

Methoden

Daniel Bergmann* und Knut Graichen

Gaußprozessregression zur Modellierung zeitvarianter Systeme

Gaussian process regression for modeling of time-varying systems

<https://doi.org/10.1515/auto-2019-0015>

Empfangen 7. Februar 2019; angenommen 29. Mai 2019

Zusammenfassung: Datenbasierte Modellbildungsverfahren haben auch für komplexe und hochdimensionale Systeme an Bedeutung gewonnen. Sie bieten den Vorteil einer deutlich verkürzten Entwicklungszeit im Vergleich zur physikalischen Modellierung. Der Nachteil vieler herkömmlicher Verfahren ist allerdings, dass selbst einfaches Wissen, wie etwa Extrapolationseigenschaften, nicht in der Modellgenerierung berücksichtigt werden können. Deshalb soll in diesem Beitrag eine Methodik vorgestellt werden, wie auf Basis der Gaußprozessregression solches Wissen auch für datenbasierte Verfahren genutzt werden kann. Desweiteren ist es in vielen physikalischen Systemen notwendig, dass die Modelle während der Laufzeit an individuell unterschiedliche Systeme angepasst werden können. Dazu wird ein Online-Adaptionsverfahren vorgestellt, welches die Unsicherheitsinformation des Gaußprozesses nutzt, um Änderungen im System zu detektieren und das Modell anzupassen.

Schlagwörter: Gaußprozessregression, datenbasierte Modellbildungsverfahren, zeitvariante Systemidentifikation

Abstract: In recent times, data based methods have gained importance especially for complex and high dimensional systems due to a short development time compared to physical modeling. The disadvantage is however, that even simple knowledge such as extrapolation behaviour can in general not be considered during model generation. In this article, a modeling scheme based on Gaussian process regression is presented, which is able to incorporate such knowledge in the model. Moreover, in many systems an adaptation of the model to the individual system is necessary. To this end, an online adaptation scheme is

presented, which uses the uncertainty information of the Gaussian process to detect changes in the model and incorporate them.

Keywords: Gaussian process regression, data based modeling, time-varying system identification

1 Einleitung

In vielen physikalischen Systemen ist eine wissensbasierte Modellbildung sehr aufwändig. Mit steigender Komplexität der Systeme ist ein physikalisches Modell außerdem teilweise nur bedingt für regelungstechnische Anwendungen geeignet, da auch die Komplexität der Regelungsstruktur mit der des Systemmodells ansteigt. Um dieser Problematik entgegenzuwirken, können datenbasierte Modellbildungsansätze herangezogen werden. Diese bieten den Vorteil, dass deren Struktur oftmals unabhängig von dem zu modellierenden System ist. Zu nennen sind hier zum einen neuronale Netze, wie beispielsweise Netze mit radialen Basisfunktionen [12], mehrschichtige Netze [3] oder auch Netze mit lokal linearen Modellen (LoLi-MoT) [7]. Zum anderen existieren diverse Ansatzfunktionen, wie etwa Polynomapproximationen, mit deren Hilfe ein System nachgebildet werden kann [14]. Eine weitere Möglichkeit zur Modellierung beliebiger Nichtlinearitäten bietet die Gaußprozessregression [10]. Da diese parameterfrei ist, ergeben sich einige Vorteile im Vergleich zu herkömmlichen Methoden, die im Folgenden genutzt werden sollen.

Alle genannten datenbasierten Modellbildungsverfahren sind grundsätzlich in der Lage, beliebige statische Nichtlinearitäten $f(\cdot)$ der Form

$$y = f(x_1, \dots, x_D) \quad (1)$$

mit Ausgang y und Eingängen x_i für $i \in \{1, \dots, D\}$ abzubilden. Neben der möglichst guten Nachbildung der Daten existieren jedoch noch andere relevante Faktoren, die die

*Korrespondenzautor: Daniel Bergmann, Institut für Mess-, Regel- und Mikrotechnik, Universität Ulm, Albert-Einstein-Allee 41, D-89081 Ulm, Deutschland, E-Mail: daniel.bergmann@uni-ulm.de
Knut Graichen, Lehrstuhl für Regelungstechnik, Friedrich-Alexander-Universität Erlangen-Nürnberg, Cauerstraße 7, D-91058 Erlangen, Deutschland, E-Mail: knut.graichen@fau.de

Modelle mehr oder weniger geeignet für regelungstechnische Anwendungen machen. Zu nennen ist hier beispielsweise die Extrapolationsfähigkeit. Hier unterscheiden sich die unterschiedlichen Verfahren deutlich. Während neuronale Netze und auch Polynomansätze in Randbereichen im Allgemeinen kein definiertes Verhalten ausweisen, extrapolieren lokal lineare Modelle linear. Die Gaußprozessregression bietet hier die Möglichkeit, ein vordefiniertes Extrapolationsverhalten in das Modell einzuprägen, wie bereits in [1] am Beispiel eines Schwerlast-Dieselmotors gezeigt wurde.

Neben der Abbildung eines beliebigen Datensatzes ist in physikalischen Systemen oftmals eine Anpassung an das individuelle System zur Laufzeit gewünscht. Dieses kann sich beispielsweise auf Grund von Alterungseffekten zeitlich ändern. Um solche Änderungen systematisch zu erfassen und zu erlernen, wird eine stochastische Information über die Modellgüte benötigt. Dazu genügt es nicht, wie in [4], [9] und [6], den Gaußprozess online hinsichtlich seiner Speichereffizienz und Hyperparameter zu verbessern. Hier zeigt sich der große Vorteil der Gaußprozessregression, deren Definition bereits eine Unsicherheitsinformation an beliebigen Punkten liefern kann und damit die Anpassung der Modelle, wie in [2] an numerischen Beispielen gezeigt, möglich macht.

In diesem Beitrag soll aufgezeigt werden, wie offline ein Nominalmodell erstellt wird und dieses online an ein zeitvariantes System angepasst werden kann. Dazu werden in Abschnitt 2 zunächst einige Grundlagen der Gaußprozessregression zusammengefasst. Daraufhin wird in Abschnitt 3 die Methodik für ein offline generiertes Modell vorgestellt. Dieses beinhaltet ein Nominalmodell mit sinnvollem Extrapolationsverhalten und ein Detailmodell für eine erhöhte Genauigkeit im Arbeitsbereich. Danach wird die Online-Adaption des Modells in Abschnitt 4 zusammengefasst. Im Anschluss wird in Abschnitt 5 anhand von numerischen Beispielen die Methodik evaluiert, bevor schließlich in Abschnitt 6 eine Zusammenfassung und ein Ausblick gegeben werden.

2 Gaußprozessregression

Gaußprozessregression ist eine nichtparametrische Modellierungsmethode, die über ihren Mittelwert $m(x)$ und ihre Kovarianzfunktion $k(x_m, x_n)$ beschrieben werden kann [10]. Die Kovarianzfunktion gibt hierbei den Zusammenhang zwischen beliebigen Punkten (x_m, x_n) im Eingangsraum an. Die gebräuchlichste Kovarianzfunktion ist

über eine *Squared-Exponential*-Funktion

$$k_{SE}(x_m, x_n) = \sigma_F^2 e^{-\frac{\|x_m - x_n\|_2^2}{2l^2}} + \sigma_M^2 \delta_{m,n} \quad (2)$$

definiert. Hierbei bezeichnet $\|\cdot\|_2$ die Euklidische Norm, σ_F^2 die Signalvarianz, σ_M^2 die Varianz des Messrauschens und $\delta_{m,n}$ das Kronecker-Delta. Der charakteristische Parameter l stellt ein Maß für die Glattheit der resultierenden Funktion dar. Die angegebene Kovarianzfunktion (2) ist durch ein konstantes Messrauschen σ_M charakterisiert. Die allgemeinere Beschreibung

$$k(x_m, x_n) = \sigma_F^2 e^{-\frac{\|x_m - x_n\|_2^2}{2l^2}} + \sum_{i=1}^L r_i \delta_{i,i} \quad (3)$$

erlaubt es, für alle Punkte im Datensatz der Länge L eine individuelle Unsicherheit $r_i \geq 0$ anzugeben [2, 5].

Da die Gaußprozessregression eine parameterfreie Methode ist, werden Funktionswerte direkt auf Basis eines Datensatzes (\mathbf{X}, \mathbf{y}) , bestehend aus Eingangsdaten \mathbf{X} und Ausgangsdaten \mathbf{y} , bestimmt. Zur Auswertung des Ausgangswertes y des Gaußprozesses wird die Verbunddichte $\mathcal{N}(\cdot)$ zwischen dem Datensatz (\mathbf{X}, \mathbf{y}) und dem Eingang \mathbf{x}

$$\begin{bmatrix} \mathbf{y} \\ \mathbf{x} \end{bmatrix} \sim \mathcal{N} \left(m(\mathbf{x}), \begin{bmatrix} k(\mathbf{X}, \mathbf{X}) & k(\mathbf{X}, \mathbf{x}) \\ k(\mathbf{x}, \mathbf{X}) & k(\mathbf{x}, \mathbf{x}) \end{bmatrix} \right) \quad (4)$$

gebildet. Die Mittelwertfunktion $m(\mathbf{x})$ wird dabei oft zu Null gewählt bzw. mithilfe von linearer Regression eine polynomiale Funktion identifiziert. Um den Ausgangswert y an einem beliebigen Eingang \mathbf{x} zu berechnen, können nun der Erwartungswert und die Varianz der Verbunddichte mit

$$E(\mathbf{x}) = m(\mathbf{x}) + k(\mathbf{x}, \mathbf{X}) (k(\mathbf{X}, \mathbf{X}))^{-1} (\mathbf{y} - m(\mathbf{X})), \quad (5)$$

$$\text{Var}(\mathbf{x}) = k(\mathbf{x}, \mathbf{x}) - k(\mathbf{x}, \mathbf{X}) (k(\mathbf{X}, \mathbf{X}))^{-1} k(\mathbf{X}, \mathbf{x}) \quad (6)$$

angegeben werden.

3 Offline-Modellerstellung

In vielen physikalischen Systemen ist Expertenwissen über die Extrapolationseigenschaften vorhanden, welches aber für eine datenbasierte Modellgenerierung kaum verwendet werden kann. Gaußprozesse bieten hier die Möglichkeit, das Wissen über solche Eigenschaften systematisch in der Modellgenerierung zu berücksichtigen, wie es in [1] am Beispiel eines Schwerlast-Dieselmotors gezeigt wurde. Dabei wird ein Nominalmodell erzeugt, welches ein global sinnvolles Extrapolationsverhalten aufweist. Das darauf aufbauende Detailmodell bildet lokale Abweichungen vom Nominalmodell nach, um die Genauigkeit zu erhöhen.

3.1 Generierung eines Nominalmodells

Im ersten Schritt der Modellerstellung wird ein Nominalmodell generiert, welches das geforderte Extrapolationsverhalten aufweist und bereits den Messdatensatz des Systems berücksichtigt. Um ein Gaußprozessmodell mit global sinnvollem Extrapolationsverhalten aufzustellen, müssen Stützstellen im gesamten Eingangsraum mit Dimension D mit normierten Eingängen $\mathbf{x} \in [0, 1]^D$ gleichmäßig platziert werden. Die Anzahl der Stützstellen pro Eingangsdimension wurde hierbei zu 3 gewählt, was einen guten Kompromiss aus Flexibilität und Komplexität darstellt. Dies führt zu einer Gitterstruktur der Stützstellen

$$\mathbf{X}_N = \begin{bmatrix} 0 & \dots & 0 & 0 \\ 0 & \dots & 0 & 0.5 \\ 0 & \dots & 0 & 1 \\ 0 & \dots & 0.5 & 0 \\ 0 & \dots & 0.5 & 0.5 \\ \vdots & & \vdots & \vdots \\ 1 & \dots & 1 & 1 \end{bmatrix} \in \{0, 0.5, 1\}^{3^D \times D}. \quad (7)$$

Da die Stützstellen keinen Messdatenpunkten entsprechen, kann für diese keine Unsicherheit angegeben werden. Daher wird die Messunsicherheit der Stützstellen zu Null gewählt. Da des Weiteren kein bekannter Mittelwert vorliegt, wird dieser ebenfalls zu Null gewählt. Damit ergibt sich die Ausgangsgleichung des nominalen Gaußprozesses zu

$$\mathbf{y}_N(\mathbf{x}) = k_N(\mathbf{x}, \mathbf{X}_N) (k_N(\mathbf{X}_N, \mathbf{X}_N))^{-1} \mathbf{y}_N, \quad (8)$$

mit den Stützstellen \mathbf{X}_N , Stützstellenausgangswerten \mathbf{y}_N und der Kovarianzfunktion

$$k_N(\mathbf{x}_m, \mathbf{x}_n) = e^{-\frac{\|\mathbf{x}_m - \mathbf{x}_n\|_2^2}{2l_N^2}}. \quad (9)$$

Mithilfe der Stützstellenausgangswerte \mathbf{y}_N als freie Parameter soll nun das Expertenwissen in die Modellbildung integriert werden. Die einfachste Form von Systemwissen kann über Monotonieigenschaften bezüglich einzelner Stellgrößen angegeben werden. Für ein allgemeines nicht-lineares System der Form (1) können die Eingänge mit monoton steigendem Einfluss auf die Ausgangsgröße in der Menge \mathcal{M}_I , für die

$$f(\mathbf{x} + \mathbf{e}_d \delta) > f(\mathbf{x}) \quad \forall d \in \mathcal{M}_I, \mathbf{x} \in \mathbb{R}^D, \delta > 0 \quad (10)$$

mit Einheitsvektor \mathbf{e}_d gilt, zusammengefasst werden. Eingänge mit monoton fallendem Einfluss werden in der Menge \mathcal{M}_D mit

$$f(\mathbf{x} + \mathbf{e}_d \delta) < f(\mathbf{x}) \quad \forall d \in \mathcal{M}_D, \mathbf{x} \in \mathbb{R}^D, \delta > 0 \quad (11)$$

zusammengefasst. Im Gegensatz zu [11] wird diese Monotonieinformation nicht über lokale virtuelle Ableitungsinformationen in das Modell integriert, sondern die Gitterstruktur dazu genutzt, Bedingungen an die Stützstellenausgangswerte zu stellen, um die geforderten Monotonien einzuhalten. So kann für monoton steigende Eingänge d_I aus der Menge \mathcal{M}_I die Ungleichungsbedingung

$$\mathbf{A}_{M,I} \mathbf{y}_N \leq 0 \quad \mathbf{A}_{M,I} \in \{-1, 0, 1\}^{3^D \times 3^D} \quad (12)$$

mit

$$\mathbf{A}_{M,I}(i, j) = \begin{cases} 1 & j = i \quad \wedge \quad \mathbf{X}_N(i, d_I) \neq 1 \\ -1 & j = i + 3^{D-d_I} \quad \wedge \quad \mathbf{X}_N(i, d_I) \neq 1 \\ 0 & \text{sonst} \end{cases}$$

angegeben werden. Analog muss für streng monoton fallende Eingänge d_D aus der Menge \mathcal{M}_D die Ungleichung

$$\mathbf{A}_{M,D} \mathbf{y}_N \leq 0 \quad \mathbf{A}_{M,D} \in \{-1, 0, 1\}^{3^D \times 3^D} \quad (13)$$

mit

$$\mathbf{A}_{M,D}(i, j) = \begin{cases} -1 & j = i \quad \wedge \quad \mathbf{X}_N(i, d_D) \neq 1 \\ 1 & j = i + 3^{D-d_D} \quad \wedge \quad \mathbf{X}_N(i, d_D) \neq 1 \\ 0 & \text{sonst} \end{cases}$$

erfüllt sein. Nun kann das Optimierungsproblem für \mathbf{y}_N , welches den quadratischen Fehler zwischen Modell und Messwerten $(\mathbf{X}_D, \mathbf{y}_D)$ unter Berücksichtigung der Monotonieigenschaften minimiert, mit

$$\begin{aligned} \min_{\mathbf{y}_N} \sum_{i=1}^L (\mathbf{y}_D(i) - k_N(\mathbf{X}_D(i), \mathbf{X}_N) k_N(\mathbf{X}_N, \mathbf{X}_N)^{-1} \mathbf{y}_N)^2 \\ \text{u. B. v. } \mathbf{A}_{M,I} \mathbf{y}_N \leq 0 \quad \forall d \in \mathcal{M}_I \\ \mathbf{A}_{M,D} \mathbf{y}_N \leq 0 \quad \forall d \in \mathcal{M}_D. \end{aligned} \quad (14)$$

als quadratisches Programm angegeben werden. Der *Length-Scale*-Parameter l_N der Kovarianzfunktion (9) ist nicht Teil des Optimierungsproblems, sondern kann aufgrund der festen Gitterstruktur im Eingang heuristisch gewählt werden. Als gute Wahl hat sich $l_N = 0.5$ identisch zum Abstand der Gitterpunkte herausgestellt. Die so gefundenen Stützstellenausgangswerte \mathbf{y}_N berücksichtigen damit das Expertenwissen und bilden den Messdatensatz bereits bestmöglich nach.

3.2 Generierung eines Detailmodells

Um die Genauigkeit des Nominalmodells zu erhöhen, wird nun ein Detailmodell gebildet, welches das Nominalmodell (8) als Mittelwert $m(\mathbf{x})$ nutzt und die Datenpunkte

$(\mathbf{X}_D, \mathbf{y}_D)$ eines Messdatensatzes des Systems abbildet. Die Ausgangsgleichung für Erwartungswert und Varianz ergeben sich damit zu

$$\mathbf{y}_D(\mathbf{x}) = \mathbf{y}_N(\mathbf{x}) + \mathbf{k}_D(\mathbf{x}, \mathbf{X}_D) \mathbf{K}_D^{-1} (\mathbf{y}_D - \mathbf{y}_N(\mathbf{X}_D)), \quad (15)$$

$$\text{Var}(\mathbf{x}) = \mathbf{k}_D(\mathbf{x}, \mathbf{x}) - \mathbf{k}_D(\mathbf{x}, \mathbf{X}_D) \mathbf{K}_D^{-1} \mathbf{k}_D(\mathbf{X}_D, \mathbf{x}) \quad (16)$$

mit Kovarianzfunktion

$$\mathbf{k}_D(\mathbf{x}_m, \mathbf{x}_n) = \sigma_F^2 e^{-\frac{\|\mathbf{x}_m - \mathbf{x}_n\|_2^2}{2l_D^2}} + \sum_{i=1}^N r_i \delta_{i,i}. \quad (17)$$

und Kovarianzmatrix $\mathbf{K}_D = \mathbf{k}_D(\mathbf{X}_D, \mathbf{X}_D)$ für $r_i = \sigma_M$. Die Signalvarianz σ_F und Messvarianz σ_M werden dabei als bekannt vorausgesetzt. Der *Length-Scale*-Parameter l_D wird ebenfalls fest gewählt. Da lokale Abweichungen vom Nominalmodell modelliert werden sollen, wird dieser kleiner als der *Length-Scale* des Nominalmodells $l_D < l_N$ gewählt. Im Detailmodell können die Monotonieeigenschaften nicht systematisch berücksichtigt werden. Diese bleiben dem Nominalmodell vorbehalten. Das so erzeugte Detailmodell erhält das Extrapolationsverhalten des Nominalmodells und modelliert lokale Abweichungen von diesem.

3.3 Reduzierung der Datenpunkte

Das Detailmodell wurde mithilfe des gesamten Datensatzes des Systems aufgestellt. Da dieser Datensatz beliebig groß sein kann, ist es für eine technische Anwendung notwendig, diesen auf eine bestimmte Anzahl an Datenpunkten zu reduzieren. Ziel ist es dabei, den Mittelwert der Funktion weiterhin möglichst gut abzubilden. Als Maß für den Einfluss einer einzelnen Stützstelle \mathbf{x}_i wird deshalb die Distanz zwischen dem Modellerwartungswert auf Basis des Gesamtdatensatzes \mathbf{X}_D im Vergleich zum reduzierten Datensatz $\mathbf{X}_D \setminus \mathbf{x}_i$ betrachtet. Für eine einfachere Notation wird hierfür der Gaußprozess mit Datensatz \mathbf{X}_D als

$$\mathbb{E}(\mathcal{GP}_{\mathbf{X}_D}(\mathbf{x})) = \mathbf{y}_N(\mathbf{x}) + \mathbf{k}_D(\mathbf{x}, \mathbf{X}_D) \mathbf{K}_D^{-1} (\mathbf{y}_D - \mathbf{y}_N(\mathbf{X}_D)), \quad (18)$$

$$\text{Var}(\mathcal{GP}_{\mathbf{X}_D}(\mathbf{x})) = \mathbf{k}_D(\mathbf{x}, \mathbf{x}) - \mathbf{k}_D(\mathbf{x}, \mathbf{X}_D) \mathbf{K}_D^{-1} \mathbf{k}_D(\mathbf{X}_D, \mathbf{x}) \quad (19)$$

geschrieben. Nun können iterativ die Punkte \mathbf{x}_i mit dem geringsten Einfluss

$$\min_i (\mathbb{E}(\mathcal{GP}_{\mathbf{X}_D}(\mathbf{x}_i)) - \mathbb{E}(\mathcal{GP}_{\mathbf{X}_D \setminus \mathbf{x}_i}(\mathbf{x}_i))) \quad (20)$$

aus dem Datensatz entfernt werden, bis die gewünschte Anzahl an Punkten erreicht ist. Damit können der Speicherbedarf und der Rechenaufwand reduziert werden und

gleichzeitig eine möglichst hohe Güte des Modells erreicht werden. Bei dem beschriebenen Verfahren handelt es sich um eine Rückwärtseliminierung, deren Resultat nicht zwingend zur optimalen Lösung führt. Diese Vorgehensweise setzt voraus, dass der Gaußprozess auf Basis des Gesamtdatensatzes berechnet werden kann. Dafür muss die Matrixinversion der positiv definiten Kovarianzmatrix berechnet werden, welche im Rechenaufwand mit $\mathcal{O}(N^3)$ für die Datensatzgröße N skaliert [13].

Die gesamte Offline-Modellerstellung wird in Algorithmus 1 zusammengefasst. Zunächst werden auf Basis des bekannten Extrapolationsverhaltens lineare Beschränkungen für die Stützstellen des Nominalmodells formuliert, welches daraufhin mithilfe der Messdatenpunkte hinsichtlich des mittleren quadratischen Fehlers optimiert wird. Nachdem das Detailmodell mit dem gesamten Messdatensatz aufgestellt wurde, kann der Speicherbedarf durch Entfernen irrelevanter Datenpunkte reduziert werden.

Algorithmus 1 Offline-Modellerstellung.

- 1: Finde Eingänge mit streng monotonem Einfluss auf den Ausgang (10), (11) und generiere lineare Beschränkungen mit (12) und (13) .
 - 2: Generiere Nominalmodell mit (14).
 - 3: Generiere Detailmodell (15), (16) aus Datensatz $(\mathbf{X}_D, \mathbf{y}_D)$.
 - 4: Reduziere die Datenpunkte mit (20).
-

4 Online-Modellanpassung

Nun soll auf Basis des offline generierten Modells eine Online-Anpassung an ein zeitveränderliches System wie in [2] durchgeführt werden. Dazu wird die Unsicherheitsinformation in Form eines Vertrauensbereichs des Modells genutzt, um zu entscheiden, ob eine neue Messung das bestehende Modell ergänzt oder eine Änderung stattgefunden hat. Dafür werden für den aktuell am System gemessenen Datenpunkt $(\mathbf{x}_M, \mathbf{y}_M)$ mit Messrauschvarianz σ_M^2 der Modellwert und dessen Varianz

$$\mathbf{y}_p = \mathbf{y}_D(\mathbf{x}_M), \quad \sigma_p^2 = \text{Var}(\mathbf{x}_M) \quad (21)$$

berechnet. Wenn die Messung innerhalb des Vertrauensbereichs

$$|\mathbf{y}_M - \mathbf{y}_p| \leq z\sigma_p \quad (22)$$

mit $z = 2.575$ liegt, was einem 99 % Konfidenzintervall für eine normalverteilte Zufallsvariable entspricht, wird das Modell erweitert. Wenn

$$|y_M - y_p| > z\sigma_p \quad (23)$$

gilt, wird von einer Änderung des bestehenden Modells ausgegangen. Die Hyperparameter des Modells bleiben von der Anpassung unberührt, um die Robustheit der Methodik zu gewährleisten. Dadurch ist die Modellanpassung aber gleichzeitig auf nicht stark strukturelle Änderungen eingeschränkt.

4.1 Erweiterung des bestehenden Modells

Für den Fall, dass die neue Messung innerhalb des Konfidenzbereichs des bestehenden Modells liegt, kann der neue Datenpunkt zum Modell hinzugefügt werden. Um Robustheit gegen Fehlmessungen zu gewährleisten, wird nicht die Messung selbst zum Modell hinzugefügt, sondern der mit den Unsicherheiten gewichtete Datenpunkt aus Messung und Modell

$$y_{\text{add}} = \frac{\sigma_p^2 \cdot y_M + \sigma_M^2 \cdot y_p}{\sigma_p^2 + \sigma_M^2}, \quad (24)$$

$$r_{\text{add}}^2 = \frac{\sigma_M^2 \cdot \sigma_p^2}{\sigma_M^2 + \sigma_p^2} \quad (25)$$

gebildet und der Datenpunkt (x_M, y_{add}) mit Standardabweichung r_{add} zum Modell hinzugefügt. Je kleiner die Messunsicherheit gegenüber der Modellunsicherheit ist, desto mehr wird der neuen Messung vertraut. Für den Fall, dass die Modellunsicherheit gering ist, wird der neue Datenpunkt in Richtung des Modellwertes korrigiert, um Messausreißer zu kompensieren. Beispielhaft ist dieses Vorgehen in Abbildung 1 dargestellt. Hier liegt der neue Messpunkt innerhalb des Konfidenzbereichs und wird dem Modell hinzugefügt.

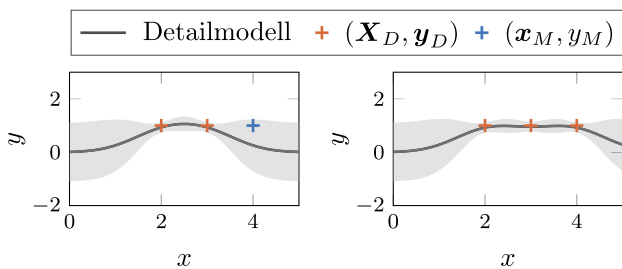


Abb. 1: Erweiterung des Modells durch die Messung (x_M, y_M) .

4.2 Änderung des bestehenden Modells

Im Falle einer detektierten Modelländerung muss entschieden werden, welche Datenpunkte des bestehenden Modells dazu führen, dass die Messung außerhalb des Konfidenzbereichs liegt. Dazu muss die Wahrscheinlichkeitsdichte

$$p_i = \frac{1}{\sqrt{2\pi\sigma_i^2}} e^{\left(-\frac{(y_i - y_M)^2}{2\sigma_i^2}\right)} \quad (26)$$

mit

$$y_i = E(\mathcal{GP}_{\mathbf{X}_D \setminus \mathbf{x}_i}(\mathbf{x}_i)), \quad (27)$$

$$\sigma_i^2 = \text{Var}(\mathcal{GP}_{\mathbf{X}_D \setminus \mathbf{x}_i}(\mathbf{x}_i)) \quad (28)$$

für jeden Punkt berechnet werden. Nun wird derjenige Datenpunkt \mathbf{x}_i aus dem Modell entfernt, der die Wahrscheinlichkeit p_i maximiert, dass die neue Messung innerhalb des Modells liegt. Dieser Schritt wird solange wiederholt, bis die neue Messung im Konfidenzbereich des Modells liegt. In Abbildung 2 ist ein Beispiel dieses Vorgehens dargestellt. Hier muss ein Datenpunkt aus dem Modell entfernt werden, damit die neue Messung innerhalb des Modells liegt.

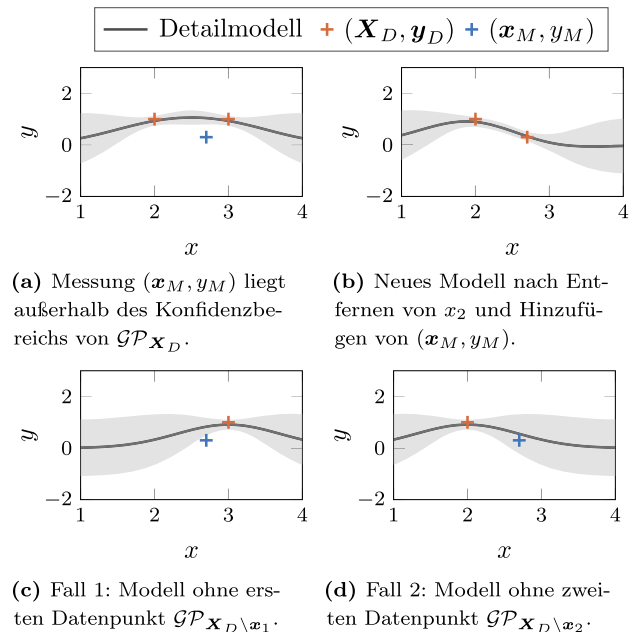


Abb. 2: Änderung des bestehenden Modells durch den neuen Messwert (x_M, y_M) .

4.3 Anpassung der Unsicherheit

Um auch langsame Änderungen des Modells gut nachbilden zu können, ist die Unterscheidung nach Vertrauens-

bereichen nicht ausreichend. Wenn die Änderung des Modells innerhalb des Vertrauensbereichs stattfindet, werden neue Datenpunkte mit Gleichung (24) hinzugefügt. Dabei wird deren Unsicherheit mit (25) künstlich erhöht, da zunächst von Messausreißern ausgegangen wird. Bestätigt eine neue Messung in deren Umgebung allerdings diesen Datenpunkt, ist es notwendig, auch die Unsicherheit der Datenpunkte zu korrigieren. Dazu wird der Gradient des Modellausgangswertes nach der Unsicherheit der einzelnen Datenpunkte r_i

$$\left. \frac{dy}{dr_i} \right|_{r_i=\bar{r}_i} = \left. \frac{dy}{d\Delta r_i} \right|_{\Delta r_i=0} \quad (29)$$

berechnet. Hierzu wird die Unsicherheit in einen konstanten und einen variablen Anteil $r_i = \bar{r}_i + \Delta r_i$ mit der derzeitigen Unsicherheit \bar{r}_i aufgeteilt. In der Modellgleichung (15) ist r_i lediglich ein Teil der Kovarianzmatrix \mathbf{K}_D , welche ebenso in zwei Teile

$$\mathbf{K}_D = \bar{\mathbf{K}}_D + \mathbf{e}_i \Delta r_i \mathbf{e}_i^T \quad (30)$$

mit dem Einheitsvektor \mathbf{e}_i aufgeteilt wird. Für die Inverse von \mathbf{K}_D wird die Woodbury-Matrix-Identität [15]

$$\begin{aligned} \mathbf{K}_D^{-1} &= (\bar{\mathbf{K}}_D + \mathbf{e}_i \Delta r_i \mathbf{e}_i^T)^{-1} \\ &= \bar{\mathbf{K}}_D^{-1} - \bar{\mathbf{K}}_D^{-1} \mathbf{e}_i \left(\frac{1}{\Delta r_i} + \mathbf{e}_i^T \bar{\mathbf{K}}_D^{-1} \mathbf{e}_i \right)^{-1} \mathbf{e}_i^T \bar{\mathbf{K}}_D^{-1} \\ &= \bar{\mathbf{K}}_D^{-1} - \frac{\Delta r_i}{1 + \Delta r_i \bar{\mathbf{K}}_D^{-1}(i, i)} \bar{\mathbf{K}}_D^{-1} \mathbf{e}_i \mathbf{e}_i^T \bar{\mathbf{K}}_D^{-1} \end{aligned} \quad (31)$$

genutzt. Der Gradient der Inverse kann damit als

$$\frac{d\mathbf{K}_D^{-1}}{d\Delta r_i} = - \frac{1}{(1 + \Delta r_i \bar{\mathbf{K}}_D^{-1}(i, i))^2} \bar{\mathbf{K}}_D^{-1} \mathbf{e}_i \mathbf{e}_i^T \bar{\mathbf{K}}_D^{-1} \quad (32)$$

geschrieben werden. Für $\Delta r_i = 0$ vereinfacht sich die Gleichung zu

$$\left. \frac{d\mathbf{K}_D^{-1}}{d\Delta r_i} \right|_{\Delta r_i=0} = -\bar{\mathbf{K}}_D^{-1} \mathbf{e}_i \mathbf{e}_i^T \bar{\mathbf{K}}_D^{-1}. \quad (33)$$

Der Gradient des Modellausgangswertes nach r_i kann durch Einsetzen von (33) in (15)

$$\begin{aligned} \frac{dy_D}{d\Delta r_i} &= \frac{d}{d\Delta r_i} \left(k_D(\mathbf{x}, \mathbf{X}_D) (\bar{\mathbf{K}}_D + \mathbf{e}_i \Delta r_i \mathbf{e}_i^T)^{-1} \bar{\mathbf{y}} \right), \\ \left. \frac{dy_D}{d\Delta r_i} \right|_{\Delta r_i=0} &= - \left(k_D(\mathbf{x}, \mathbf{X}_D) \bar{\mathbf{K}}_D^{-1} \mathbf{e}_i \mathbf{e}_i^T \bar{\mathbf{K}}_D^{-1} \bar{\mathbf{y}} \right) \end{aligned} \quad (34)$$

mit $\bar{\mathbf{y}} = \mathbf{y}_D - \mathbf{y}_N(\mathbf{X}_D)$ ausgewertet werden. Die Anpassung der Unsicherheit wird auf die umliegenden Datenpunkte

eingeschränkt. Als Schwellenwert für die Adaption wird der Euklidische Abstand zum Messwert auf

$$\|\mathbf{x}_M - \mathbf{x}_i\|_2 \leq \frac{1}{2} l_D \quad (35)$$

gewählt. Nun wird die Unsicherheit durch

$$r_i = \max \left(0, r_i + \lambda k_D(\mathbf{x}_M, \mathbf{x}_i) \frac{dy_D(\mathbf{x}_M)}{d\Delta r_i} \Big|_{\Delta r_i=0} (y_M - y_p) \right) \quad (36)$$

angepasst. Dabei wird die Adaption durch die Kovarianz skaliert. Dies führt dazu, dass nah gelegene Datenpunkte stärker angepasst werden als weiter entfernte. Außerdem führt ein höherer Prädiktionsfehler ($y_M - y_p$) zu einer stärkeren Anpassung. Der Skalierungsparameter λ stellt nun einen Kompromiss aus Robustheit und schneller Adaption dar. Für verschiedene numerische Beispiele hat sich $\lambda = 0.01$ als gute Wahl herausgestellt. Beispielhaft ist dieses Vorgehen in Abbildung 3 dargestellt. Hier kann die Unsicherheit eines Datenpunktes durch eine neue Messung deutlich reduziert werden.

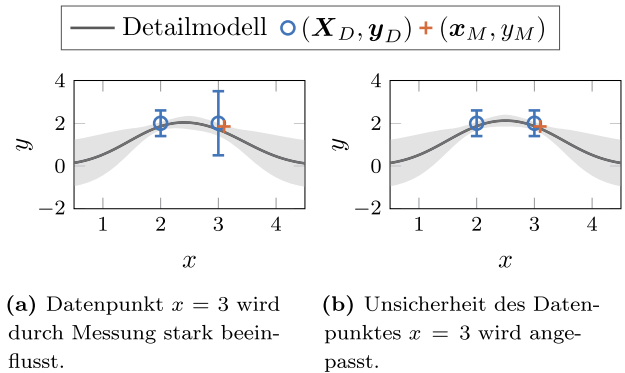


Abb. 3: Anpassung der Unsicherheit durch neue Messung.

4.4 Rückanpassung Nominalmodell

Um Änderungen im Detailmodell auch auf das Nominalmodell übertragen zu können, muss eine Rückanpassung vorgenommen werden. Dazu werden Gitterpunkte, die von Messdatenpunkten stark beeinflusst werden, auf ihren vorhergesagten Wert adaptiert. Die Unterscheidung wird auf Basis der Varianz des Detailmodells getroffen. Wenn diese unterhalb der Schwelle

$$\text{Var}(x) < c\sigma_F^2 \quad (37)$$

liegt, kann die Gitterstelle angepasst werden. Der Faktor c ist dabei ein Kompromiss aus Robustheit und Flexibilität der Anpassung. An numerischen Beispielen hat sich

ein Faktor $c = \frac{1}{4}$ als gute Wahl herausgestellt. Nun werden alle zu adaptierenden Gitterstellen im Vektor \mathbf{I} zusammengefasst, alle anderen im Vektor $\bar{\mathbf{I}}$. Die Rückanpassung soll nun sicherstellen, dass die Auswertung des Nominalmodells an den Gitterstellen dem Detailmodell entspricht, d. h.

$$\mathbf{y}_N(\mathbf{I}) \stackrel{!}{=} \mathbf{E}(\mathbf{X}_N(\mathbf{I})), \quad (38)$$

$$\mathbf{y}_N(\mathbf{I}) = \mathbf{m}(\mathbf{X}_N(\mathbf{I})) + \mathbf{k}_D(\mathbf{X}_N(\mathbf{I}), \mathbf{X}_D)\boldsymbol{\alpha}_D, \quad (39)$$

$$\mathbf{y}_N(\mathbf{I}) = \mathbf{y}_N(\mathbf{I}) + \mathbf{k}_D(\mathbf{X}_N(\mathbf{I}), \mathbf{X}_D)\boldsymbol{\alpha}_D. \quad (40)$$

Dies führt zu

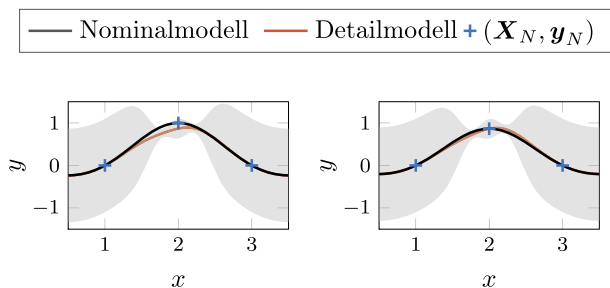
$$0 = \underbrace{\mathbf{k}_D(\mathbf{X}_N(\mathbf{I}), \mathbf{X}_D)\mathbf{K}_D^{-1}}_{\mathbf{A}} (\mathbf{y}_D - \mathbf{m}(\mathbf{X}_D)), \quad (41)$$

$$0 = \mathbf{A}\mathbf{y}_D - \underbrace{\mathbf{A}\mathbf{k}_N(\mathbf{X}_D, \mathbf{X}_N)(\mathbf{k}_N(\mathbf{X}_N, \mathbf{X}_N))^{-1}}_{\mathbf{B}} \mathbf{y}_N. \quad (42)$$

Gleichung (42) kann nun in beeinflusste und unbeeinflusste Gitterpunkte aufgeteilt werden. Das lineare Gleichungssystem für die Rückanpassung kann damit zu

$$\mathbf{AB}(\mathbf{I})\mathbf{y}_N(\mathbf{I}) = \mathbf{A}\mathbf{y}_D - \mathbf{AB}(\bar{\mathbf{I}})\mathbf{y}_N(\bar{\mathbf{I}}) \quad (43)$$

zusammengefasst werden. In Abbildung 4 ist ein beispielhaftes Vorgehen dargestellt. Hier kann der beeinflusste mittlere Gitterpunkt des Nominalmodells angepasst werden und so die Modellgüte erhöht werden. Da im Online-Betrieb im Vergleich zur Offline-Modellerstellung nur ein stark reduzierter Datensatz zur Verfügung steht, wird das Modell nur in der Nähe der Datenpunkte korrigiert, um Robustheit zu gewährleisten. Dadurch können allerdings die Monotonieigenschaften nicht mehr systematisch berücksichtigt werden.



(a) Gitterpunkt $x = 2$ wird stark durch Detailmodell beeinflusst.

(b) Gitterpunkt $x = 2$ wird rückangepasst.

Abb. 4: Vorgehen bei Rückanpassung des Nominalmodells.

4.5 Reduzieren der Datenpunkte auf Basis zeitlicher Bewertung

Neben dem absoluten Einfluss von Stützstellen auf das Modell ist es von Vorteil, auch eine zeitliche Bewertung der Stützstellen vorzunehmen. Die Intention dabei ist, dass Bereiche der zu modellierenden Funktion, in denen lange keine Messwerte vorlagen, an Unsicherheit zunehmen. Grundsätzlich bestehen hierbei zwei Möglichkeiten. Zum einen kann die Unsicherheitsinformation r_i der Modell-datenpunkte zeitlich verändert werden. Da dies aber zu einem deutlich erhöhtem Rechenaufwand führt, muss in Hinblick auf eine echtzeitfähige Implementierung davon Abstand genommen werden. Alternativ wird die zeitliche Bewertung der einzelnen Stützstellen separat in einem Score-Vektor \mathbf{s} hinterlegt. Der Score \mathbf{s} unterliegt einer zeitlichen Abnahme in Form eines Verzögerungsgliedes erster Ordnung

$$\dot{\mathbf{s}}_i = \frac{1}{T} \left(\frac{k_D(\mathbf{x}_M, \mathbf{x}_i)}{\sigma_F^2} - \mathbf{s}_i \right) \quad (44)$$

mit der Zeitkonstante T . Den Eingang bildet dabei die normierte Kovarianz $k_D(\mathbf{x}_M, \mathbf{x}_i)/\sigma_F^2$ des Datenpunktes zur aktuellen Messung. Damit ist sichergestellt, dass der Score in der Nähe der Messungen zunimmt, während Datenpunkte, in deren Nähe lange keine Messung stattgefunden hat, im Score abnehmen. Die Zeitkonstante T soll dabei abhängig von der Änderungsrate des Systems sein. Eine Maßzahl hierfür bildet die Differenz zwischen den Modellwerten vor einem Adaptionsschritt y^- und nach der Adaption y^+ . Die Adaptionsrate ΔM ist dann durch

$$\Delta M = (y^+ - y^-)^2 \quad (45)$$

gegeben. Da die Adaptionsrate starken zeitlichen Schwankungen unterliegt, muss diese noch zeitlich gefiltert werden. Die gefilterte Adaptionsrate ΔM_f wird nun als Zeitkonstante für die Änderung des Scores

$$T = \frac{1}{\Delta M_f} \quad (46)$$

genutzt. Damit ergibt sich für eine große Änderungsrate eine schnelle Abnahme des Scores.

Während der Online-Adaption kann der Score nun genutzt werden, um zu unterscheiden, welche Datenpunkte bereits veraltet sind und aus dem Modell entfernt werden sollten. Neben der Bedingung, dass Datenpunkte aufgrund ihres Einflusses auf den Mittelwert des Modells im Modell gehalten werden, wie in Abschnitt 3.3 erläutert, bildet dies ein zweites Kriterium für die Relevanz von Datenpunkten. Zur Verknüpfung der beiden Kriterien wird das

Minimum des Produktes

$$\min_i \mathbf{s}_i \cdot (\mathbb{E}(\mathcal{GP}_{\mathbf{X}_D}(\mathbf{x}_i)) - \mathbb{E}(\mathcal{GP}_{\mathbf{X}_D \setminus \mathbf{x}_i}(\mathbf{x}_i))) \quad (47)$$

gesucht. Damit werden Datenpunkte sowohl dann entfernt, wenn die Wahrscheinlichkeit hoch ist, dass sie nicht mehr das aktuelle Systemverhalten abbilden, als auch dann, wenn ihr Einfluss auf das Modell gering ist.

Die gesamte Online-Adaption ist in Algorithmus 2 zusammengefasst. Zunächst wird auf Basis des Unsicherheitsbereichs des Modells entschieden, ob ein Messpunkt innerhalb oder außerhalb des Modells liegt. Für den Fall, dass der Messpunkt im Konfidenzbereich liegt, wird der neue Punkt hinzugefügt und die Unsicherheit angepasst. Falls der Messpunkt außerhalb liegt, werden diejenigen Punkte entfernt, die durch die neue Messung als Änderung detektiert worden sind, und der neue Punkt wird hinzugefügt. Im Anschluss werden die Schritte der Rückanpassung des Nominalmodells analog zur Offline-Modellerstellung durchgeführt. Bei der Reduzierung der Datenpunkte wird im Gegensatz zur Offline-Modellerstellung eine zeitliche Bewertung der Datenpunkte berücksichtigt, die in jedem Zeitschritt vorgenommen werden muss.

Algorithmus 2 Online-Modellanpassung.

- 1: Berechne u_p, y_p mit Gleichung (15), (16).
 - 2: **if** $y \in$ Konfidenzintervall (22) **then**
 - 3: Unsicherheit anpassen mit (36).
 - 4: Punkt hinzufügen mit (24), (25).
 - 5: **else**
 - 6: **while** $y \notin$ Konfidenzintervall (23) **do**
 - 7: Punkt entfernen mit (26).
 - 8: **end while**
 - 9: Punkt hinzufügen mit (24), (25).
 - 10: **end if**
 - 11: Rückanpassung des Nominalmodells mit (43).
 - 12: Reduzieren der Datenpunkte mit (47).
 - 13: Anpassung des Scores mit (46), (44).
-

5 Validierung an Testfunktionen

Zur Validierung der vorgestellten Methodik wird diese nun zur leichteren Veranschaulichung an zufälligen zweidimensionalen Funktionen getestet. Zur Generierung von solchen Funktionen wird die zweidimensionale Formulierung von Perlin-Rauschen genutzt, welche in [8] vorgestellt wird. Um strenge Monotonie in einem Eingang zu

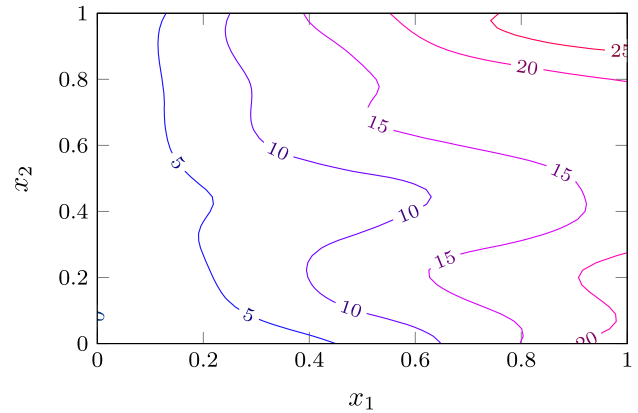


Abb. 5: Zweidimensionale Zufallstestfunktion $f(x_1, x_2)$.

simulieren, wird das Integral über die erste Eingangsdimension gebildet. Eine Beispielfunktion ist in Abbildung 5 gezeigt, welche im Folgenden als Testfunktion verwendet wird. Die Modelle haben damit die Form

$$y = f(x_1, x_2). \quad (48)$$

Die Monotonieeigenschaften werden in den Mengen

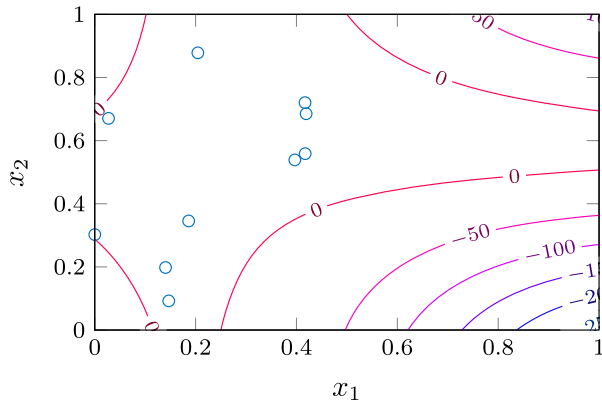
$$\mathcal{M}_I = \{1\} \quad \mathcal{M}_D = \{\} \quad (49)$$

zusammengefasst. Zur Erstellung des Nominalmodells, wie in Abschnitt 3 beschrieben, wird der Parameter $l_N = 0.5$ identisch zur Schrittweite des Gitters gewählt. Der *Length-Scale*-Parameter l_D des Detailmodells wird zu $l_D = l_N/4$ gewählt. Für die Wahl von σ_F hat sich die heuristische Regel von $\sigma_F = 0.1\Delta y_D$ bewährt, wobei mit Δy_D die Differenz aus dem Maximum und dem Minimum des Gesamtdatensatzes y_D bezeichnet wird. Das normalverteilte Messrauschen wird mit einer Standardabweichung von $\sigma_M = 0.1$ simuliert.

5.1 Validierung Erstellung Nominalmodell

Für die Veranschaulichung des Extrapolationsverhaltens werden nun 10 zufällige Datenpunkte aus der Funktion gezogen und damit das Nominalmodell zunächst ohne Wissen über das Extrapolationsverhalten und anschließend unter Verwendung dieses Wissens generiert. In Abbildung 6 sind die resultierenden Modelle dargestellt. Zu sehen ist, dass die Modellgüte in Bereichen, an denen keine Messwerte vorlagen, durch das Modell mit Monotoniewissen wesentlich besser abgebildet werden kann. Das Modell ohne Berücksichtigung der Monotonieeigenschaften dagegen weist in Randbereichen kein sinnvolles Extrapolationsverhalten auf.

(a) Nominalmodell ohne Monotonieinformation.



(b) Nominalmodell mit Monotonieinformation.

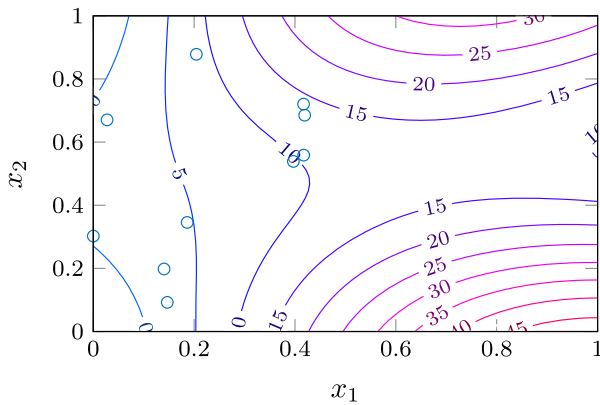


Abb. 6: Generierung Nominalmodell mit 10 Datenpunkten.

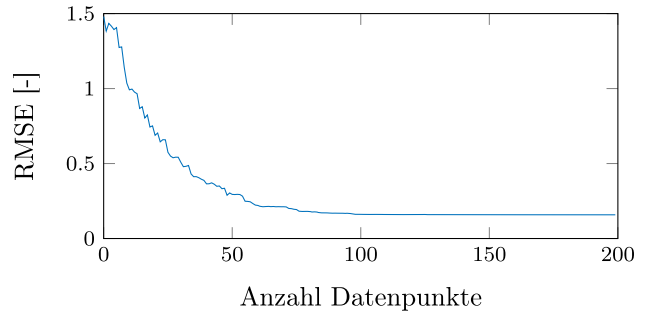


Abb. 7: Iterative Reduzierung der Datenpunkte.

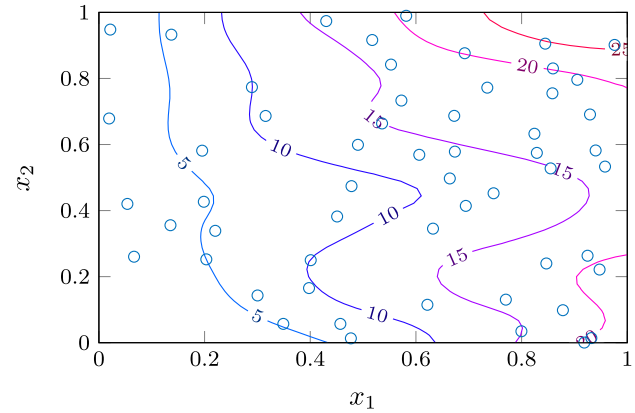


Abb. 8: Detailmodell mit 55 Datenpunkten.

5.2 Validierung Offline-Modellerstellung

Nach der Validierung des Extrapolationsverhaltens soll nun die Offline-Modellerstellung auf Basis eines großen Offline-Datensatzes mit 200 zufälligen Datenpunkten veranschaulicht werden. Dazu wird zunächst ein Nominalmodell mit Monotonieeigenschaften in x_1 erzeugt. Das Detailmodell wird daraufhin mit allen Datenpunkten generiert, bevor diese schrittweise mit Gleichung (20) reduziert werden. Von besonderem Interesse ist hier die Abnahme der Modellgüte im Verlauf der Datenpunktreduzierung. Diese wird über den mittleren quadratischen Fehler

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2} \quad (50)$$

mit dem Wert der Testfunktion y_i und dem Modellwert \hat{y}_i bewertet. In Abbildung 7 ist zu sehen, dass bei der Reduzierung von 200 auf 100 Datenpunkte die Modellgüte nahezu konstant bleibt. Dies ist darauf zurückzuführen, dass die reduzierten 100 Datenpunkte kaum Einfluss auf den

Mittelwert des Modells haben und ohne Güteverlust entfernt werden können. Erst bei weiterer Reduzierung steigt der Fehler langsam an. Bei 55 Datenpunkten hat der Fehler um 10 % zugenommen. Das zugehörige Modell ist in Abbildung 8 dargestellt.

5.3 Validierung Online-Modellanpassung

Zur Validierung der Online-Adaption wird eine zweite zufällige Testfunktion auf Basis von Perlin-Rauschen generiert. Beide Testfunktionen liegen dann als zweidimensionale Kennfelder vor. Eine zeitliche Änderung wird durch die lineare Interpolation beider Modelle erreicht. Als TestszENARIO werden zunächst 500 Punkte aus Modell 1 gezogen, mit welchem auch die Offline-Modellgenerierung durchgeführt wurde. Im Anschluss wird in 500 Teilschritten linear auf Modell 2 gewechselt, bevor dieses für 500 Schritte konstant gehalten wird. Nun wird das Modell sprunghaft zurück auf Modell 1 geschaltet und weitere 500

Adaptionsschritte durchgeführt. Der Zyklus lässt sich als

$$\text{Modell} = \begin{cases} 1 & \text{for } k < 500 \\ 1 \rightarrow 2 & \text{for } 500 \leq k < 1000 \\ 2 & \text{for } 1000 \leq k < 1500 \\ 1 & \text{for } 1500 \leq k < 2000 \end{cases} \quad (51)$$

mit dem linearen Übergang \rightarrow zwischen den Modellen beschreiben. Die Adaptionroutine (Algorithmus 2) wird in jedem Zeitschritt einmal aufgerufen. In Abbildung 9 ist der Verlauf des Modellfehlers über den gesamten Zyklus zu sehen. Zu beobachten ist, dass der Fehler zunächst konstant bleibt, da das Modell bereits auf Testfunktion 1 trainiert war. Im Abschnitt des linearen Modellwechsels steigt der Fehler dann zunächst langsam an, bevor er für das konstante Testmodell 2 wieder reduziert wird. Beim Modellsprung springt auch der Fehler zunächst, kann aber zügig wieder auf das Anfangsniveau gebracht werden. Zu beobachten sind diese Effekte sowohl im Nominalmodell durch die Rückanpassung als auch im Detailmodell durch das Hinzufügen und Entfernen von Punkten. Eine weitere Größe zur Beurteilung des Adaptionsverhaltens ist die detektierte Änderungsrate ΔM_f in Abbildung 10. Auch hier

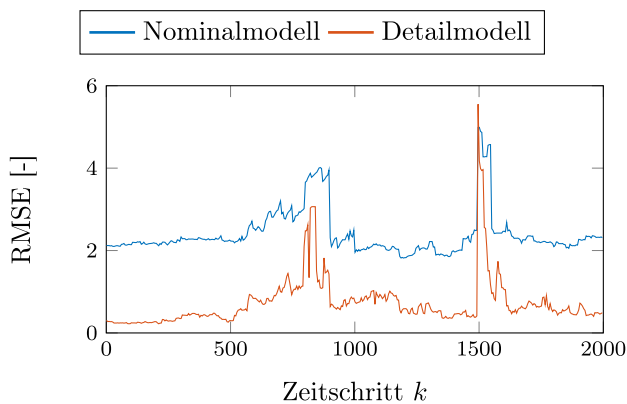


Abb. 9: Modellfehler im Verlauf der Online-Adaption.

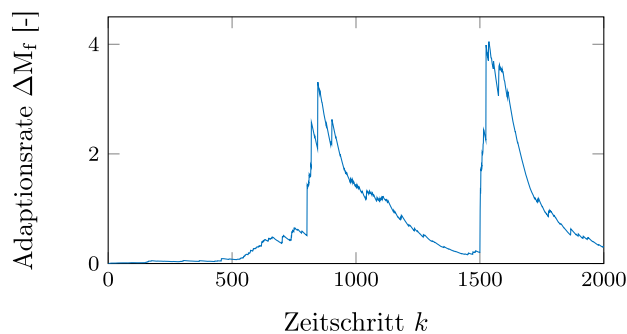


Abb. 10: Modellfehler im Verlauf der Online-Adaption.

sind die Modelländerungen gut erkennbar. Für schnelle Modelländerungen nimmt auch die detektierte Modelländerungsrate zu.

6 Zusammenfassung und Ausblick

In diesem Beitrag wurde eine datenbasierte Modellbildungsmethode vorgestellt, die zum einen systematisch Expertenwissen über ein System in Form von Monotonieeigenschaften in die Modellbildung integriert und sich zum anderen in der Laufzeit an ein zeitveränderliches System durch einen Online-Adaptionsverfahren anpasst. Damit ist es möglich, Modelle zu generieren, die ein sinnvolles Extrapolationsverhalten in Bereichen aufweisen, in denen keine Messwerte vorliegen. Dies ist besonders im Hinblick auf eine regelungstechnische Anwendung von großem Vorteil, wie es in [1] am Beispiel eines Schwerlast-Dieselmotors gezeigt wird. Desweiteren wurde ein Verfahren vorgestellt, welches auf Basis der Gaußprozessregression eine Online-Adaption des Modells durchführen kann, wie es bereits in [2] an numerischen Beispielen und in [1] an einem realen System gezeigt wurde. Dazu wird die Eigenschaft des Gaußprozesses genutzt, der über eine analytisch auswertbare Unsicherheitsinformation verfügt. Schließlich wurde auch eine zeitliche Bewertung der einzelnen Datenpunkte durchgeführt, die es möglich macht, Aussagen über die Geschwindigkeit der Modelländerung zu machen. Sowohl die Offline-Modellerstellung als auch die Online-Anpassung wurden an zufälligen Funktionen getestet, um deren vielfältigen Einsatzmöglichkeiten aufzuzeigen.

In folgenden Arbeiten soll der Online-Algorithmus im Hinblick auf seine Echtzeitfähigkeit hin weiterentwickelt werden und an weiteren realen technischen Systemen getestet werden.

Literatur

1. D. Bergmann, R. Geiselhart and K. Graichen. "Modelling and control of a heavy-duty Diesel engine gas path with Gaussian process regression". In: *Proc. European Control Conference (angenommen)* (2019).
2. D. Bergmann, M. Buchholz, J. Niemeyer, J. Remele and K. Graichen. "Gaussian process regression for nonlinear time-varying system identification". In: *Proc. Conference on Decision and Control* (2018), S. 3025–3031.
3. David S. Broomhead and David Lowe. *Radial basis functions, multi-variable functional interpolation and adaptive networks*. Technical report. Royal Signals and Radar Establishment Malvern (United Kingdom), 1988.

4. Lehel Csató and Manfred Opper. "Sparse representation for Gaussian process models". In: *Advances in Neural Information Processing Systems*, 2001, S. 444–450.
5. Paul W. Goldberg, Christopher Williams and Christopher M. Bishop. "Regression with input-dependent noise: A Gaussian process treatment". In: *Advances in Neural Information Processing Systems*. 1998. S. 493–499.
6. Marco F. Huber. "Recursive Gaussian process: On-line regression and learning". In: *Pattern Recognition Letters* 45 (2014), S. 85–91.
7. Oliver Nelles and Rolf Isermann. "Basis function networks for interpolation of local linear models". In: *Proc. 35th IEEE Conference on Decision and Control*, 1996. Bd. 1, S. 470–475.
8. Ken Perlin. "Improving noise". In: *ACM Transactions on Graphics (TOG)*. Bd. 21. 3. ACM. 2002, S. 681–682.
9. Ananth Ranganathan, Ming-Hsuan Yang and Jeffrey Ho. "Online sparse Gaussian process regression and its applications". In: *IEEE Transactions on Image Processing* 20.2 (2011), S. 391–404.
10. Carl Edward Rasmussen and Christopher Williams. *Gaussian Processes for Machine Learning*, Bd. 1. MIT Press Cambridge, 2006.
11. Jaakko Riihimäki and Aki Vehtari. "Gaussian processes with monotonicity information". In: *Proc. International Conference on Artificial Intelligence and Statistics*. 2010, S. 645–652.
12. Frank Rosenblatt. *Principles of neurodynamics. perceptrons and the theory of brain mechanisms*, Technical report. Cornell Aeronautical Lab Inc. Buffalo NY, 1961.
13. Hans-Rudolf Schwarz und Norbert Köckler. *Numerische Mathematik*. Springer-Verlag, 2013.
14. Aleksandr Filippovich Timan, *Theory of approximation of functions of a real variable*. Bd. 34. Elsevier, 2014.
15. Max A. Woodbury. "Inverting modified matrices". *Memorandum Report* 42.106 (1950), S. 336.

Autoreninformationen



M. Sc. Daniel Bergmann

Institut für Mess-, Regel- und Mikrotechnik,
Universität Ulm, Albert-Einstein-Allee 41,
D-89081 Ulm, Deutschland
daniel.bergmann@uni-ulm.de

M. Sc. Daniel Bergmann ist Mitarbeiter am Institut für Mess-, Regel- und Mikrotechnik der Universität Ulm. Hauptarbeitsgebiete: Modellbildung und Identifikation für regelungstechnische Anwendungen, Echtzeitfähige Lernverfahren, Modellbildung von Schwerlast-Dieselmotoren.



Prof. Dr.-Ing. Knut Graichen

Lehrstuhl für Regelungstechnik,
Friedrich-Alexander-Universität
Erlangen-Nürnberg, Cauerstraße 7, D-91058
Erlangen, Deutschland
knut.graichen@fau.de

Prof. Dr.-Ing. Knut Graichen ist Leiter des Lehrstuhls für Regelungstechnik der Friedrich-Alexander-Universität Erlangen-Nürnberg. Hauptarbeitsgebiete: optimale und modellprädiktive Regelung, nichtlineare Steuerungs- und Regelungsverfahren, eingebettete Umsetzung von optimierungsbasierten Verfahren für mechatronische und vernetzte Systeme.