned angles

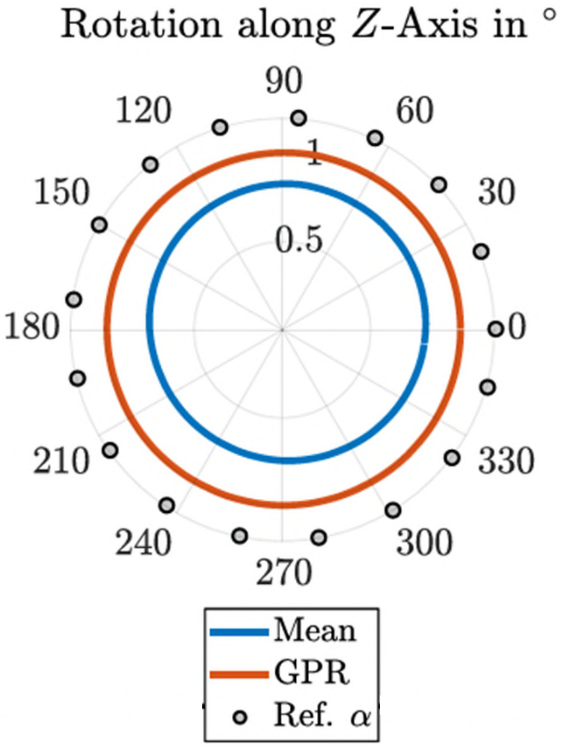


Magnet at array edge

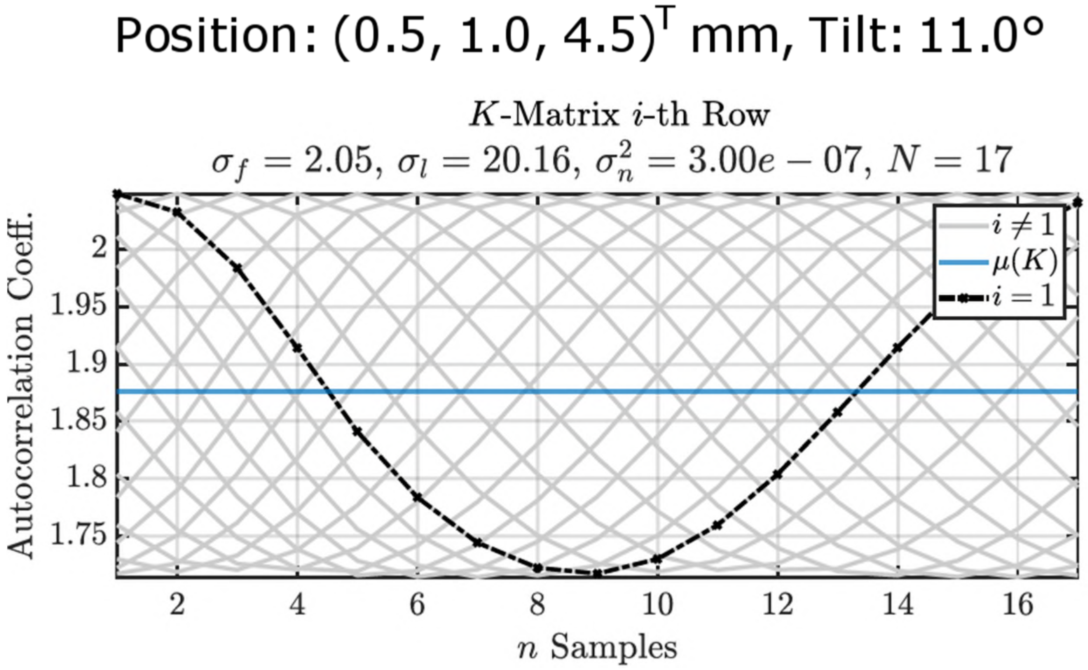
Testing Experiments

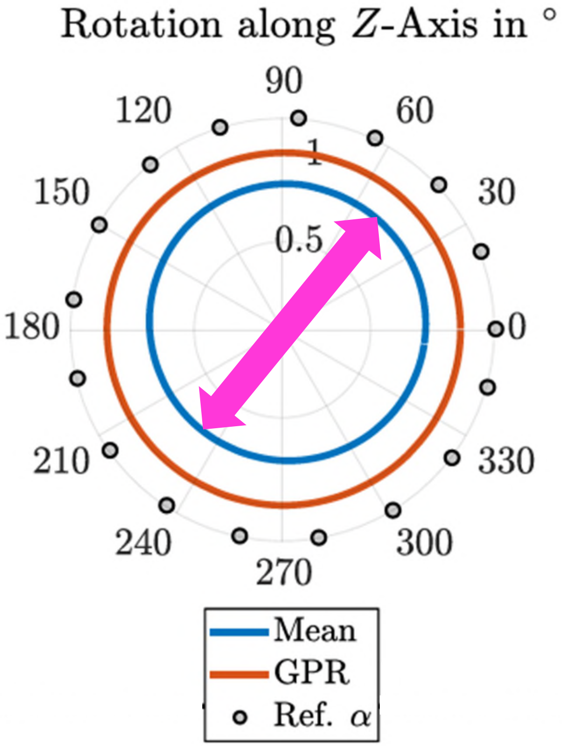
Position: $(0.5, 1.0, 4.5)^T$ mm, Tilt: 11.0°





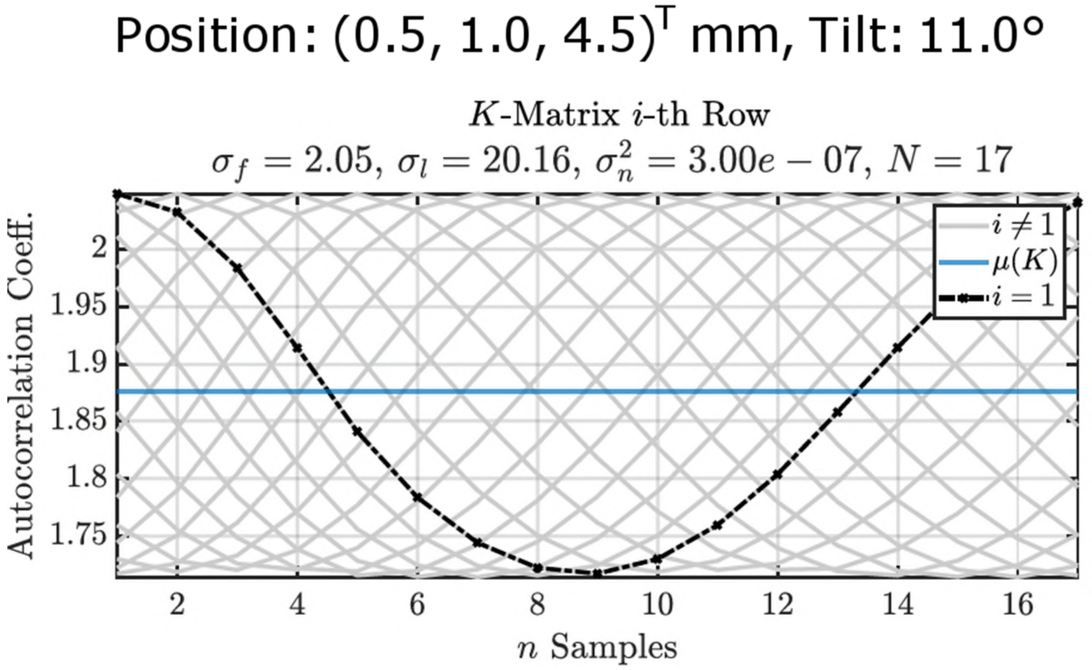
Testing Experiments

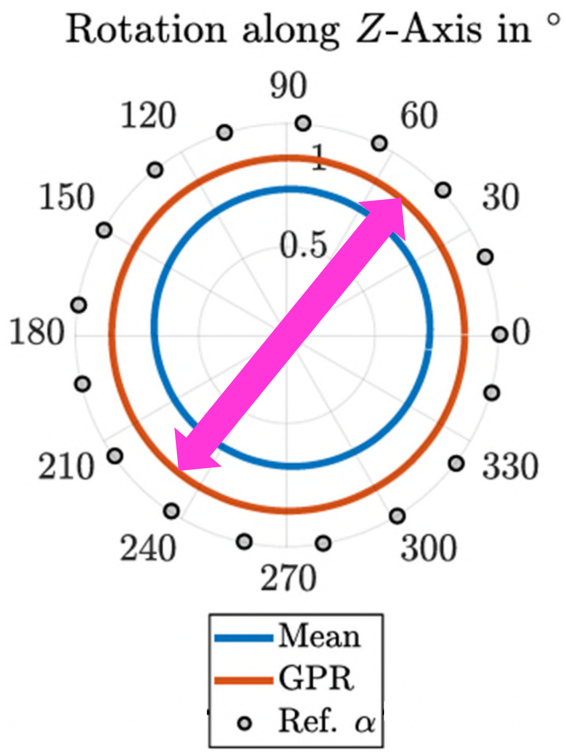




- Simple mean is drifted

Testing Experiments

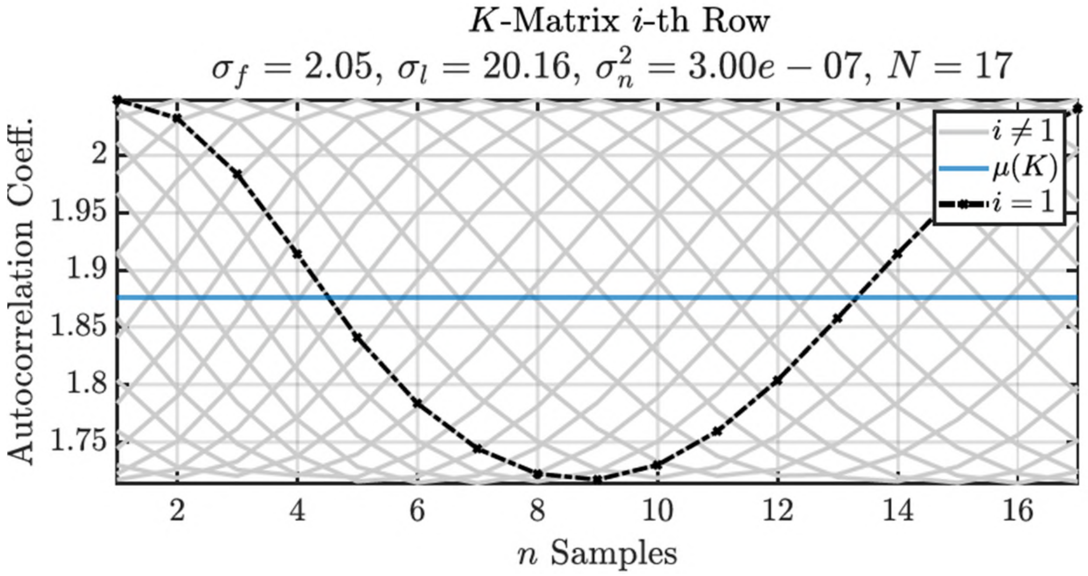


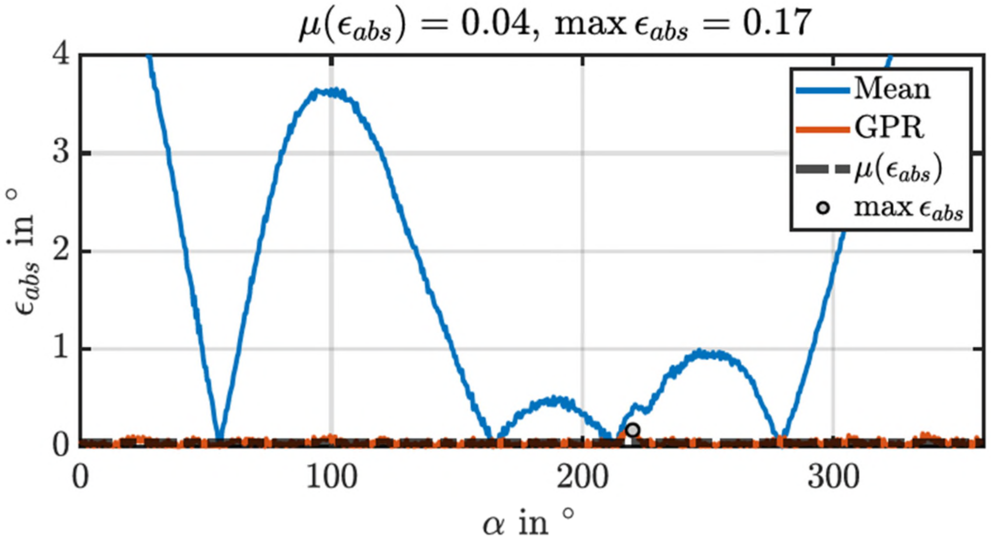


- Simple mean is drifted
- Regression balanced out

Testing Experiments

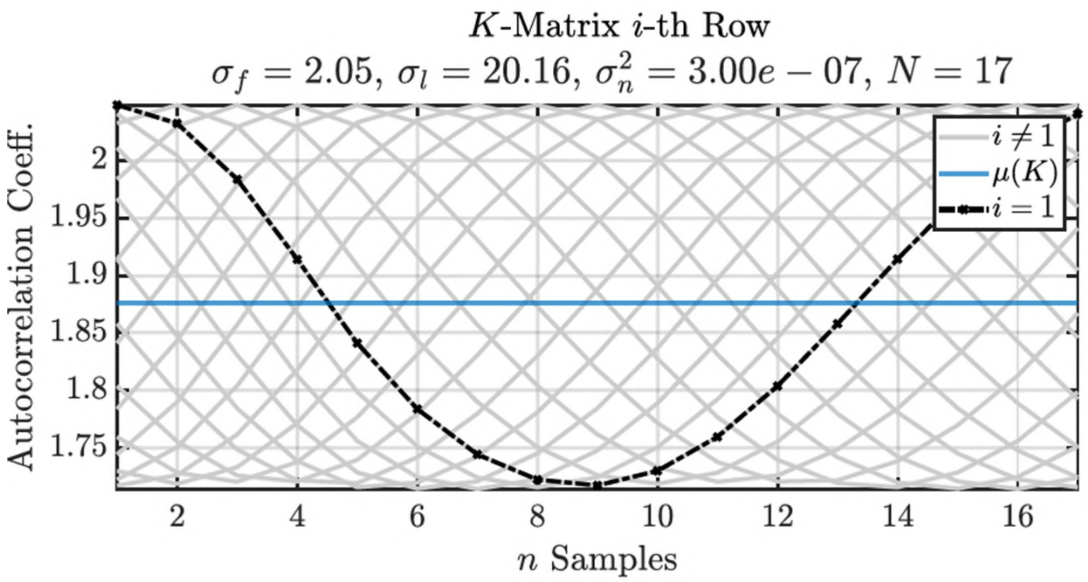
Position: $(0.5, 1.0, 4.5)^T$ mm, Tilt: 11.0°



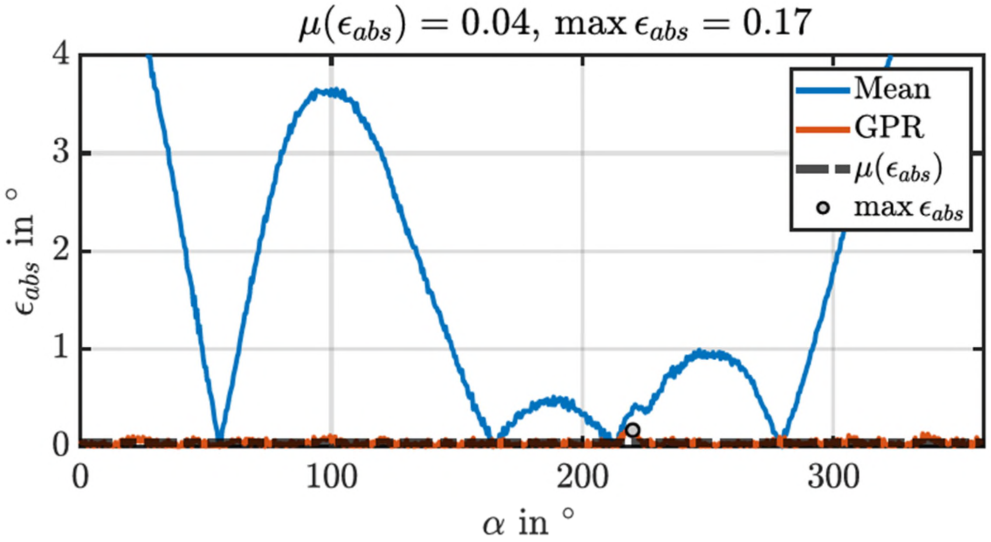


Testing Experiments

Position: $(0.5, 1.0, 4.5)^T$ mm, Tilt: 11.0°

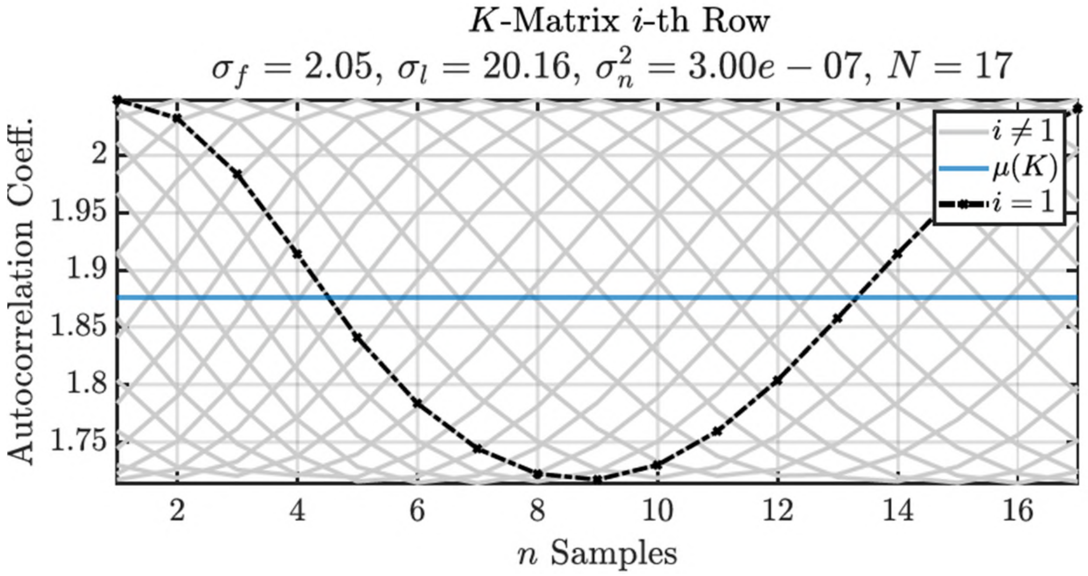


Very small angle errors over full rotation

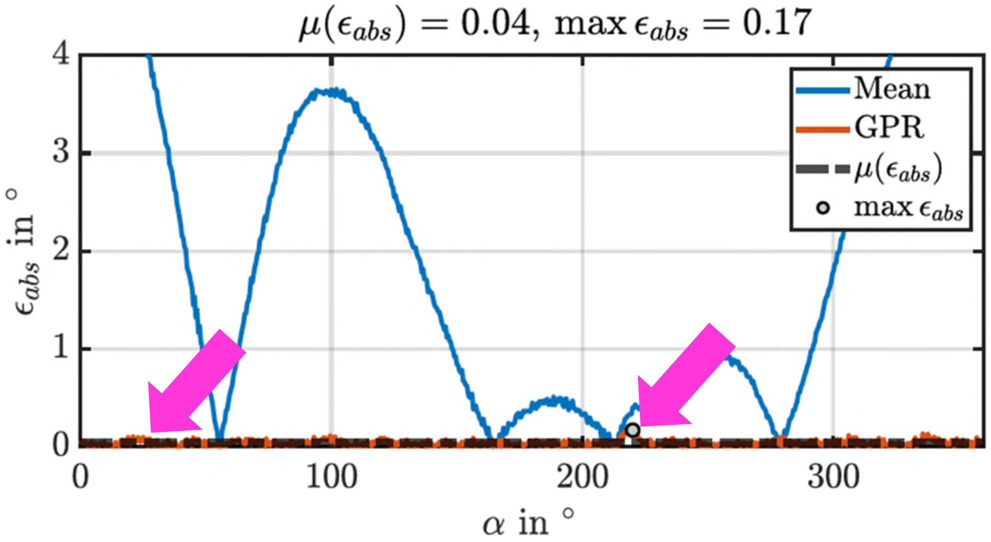


Testing Experiments

Position: $(0.5, 1.0, 4.5)^T$ mm, Tilt: 11.0°

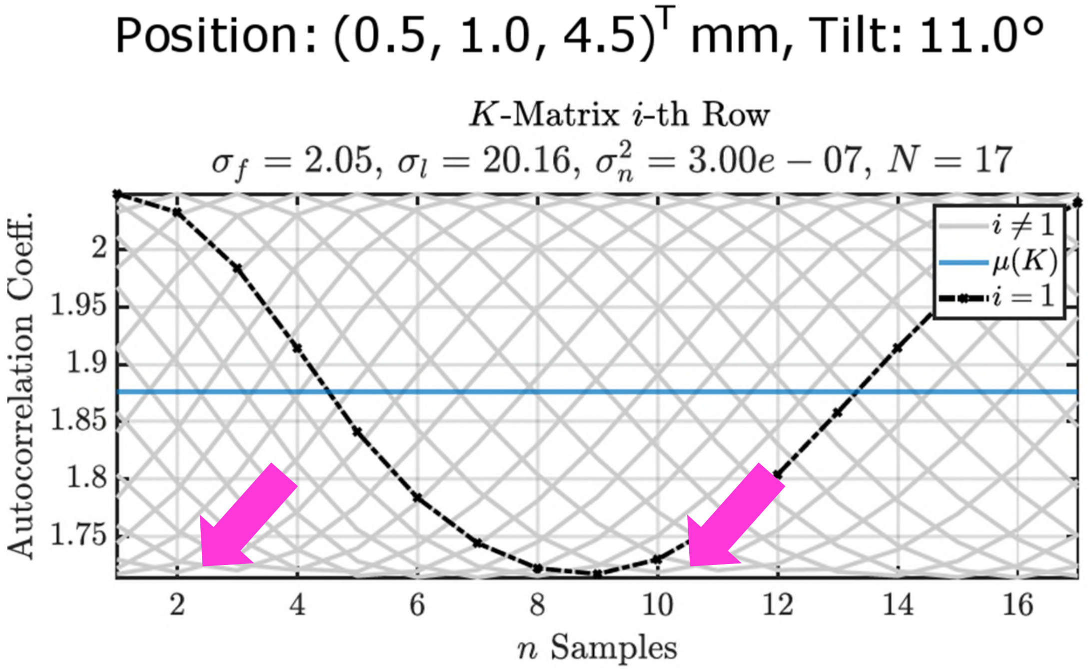


Very small angle errors over full rotation



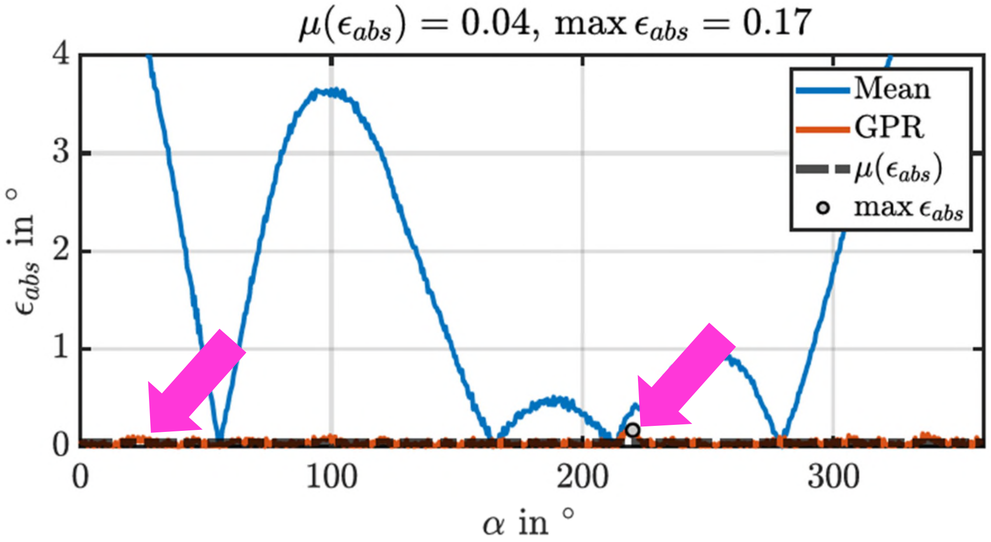
Small variation

Testing Experiments



Small lift, leak of coverage

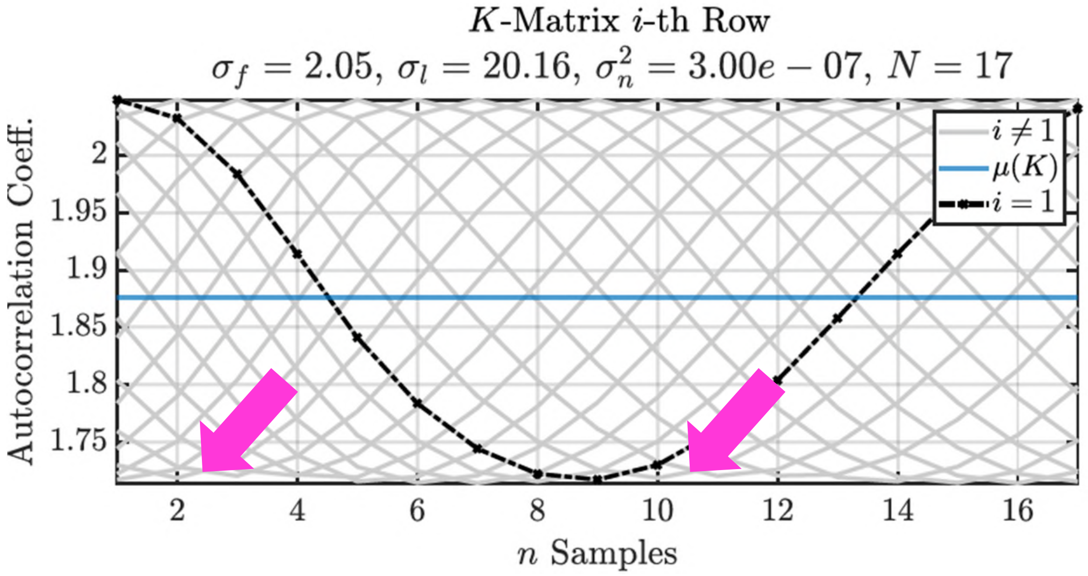
Very small angle errors over full rotation

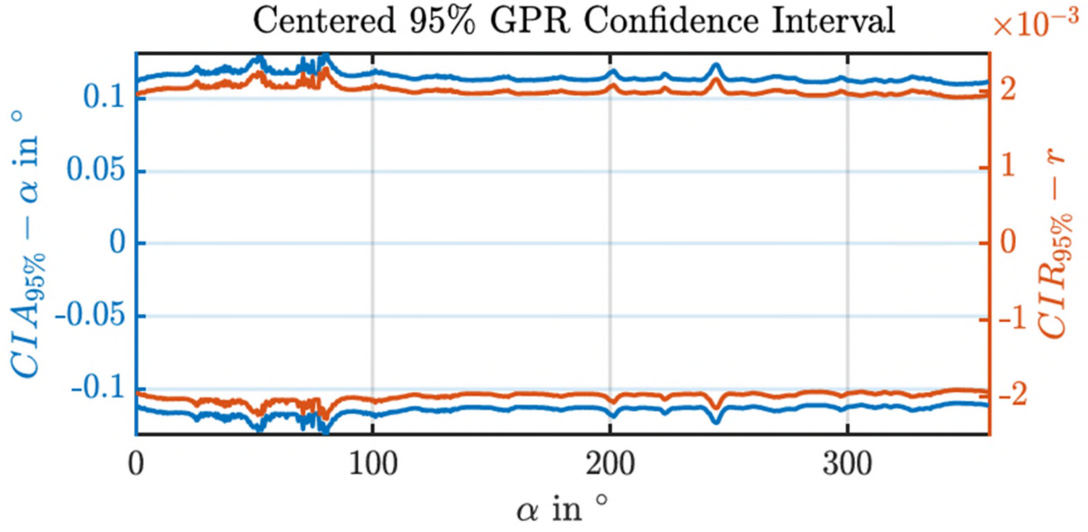


Small variation

Testing Experiments

Position: (0.5, 1.0, 4.5)^T mm, Tilt: 11.0°





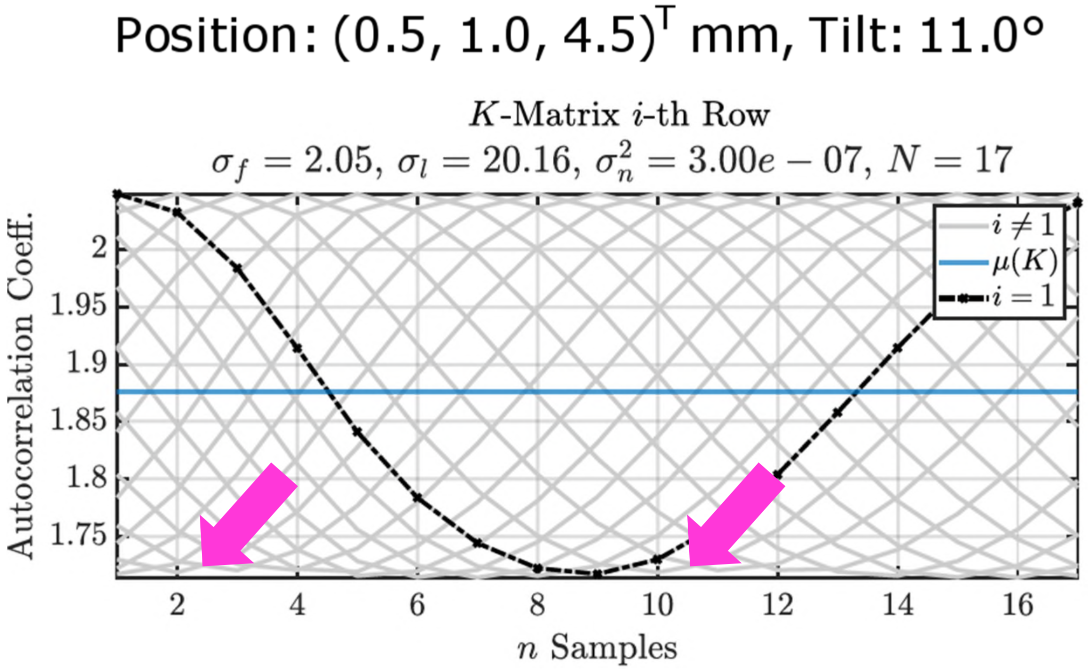
Small lift, leak of coverage

High trust, tight intervals

CIA – Confidence Interval Angle
CIR – Confidence Interval Radius

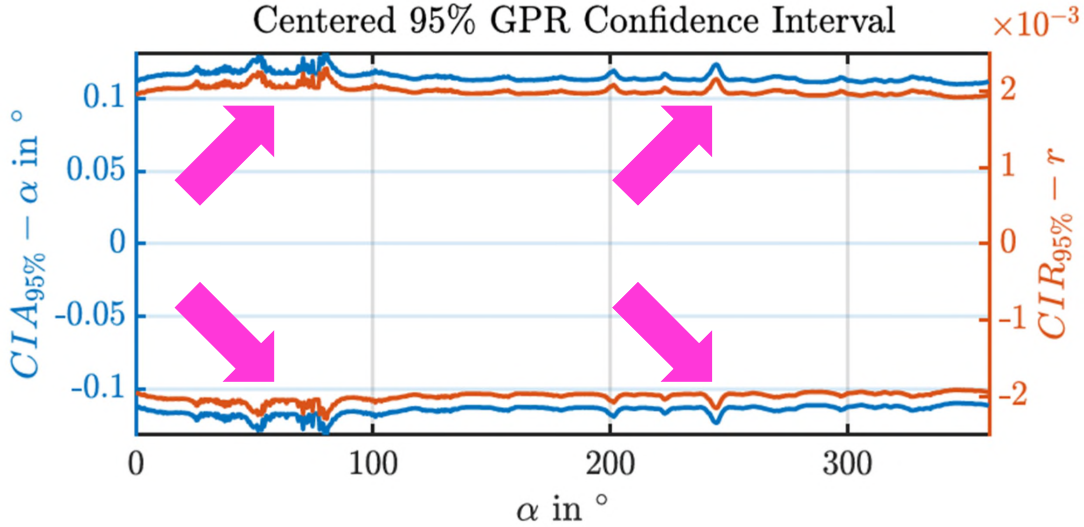
Testing Experiments

HAW HAMBURG



Small lift, leak of coverage

Showing the leakage



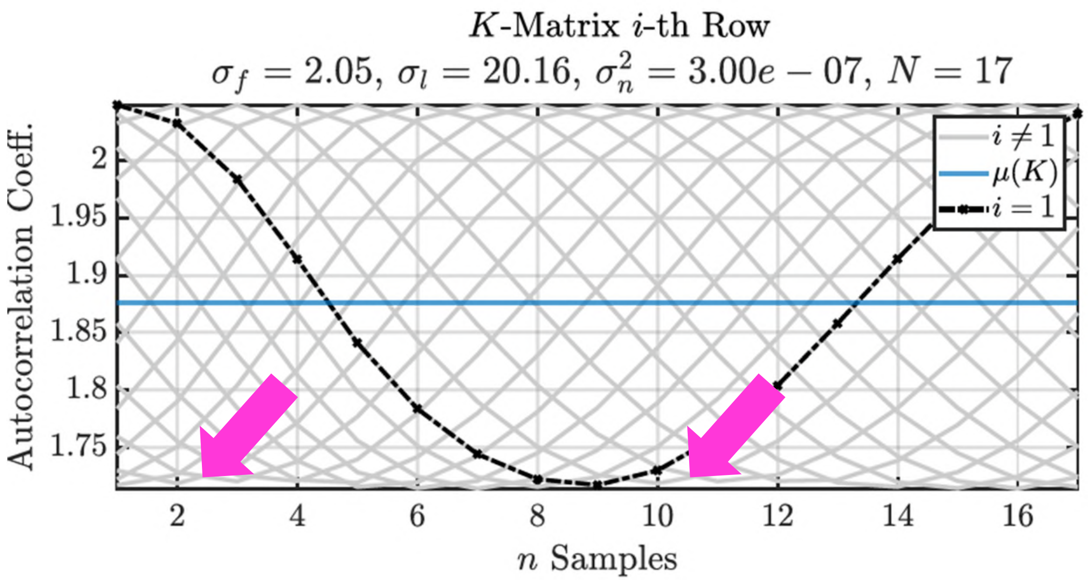
High trust, tight intervals

CIA – Confidence Interval Angle
CIR – Confidence Interval Radius

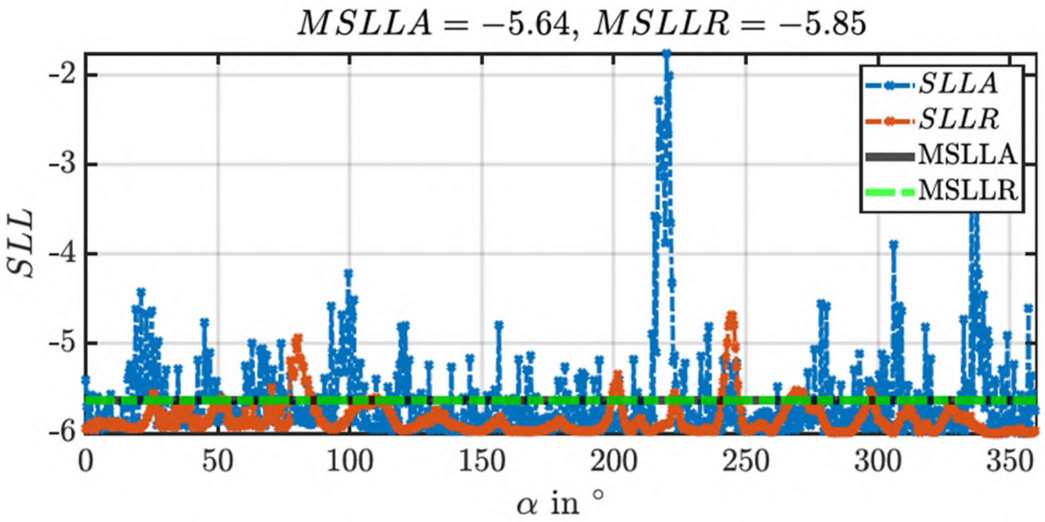
Testing Experiments

HAW HAMBURG

Position: $(0.5, 1.0, 4.5)^T$ mm, Tilt: 11.0°

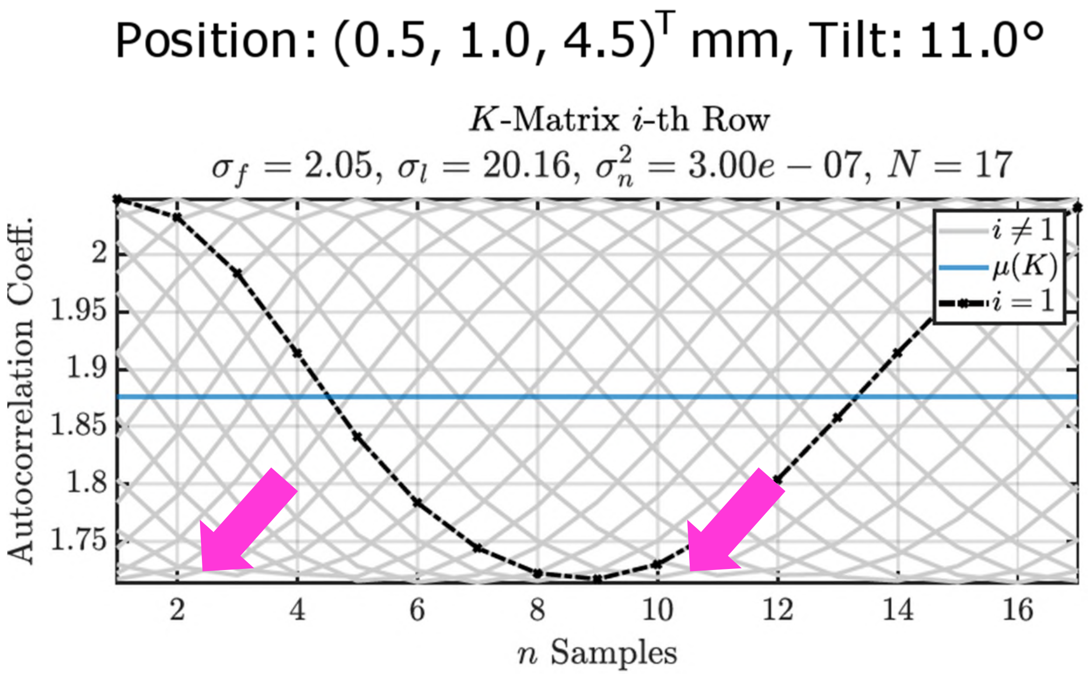


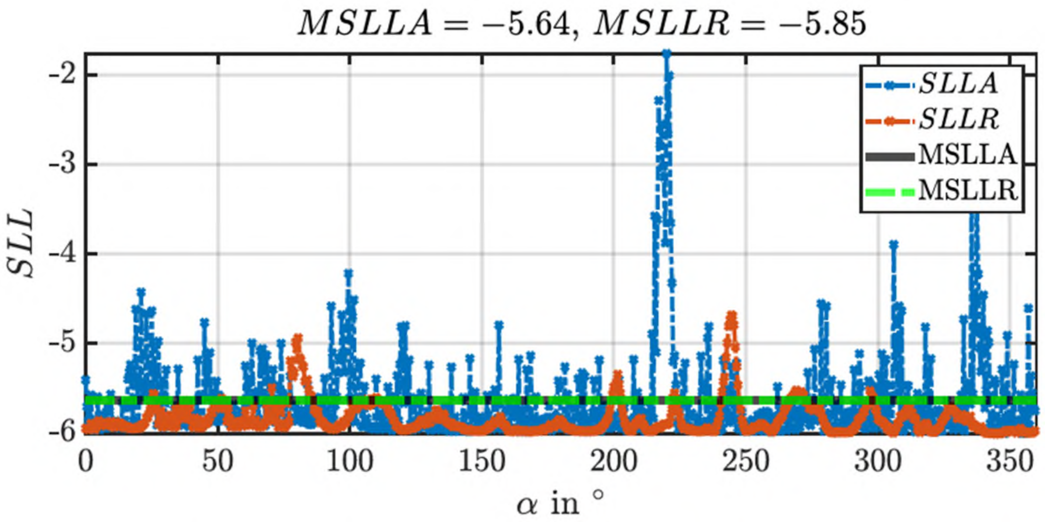
Small lift, leak of coverage



SLLA – Standardized Logarithmic Loss Angle
SLLR – Standardized Logarithmic Loss Radius
MSLLA – Mean Standardized Logarithmic Loss Angle
MSLLR – Mean Standardized Logarithmic Loss Radius

Testing Experiments





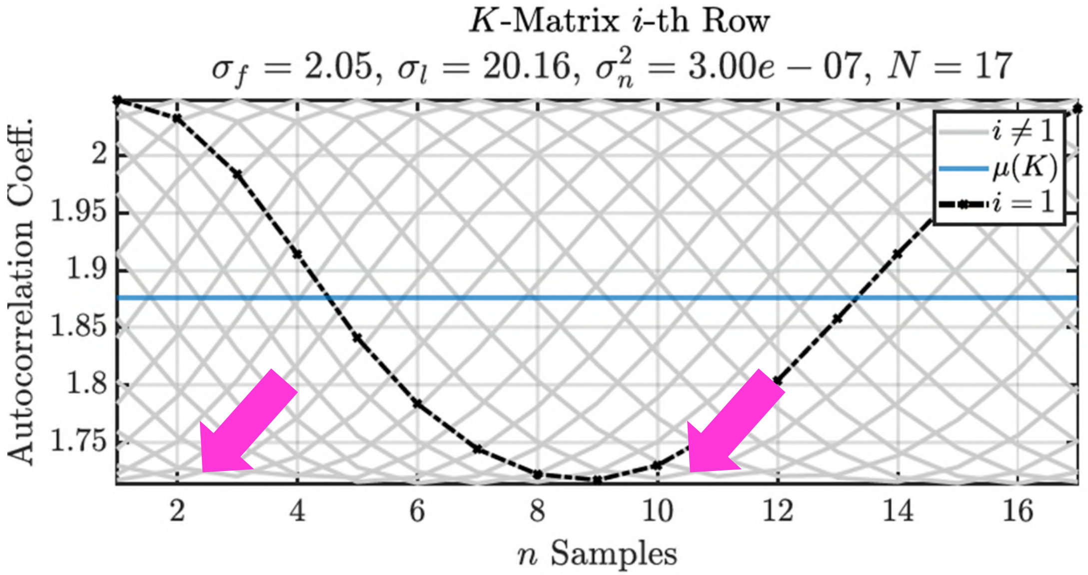
Small lift, leak of coverage

Strong generalization

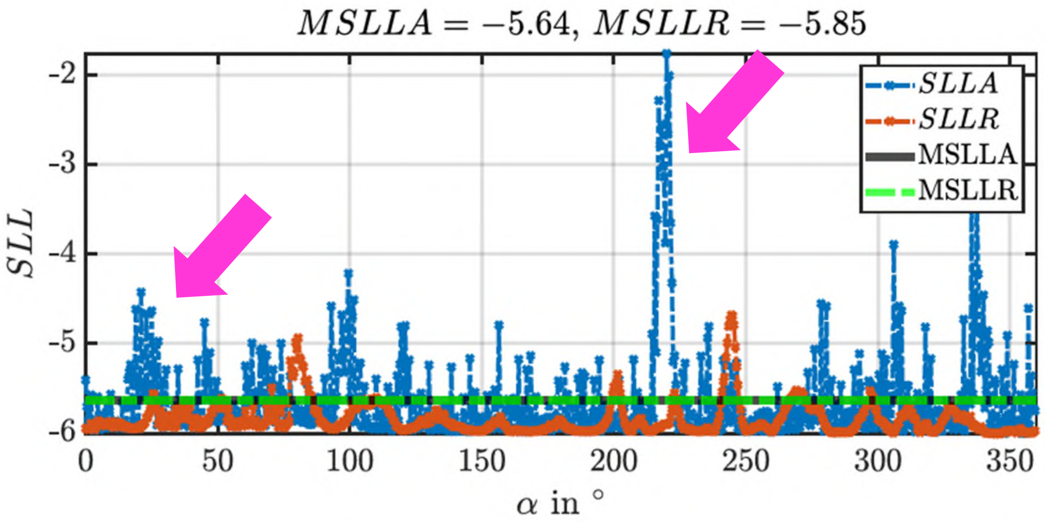
SLLA – Standardized Logarithmic Loss Angle
SLLR – Standardized Logarithmic Loss Radius
MSLLA – Mean Standardized Logarithmic Loss Angle
MSLLR – Mean Standardized Logarithmic Loss Radius

Testing Experiments

Position: $(0.5, 1.0, 4.5)^T$ mm, Tilt: 11.0°



Leakage weakened the generalization



Small lift, leak of coverage

Strong generalization

SLLA – Standardized Logarithmic Loss Angle
SLLR – Standardized Logarithmic Loss Radius
MSLLA – Mean Standardized Logarithmic Loss Angle
MSLLR – Mean Standardized Logarithmic Loss Radius

Summary

- **Achievements**
- **To do's**

Summary

- **Achievements**
  - Simulation Framework
  - Expandability
- **To do's**

Summary

HAW HAMBURG

- **Achievements**
  - Simulation Framework
  - Expandability
  - Kernel Scaling
  - Generalization
- **To do's**

Summary

- **Achievements**
  - Simulation Framework
  - Expandability
  - Kernel Scaling
  - Generalization
  - Tolerance compensation
- **To do's**

Summary

- **Achievements**
  - Simulation Framework
  - Expandability
  - Kernel Scaling
  - Generalization
  - Tolerance compensation
- **To do's**
  - Process minimization and limitation

Summary

- **Achievements**
  - Simulation Framework
  - Expandability
  - Kernel Scaling
  - Generalization
  - Tolerance compensation
- **To do's**
  - Process minimization and limitation
  - Hardware libs in C
  - Route to real data

Summary

- **Achievements**
  - Simulation Framework
  - Expandability
  - Kernel Scaling
  - Generalization
  - Tolerance compensation
- **To do's**
  - Process minimization and limitation
  - Hardware libs in C
  - Route to real data
  - Full characterization

Summary

- **Achievements**
  - Simulation Framework
  - Expandability
  - Kernel Scaling
  - Generalization
  - Tolerance compensation
- **To do's**
  - Process minimization and limitation
  - Hardware libs in C
  - Route to real data
  - Full characterization
  - Circular Statistics

The End

Thank you for your attention!