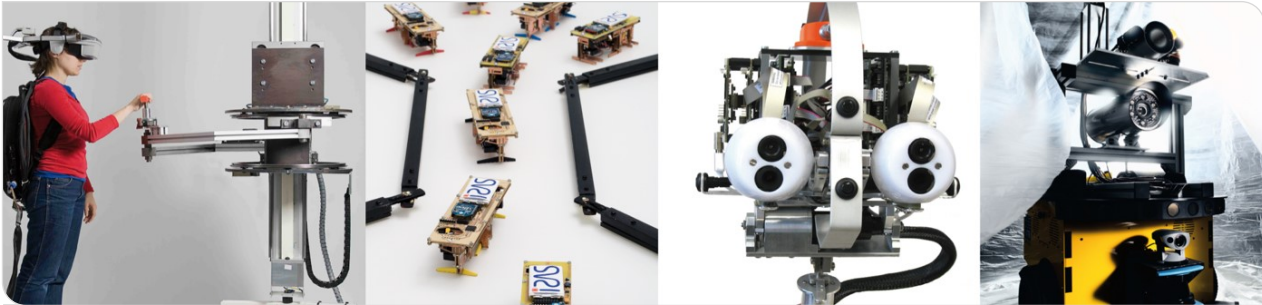


Model Predictive Control with Bayesian Last Layer Trust Regions

Johannes Gaus | 27. Juni 2024

Betreuer: Markus Walker



Inhaltsübersicht

- Motivation
 - Einführung und Funktionsweise der modellprädiktiven Regelung
 - Einführung und Funktionsweise von Bayesian Neuronalen Netzen
 - Einführung und Funktionsweise von Bayesian Last Layer Trust Regions
 - Integration von Bayesian Last Layer Trust Regions in die modellprädiktive Regelung
 - Fazit und Ausblick

Inhaltsübersicht

- Motivation
- Einführung und Funktionsweise der modellprädiktiven Regelung
- Einführung und Funktionsweise von Bayesian Neuronalen Netzen
- Einführung und Funktionsweise von Bayesian Last Layer Trust Regions
- Integration von Bayesian Last Layer Trust Regions in die modellprädiktive Regelung
- Fazit und Ausblick

Inhaltsübersicht

- Motivation
- Einführung und Funktionsweise der modellprädiktiven Regelung
- Einführung und Funktionsweise von Bayesian Neuronalen Netzen
- Einführung und Funktionsweise von Bayesian Last Layer Trust Regions
- Integration von Bayesian Last Layer Trust Regions in die modellprädiktive Regelung
- Fazit und Ausblick

Inhaltsübersicht

- Motivation
- Einführung und Funktionsweise der modellprädiktiven Regelung
- Einführung und Funktionsweise von Bayesian Neuronalen Netzen
- Einführung und Funktionsweise von Bayesian Last Layer Trust Regions
- Integration von Bayesian Last Layer Trust Regions in die modellprädiktive Regelung
- Fazit und Ausblick

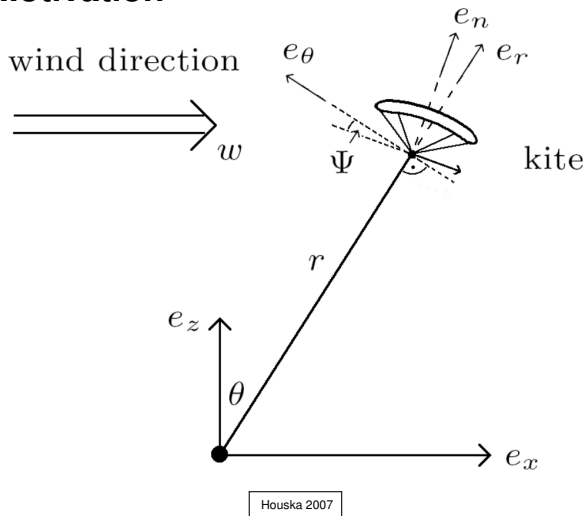
Inhaltsübersicht

- Motivation
- Einführung und Funktionsweise der modellprädiktiven Regelung
- Einführung und Funktionsweise von Bayesian Neuronalen Netzen
- Einführung und Funktionsweise von Bayesian Last Layer Trust Regions
- Integration von Bayesian Last Layer Trust Regions in die modellprädiktive Regelung
- Fazit und Ausblick

Inhaltsübersicht

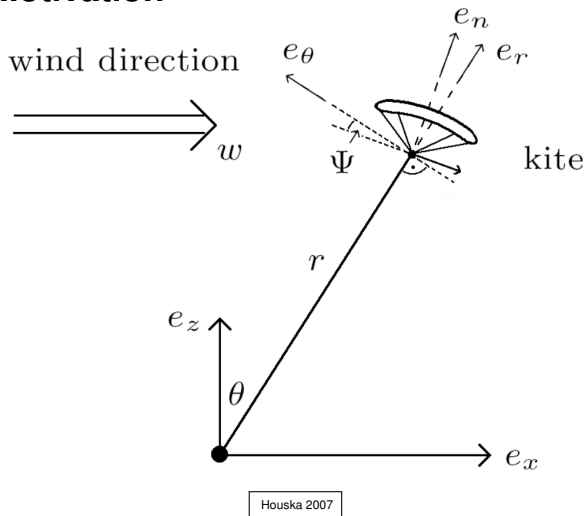
- Motivation
- Einführung und Funktionsweise der modellprädiktiven Regelung
- Einführung und Funktionsweise von Bayesian Neuronalen Netzen
- Einführung und Funktionsweise von Bayesian Last Layer Trust Regions
- Integration von Bayesian Last Layer Trust Regions in die modellprädiktive Regelung
- Fazit und Ausblick

Motivation



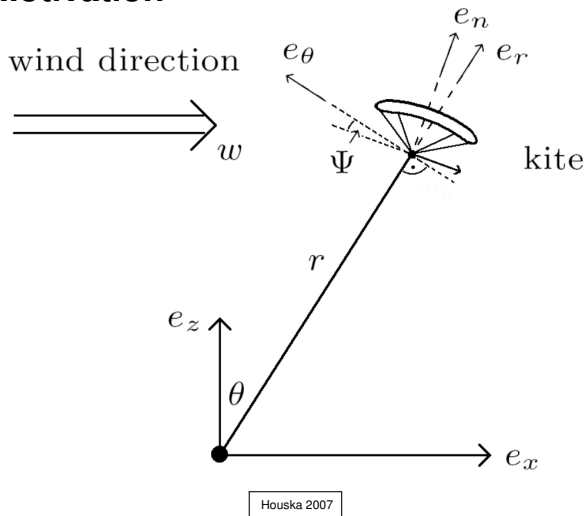
- Welche Stellgrößen können wir hier verändern, um uns sicher in eine Richtung zu bewegen?
- θ (Zenitwinkel): Winkel des Kites relativ zum Ankerpunkt
- ϕ (Azimutwinkel): Winkel des Kites in der horizontalen Ebene
- ψ (Ausrichtung): Yaw-Ausrichtung des Kites

Motivation



- Welche Stellgrößen können wir hier verändern, um uns sicher in eine Richtung zu bewegen?
- θ (Zenitwinkel): Winkel des Kites relativ zum Ankerpunkt
- ϕ (Azimutwinkel): Winkel des Kites in der horizontalen Ebene
- ψ (Ausrichtung): Yaw-Ausrichtung des Kites

Motivation



Externe Eingaben:

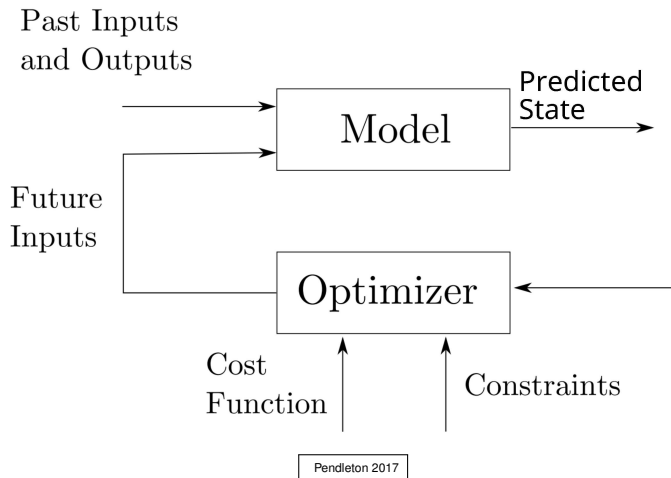
- Glide Ratio: Verhältnis zwischen horizontaler und vertikaler Geschwindigkeit des Kites
- Windgeschwindigkeit: Umgebungswindgeschwindigkeit, die die Flugdynamik beeinflusst

Motivation

Was ist MPC?

- Modellprädiktive Regelung (MPC) → Regelstrategie, die zukünftige Zustände eines Systems vorhersagt um optimale Stellgrößen zu bestimmen
- Effiziente Nutzung von Ressourcen und Optimierung der Systemleistung durch präzise Vorhersagen und Anpassungen

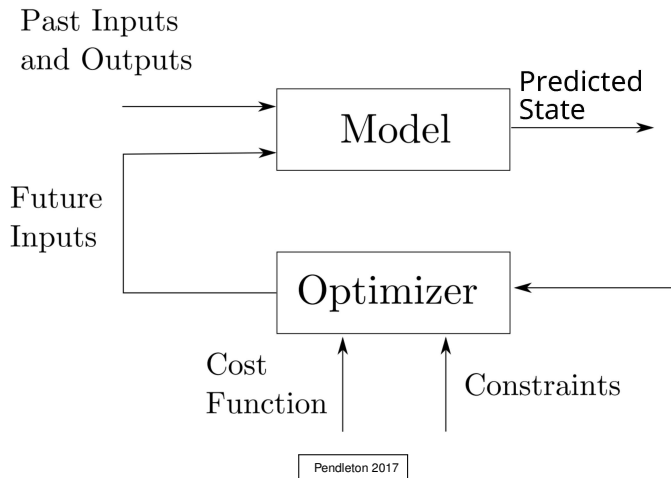
Block Diagramm



■ Optimierungsproblem:

- Ziel: optimale Stellgrößen über einen festgelegten Vorhersagehorizont bestimmen
- Nebenbedingungen, wie z.B. Mindestwerte oder Intervalle für Zustände und Stellgrößen

Block Diagramm



■ Optimierungsproblem:

- Ziel: optimale Stellgrößen über einen festgelegten Vorhersagehorizont bestimmen
- Nebenbedingungen, wie z.B. Mindestwerte oder Intervalle für Zustände und Stellgrößen

MPC Optimierungsproblem

Ziel: Finde die besten Stellgrößen über einen Vorhersagehorizont, um die Kosten zu minimieren

Formulierung:

$$\min_{\mathbf{u}} J(\mathbf{u}) = \sum_{k=0}^{N-1} L(\mathbf{s}(k), \mathbf{u}(k)) + M(\mathbf{s}(N))$$

unter den Nebenbedingungen:

$$\mathbf{s}(k+1) = f(\mathbf{s}(k), \mathbf{u}(k)),$$

$$k = 0, 1, \dots, N-1,$$

$$\mathbf{s}(0) = \mathbf{s}_0,$$

$$\mathbf{u}(k) \in \mathcal{U}, \quad k = 0, 1, \dots, N-1$$

- $\mathbf{s}(k)$: Zustand des Systems zum Zeitpunkt k
- $\mathbf{u}(k)$: Steuereingang zum Zeitpunkt k
- N : Länge des Vorhersagehorizonts
- $L(\cdot)$: Kostenfunktion für jeden Zeitschritt
- $M(\cdot)$: Endkostenfunktion nach dem Horizont
- $f(\cdot)$: Systemdynamik (wie sich der Zustand ändert)
- \mathbf{s}_0 : Anfangszustand des Systems
- \mathcal{U} : Zulässige Stellgrößen

MPC Optimierungsproblem

Ziel: Finde die besten Stellgrößen über einen Vorhersagehorizont, um die Kosten zu minimieren

Formulierung:

$$\min_{\mathbf{u}} J(\mathbf{u}) = \sum_{k=0}^{N-1} L(\mathbf{s}(k), \mathbf{u}(k)) + M(\mathbf{s}(N))$$

unter den Nebenbedingungen:

$$\mathbf{s}(k+1) = f(\mathbf{s}(k), \mathbf{u}(k)),$$

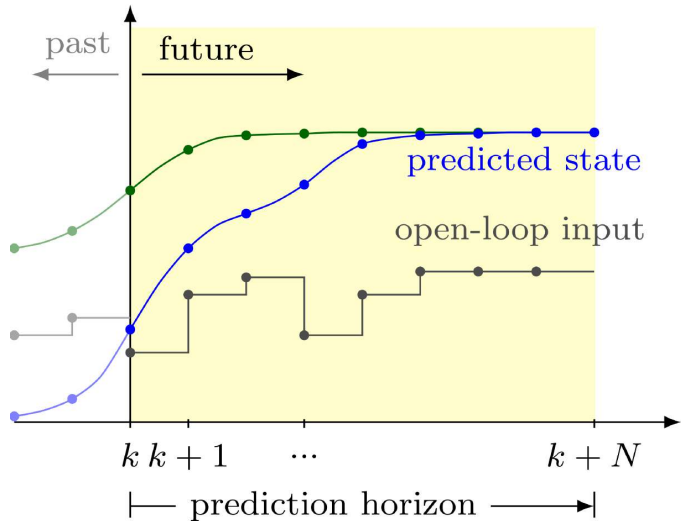
$$k = 0, 1, \dots, N-1,$$

$$\mathbf{s}(0) = \mathbf{s}_0,$$

$$\mathbf{u}(k) \in \mathcal{U}, \quad k = 0, 1, \dots, N-1$$

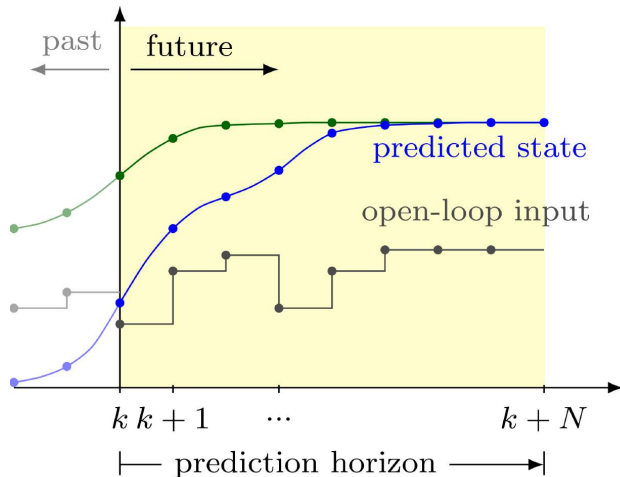
- $\mathbf{s}(k)$: Zustand des Systems zum Zeitpunkt k
- $\mathbf{u}(k)$: Steuereingang zum Zeitpunkt k
- N : Länge des Vorhersagehorizonts
- $L(\cdot)$: Kostenfunktion für jeden Zeitschritt
- $M(\cdot)$: Endkostenfunktion nach dem Horizont
- $f(\cdot)$: Systemdynamik (wie sich der Zustand ändert)
- \mathbf{s}_0 : Anfangszustand des Systems
- \mathcal{U} : Zulässige Stellgrößen

How to MPC



Ebenbauer 2024

How to MPC



Ebenbauer 2024

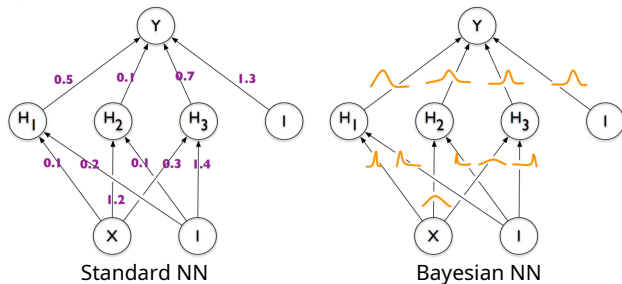
■ Vorhersage:

- Basierend auf dem aktuellen Zustand k und dem Systemmodell werden zukünftige Zustände für einen Vorhersagehorizont $k + N$ berechnet

■ Optimierung:

- Ein Optimierungsproblem wird gelöst, um Stellgrößen k_0, k_1, \dots, k_{N-1} zu finden die Kostenfunktion minimieren und alle Beschränkungen einhalten

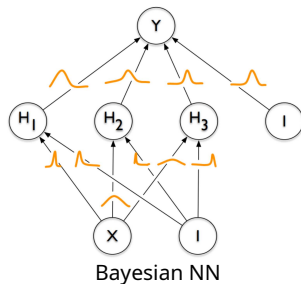
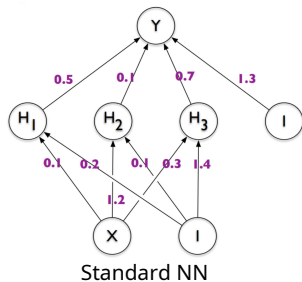
Bayesian Neural Network (BNN)



Blundell u. a. 2015

- Standard NNs → deterministische Ausgabe mit deterministischen Gewichten
- BNNs → probabilistische Ausgabe und Gewichte als Wahrscheinlichkeitsverteilung
- Die Gewichte werden mithilfe des Bayes-Theorems berechnet
- Besserer Umgang mit begrenzten Daten → Nutzen der Modellunsicherheit
- Bayesian Last Layer → nur die Ausgangsschicht ist probabilistisch, alle anderen Schichten bleiben deterministisch

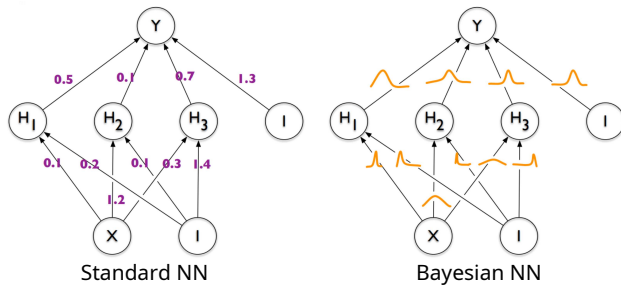
Bayesian Neural Network (BNN)



Blundell u. a. 2015

- Standard NNs → deterministische Ausgabe mit deterministischen Gewichten
- BNNs → probabilistische Ausgabe und Gewichte als Wahrscheinlichkeitsverteilung
- Die Gewichte werden mithilfe des Bayes-Theorems berechnet
- Besserer Umgang mit begrenzten Daten → Nutzen der Modellunsicherheit
- Bayesian Last Layer → nur die Ausgangsschicht ist probabilistisch, alle anderen Schichten bleiben deterministisch

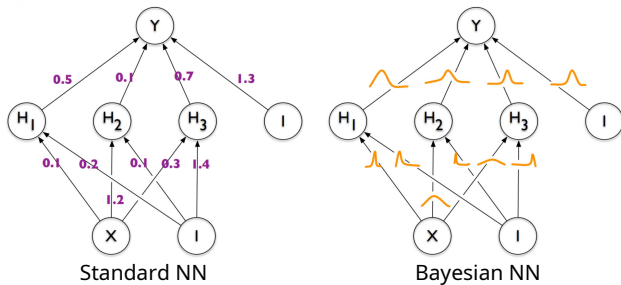
Bayesian Neural Network (BNN)



Blundell u. a. 2015

- Standard NNs → deterministische Ausgabe mit deterministischen Gewichten
- BNNs → probabilistische Ausgabe und Gewichte als Wahrscheinlichkeitsverteilung
- Die Gewichte werden mithilfe des Bayes-Theorems berechnet
- Besserer Umgang mit begrenzten Daten → Nutzen der Modellunsicherheit
- Bayesian Last Layer → nur die Ausgangsschicht ist probabilistisch, alle anderen Schichten bleiben deterministisch

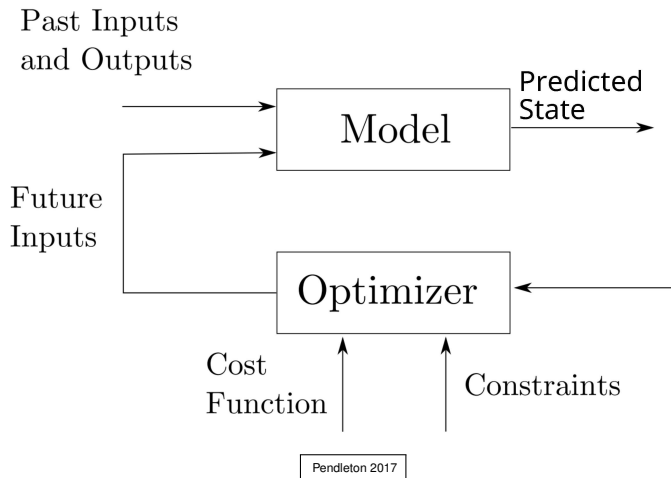
Bayesian Neural Network (BNN)



Blundell u. a. 2015

- Standard NNs → deterministische Ausgabe mit deterministischen Gewichten
- BNNs → probabilistische Ausgabe und Gewichte als Wahrscheinlichkeitsverteilung
- Die Gewichte werden mithilfe des Bayes-Theorems berechnet
- Besserer Umgang mit begrenzten Daten → Nutzen der Modellunsicherheit
- Bayesian Last Layer → nur die Ausgangsschicht ist probabilistisch, alle anderen Schichten bleiben deterministisch

Block Diagramm

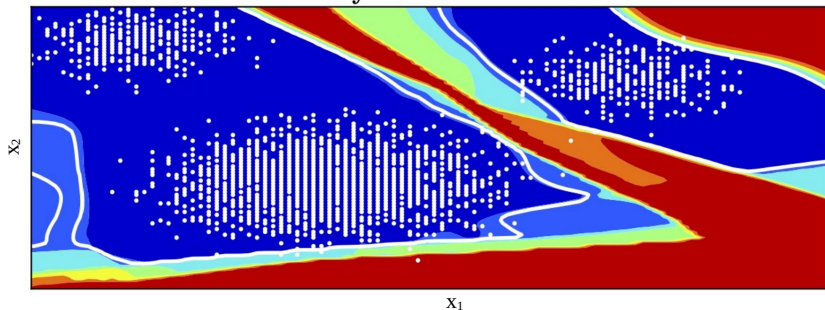


- Anwendung von ML im Systemmodell:
 - Bayesian Last Layer für Systemmodell
 - Bayesian Last Layer Trust Regions

Trust Regions

Neural Network

BLL covariance $\Sigma_{\hat{y}}^{NN}$ and trust region threshold



Fiedler und Lucia 2022

- Bereiche im Eingaberaum, in denen die Vorhersagen des Modells als zuverlässig gelten

■ Bsp: MPC mit Trust Regions

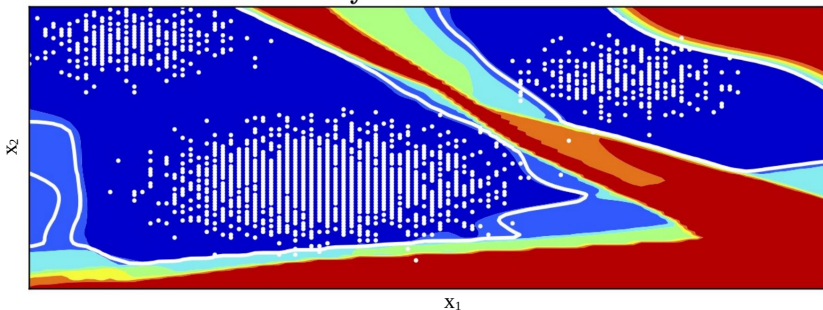


$$C_{\phi} \leq C_{ub}$$

Trust Regions

Neural Network

BLL covariance $\Sigma_{\hat{y}}^{NN}$ and trust region threshold



Fiedler und Lucia 2022

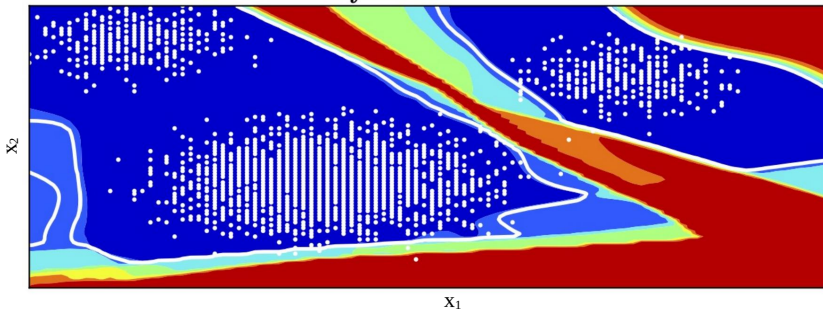
- Bereiche im Eingaberaum, in denen die Vorhersagen des Modells als zuverlässig gelten
- Bsp: MPC mit Trust Regions

$$C_{\phi} \leq C_{ub}$$

Trust Regions

Neural Network

BLL covariance $\Sigma_{\hat{y}}^{NN}$ and trust region threshold



Fiedler und Lucia 2022

- Bereiche im Eingaberaum, in denen die Vorhersagen des Modells als zuverlässig gelten
- Bsp: MPC mit Trust Regions
-

$$\mathbf{C}_{\phi} \leq c_{ub}$$

MPC Optimierungsproblem mit Trust Regions

Trust Regions Zusätzliche Constraints in Form von
Soft Constraints mit Slack-Variablen

Formulierung:

$$\min_{\mathbf{u}, \delta} J(\mathbf{u}) = \sum_{k=0}^{N-1} L(\mathbf{s}(k), \mathbf{u}(k)) + M(\mathbf{s}(N)) + \sigma_{\delta} \sum_{k=0}^{N-1} \delta_i^2,$$
$$\mathbf{C}_{\phi} \leq c_{ub}$$

- \mathbf{C} : Kovarianz aus dem Bayesian Last Layer Ansatz
- δ_i : Slack-Variable, die als Soft Constraint eingeführt wird
- c_{ub} : Vorgegebene Schranke, die die Vertrauenswürdigkeit der Vorhersage angibt

MPC Optimierungsproblem mit Trust Regions

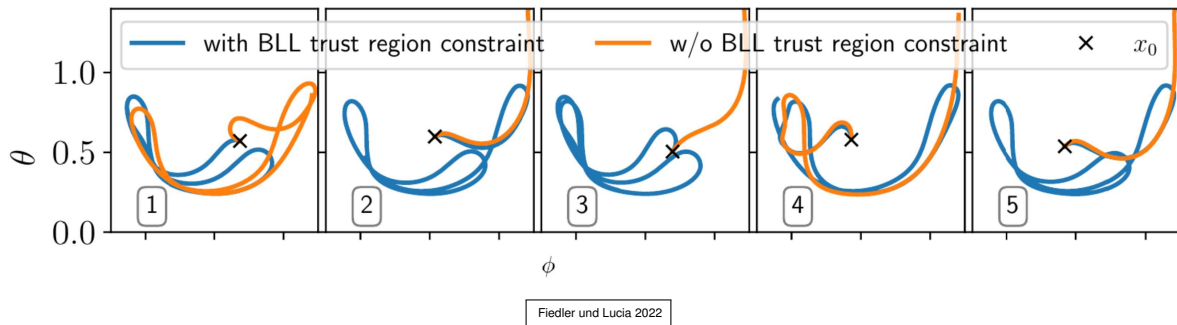
Trust Regions Zusätzliche Constraints in Form von
Soft Constraints mit Slack-Variablen

Formulierung:

$$\min_{\mathbf{u}, \delta} J(\mathbf{u}) = \sum_{k=0}^{N-1} L(\mathbf{s}(k), \mathbf{u}(k)) + M(\mathbf{s}(N)) + \sigma_{\delta} \sum_{k=0}^{N-1} \delta_i^2,$$
$$\mathbf{C}_{\phi} \leq c_{ub}$$

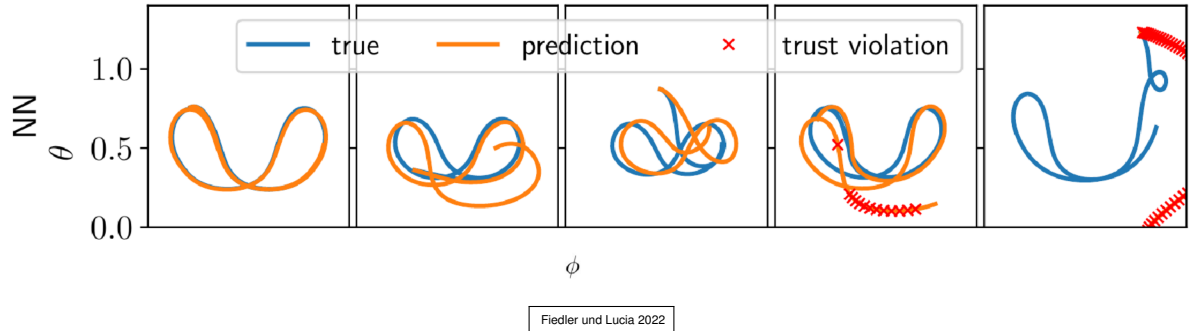
- \mathbf{C} : Kovarianz aus dem Bayesian Last Layer Ansatz
- δ_i : Slack-Variable, die als Soft Constraint eingeführt wird
- c_{ub} : Vorgegebene Schranke, die die Vertrauenswürdigkeit der Vorhersage angibt

Trust Region Case Study w & w/o trust regions



- Trajektorie des Kite-Segels aus der Case Study [Fiedler und Lucia 2022] für Test Case 1 bis 5

Trust Region Case Study true vs. prediction



- Trajektorie des Kite-Segels aus der Case Study [Fiedler und Lucia 2022] für Test Case 1 bis 5

Trust Region Schwellwerte

c_{ub}	M1	M2	M3	M4	M5	M6	mean
0.0001	20	20	20	20	20	20	20
0.0005	20	20	20	20	20	20	20
0.001	20	20	20	20	20	20	20
0.005	20	20	20	20	20	20	20
0.0075	20	20	20	19	20	20	19.8
0.01	20	16	18	15	18	18	17.5
0.02	14	8	9	5	10	7	8.8
0.03	7	3	5	2	5	4	4.3
0.04	6	2	5	2	3	4	3.6
0.05	4	2	5	1	3	3	3.0
0.06	3	2	3	2	1	2	2.2
0.07	2	1	3	2	1	2	1.8
0.08	0	1	3	2	1	1	1.4
0.09	0	1	3	1	0	1	1.0

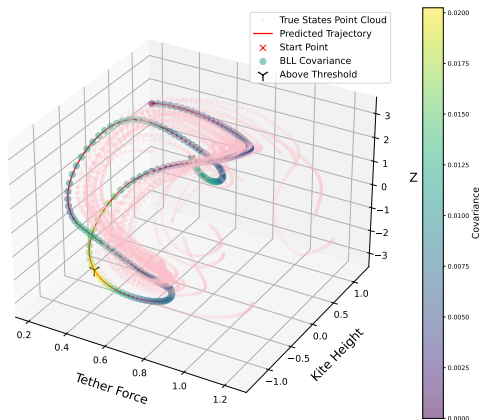
- Zusammenfassung der verschiedenen Schwellwerte die getestet wurden
 - Auf der linken Seite verschiedene Schwellwerte und verschiedene Modelle (M1 bis M6)
 - Die Anzahl der Fälle außerhalb des definierten Schwellwertes für jedes Modell (M1 bis M6) in der Tabelle
 - Mit dem dazugehöriger Mittelwert rechts

Trust Region Schwellwerte

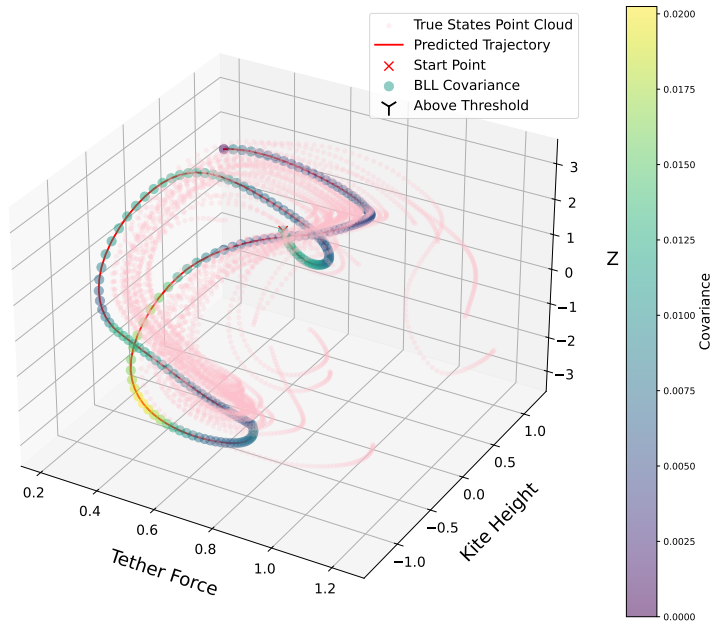
c_{ub}	M1	M2	M3	M4	M5	M6	mean
0.0001	20	20	20	20	20	20	20
0.0005	20	20	20	20	20	20	20
0.001	20	20	20	20	20	20	20
0.005	20	20	20	20	20	20	20
0.0075	20	20	20	19	20	20	19.8
0.01	20	16	18	15	18	18	17.5
0.02	14	8	9	5	10	7	8.8
0.03	7	3	5	2	5	4	4.3
0.04	6	2	5	2	3	4	3.6
0.05	4	2	5	1	3	3	3.0
0.06	3	2	3	2	1	2	2.2
0.07	2	1	3	2	1	2	1.8
0.08	0	1	3	2	1	1	1.4
0.09	0	1	3	1	0	1	1.0

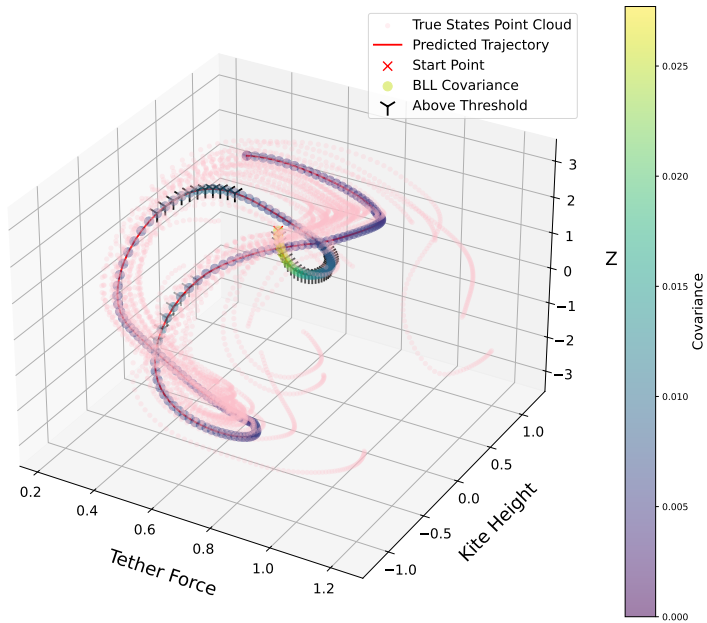
- Zusammenfassung der verschiedenen Schwellwerte die getestet wurden
 - Auf der linken Seite verschiedene Schwellwerte und verschiedene Modelle (M1 bis M6)
 - Die Anzahl der Fälle außerhalb des definierten Schwellwertes für jedes Modell (M1 bis M6) in der Tabelle
 - Mit dem dazugehöriger Mittelwert rechts

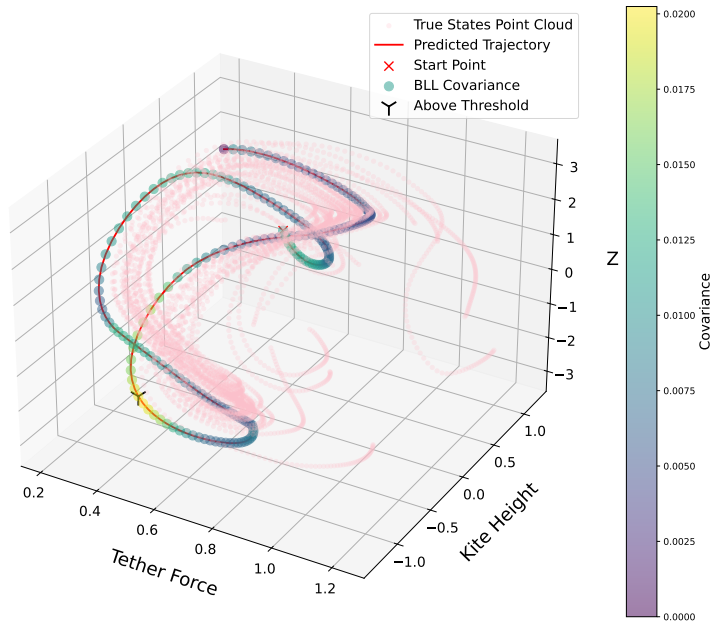
Trust Regions Wahl des Schwellwertes



- Einzelner test case
- Abbildung der Unsicherheit sowie der trust region violations
- Wahl des Schwellwertes?







Fazit & Ausblick

■ Fazit

- Bayesian Last Layer mit Trust Regions in MPC bietet einen leistungsstarken Ansatz für die Regelung in unsicheren Umgebungen
- Möglichkeit, geschickte Trust Regions zu wählen und Unsicherheiten im Kontrollproblem bei der Optimierung zu berücksichtigen

■ Ausblick

- Auswahl des Schwellwertes erscheint willkürlich und ohne Begründung
- Weitere Herangehensweisen zur Wahl des thresh value Werts?
- Auch Unsicherheiten in Zuständen und Messungen in das MPC-Problem miteinbeziehen

Fazit & Ausblick

■ Fazit

- Bayesian Last Layer mit Trust Regions in MPC bietet einen leistungsstarken Ansatz für die Regelung in unsicheren Umgebungen
- Möglichkeit, geschickte Trust Regions zu wählen und Unsicherheiten im Kontrollproblem bei der Optimierung zu berücksichtigen

■ Ausblick

- Auswahl des Schwellwertes erscheint willkürlich und ohne Begründung
- Weitere Herangehensweisen zur Wahl des thresh value Werts?
- Auch Unsicherheiten in Zuständen und Messungen in das MPC-Problem miteinbeziehen

Fazit & Ausblick

■ Fazit

- Bayesian Last Layer mit Trust Regions in MPC bietet einen leistungsstarken Ansatz für die Regelung in unsicheren Umgebungen
- Möglichkeit, geschickte Trust Regions zu wählen und Unsicherheiten im Kontrollproblem bei der Optimierung zu berücksichtigen

■ Ausblick

- Auswahl des Schwellwertes erscheint willkürlich und ohne Begründung
- Weitere Herangehensweisen zur Wahl des thresh value Werts?
- Auch Unsicherheiten in Zuständen und Messungen in das MPC-Problem miteinbeziehen

Fazit & Ausblick

■ Fazit

- Bayesian Last Layer mit Trust Regions in MPC bietet einen leistungsstarken Ansatz für die Regelung in unsicheren Umgebungen
- Möglichkeit, geschickte Trust Regions zu wählen und Unsicherheiten im Kontrollproblem bei der Optimierung zu berücksichtigen

■ Ausblick

- Auswahl des Schwellwertes erscheint willkürlich und ohne Begründung
- Weitere Herangehensweisen zur Wahl des thresh value Werts?
- Auch Unsicherheiten in Zuständen und Messungen in das MPC-Problem miteinbeziehen

Fazit & Ausblick

■ Fazit

- Bayesian Last Layer mit Trust Regions in MPC bietet einen leistungsstarken Ansatz für die Regelung in unsicheren Umgebungen
- Möglichkeit, geschickte Trust Regions zu wählen und Unsicherheiten im Kontrollproblem bei der Optimierung zu berücksichtigen

■ Ausblick

- Auswahl des Schwellwertes erscheint willkürlich und ohne Begründung
- Weitere Herangehensweisen zur Wahl des thresh value Werts?
- Auch Unsicherheiten in Zuständen und Messungen in das MPC-Problem miteinbeziehen

Bibliographie I

- [1] Charles Blundell u. a. „Weight Uncertainty in Neural Networks“. In: (2015). DOI: 10.48550/ARXIV.1505.05424. URL: <https://arxiv.org/abs/1505.05424> (besucht am 19.06.2024).
- [2] Ebenbauer. *Model Predictive Control*. Chair of Intelligent Control Systems RWTH Aachen. 2024. URL: <https://www.ic.rwth-aachen.de/cms/ic/forschung/~qxawy/modellpraediktive-regelung/?lidx=1>.
- [3] Felix Fiedler und Sergio Lucia. „Model predictive control with neural network system model and Bayesian last layer trust regions“. In: Naples, Italy: IEEE, 27. Juni 2022, S. 141–147. ISBN: 9781665495721.
- [4] Boris Houska. *Robustness and Stability Optimization of Open-Loop Controlled Power Generating Kites*. 2007. URL: [:~https://www.researchgate.net/publication/230872992](https://www.researchgate.net/publication/230872992).
- [5] Scott Pendleton u. a. „Perception, Planning, Control, and Coordination for Autonomous Vehicles“. In: *Machines* 5.1 (), S. 6. ISSN: 2075-1702. DOI: 10.3390/machines5010006. URL: <https://www.mdpi.com/2075-1702/5/1/6>.

Model Predictive Control mit Trust Regions

- **Grundlagen:** MPC optimiert die Steuerung eines Systems über einen bestimmten Zeithorizont, basierend auf einem Prädiktionsmodell
- **Integration von Trust Regions:** Unsicherheiten des Modells werden durch Trust Regions in die MPC-Optimierung einbezogen
- **Vorgehensweise:**
 - 1 **Modelltraining:** Ein neuronales Netz wird trainiert, um die Systemdynamik zu approximieren
 - 2 **Kovarianzberechnung:** Die Kovarianzmatrix der letzten Schicht wird berechnet
 - 3 **Trust Region Definition:** Trust Regions werden basierend auf der Kovarianzmatrix definiert
 - 4 **Optimierung:** Die MPC-Optimierung berücksichtigt die Trust Regions, um robuste Steuerungsentscheidungen zu treffen
- **Nutzen:** Die Einbeziehung von Trust Regions verbessert die Zuverlässigkeit und Stabilität der Steuerungsentscheidungen

Bayesian vs Frequentist

- Normale Ansätze: Bieten deterministische Schätzungen und betrachten Modellparameter als feste Werte
- Sie sind oft einfacher zu implementieren und zu verstehen, bieten jedoch keine direkte Quantifizierung der Unsicherheit
- Bayesische Ansätze: Bieten probabilistische Schätzungen und betrachten Modellparameter als Zufallsvariablen mit Wahrscheinlichkeitsverteilungen
- Liefern eine explizite Quantifizierung der Unsicherheit, was zu robusteren und zuverlässigeren Modellen führen kann

Bayesian vs Frequentist

- Normale Ansätze: Bieten deterministische Schätzungen und betrachten Modellparameter als feste Werte
- Sie sind oft einfacher zu implementieren und zu verstehen, bieten jedoch keine direkte Quantifizierung der Unsicherheit
- Bayesische Ansätze: Bieten probabilistische Schätzungen und betrachten Modellparameter als Zufallsvariablen mit Wahrscheinlichkeitsverteilungen
- Liefern eine explizite Quantifizierung der Unsicherheit, was zu robusteren und zuverlässigeren Modellen führen kann

Bayesian vs Frequentist Bsp.

- Wir wollen das Gewicht von Äpfeln vorhersagen
 - Normales Frequentist Modell: Der Apfel wiegt 150 Gramm
 - Bayesisches Modell: Es gibt eine 95% Wahrscheinlichkeit, dass das Gewicht des Apfels zwischen 145 und 155 Gramm liegt, und eine 5% Wahrscheinlichkeit, dass es außerhalb dieses Bereichs liegt

Bayesian vs Frequentist Bsp.

- Wir wollen das Gewicht von Äpfeln vorhersagen
 - Normales Frequentist Modell: Der Apfel wiegt 150 Gramm
 - Bayesisches Modell: Es gibt eine 95% Wahrscheinlichkeit, dass das Gewicht des Apfels zwischen 145 und 155 Gramm liegt, und eine 5% Wahrscheinlichkeit, dass es außerhalb dieses Bereichs liegt

Bayesian Last Layer Details

- **Grundlagen:** BLL basiert auf der Annahme, dass die letzte Schicht eines neuronalen Netzes als linearer Modellteil interpretiert werden kann
- **Parameterunsicherheit:** Die Gewichte der letzten Schicht werden als Zufallsvariablen modelliert, um die Unsicherheit in den Vorhersagen zu erfassen
- **Bayesian Inferenz:** Durch Anwendung der Bayesschen Inferenz werden Verteilungen für die Gewichte der letzten Schicht bestimmt
- **Formel:** Die Verteilung der Gewichte w wird als Normalverteilung $\mathcal{N}(w \mid \mu, \Sigma)$ modelliert, wobei μ der Mittelwert und Σ die Kovarianzmatrix ist
- **Anwendung:** BLL wird genutzt, um die Unsicherheit in den Vorhersagen zu quantifizieren und dadurch fundierte Entscheidungen zu ermöglichen

Bayesian layer berechnen

$$\text{Posterior Probability} \quad \text{Likelihood of Observations} \quad \text{Prior Probability}$$
$$\Pr(\theta|y) = \frac{\Pr(y|\theta)\Pr(\theta)}{\Pr(y)}$$

Normalizing Constant

- wobei y die Daten sind
- $P(\theta|y)$ die Wahrscheinlichkeit ist, y zu beobachten, gegeben die Gewichte θ
- $P(\theta)$ die vorherige Wkeit. der Gewichte
- $P(y)$ die Wahrscheinlichkeit der realen Daten darstellt

Bayesian Neural Network

Anwendung von BNNs in der Robotik

- **Autonome Navigation:** Bayesian Neural Networks (BNNs) helfen bei der Navigation autonomer Roboter durch Unsicherheiten in Umgebungsmodellen zu quantifizieren
- **Manipulation:** BNNs werden verwendet, um Unsicherheiten bei der Greifplanung und Objekterkennung zu berücksichtigen, was zu robusteren Manipulationsstrategien führt
- **Bewegungsplanung:** Integration von BNNs in Bewegungsplanungsalgorithmen, um probabilistische Trajektorien zu erzeugen, die Hindernissen und dynamischen Umgebungen ausweichen
- **Beispielprojekte:** Projekte wie das Google DeepMind's AlphaGo nutzen BNNs zur strategischen Entscheidungsfindung unter Unsicherheit

Vergleich von BLL mit anderen Unsicherheitsmodellen

- **Ensemble Methoden:** Mehrere Modelle werden trainiert und deren Vorhersagen kombiniert, um Unsicherheiten zu schätzen
- **Monte Carlo Dropout:** Dropout wird während der Inferenzphase mehrmals angewendet, um Unsicherheitsvorhersagen zu erhalten
- **Gaussian Processes:** Verwenden eine bayessche Methodik zur Modellierung von Unsicherheiten, sind jedoch oft rechnerisch intensiver als BLL
- **Vorteile von BLL:**
 - Einfache Integration in vorhandene neuronale Netze
 - Rechenaufwand meist geringer als bei vollständigen bayesschen Modellen
 - Flexibel und gut skalierbar für große Datensätze
- **Nachteile von BLL:**
 - Unsicherheitsquantifizierung ist auf die letzte Schicht beschränkt
 - Kann in hochkomplexen Modellen an Genauigkeit verlieren

Vergleich von BLL mit anderen Unsicherheitsmodellen

- **Ensemble Methoden:** Mehrere Modelle werden trainiert und deren Vorhersagen kombiniert, um Unsicherheiten zu schätzen
- **Monte Carlo Dropout:** Dropout wird während der Inferenzphase mehrmals angewendet, um Unsicherheitsvorhersagen zu erhalten
- **Gaussian Processes:** Verwenden eine bayessche Methodik zur Modellierung von Unsicherheiten, sind jedoch oft rechnerisch intensiver als BLL
- **Vorteile von BLL:**
 - Einfache Integration in vorhandene neuronale Netze
 - Rechenaufwand meist geringer als bei vollständigen bayesschen Modellen
 - Flexibel und gut skalierbar für große Datensätze
- **Nachteile von BLL:**
 - Unsicherheitsquantifizierung ist auf die letzte Schicht beschränkt
 - Kann in hochkomplexen Modellen an Genauigkeit verlieren

Optimierung mit Unsicherheiten in MPC

- **Unsicherheitsbewusste Zielfunktion:** Die Zielfunktion der MPC wird modifiziert, um Unsicherheiten zu berücksichtigen und Risiken zu minimieren
- **Robuste Optimierung:** Berücksichtigt schlimmste Szenarien (Worst-Case) innerhalb der Unsicherheitsbereiche, um robuste Lösungen zu finden
- **Stochastische Optimierung:** Integriert Wahrscheinlichkeitsverteilungen der Unsicherheiten direkt in die Optimierung
- **Beispielmethoden:**
 - **Scenario-Based MPC:** Nutzt mehrere mögliche Zukunftsszenarien, um die Optimierung durchzuführen
 - **Chance-Constrained MPC:** Beschränkt die Wahrscheinlichkeit von Verletzungen bestimmter Bedingungen auf ein akzeptables Niveau
- **Nutzen:** Verbesserte Stabilität und Leistung des Kontrollsystems unter Unsicherheit Pendleton u. a. o. D.

Optimierung mit Unsicherheiten in MPC

- **Unsicherheitsbewusste Zielfunktion:** Die Zielfunktion der MPC wird modifiziert, um Unsicherheiten zu berücksichtigen und Risiken zu minimieren
- **Robuste Optimierung:** Berücksichtigt schlimmste Szenarien (Worst-Case) innerhalb der Unsicherheitsbereiche, um robuste Lösungen zu finden
- **Stochastische Optimierung:** Integriert Wahrscheinlichkeitsverteilungen der Unsicherheiten direkt in die Optimierung
- **Beispielmethoden:**
 - **Scenario-Based MPC:** Nutzt mehrere mögliche Zukunftsszenarien, um die Optimierung durchzuführen
 - **Chance-Constrained MPC:** Beschränkt die Wahrscheinlichkeit von Verletzungen bestimmter Bedingungen auf ein akzeptables Niveau
- **Nutzen:** Verbesserte Stabilität und Leistung des Kontrollsystems unter Unsicherheit Pendleton u. a. o. D.