

Gauß-Prozess-Regression-Optimierung

Tobias Wulf

Hochschule für Angewandte Wissenschaften Hamburg

tobias.wulf@haw-hamburg.de

26. März 2021

Übersicht

1 Klassischer Anwendungsfall

- Winkelmessung
- Kreisdarstellung der Winkelmessung
- Euklidischer Abstand als Winkelabstand

2 Kreisdarstellung in Norm-Notation

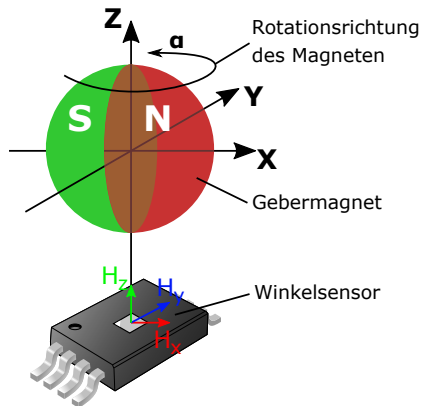
- Vektor-2-Norm für einfache Vektorfelder
- Überführung in höheren Norm-Raum für Vektormatrizen
- Gegenüberstellung

3 GPR-Verfahren

- Skalierung der Kovarianzfunktion
- Kreisdarstellung - GPR ohne Mittelwertkorrektur
- Kreisdarstellung - GPR mit Mittelwertkorrektur
- GPR-Trainingsphase
- GPR-Arbeitsphase

Klassischer Anwendungsfall

Winkelmessung

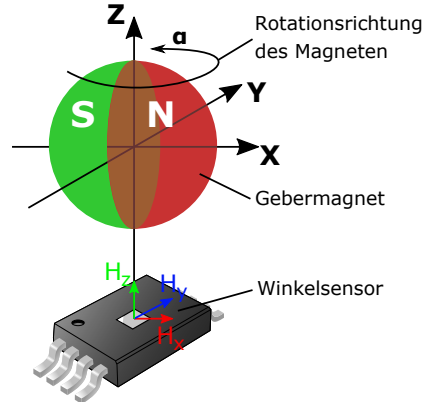


Klassischer Anwendungsfall

Winkelmessung

Winkelmessung

$$\underbrace{\begin{pmatrix} H_x(\alpha) \\ H_y(\alpha) \end{pmatrix}}_{\text{Gebermagnetfeld}} \Rightarrow \underbrace{\begin{pmatrix} V_{\cos}(H_x, H_y) \\ V_{\sin}(H_x, H_y) \end{pmatrix}}_{\text{Winkelsensormesswerte}} = \underbrace{\begin{pmatrix} r \cdot \cos(\alpha) \\ r \cdot \sin(\alpha) \end{pmatrix}}_{\text{Kreisdarstellung}} = \underbrace{\begin{pmatrix} a_x \\ a_y \end{pmatrix}}_{\text{Winkelmessung}} = \mathbf{A}(\alpha)$$



Klassischer Anwendungsfall

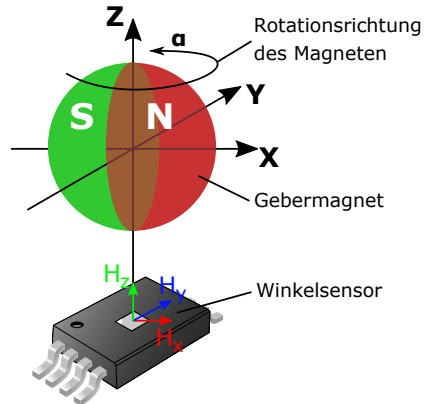
Winkelmessung

Winkelmessung

$$\underbrace{\begin{pmatrix} H_x(\alpha) \\ H_y(\alpha) \end{pmatrix}}_{\text{Gebermagnetfeld}} \Rightarrow \underbrace{\begin{pmatrix} V_{\cos}(H_x, H_y) \\ V_{\sin}(H_x, H_y) \end{pmatrix}}_{\text{Winkelsensormesswerte}} = \underbrace{\begin{pmatrix} r \cdot \cos(\alpha) \\ r \cdot \sin(\alpha) \end{pmatrix}}_{\text{Kreisdarstellung}} = \underbrace{\begin{pmatrix} a_x \\ a_y \end{pmatrix}}_{\text{Winkelmessung}} = \mathbf{A}(\alpha)$$

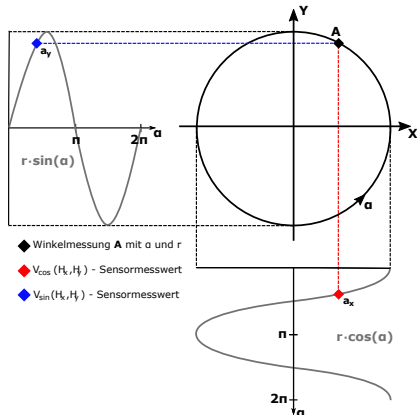
Orthogonalität der Messwerte

$$V_{\cos}(H_x, H_y) \perp V_{\sin}(H_x, H_y) \Leftrightarrow \mathbf{A} \mapsto \alpha$$



Klassischer Anwendungsfall

Kreisdarstellung der Winkelmessung

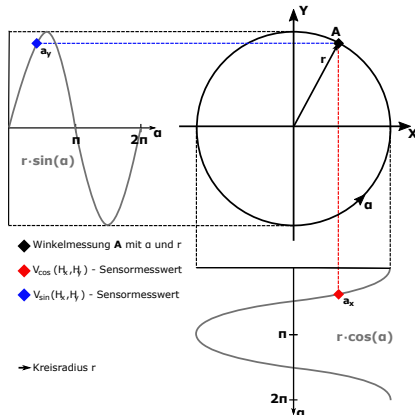


Klassischer Anwendungsfall

Kreisdarstellung der Winkelmessung

Radius

$$r = |\mathbf{A}|$$

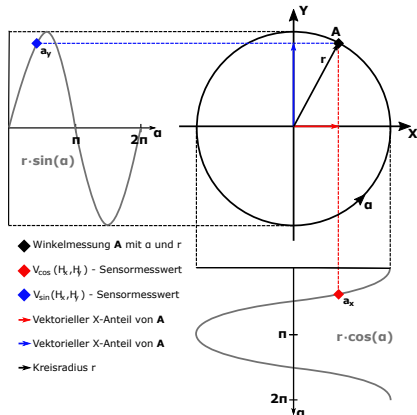


Klassischer Anwendungsfall

Kreisdarstellung der Winkelmessung

Radius

$$r = |\mathbf{A}| = \sqrt{a_x^2 + a_y^2}$$



Klassischer Anwendungsfall

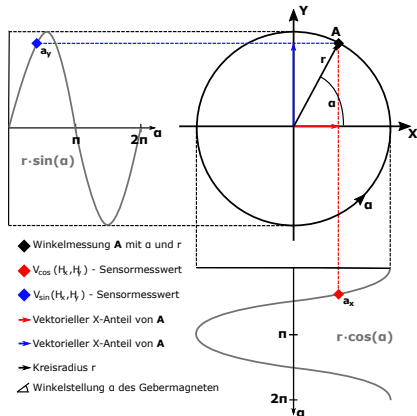
Kreisdarstellung der Winkelmessung

Radius

$$r = |\mathbf{A}| = \sqrt{a_x^2 + a_y^2}$$

Winkel

$$\alpha = \begin{cases} \arctan2(a_y, a_x) & \text{f. } a_y > 0 \\ \pi & \text{f. } a_y = 0 \\ \arctan2(a_y, a_x) + 2\pi & \text{f. } a_y < 0 \end{cases}$$

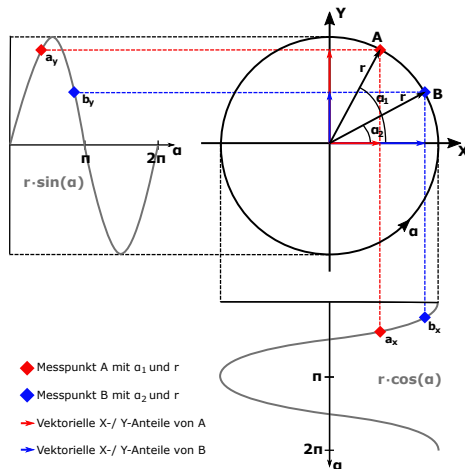


Klassischer Anwendungsfall

Euklidischer Abstand als Winkelabstand

Winkelmessungen

$A \mapsto \alpha_1$ $B \mapsto \alpha_2$



Klassischer Anwendungsfall

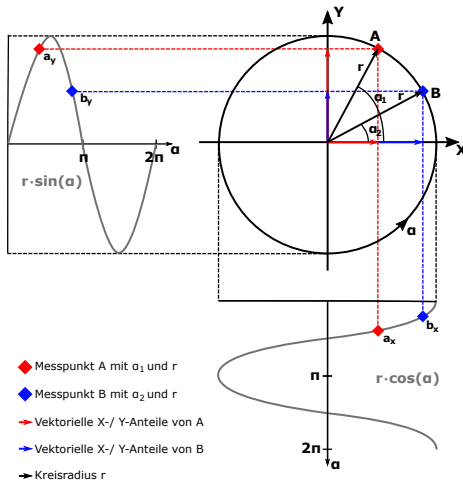
Euklidischer Abstand als Winkelabstand

Winkelmessungen

$$A \mapsto \alpha_1 \quad B \mapsto \alpha_2$$

Radius

$$r = |A| = |B| = \text{konst.}$$



Klassischer Anwendungsfall

Euklidischer Abstand als Winkelabstand

Winkelmessungen

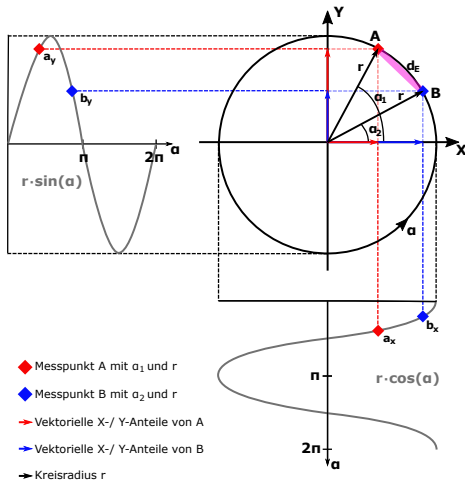
$$\mathbf{A} \mapsto \alpha_1 \quad \mathbf{B} \mapsto \alpha_2$$

Radius

$$r = |\mathbf{A}| = |\mathbf{B}| = \text{konst.}$$

Euklidischer Abstand

$$d_E\langle \mathbf{A}, \mathbf{B} \rangle =$$



Klassischer Anwendungsfall

Euklidischer Abstand als Winkelabstand

Winkelmessungen

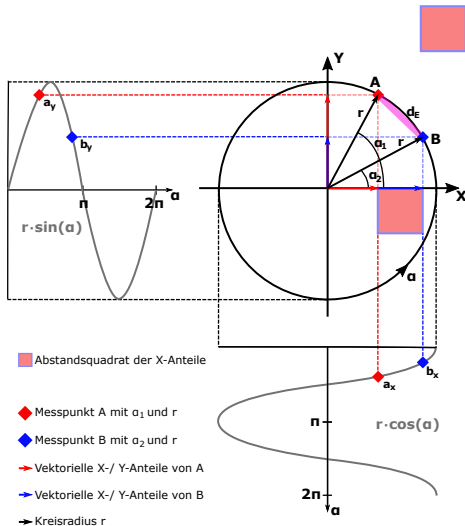
$$\mathbf{A} \mapsto \alpha_1 \quad \mathbf{B} \mapsto \alpha_2$$

Radius

$$r = |\mathbf{A}| = |\mathbf{B}| = \text{konst.}$$

Euklidischer Abstand

$$d_E(\mathbf{A}, \mathbf{B}) = \sqrt{(a_x - b_x)^2 + (a_y - b_y)^2}$$



Klassischer Anwendungsfall

Euklidischer Abstand als Winkelabstand

Winkelmessungen

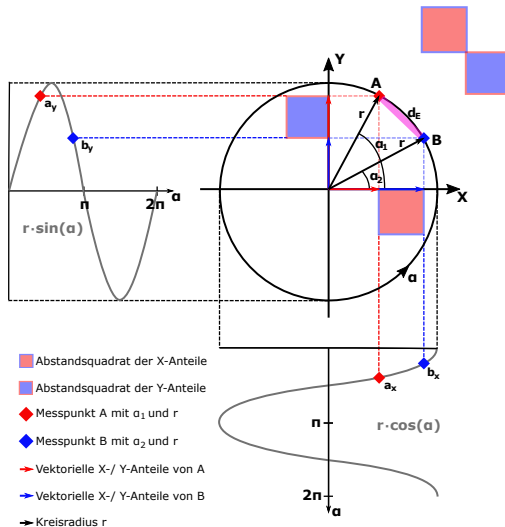
$$\mathbf{A} \mapsto \alpha_1 \quad \mathbf{B} \mapsto \alpha_2$$

Radius

$$r = |\mathbf{A}| = |\mathbf{B}| = \text{konst.}$$

Euklidischer Abstand

$$d_E\langle \mathbf{A}, \mathbf{B} \rangle = \sqrt{(a_x - b_x)^2 + (a_y - b_y)^2}$$



Klassischer Anwendungsfall

Euklidischer Abstand als Winkelabstand

Winkelmessungen

$$\mathbf{A} \mapsto \alpha_1 \quad \mathbf{B} \mapsto \alpha_2$$

Radius

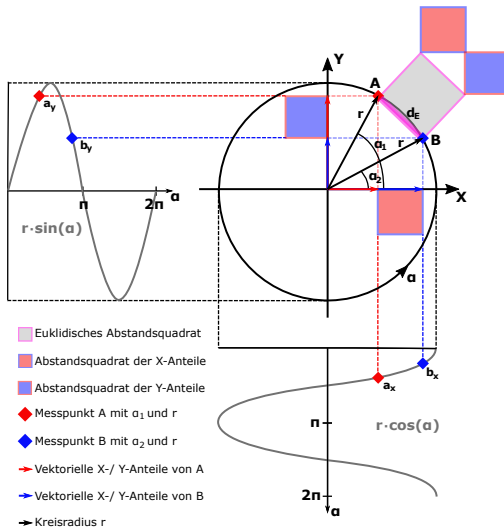
$$r = |\mathbf{A}| = |\mathbf{B}| = \text{konst.}$$

Euklidischer Abstand

$$d_E\langle \mathbf{A}, \mathbf{B} \rangle = \sqrt{(a_x - b_x)^2 + (a_y - b_y)^2}$$

Abstandsquadrat

$$d_E^2\langle \mathbf{A}, \mathbf{B} \rangle = (a_x - b_x)^2 + (a_y - b_y)^2$$



Kreisdarstellung in Norm-Notation

Vektor-2-Norm für einfache Vektorfelder

- Zweidimensionales Problem

Kreisdarstellung in Norm-Notation

Vektor-2-Norm für einfache Vektorfelder

- Zweidimensionales Problem
- Vektorfeld in 1. Dimension
$$\mathbf{A} = \begin{pmatrix} a_x \\ a_y \end{pmatrix}$$
- $\mathbf{A} \mapsto \alpha$ bildet 2. Dimension

Kreisdarstellung in Norm-Notation

Vektor-2-Norm für einfache Vektorfelder

- Zweidimensionales Problem
- Vektorfeld in 1. Dimension
 $\mathbf{A} = \begin{pmatrix} a_x \\ a_y \end{pmatrix}$
- $\mathbf{A} \mapsto \alpha$ bildet 2. Dimension
- Beträge als Norm

Radius

$$r := |\mathbf{A}| = \sqrt{\sum_{i=1}^n |A_i|^2} = \|\mathbf{A}\|_2$$

Kreisdarstellung in Norm-Notation

Vektor-2-Norm für einfache Vektorfelder

- Zweidimensionales Problem
- Vektorfeld in 1. Dimension
 $\mathbf{A} = \begin{pmatrix} a_x \\ a_y \end{pmatrix}$
- $\mathbf{A} \mapsto \alpha$ bildet 2. Dimension
- Beträge als Norm
- Abstand als Differenznorm

Radius

$$r := |\mathbf{A}| = \sqrt{\sum_{i=1}^n |A_i|^2} = \|\mathbf{A}\|_2$$

Euklidischer Abstand

$$d_E\langle \mathbf{A}, \mathbf{B} \rangle := \sqrt{\sum_{i=1}^n (A_i - B_i)^2} = \|\mathbf{A} - \mathbf{B}\|_2$$

Kreisdarstellung in Norm-Notation

Vektor-2-Norm für einfache Vektorfelder

- Zweidimensionales Problem
- Vektorfeld in 1. Dimension
 $\mathbf{A} = \begin{pmatrix} a_x \\ a_y \end{pmatrix}$
- $\mathbf{A} \mapsto \alpha$ bildet 2. Dimension
- Beträge als Norm
- Abstand als Differenznorm
- Es gilt: Dreiecksungleichung

Radius

$$r := |\mathbf{A}| = \sqrt{\sum_{i=1}^n |A_i|^2} = \|\mathbf{A}\|_2$$

Euklidischer Abstand

$$d_E\langle \mathbf{A}, \mathbf{B} \rangle := \sqrt{\sum_{i=1}^n (A_i - B_i)^2} = \|\mathbf{A} - \mathbf{B}\|_2$$

Dreiecksungleichung

$$|\|\mathbf{A}\|_2 - \|\mathbf{B}\|_2| \leq \|\mathbf{A} \pm \mathbf{B}\|_2 \leq \|\mathbf{A}\|_2 + \|\mathbf{B}\|_2$$

Kreisdarstellung in Norm-Notation

Überführung in höheren Norm-Raum für Vektormatrizen

- Vierdimensionales Problem

Kreisdarstellung in Norm-Notation

Überführung in höheren Norm-Raum für Vektormatrizen

- Vierdimensionales Problem
- 1. + 2. Dimension durch
Array-Geometrie

Kreisdarstellung in Norm-Notation

Überführung in höheren Norm-Raum für Vektormatrizen

- Vierdimensionales Problem
- 1. + 2. Dimension durch Array-Geometrie
- $\mathbf{A} \mapsto \alpha$ bildet 3. Dimension

Kreisdarstellung in Norm-Notation

Überführung in höheren Norm-Raum für Vektormatrizen

- Vierdimensionales Problem
- 1. + 2. Dimension durch Array-Geometrie
- $\mathbf{A} \mapsto \alpha$ bildet 3. Dimension
- Vektorfelder in 4. Dimension
 $\mathbf{A} = [\mathbf{A}_x, \mathbf{A}_y]$

Kreisdarstellung in Norm-Notation

Überführung in höheren Norm-Raum für Vektormatrizen

- Vierdimensionales Problem
- 1. + 2. Dimension durch Array-Geometrie
- $\mathbf{A} \mapsto \alpha$ bildet 3. Dimension
- Vektorfelder in 4. Dimension
 $\mathbf{A} = [\mathbf{A}_x, \mathbf{A}_y]$
- Frobenius Norm - Matrix als langer Vektor

Vektor-2-Norm² f. j -te Spalte

$$\|\mathbf{A}_x\|_F = \sqrt{\sum_{j=1}^n \|A_{xj}\|_2^2}$$

Kreisdarstellung in Norm-Notation

Überführung in höheren Norm-Raum für Vektormatrizen

- Vierdimensionales Problem
- 1. + 2. Dimension durch Array-Geometrie
- $\mathbf{A} \mapsto \alpha$ bildet 3. Dimension
- Vektorfelder in 4. Dimension
 $\mathbf{A} = [\mathbf{A}_x, \mathbf{A}_y]$
- Frobenius Norm - Matrix als langer Vektor
- Nutzen von Ähnlichkeit

Vektor-2-Norm² f. j -te Spalte

$$\|\mathbf{A}_x\|_F = \sqrt{\sum_{j=1}^n \|A_{xj}\|_2^2}$$

Ungleichung: Vektor-2-Norm

$$|\|\mathbf{A}\|_2 - \|\mathbf{B}\|_2| \leq \|\mathbf{A} \pm \mathbf{B}\|_2 \leq \|\mathbf{A}\|_2 + \|\mathbf{B}\|_2$$



Kreisdarstellung in Norm-Notation

Überführung in höheren Norm-Raum für Vektormatrizen

- Vierdimensionales Problem
- 1. + 2. Dimension durch Array-Geometrie
- $\mathbf{A} \mapsto \alpha$ bildet 3. Dimension
- Vektorfelder in 4. Dimension
 $\mathbf{A} = [\mathbf{A}_x, \mathbf{A}_y]$
- Frobenius Norm - Matrix als langer Vektor
- Nutzen von Ähnlichkeit
- Es muss Dreiecksungleichung gelten

Vektor-2-Norm² f. j -te Spalte

$$\|\mathbf{A}_x\|_F = \sqrt{\sum_{j=1}^n \|A_{xj}\|_2^2}$$

Ungleichung: Vektor-2-Norm

$$|\|\mathbf{A}\|_2 - \|\mathbf{B}\|_2| \leq \|\mathbf{A} \pm \mathbf{B}\|_2 \leq \|\mathbf{A}\|_2 + \|\mathbf{B}\|_2$$



Ungleichung: Frobenius Norm

$$|\|\mathbf{A}_x\|_F - \|\mathbf{B}_x\|_F| \leq \|\mathbf{A}_x \pm \mathbf{B}_x\|_F \leq \|\mathbf{A}_x\|_F + \|\mathbf{B}_x\|_F$$

Kreisdarstellung in Norm-Notation

Überführung in höheren Norm-Raum für Vektormatrizen

Aufbau über euklidischen Abstand

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

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

Kreisdarstellung in Norm-Notation

Überführung in höheren Norm-Raum für Vektormatrizen

- Einsetzen v. Einzelnormen,
"runter brechen" d. Komplexität

Aufbau über euklidischen Abstand

$$d_E^2\langle \mathbf{A}, \mathbf{B} \rangle = (\|\mathbf{A}_x\|_F - \|\mathbf{B}_x\|_F)^2 + (\|\mathbf{A}_y\|_F - \|\mathbf{B}_y\|_F)^2$$

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

Kreisdarstellung in Norm-Notation

Überführung in höheren Norm-Raum für Vektormatrizen

- Einsetzen v. Einzelnormen,
"runter brechen" d. Komplexität
- Neue normierte Kreisbahn $\|r\|_F$

Aufbau über euklidischen Abstand

$$d_E^2\langle \mathbf{A}, \mathbf{B} \rangle = (\|\mathbf{A}_x\|_F - \|\mathbf{B}_x\|_F)^2 + (\|\mathbf{A}_y\|_F - \|\mathbf{B}_y\|_F)^2$$

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

Kreisdarstellung in Norm-Notation

Überführung in höheren Norm-Raum für Vektormatrizen

- Einsetzen v. Einzelnormen, "runter brechen" d. Komplexität
- Dreiecksungleichung
- Neue normierte Kreisbahn $\|r\|_F$

Aufbau über euklidischen Abstand

$$\begin{aligned} d_E^2\langle \mathbf{A}, \mathbf{B} \rangle &= (\|\mathbf{A}_x\|_F - \|\mathbf{B}_x\|_F)^2 + (\|\mathbf{A}_y\|_F - \|\mathbf{B}_y\|_F)^2 \\ &\leq \\ d_F^2\langle \mathbf{A}, \mathbf{B} \rangle &= \end{aligned}$$

Kreisdarstellung in Norm-Notation

Überführung in höheren Norm-Raum für Vektormatrizen

- Einsetzen v. Einzelnormen,
"runter brechen" d. Komplexität
- Neue normierte Kreisbahn $\|r\|_F$
- Dreiecksungleichung
- Vergleich der Einzelkreisbahnen
 $r_{i,j}$ f. $d_F^2\langle \mathbf{A}, \mathbf{B} \rangle$

Aufbau über euklidischen Abstand

$$d_E^2\langle \mathbf{A}, \mathbf{B} \rangle = (\|\mathbf{A}_x\|_F - \|\mathbf{B}_x\|_F)^2 + (\|\mathbf{A}_y\|_F - \|\mathbf{B}_y\|_F)^2 \quad (1)$$

\leq

$$d_F^2\langle \mathbf{A}, \mathbf{B} \rangle = \|\mathbf{A}_x - \mathbf{B}_x\|_F^2 + \|\mathbf{A}_y - \mathbf{B}_y\|_F^2 = \|\mathbf{A} - \mathbf{B}\|_F^2 \quad (2)$$

Kreisdarstellung in Norm-Notation

Gegenüberstellung

Vektor-2-Norm	\Rightarrow	Frobenius Norm
$\mathbf{A} = (a_x, a_y)$	\Rightarrow	$\mathbf{A} = [\mathbf{A}_x, \mathbf{A}_y]$
\Downarrow		\Downarrow
$\ \mathbf{A}\ _2 = \sqrt{\sum_{i=1}^n A_i ^2}$	\Rightarrow	$\ \mathbf{A}_x\ _F = \sqrt{\sum_{j=1}^n \ A_{xj}\ _2^2}$ f. j -te Spalte
\Downarrow		\Downarrow
$d_E^2 \langle \mathbf{A}, \mathbf{B} \rangle$	\Rightarrow	$d_F^2 \langle \mathbf{A}, \mathbf{B} \rangle$
$=$		$=$
$\ \mathbf{A} - \mathbf{B}\ _2^2$	\Rightarrow	$\ \mathbf{A} - \mathbf{B}\ _F^2$
$=$		$=$
\vdots		$\ \mathbf{A}_x - \mathbf{B}_x\ _F^2 + \ \mathbf{A}_y - \mathbf{B}_y\ _F^2$
\vdots		\geq
$(a_x - b_x)^2 + (a_y - b_y)^2$	\Leftrightarrow	$(\ \mathbf{A}_x\ _F - \ \mathbf{B}_x\ _F)^2 + (\ \mathbf{A}_y\ _F - \ \mathbf{B}_y\ _F)^2$
\Downarrow		\Downarrow
$r = \mathbf{A} = \sqrt{a_x^2 + a_y^2}$	\Rightarrow	$\ r\ _F = \ \mathbf{A}\ _F = \sqrt{\ \mathbf{A}_x\ _F^2 + \ \mathbf{A}_y\ _F^2}$

Kreisdarstellung in Norm-Notation

Gegenüberstellung

Kernel: *QFCAPX*

Implementierung nach (1)

Kovarianz:

Kernel: *QFC*

Implementierung nach (2)

Kovarianz:

Kreisdarstellung in Norm-Notation

Gegenüberstellung

Kernel: *QFCAPX*

Implementierung nach (1)

- Approximierte Lösung

Kovarianz:

Kernel: *QFC*

Implementierung nach (2)

- Genaue Lösung

Kovarianz:

Kreisdarstellung in Norm-Notation

Gegenüberstellung

Kernel: QFCAPX

Implementierung nach (1)

- Approximierte Lösung
- Frobenius Norm vor Kovarianzfunktion

Kovarianz:

$$\text{cov}(\mathbf{A}, \mathbf{B}) = \frac{a}{b + d_E^2 \langle \mathbf{A}, \mathbf{B} \rangle}$$

Kernel: QFC

Implementierung nach (2)

- Genaue Lösung
- Frobenius Norm in Kovarianzfunktion

Kovarianz:

$$\text{cov}(\mathbf{A}, \mathbf{B}) = \frac{a}{b + d_F^2 \langle \mathbf{A}, \mathbf{B} \rangle}$$

Kreisdarstellung in Norm-Notation

Gegenüberstellung

Kernel: *QFCAPX*

Implementierung nach (1)

- Approximierte Lösung
- Frobenius Norm vor Kovarianzfunktion
- Vektoren als Trainingsdaten

Kovarianz:

$$\text{cov}(\mathbf{A}, \mathbf{B}) = \frac{a}{b + d_E^2 \langle \mathbf{A}, \mathbf{B} \rangle}$$

Kernel: *QFC*

Implementierung nach (2)

- Genaue Lösung
- Frobenius Norm in Kovarianzfunktion
- Matrizen als Trainingsdaten

Kovarianz:

$$\text{cov}(\mathbf{A}, \mathbf{B}) = \frac{a}{b + d_F^2 \langle \mathbf{A}, \mathbf{B} \rangle}$$

Kreisdarstellung in Norm-Notation

Gegenüberstellung

Kernel: *QFCAPX*

Implementierung nach (1)

- Approximierte Lösung
- Frobenius Norm vor Kovarianzfunktion
- Vektoren als Trainingsdaten
- Informationsverlust

Kovarianz:

$$\text{cov}(\mathbf{A}, \mathbf{B}) = \frac{a}{b + d_E^2 \langle \mathbf{A}, \mathbf{B} \rangle}$$

Kernel: *QFC*

Implementierung nach (2)

- Genaue Lösung
- Frobenius Norm in Kovarianzfunktion
- Matrizen als Trainingsdaten
- Höherer Speicherbedarf

Kovarianz:

$$\text{cov}(\mathbf{A}, \mathbf{B}) = \frac{a}{b + d_F^2 \langle \mathbf{A}, \mathbf{B} \rangle}$$

GPR-Verfahren

Skalierung der Kovarianzfunktion

- Skalierung nach RBF-Vorbild

GPR-Verfahren

Skalierung der Kovarianzfunktion

- Skalierung nach RBF-Vorbild

RBF Kernel

$$\sigma_f^2 \cdot e^{-\frac{d_x^2 \langle \mathbf{A}, \mathbf{B} \rangle}{2\sigma_l^2}}$$

- Vorbild

GPR-Verfahren

Skalierung der Kovarianzfunktion

- Skalierung nach RBF-Vorbild

RBF Kernel

$$\sigma_f^2 \cdot e^{-\frac{d_x^2 \langle \mathbf{A}, \mathbf{B} \rangle}{2\sigma_l^2}}$$



Fractional Kernel

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

- Vorbild

GPR-Verfahren

Skalierung der Kovarianzfunktion

- Skalierung nach RBF-Vorbild
- Bei Auslöschung $\text{cov}(\mathbf{A}, \mathbf{B}) = \sigma_f^2$ f. $d_x^2 \langle \mathbf{A}, \mathbf{B} \rangle = 0$

RBF Kernel

$$\sigma_f^2 \cdot e^{-\frac{d_x^2 \langle \mathbf{A}, \mathbf{B} \rangle}{2\sigma_l^2}}$$

- Vorbild

Fractional Kernel

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

- Übertragen

GPR-Verfahren

Skalierung der Kovarianzfunktion

- Skalierung nach RBF-Vorbild
- Bei Auslöschung $\text{cov}(\mathbf{A}, \mathbf{B}) = \sigma_f^2$ f. $d_x^2 \langle \mathbf{A}, \mathbf{B} \rangle = 0$

RBF Kernel

$$\sigma_f^2 \cdot e^{-\frac{d_x^2 \langle \mathbf{A}, \mathbf{B} \rangle}{2\sigma_l^2}}$$

- Vorbild

Fractional Kernel

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

- Übertragen

Fractional Kernel

$$\frac{a}{b + d_x^2 \langle \mathbf{A}, \mathbf{B} \rangle}$$

GPR-Verfahren

Skalierung der Kovarianzfunktion

- Skalierung nach RBF-Vorbild
- Bei Auslöschung $\text{cov}(\mathbf{A}, \mathbf{B}) = \sigma_f^2$ f. $d_x^2 \langle \mathbf{A}, \mathbf{B} \rangle = 0$
- $a = \sigma_f^2 \cdot 2\sigma_l^2$
- $b = 2\sigma_l^2$

RBF Kernel

$$\sigma_f^2 \cdot e^{-\frac{d_x^2 \langle \mathbf{A}, \mathbf{B} \rangle}{2\sigma_l^2}}$$

- Vorbild

Fractional Kernel

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

- Übertragen

Fractional Kernel

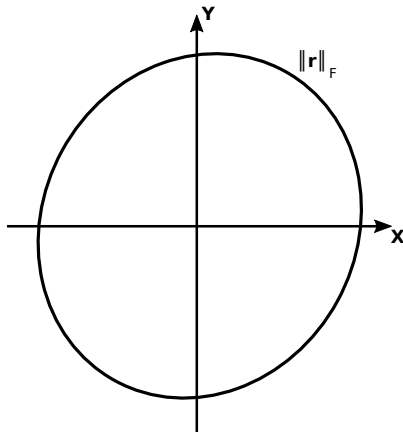
$$\frac{a}{b + d_x^2 \langle \mathbf{A}, \mathbf{B} \rangle}$$

- Umstellen

GPR-Verfahren

Kreisdarstellung - GPR ohne Mittelwertkorrektur

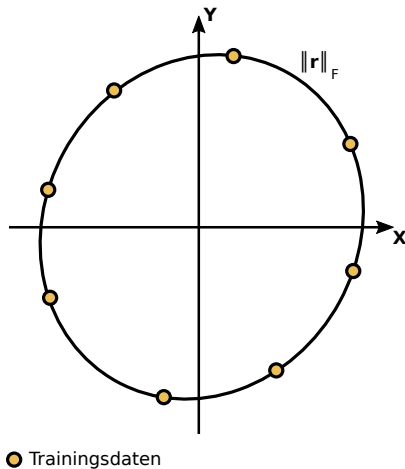
- Normierung \rightarrow Ellipse



GPR-Verfahren

Kreisdarstellung - GPR ohne Mittelwertkorrektur

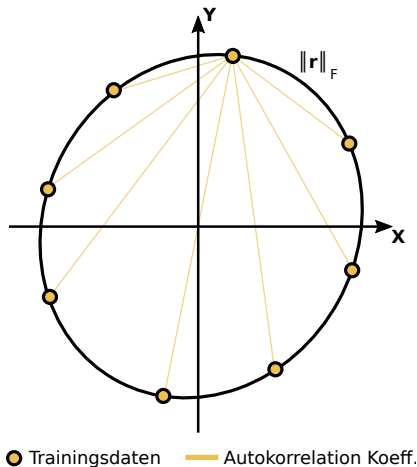
- Normierung \rightarrow Ellipse
- Trainingsdaten, Fehler $\rightarrow \alpha, r$



GPR-Verfahren

Kreisdarstellung - GPR ohne Mittelwertkorrektur

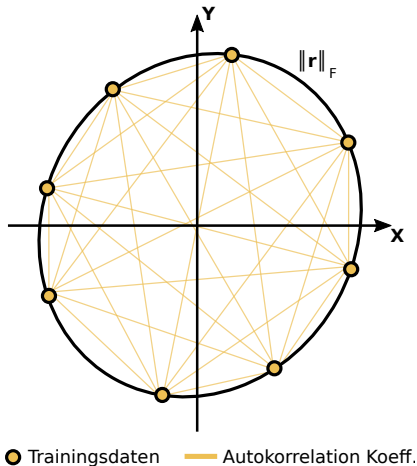
- Normierung \rightarrow Ellipse
- Trainingsdaten, Fehler $\rightarrow \alpha, r$
- K-Matrix liefert Autokorrelation



GPR-Verfahren

Kreisdarstellung - GPR ohne Mittelwertkorrektur

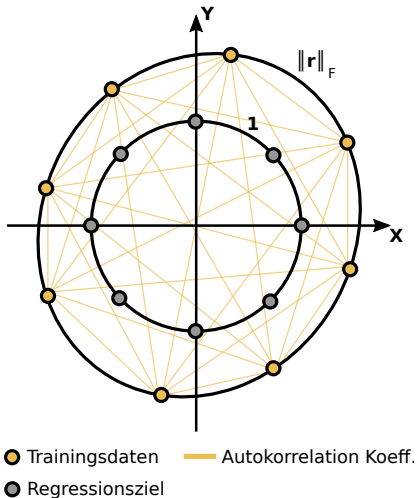
- Normierung \rightarrow Ellipse
- Trainingsdaten, Fehler $\rightarrow \alpha, r$
- K-Matrix liefert Autokorrelation



GPR-Verfahren

Kreisdarstellung - GPR ohne Mittelwertkorrektur

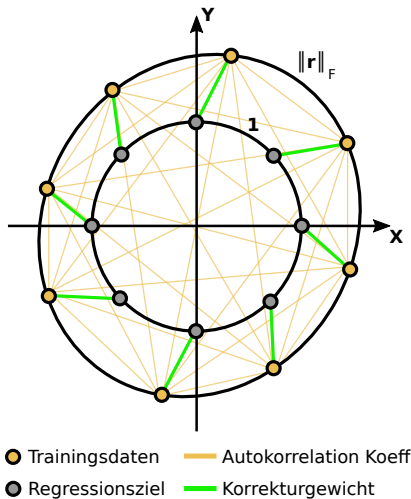
- Normierung \rightarrow Ellipse
- Trainingsdaten, Fehler $\rightarrow \alpha, r$
- K-Matrix liefert Autokorrelation
- Regression zielt auf Einheitskreis



GPR-Verfahren

Kreisdarstellung - GPR ohne Mittelwertkorrektur

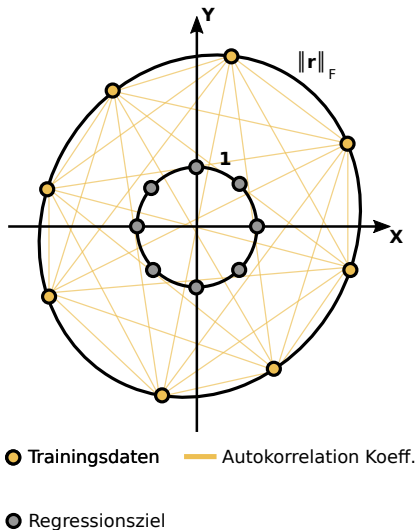
- Normierung \rightarrow Ellipse
- Trainingsdaten, Fehler $\rightarrow \alpha, r$
- K-Matrix liefert Autokorrelation
- Regression zielt auf Einheitskreis
- GPR richtet sich direkt auf Regressionsziele
- GPR direkt durch Gewichte gestützt



GPR-Verfahren

Kreisdarstellung - GPR mit Mittelwertkorrektur

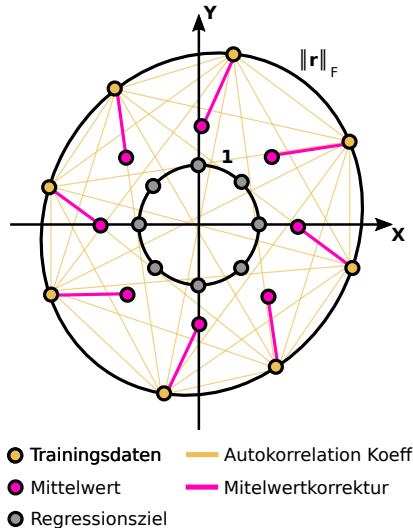
- Trainingsdaten, Autokorrelation identisch



GPR-Verfahren

Kreisdarstellung - GPR mit Mittelwertkorrektur

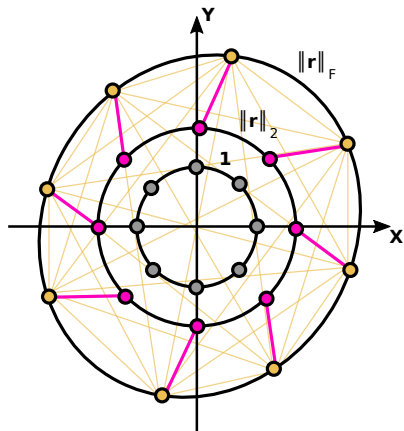
- Trainingsdaten, Autokorrelation identisch
- Mittelwertbildung üb.
 $m(X) = H'(X) \cdot \beta$



GPR-Verfahren

Kreisdarstellung - GPR mit Mittelwertkorrektur

- Trainingsdaten, Autokorrelation identisch
- Mittelwertbildung üb.
 $m(X) = H'(X) \cdot \beta$
- 2. Gewichtsniveau durch Mittelwert
- Vorwiegend Amplituden-/ Offset-Korrektur

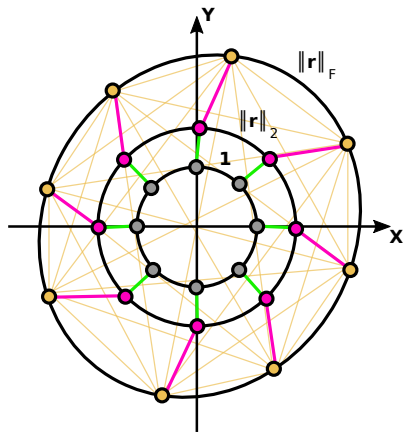


- Trainingsdaten
- Mittelwert
- Regressionsziel
- Autokorrelation Koeff.
- Mittelwertkorrektur

GPR-Verfahren

Kreisdarstellung - GPR mit Mittelwertkorrektur

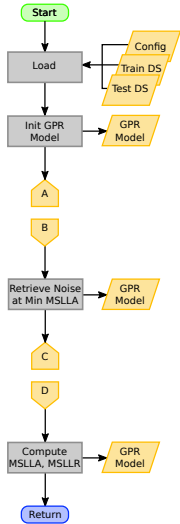
- Trainingsdaten, Autokorrelation identisch
- Mittelwertbildung üb.
 $m(X) = H'(X) \cdot \beta$
- 2. Gewichtsniveau durch Mittelwert
- Vorwiegend Amplituden-/ Offset-Korrektur
- Guter Mittelwert, kleine Korrekturgewichte
- Höherer Rechenaufwand



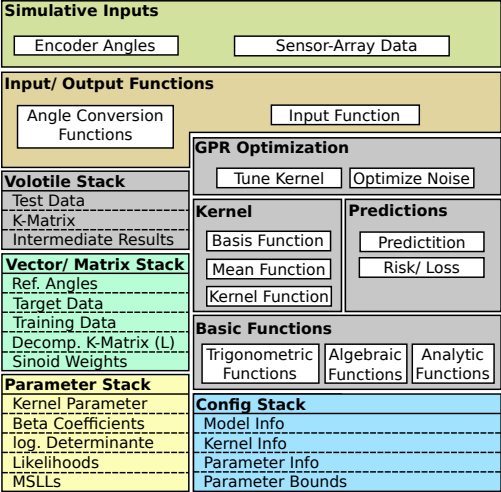
- | | |
|-------------------|--------------------------|
| ● Trainingsdaten | — Autokorrelation Koeff. |
| ● Mittelwert | — Mittelwertkorrektur |
| ● Regressionsziel | — Korrekturgewicht |

GPR-Verfahren

GPR-Trainingsphase

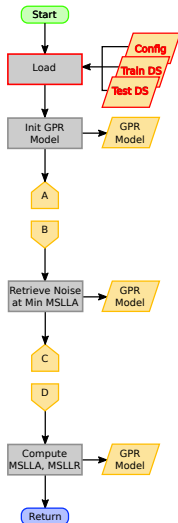


GPR Training



GPR-Verfahren

GPR-Trainingsphase



GPR Training

Simulative Inputs

Encoder Angles

Sensor-Array Data

Input/ Output Functions

Angle Conversion Functions

Input Function

GPR Optimization

Tune Kernel

Optimize Noise

Kernel

Basis Function

Mean Function

Kernel Function

Predictions

Prediction

Risk/ Loss

Volatile Stack

Test Data

K-Matrix

Intermediate Results

Vector/ Matrix Stack

Ref. Angles

Target Data

Training Data

Decomp. K-Matrix (L)

Sinoid Weights

Parameter Stack

Kernel Parameter

Beta Coefficients

log. Determinante

Likelihoods

MSLLs

Basic Functions

Trigonometric Functions

Algebraic Functions

Analytic Functions

Config Stack

Model Info

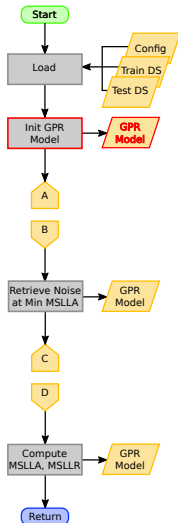
Kernel Info

Parameter Info

Parameter Bounds

GPR-Verfahren

GPR-Trainingsphase



GPR Training

Simulative Inputs

Encoder Angles

Sensor-Array Data

Input/ Output Functions

Angle Conversion Functions

Input Function

GPR Optimization

Tune Kernel

Optimize Noise

Kernel

Basis Function

Mean Function

Kernel Function

Predictions

Prediction

Risk/ Loss

Basic Functions

Trigonometric Functions

Algebraic Functions

Analytic Functions

Config Stack

Model Info

Kernel Info

Parameter Info

Parameter Bounds

Volatile Stack

Test Data

K-Matrix

Intermediate Results

Vector/ Matrix Stack

Ref. Angles

Target Data

Training Data

Decomp. K-Matrix (L)

Sinoid Weights

Parameter Stack

Kernel Parameter

Beta Coefficients

log. Determinante

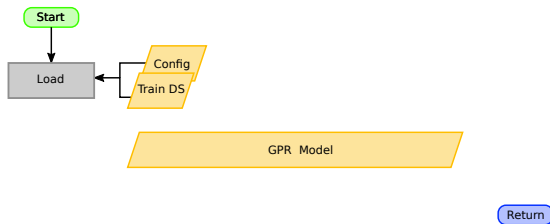
Likelihoods

MSLLs

GPR-Verfahren

GPR-Trainingsphase

Init GPR:



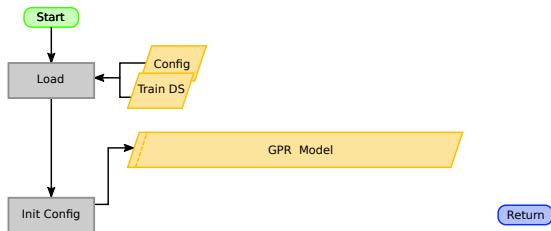
- MAT-Dateien, Struct

GPR-Verfahren

GPR-Trainingsphase

Init GPR:

- MAT-Dateien, Struct
- Infos, Bounds, etc.

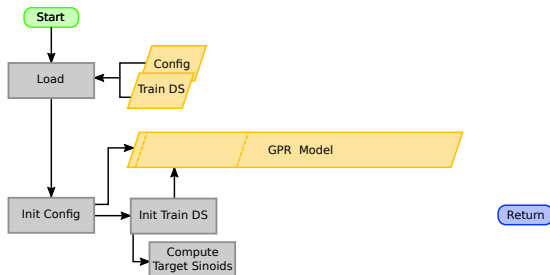


GPR-Verfahren

GPR-Trainingsphase

Init GPR:

- MAT-Dateien, Struct
- Infos, Bounds, etc.
- Einheitskreis, Bezüge

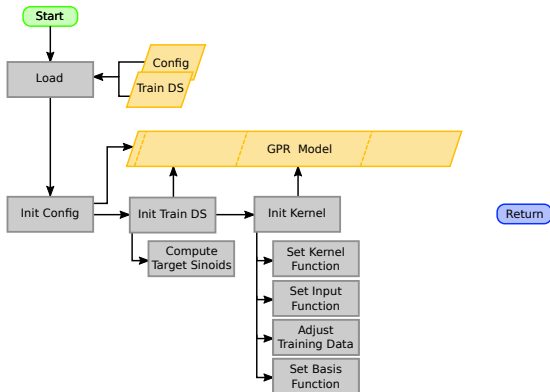


GPR-Verfahren

GPR-Trainingsphase

Init GPR:

- MAT-Dateien, Struct
- Infos, Bounds, etc.
- Einheitskreis, Bezüge
- Module, Data-Fit

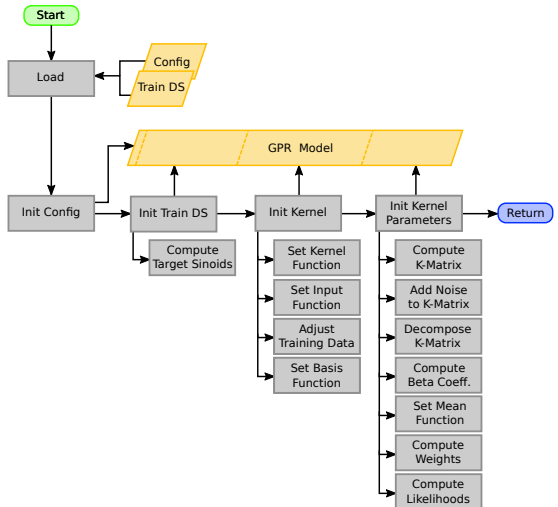


GPR-Verfahren

GPR-Trainingsphase

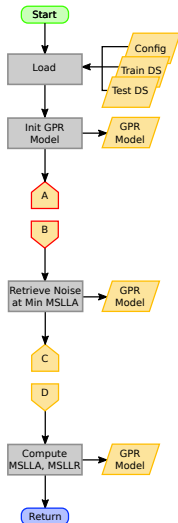
Init GPR:

- MAT-Dateien, Struct
- Infos, Bounds, etc.
- Einheitskreis, Bezüge
- Module, Data-Fit
- GPR Parametrierung



GPR-Verfahren

GPR-Trainingsphase



GPR Training

Simulative Inputs

Encoder Angles

Sensor-Array Data

Input/ Output Functions

Angle Conversion Functions

Input Function

GPR Optimization

Tune Kernel

Optimize Noise

Volatile Stack

Test Data

K-Matrix

Intermediate Results

Kernel

Basis Function

Mean Function

Kernel Function

Predictions

Prediction

Risk/ Loss

Vector/ Matrix Stack

Ref. Angles

Target Data

Training Data

Decomp. K-Matrix (L)

Sinoid Weights

Basic Functions

Trigonometric Functions

Algebraic Functions

Analytic Functions

Parameter Stack

Kernel Parameter

Beta Coefficients

log. Determinante

Likelihoods

MSLLs

Config Stack

Model Info

Kernel Info

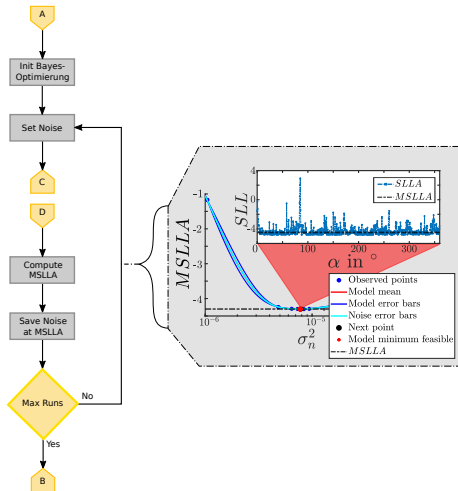
Parameter Info

Parameter Bounds

GPR-Verfahren

GPR-Trainingsphase

Äußere Optimierung (Modell):

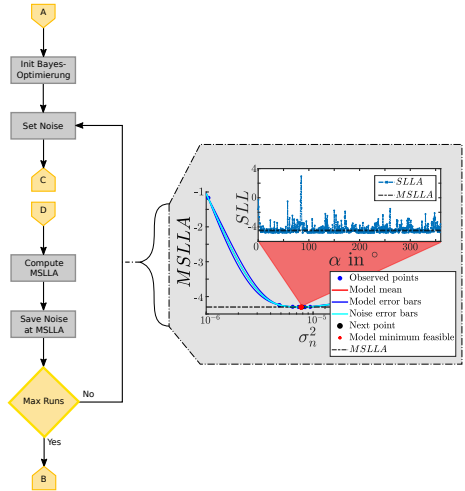


GPR-Verfahren

GPR-Trainingsphase

Äußere Optimierung (Modell):

- Bayes-Optimierung, Std.-Verfahren
- Improve-Per-2nd +

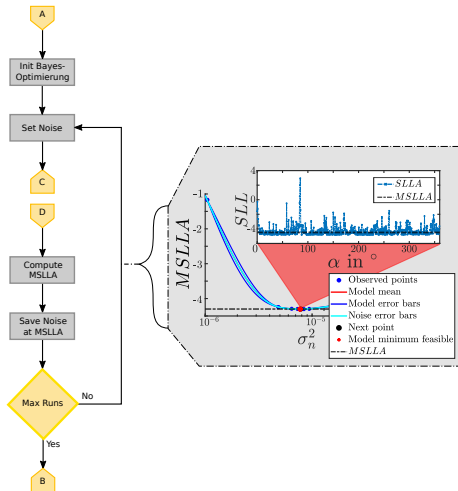


GPR-Verfahren

GPR-Trainingsphase

Äußere Optimierung (Modell):

- Bayes-Optimierung, Std.-Verfahren
- Improve-Per-2nd +
- Anpassung üb. alle Winkel

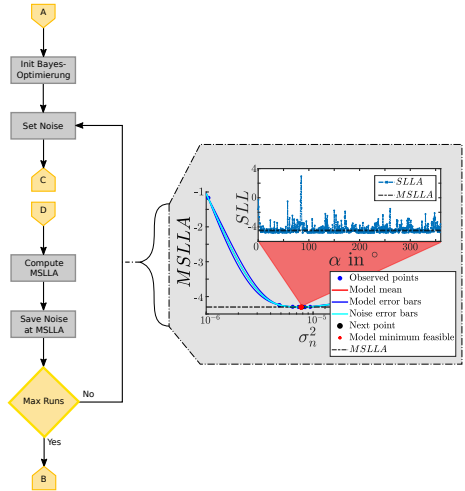


GPR-Verfahren

GPR-Trainingsphase

Äußere Optimierung (Modell):

- Bayes-Optimierung, Std.-Verfahren
- Improve-Per-2nd +
- Anpassung üb. alle Winkel
- Vorgabe von $\sigma_n^2 = konst.$

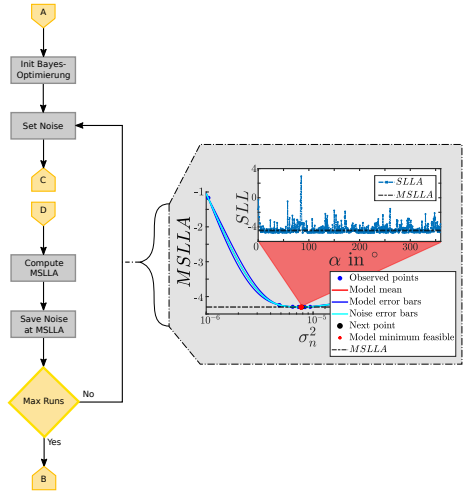


GPR-Verfahren

GPR-Trainingsphase

Äußere Optimierung (Modell):

- Bayes-Optimierung, Std.-Verfahren
- Improve-Per- $2^{nd}+$
- Anpassung üb. alle Winkel
- Vorgabe von $\sigma_n^2 = konst.$
- Durchlaufzahl entscheidend

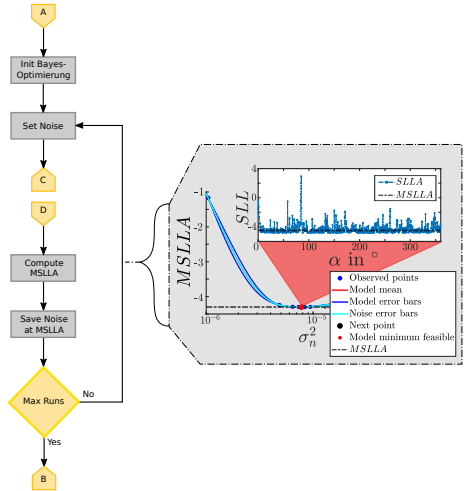


GPR-Verfahren

GPR-Trainingsphase

Äußere Optimierung (Modell):

- Bayes-Optimierung, Std.-Verfahren
- Improve-Per- $2^{nd}+$
- Anpassung üb. alle Winkel
- Vorgabe von $\sigma_n^2 = konst.$
- Durchlaufzahl entscheidend
- Modellsuche



GPR-Verfahren

GPR-Trainingsphase

Äußere Optimierung (Modell):

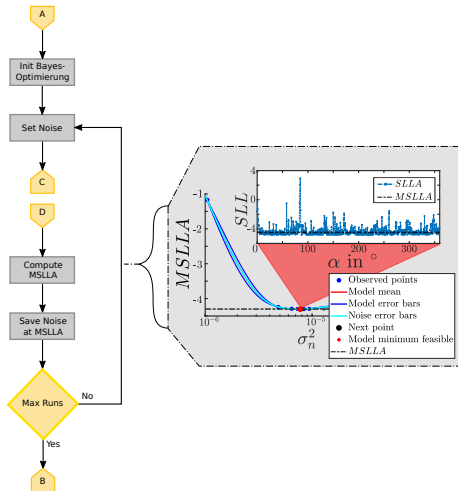
- Bayes-Optimierung, Std.-Verfahren
- Improve-Per-2nd +
- Anpassung üb. alle Winkel
- Vorgabe von $\sigma_n^2 = konst.$
- Durchlaufzahl entscheidend
- Modellsuche

Min.-Kriterium

$$\sigma_n^2 | \mathcal{D}, \alpha_* = \arg \min MSLLA(\sigma_n^2 | \mathcal{D}, \alpha_*)$$

$$SLLA = 0,5 \cdot \left(\log(2\pi\sigma_*^2) + \frac{(\alpha - \alpha_*)^2}{\sigma_*^2} \right)$$

$$\sigma_*^2 = \sigma_n^2 + V_*$$



GPR-Verfahren

GPR-Trainingsphase

Äußere Optimierung (Modell):

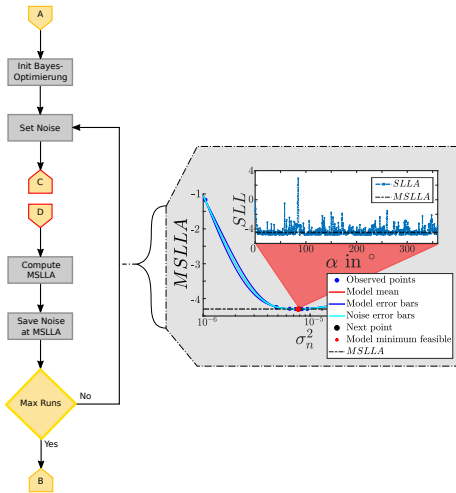
- Bayes-Optimierung, Std.-Verfahren
- Improve-Per-2nd +
- Anpassung üb. alle Winkel
- Vorgabe von $\sigma_n^2 = konst.$
- Durchlaufzahl entscheidend
- Modellsuche

Min.-Kriterium

$$\sigma_n^2 | \mathcal{D}, \alpha_* = \arg \min MSLLA(\sigma_n^2 | \mathcal{D}, \alpha_*)$$

$$SLLA = 0,5 \cdot \left(\log(2\pi\sigma_*^2) + \frac{(\alpha - \alpha_*)^2}{\sigma_*^2} \right)$$

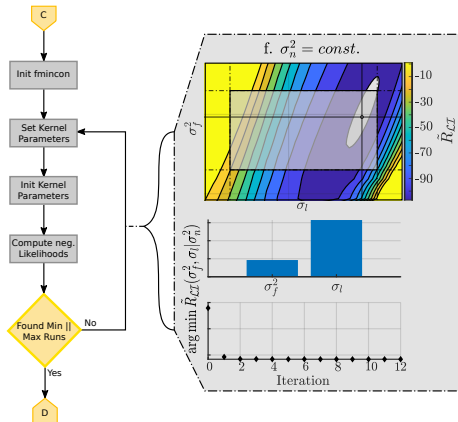
$$\sigma_*^2 = \sigma_n^2 + V_*$$



GPR-Verfahren

GPR-Trainingsphase

Innere Optimierung (Fit):

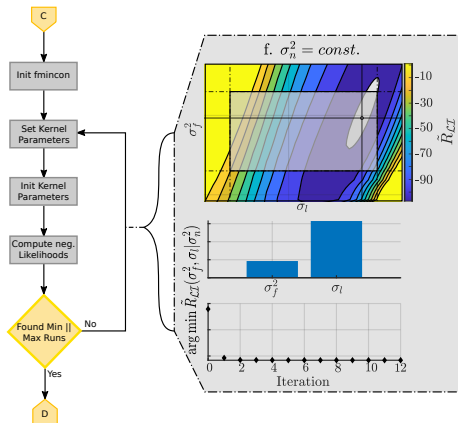


GPR-Verfahren

GPR-Trainingsphase

Innere Optimierung (Fit):

- fmincon, Std.-Verfahren
- SQP, klein- bis mittlere Probleme

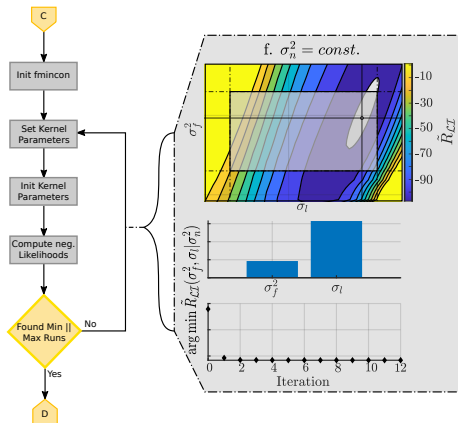


GPR-Verfahren

GPR-Trainingsphase

Innere Optimierung (Fit):

- fmincon, Std.-Verfahren
- SQP, klein- bis mittlere Probleme
- Fit auf Trainingsdaten

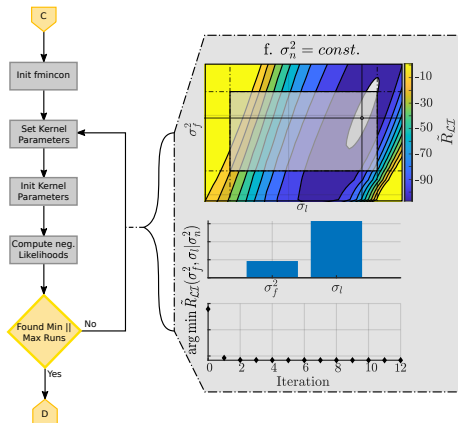


GPR-Verfahren

GPR-Trainingsphase

Innere Optimierung (Fit):

- fmincon, Std.-Verfahren
- SQP, klein- bis mittlere Probleme
- Fit auf Trainingsdaten
- Teilreinitialisierung

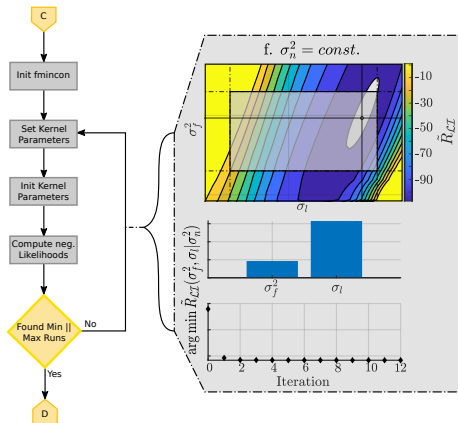


GPR-Verfahren

GPR-Trainingsphase

Innere Optimierung (Fit):

- fmincon, Std.-Verfahren
- SQP, klein- bis mittlere Probleme
- Fit auf Trainingsdaten
- Teilreinitialisierung
- Bounds entscheidend

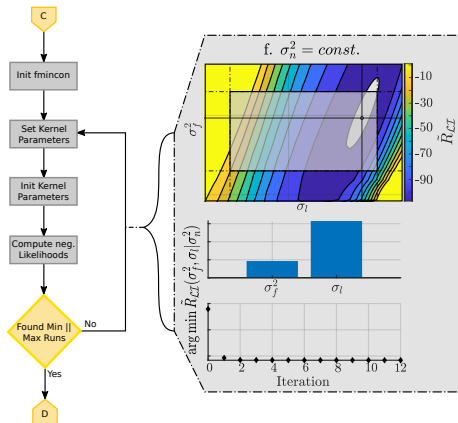


GPR-Verfahren

GPR-Trainingsphase

Innere Optimierung (Fit):

- fmincon, Std.-Verfahren
- SQP, klein- bis mittlere Probleme
- Fit auf Trainingsdaten
- Teilreinitialisierung
- Bounds entscheidend
- Best-Fit-Suche



GPR-Verfahren

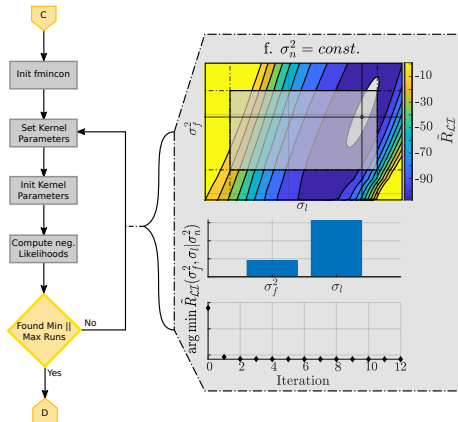
GPR-Trainingsphase

Innere Optimierung (Fit):

- fmincon, Std.-Verfahren
- SQP, klein- bis mittlere Probleme
- Fit auf Trainingsdaten
- Teilreinitialisierung
- Bounds entscheidend
- Best-Fit-Suche

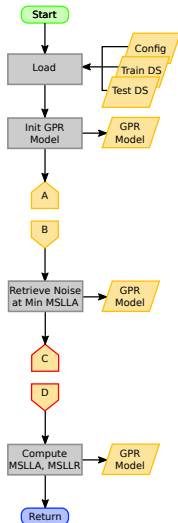
Min.-Kriterium

$$\sigma_l, \sigma_f^2 | \sigma_n^2 = \arg \min \tilde{R}_{\mathcal{L}}(\sigma_l, \sigma_f^2 | \sigma_n^2)$$
$$\tilde{R}_{\mathcal{L}}(\sigma_l, \sigma_f^2 | \sigma_n^2) = -(\log p(y_{\cos} | X_{\cos}, \sigma_l, \sigma_f^2, \sigma_n^2) + \log p(y_{\sin} | X_{\sin}, \sigma_l, \sigma_f^2, \sigma_n^2))$$



GPR-Verfahren

GPR-Trainingsphase



GPR Training

Simulative Inputs

Encoder Angles

Sensor-Array Data

Input/ Output Functions

Angle Conversion Functions

Input Function

GPR Optimization

Tune Kernel

Optimize Noise

Kernel

Basis Function

Mean Function

Kernel Function

Predictions

Prediction

Risk/ Loss

Basic Functions

Trigonometric Functions

Algebraic Functions

Analytic Functions

Config Stack

Model Info

Kernel Info

Parameter Info

Parameter Bounds

Volatile Stack

Test Data

K-Matrix

Intermediate Results

Vector/ Matrix Stack

Ref. Angles

Target Data

Training Data

Decomp. K-Matrix (L)

Sinoid Weights

Parameter Stack

Kernel Parameter

Beta Coefficients

log. Determinante

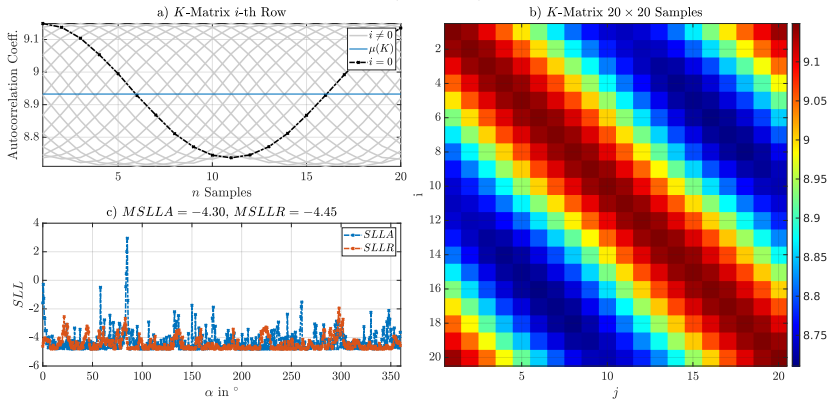
Likelihoods

MSLLs

GPR-Verfahren

GPR-Trainingsphase

Kernel QFCAPX: $\sigma_f = 9.28$, $\sigma_l = 31.48$, $\sigma_n^2 = 7.77e-06$, $N = 20$
8 × 8 Sensor-Array, Position: (0.0, 0.0, -7.0) mm, Magnet Tilt: 0.0°

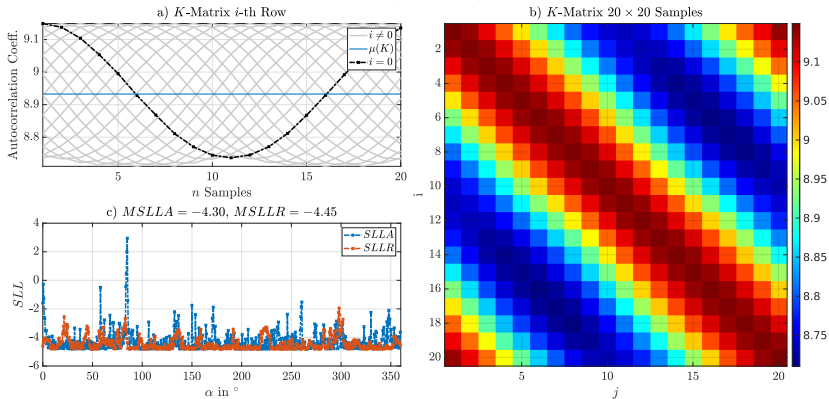


- Fit abhängig von Auflösung

GPR-Verfahren

GPR-Trainingsphase

Kernel QFCAPX: $\sigma_f = 9.28$, $\sigma_l = 31.48$, $\sigma_n^2 = 7.77e-06$, $N = 20$
8 × 8 Sensor-Array, Position: (0.0, 0.0, -7.0) mm, Magnet Tilt: 0.0°

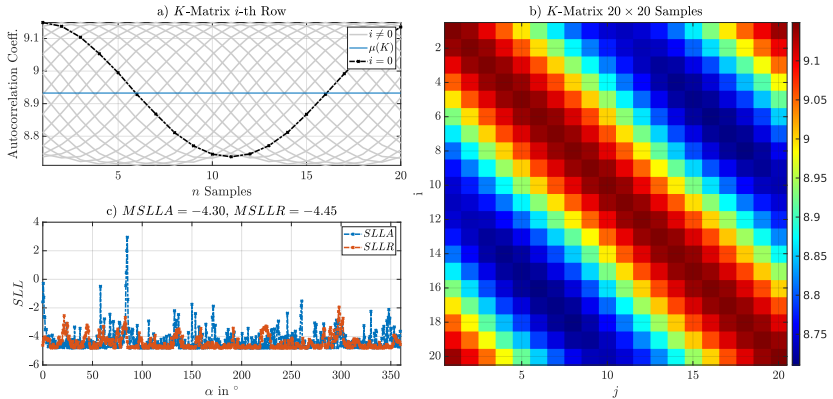


- Fit abhängig von Auflösung
- Ausgleich $\mu(K)$, mittige Platzierung

GPR-Verfahren

GPR-Trainingsphase

Kernel QFCAPX: $\sigma_f = 9.28$, $\sigma_l = 31.48$, $\sigma_n^2 = 7.77e-06$, $N = 20$
8 × 8 Sensor-Array, Position: (0.0, 0.0, -7.0) mm, Magnet Tilt: 0.0°



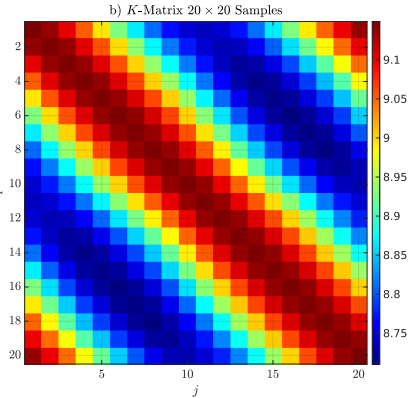
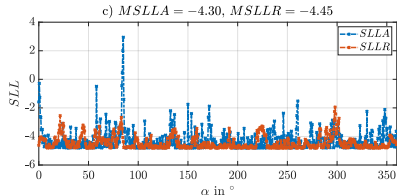
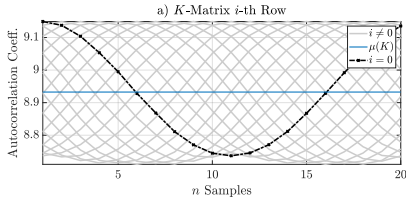
- Fit abhängig von Auflösung
- Ausgleich $\mu(K)$, mittige Platzierung

- SLLA nur in Trainingsphase

GPR-Verfahren

GPR-Trainingsphase

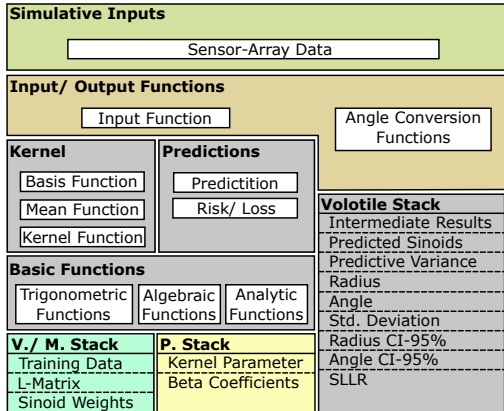
Kernel QFCAPX: $\sigma_f = 9.28$, $\sigma_l = 31.48$, $\sigma_n^2 = 7.77e-06$, $N = 20$
8 × 8 Sensor-Array, Position: (0.0, 0.0, -7.0) mm, Magnet Tilt: 0.0°



- Fit abhängig von Auflösung
- Ausgleich $\mu(K)$, mittige Platzierung

- SLLA nur in Trainingsphase
- SLLR in beiden Phasen

GPR Working

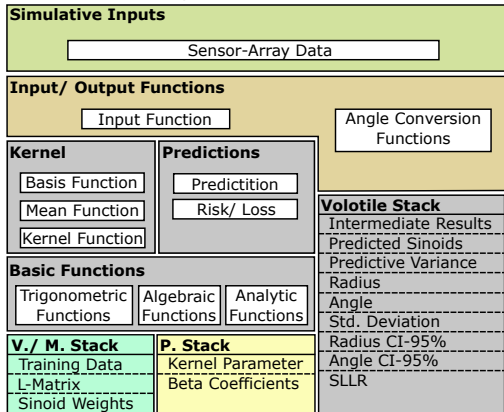


GPR-Verfahren

GPR-Arbeitsphase

- Umlegen d. Ausrichtung

GPR Working

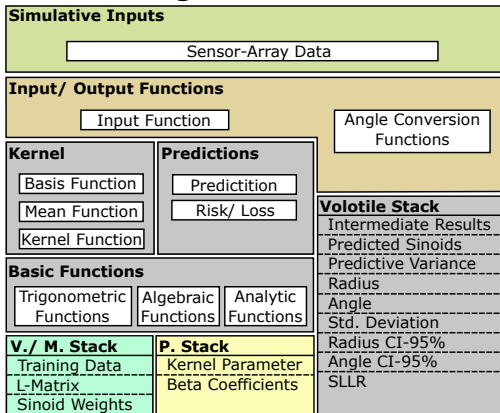


GPR-Verfahren

GPR-Arbeitsphase

- Umlegen d. Ausrichtung
- Min. Parametrierung

GPR Working

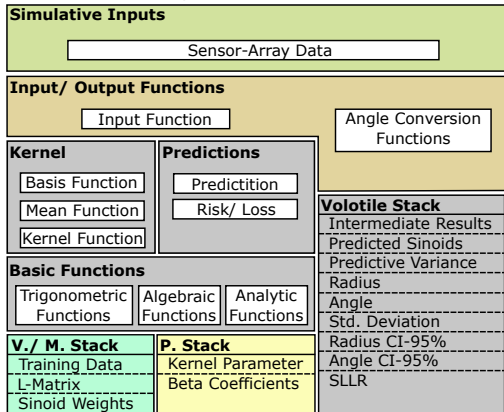


GPR-Verfahren

GPR-Arbeitsphase

- Umlegen d. Ausrichtung
- Min. Parametrierung
- Freie Ressourcen

GPR Working

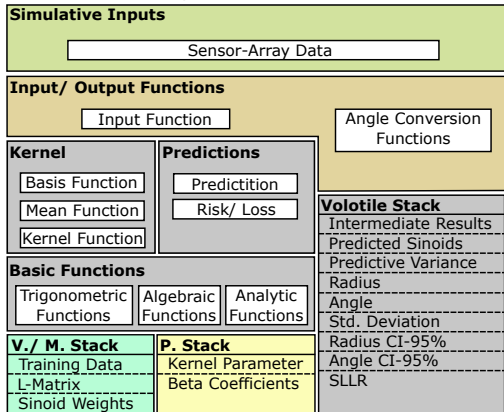


GPR-Verfahren

GPR-Arbeitsphase

- Umlegen d. Ausrichtung
- Min. Parametrierung
- Freie Ressourcen
- Vorhersage Sinoide

GPR Working

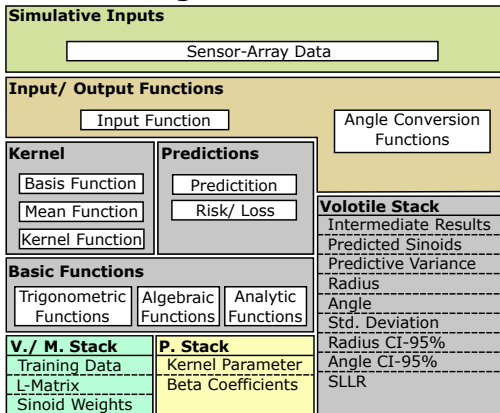


GPR-Verfahren

GPR-Arbeitsphase

- Umlegen d. Ausrichtung
- Min. Parametrierung
- Freie Ressourcen
- Vorhersage Sinoide
- Winkel, Radius abgeleitet

GPR Working

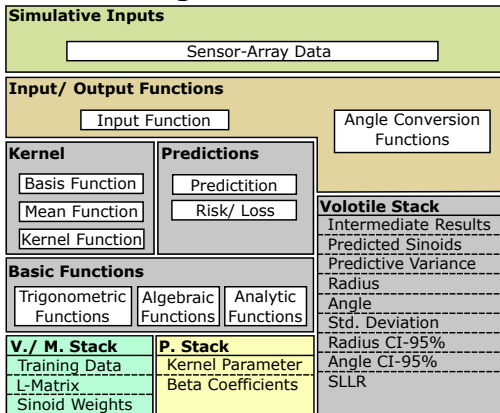


GPR-Verfahren

GPR-Arbeitsphase

- Umlegen d. Ausrichtung
- Min. Parametrierung
- Freie Ressourcen
- Vorhersage Sinoide
- Winkel, Radius abgeleitet
- Mehrere Qualitätskriterien
 - ▶ Std.-Abweichung

GPR Working

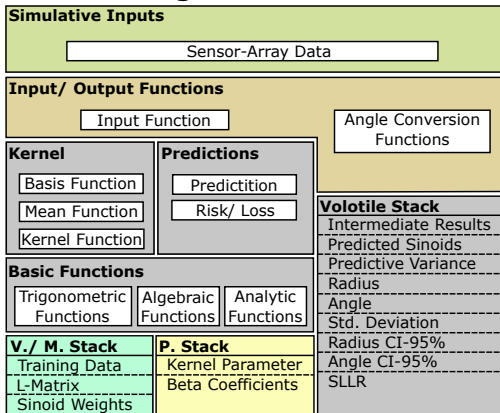


GPR-Verfahren

GPR-Arbeitsphase

- Umlegen d. Ausrichtung
- Min. Parametrierung
- Freie Ressourcen
- Vorhersage Sinoide
- Winkel, Radius abgeleitet
- Mehrere Qualitätskriterien
 - ▶ Std.-Abweichung
 - ▶ Konfidenzintervall Winkel
 - ▶ Konfidenzintervall Radius

GPR Working

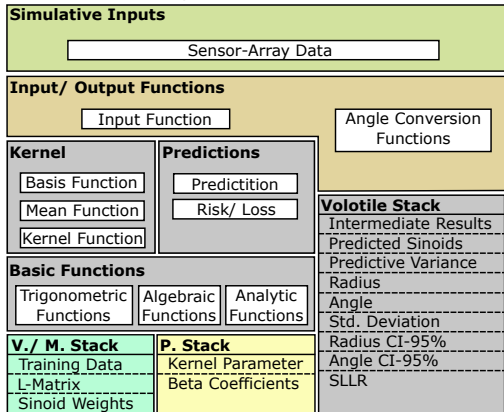


GPR-Verfahren

GPR-Arbeitsphase

- Umlegen d. Ausrichtung
- Min. Parametrierung
- Freie Ressourcen
- Vorhersage Sinoide
- Winkel, Radius abgeleitet
- Mehrere Qualitätskriterien
 - ▶ Std.-Abweichung
 - ▶ Konfidenzintervall Winkel
 - ▶ Konfidenzintervall Radius
 - ▶ Modellanpassung Radius

GPR Working

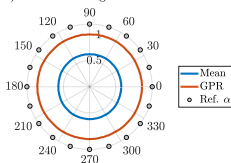


GPR-Verfahren

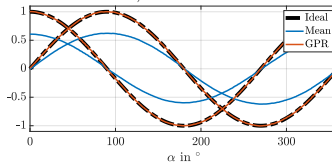
GPR-Arbeitsphase

Kernel QFCAPX: $\sigma_f = 9.28$, $\sigma_l = 31.48$, $\sigma_n^2 = 7.77e-06$, $N = 20$
8 x 8 Sensor-Array, Position: (0.0, 0.0, -7.0) mm, Magnet Tilt: 0.0°

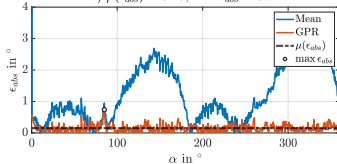
a) Rotation along Z-Axis in °



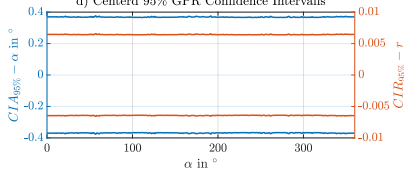
b) Sine and Cosine



c) $\mu(\epsilon_{abs}) = 0.15$, $\max \epsilon_{abs} = 0.74$



d) Centerd 95% GPR Confidence Intervals



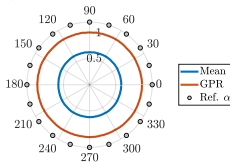
● 20 Referenzwinkel

GPR-Verfahren

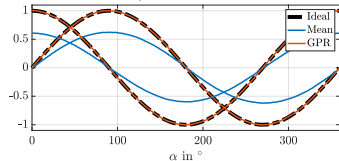
GPR-Arbeitsphase

Kernel QFCAPX: $\sigma_f = 9.28$, $\sigma_l = 31.48$, $\sigma_n^2 = 7.77e-06$, $N = 20$
8 × 8 Sensor-Array, Position: (0.0, 0.0, -7.0) mm, Magnet Tilt: 0.0°

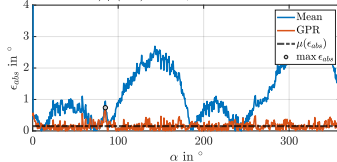
a) Rotation along Z-Axis in °



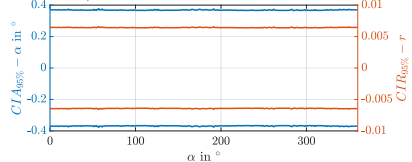
b) Sine and Cosine



c) $\mu(\epsilon_{abs}) = 0.15$, $\max \epsilon_{abs} = 0.74$



d) Centerd 95% GPR Confidence Intervals



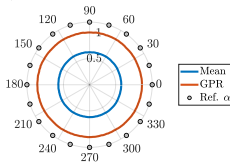
- 20 Referenzwinkel
- 720 Winkel, 0.5° Auflösung

GPR-Verfahren

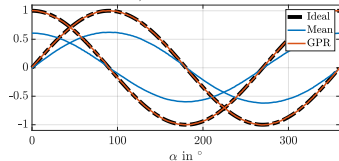
GPR-Arbeitsphase

Kernel QFCAPX: $\sigma_f = 9.28$, $\sigma_l = 31.48$, $\sigma_n^2 = 7.77e-06$, $N = 20$
 8×8 Sensor-Array, Position: (0.0, 0.0, -7.0) mm, Magnet Tilt: 0.0°

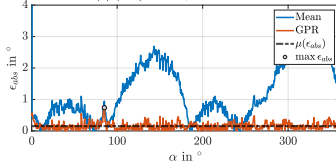
a) Rotation along Z-Axis in °



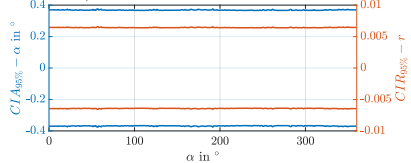
b) Sine and Cosine



c) $\mu(\epsilon_{abs}) = 0.15$, $\max \epsilon_{abs} = 0.74$



d) Centerd 95% GPR Confidence Intervals



- 20 Referenzwinkel
- 720 Winkel, 0.5° Auflösung

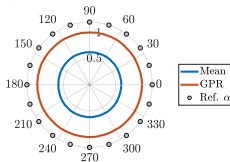
- Approximierter Kernel (1)

GPR-Verfahren

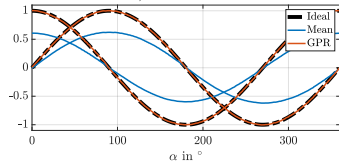
GPR-Arbeitsphase

Kernel QFCAPX: $\sigma_f = 9.28$, $\sigma_l = 31.48$, $\sigma_n^2 = 7.77e-06$, $N = 20$
8 × 8 Sensor-Array, Position: (0.0, 0.0, -7.0) mm, Magnet Tilt: 0.0°

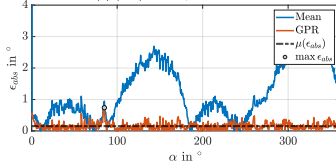
a) Rotation along Z-Axis in °



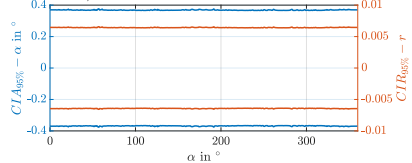
b) Sine and Cosine



c) $\mu(\epsilon_{abs}) = 0.15$, $\max \epsilon_{abs} = 0.74$



d) Centerd 95% GPR Confidence Intervals

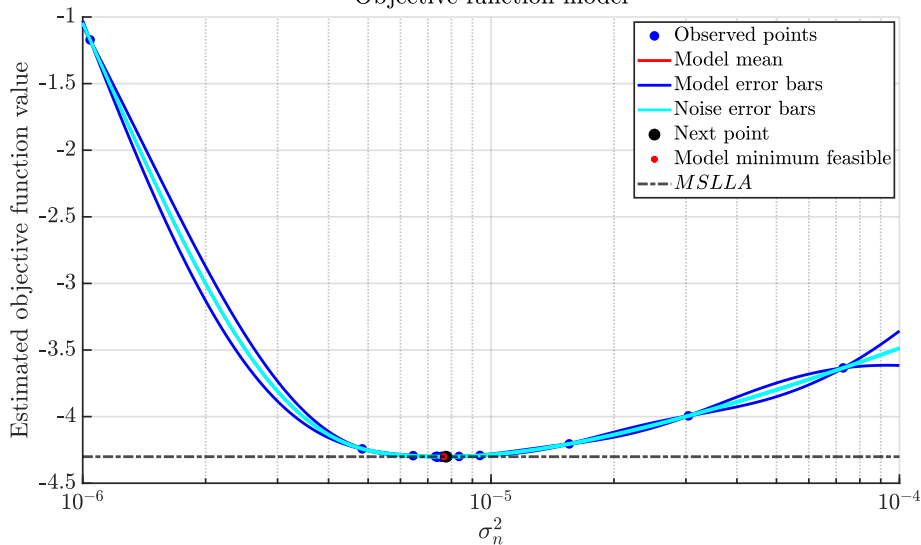


- 20 Referenzwinkel
- 720 Winkel, 0.5° Auflösung

- Approximierter Kernel (1)
- Konfidenzintervall konst.

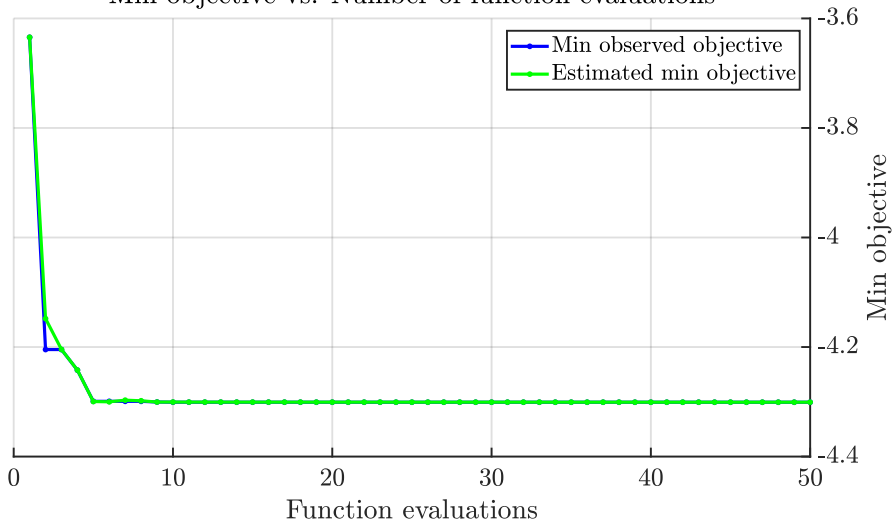
Vielen Dank!

Kernel QFCAPX: $\sigma_f = 9.15$, $\sigma_l = 31.33$, $\sigma_n^2 = 7.75e - 06$, $N = 20$
 8×8 Sensor-Array, Position: $(0.0, 0.0, -7.0)$ mm, Magnet Tilt: 0.0°
 Objective function model



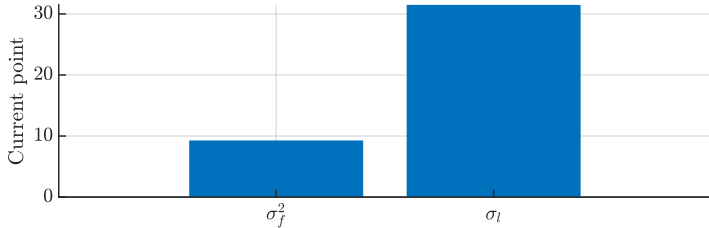
Kernel QFCAPX: $\sigma_f = 9.15$, $\sigma_l = 31.33$, $\sigma_n^2 = 7.75e - 06$, $N = 20$
 8×8 Sensor-Array, Position: $(0.0, 0.0, -7.0)$ mm, Magnet Tilt: 0.0°

Min objective vs. Number of function evaluations



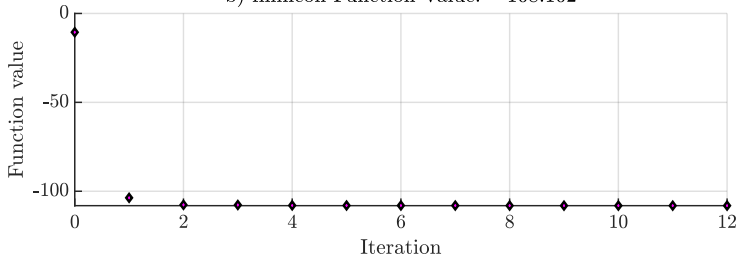
Kernel QFCAPX: $\sigma_f = 9.15$, $\sigma_l = 31.33$, $\sigma_n^2 = 7.75e - 06$, $N = 20$
8 × 8 Sensor-Array, Position: (0.0, 0.0, -7.0) mm, Magnet Tilt: 0.0°

a) fmincon Parameter



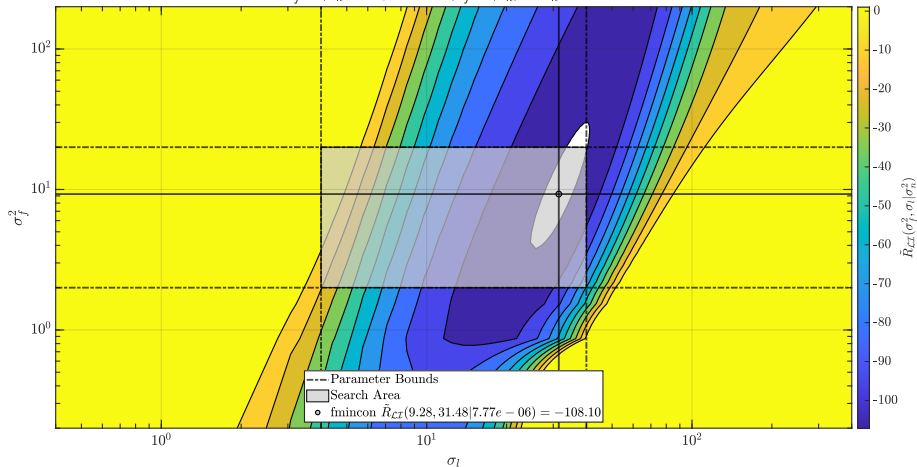
Number of variables: 2

b) fmincon Function Value: -108.102



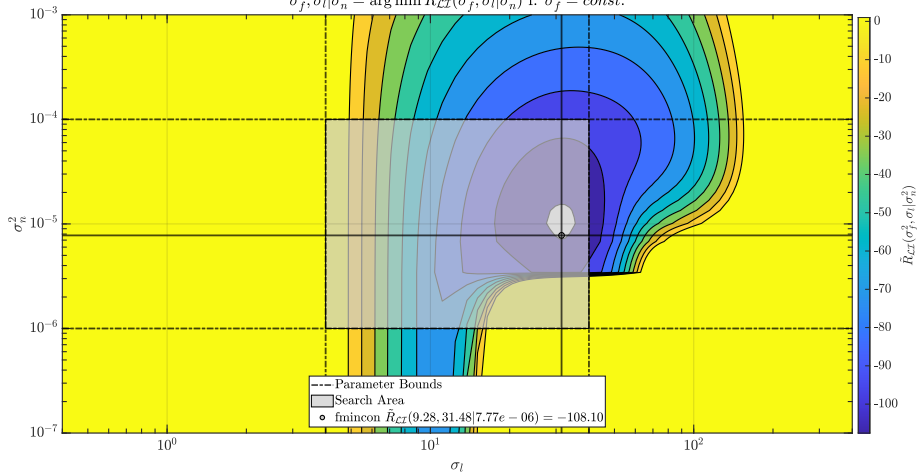
Kernel QFCAPX: $\sigma_f = 9.28$, $\sigma_l = 31.48$, $\sigma_n^2 = 7.77e-06$, $N = 20$
 8×8 Sensor-Array, Position: $(0.0, 0.0, -7.0)$ mm, Magnet Tilt: 0.0°

$$\sigma_f^2, \sigma_l | \sigma_n^2 = \arg \min \tilde{R}_{\mathcal{LX}}(\sigma_f^2, \sigma_l | \sigma_n^2) \text{ f. } \sigma_n^2 = \text{const.}$$



Kernel QFCAPX: $\sigma_f = 9.28$, $\sigma_l = 31.48$, $\sigma_n^2 = 7.77e-06$, $N = 20$
 8×8 Sensor-Array, Position: $(0.0, 0.0, -7.0)$ mm, Magnet Tilt: 0.0°

$$\sigma_f^2, \sigma_l | \sigma_n^2 = \arg \min \hat{R}_{\mathcal{CI}}(\sigma_f^2, \sigma_l | \sigma_n^2) \text{ f. } \sigma_f^2 = \text{const.}$$



Kernel QFCAPX: $\sigma_f = 9.28$, $\sigma_l = 31.48$, $\sigma_n^2 = 7.77e-06$, $N = 20$
 8×8 Sensor-Array, Position: (0.0, 0.0, -7.0) mm, Magnet Tilt: 0.0°

$$\sigma_f^2, \sigma_l | \sigma_n^2 = \arg \min \tilde{R}_{\mathcal{CI}}(\sigma_f^2, \sigma_l | \sigma_n^2) \text{ f. } \sigma_l = \text{const.}$$

