UNIVERSITÄT PADERBORN
Die Universität der Informationsgesellschaft

LEA

Systemidentifikation

Dr.-Ing. Oliver Wallscheid

Skript zur Vorlesung
Version: 12. Februar 2022

Universität Paderborn
Fachgebiet Leistungselektronik und Elektrische Antriebstechnik (LEA)

Vorwort

Das vorliegende Skriptum gibt die wichtigsten Inhalte der Vorlesung *Systemidentifikation* wieder. Dieses soll daher nicht als umfassendes Lehrbuch verstanden werden, sondern als Orientierungs- und Gedächtnisstütze dienen. Ich empfehle den Studentinnen und Studenten, auf eigene Notizen nicht zu verzichten und darüber hinaus neben den Vorlesungs- und Übungsunterlagen auch weitere Lehrbücher zu nutzen – hierzu befindet sich u. a. am Ende des Skriptums ein Literaturverzeichnis, in dem empfehlenswerte Lehrbuchtitel oder auch Forschungsaufsätze aufgelistet sind. Es vertieft das Verständnis, einen Sachverhalt von einer/einem anderen Autorin/Autor, in einer anderen Sichtweise, auch andere Schreibweisen kennen zu lernen. Selbst das Entdecken vermeintlicher oder tatsächlicher Widersprüche ist von Wert, wenn man gewillt ist, sich mit den Inhalten und Zusammenhängen auseinander zu setzen, und nicht nur eine „Formelsammlung“ erwartet.

Sachdienliche Hinweise auf typographische oder inhaltliche Unzulänglichkeiten in den begleitenden Veranstaltungsunterlagen werden sehr geschätzt und gerne angenommen. Auch Rückmeldungen zur allgemeinen Struktur und Durchführung der Lehrveranstaltung sind stets willkommen.

Paderborn, im Februar 2022

Inhaltsverzeichnis

1 Einleitung

1.1 Was ist Systemidentifikation?

Zur Einleitung in die *Systemidentifikation* werden die nachfolgenden Literaturzitate herangezogen, welche das Themenfeld der Lehrveranstaltung beleuchten und zu dessen Eingrenzung beitragen:

Zitat 1.1: Systemidentifikation nach L. Ljung: System Identification: Theory for the User, Prentice Hall, 1999

„Inferring models from observations and studying their properties is really what science is about. The models may be of more or less formal character, but they have the basic feature that they attempt to link observations together in some pattern. System identification deals with the problem of building mathematical models of dynamical systems based on observed data from the system.“ [Lju99]

Zitat 1.2: Systemidentifikation nach R. Isermann: Identification of Dynamic Systems, Springer-Verlag, 2011

„The basic static and dynamic behavior [of processes] can be obtained by theoretical or physical modeling, if the underlying physical laws (first principles) are known in analytical form. If, however, these laws are not known or are only partially known, or if significant parameters are not known precisely enough, one has to perform an experimental modeling, which is called process or system identfication. Then, measured signals are used and process or system models are determined within selected classes of mathematical models.“ [IM11]

Zitat 1.3: Systemidentifikation nach Wikipedia

„Systemidentifikation (auch Systemidentifizierung) ist die theoretische oder/und experimentelle Ermittlung der quantitativen Abhängigkeit der Ausgangs- von den Eingangsgrößen eines Systems. Dazu wird das System mit definierten Testsignalen (Sprung, Impuls, Rampe o. Ä.) angeregt und der Ausgang aufgezeichnet. Die zur mathematischen Auswertung angewandten Verfahren können deterministisch oder stochastisch sein.“ [Wik18a]

Die Systemidentifikation stellt somit eine Teildisziplin der ingenieurwissenschaftlichen Systemtheorie dar, welche die empirische Modellbildung und -validierung beschreibt. Hieraus abgeleitete Modelle können im weiteren systemtheoretischen Sinne für die Regelungssynthese bzw. (numerische) Simulation verwendet werden (s. Abb. 1.1).

Nichtsdestotrotz steht neben der Systemidentifikation als empirisches Modellbildungswerkzeug noch die analytische Modellierung zur Verfügung – siehe Abb. 1.2 zur Darstellung der grund-

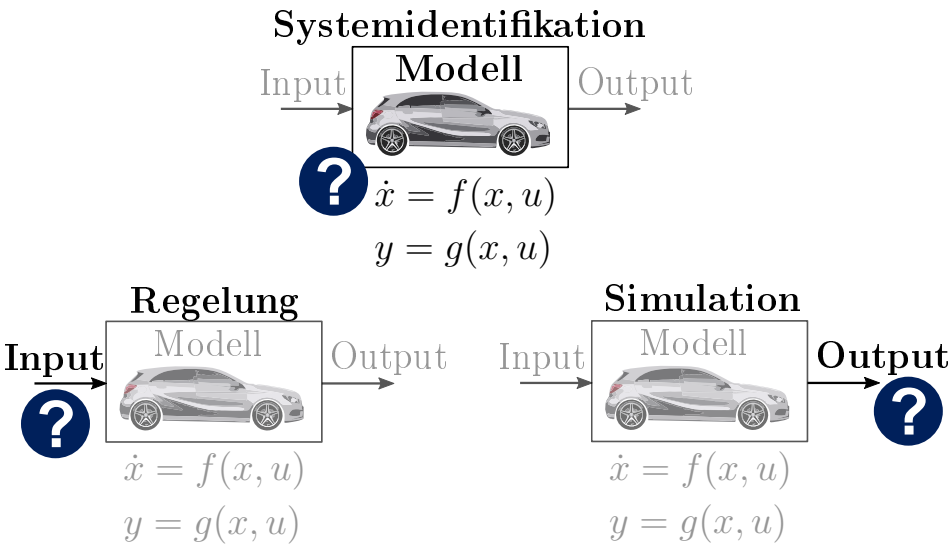


Abb. 1.1: Differenzierung der verschiedenen Teildisziplinen innerhalb der ingenieurwissenschaftlichen Systemtheorie

sätzlichen Vorgehensweise. Je nach Anwendung und Rahmenbedingungen, z. B. verfügbare Zeit für die Modellbildung oder Expertenvorwissen des/der verantwortlichen Ingenieurs/in, stehen beide Modellierungszweige in direkter Konkurrenz zueinander – häufig kann aber auch ein hybrider Lösungsweg aus beiden Ansätzen sinnvoll sein. Zur weiteren Abgrenzung der Vor- und Nachteile einer experimentelle Systemidentifikation sei auf Tab. 1.1 verwiesen.

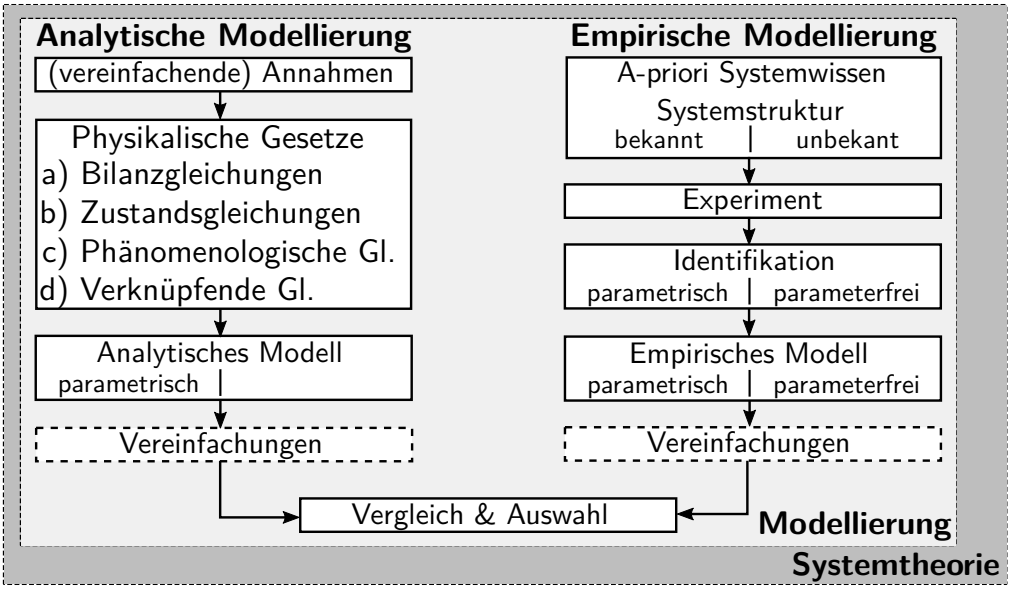


Abb. 1.2: Vorgehen zur analytischen und empirischen Modellbildung (vgl. [IM11])

Neben der direkten Abgrenzung von analytischen und empirischen Modellierungsmethoden ist demgegenüber auch eine ergänzende Modellbildung durch die Verschmelzung beider Methoden als sog. Grey-Box-Modelle möglich. Dies ist exemplarisch anhand des elektromagnetischen Verhaltens eines Elektromotors in Abb. 1.3 dargestellt. Hier stellen Black- und White-Box-Modelle die beiden Extrema dar: Während bei Ersterem eine rein datengetriebene Identifikation ohne

Vorteile	Nachteile
• **Schnell** Insbesondere für große Systeme oder solche mit komplexer innerer Struktur • **Präzise** Falls interne Systemstruktur unbekannt oder schlecht beschreibbar • **Transferierbar** Methodik auf verschiedene Domänen anwendbar • **Einfach** Nur wenig anwendungsspezifisches Wissen notwendig • **Skalierbar** Modellordnung kann anforderungsspezifisch angepasst werden	• **Experiment erforderlich** Eventuell aufwendig, kostenintensiv oder sogar nicht realisierbar (z. B. während technischer Neuentwicklungen) • **Begrenzte Gültigkeit** Extrapolation außerhalb des identifizierten Arbeitsbereich ggf. ungenau oder sogar instabil • **Begrenzte Aussagekraft** Technisch-physikalische Interpretation aufgrund Abstraktion nicht oder nur eingeschränkt möglich

Tab. 1.1: Vor- und Nachteile der Systemidentifikation gegenüber der analytischen Modellbildung

jegliches technisch-physikalisches Systemwissen stattfindet, erfolgt die Modellierung bei Letzterem durch expertengetriebenes, anwendungsspezifisches Wissen auf analytischem Wege. Die hybride Nutzung beider Ansätze findet in der Praxis besonders häufig statt: Analytische Modellierungstechniken bringen Robustheit und Strukturierung in das Modellierungsproblem ein, da diese auf dem gesicherten Fachwissen der jeweiligen Anwendungsdomäne basieren. Demgegenüber können empirische Identifikationstechniken die Genauigkeit der Modelle erhöhen, da in praktischen Aufbauten häufig Systemparameter nicht oder nur unzureichend genau bekannt sind bzw. deren exakte analytische Modellbildung zu zeitaufwendig wäre. Einige Identifikationsprobleme entsprechen zudem nichtlinearen und mehrdimensionalen Optimierungsproblemen, welche neben dem gesuchten globalen Optimum häufig noch suboptimale, lokale Optima aufweisen. Um derartige Optimierungsprobleme in endlicher Zeit und mit zielführenden Ergebnissen lösen zu können, müssen hierfür geeignete Startwerte als auch Parametergrenzen im Suchraum vorgegeben werden – hier kann eine analytische (Grob-)Modellierung helfen, um entsprechende Vorgaben an das Optimierungsproblem zu stellen.

Ein typisches Grey-Box-Modellierungsbeispiel aus Abb. 1.3 stellt die Nutzung von Verlustleistungskoeffizienten in einer Finiten-Elemente-Analyse (FEA) für elektromagnetische Systeme wie Elektromotoren oder Transformatoren dar: Diese können a-priori nur grob abgeschätzt werden, da diese nicht nur von den verwendeten Materialien, sondern auch von der Bauteilgeometrie, Verarbeitungsprozessen in der Produktion und Temperaturverteilung im Betrieb abhängen. Steht ein Prototyp des betrachteten Systems zur Verfügung, können besagte Koeffizienten durch Abgleich mit Verlustleistungsmessdaten korrigiert werden, um so die Modellgüte zu erhöhen.

Das grundsätzliche Vorgehen bei der Identifikation ist im Ablaufdiagramm aus Abb. 1.4 dargestellt. Hier sei auf einige Aspekte besonders hingewiesen: Jede Identifikationsaufgabe unterliegt einem begrenzten Zeit- und Kostenbudget und das zu ermittelnde Modell folgt stets einem anwendungsbezogenen Zweck. Hieraus lassen sich Anforderungen an das zu identifizierende Modell stellen, z. B. definiert durch quantifizierbare Gütemaße hinsichtlich der Abweichungen zwischen Modellausgang und realem Prozess. Im Zuge der (Kreuz-)Validierung ist dann zu prüfen, ob ein

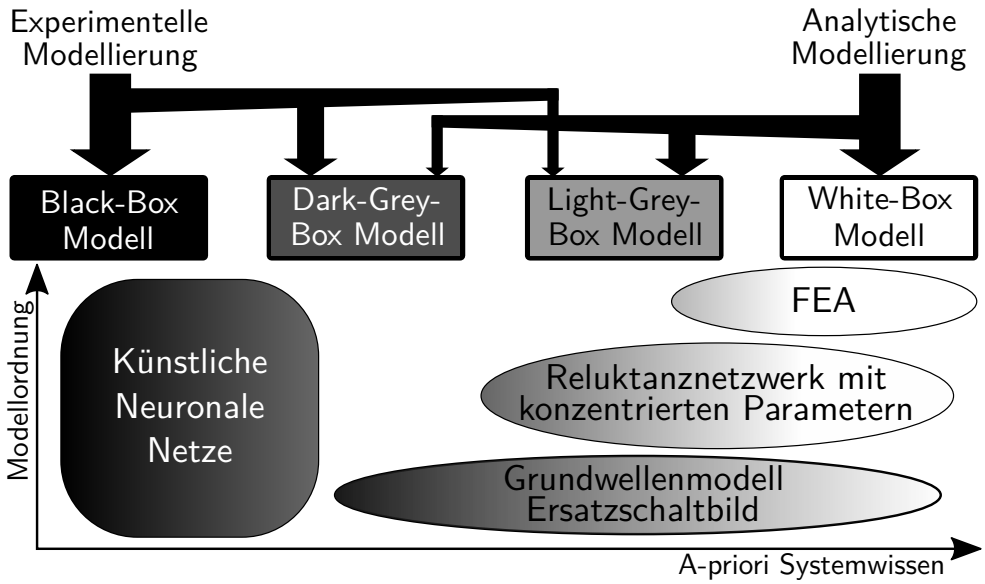


Abb. 1.3: Verschiedene Modellierungsansätze beispielhaft skizziert für das elektromagnetische Verhalten eines Elektromotors

gefundenes Modell diesen Anforderungen genügt – falls dies der Fall ist, ist die Aufgabe gelöst. Andernfalls müssen Veränderungen innerhalb des Identifikationsvorgangs vorgenommen werden, wie z. B. eine Anpassung der Modellstruktur oder der Systemanregung. Der beschriebene Vorgang kann daher als Regelschleife interpretiert werden, welche durch die/den verantwortliche/n Ingenieur/in geschlossen wird – je nach Anwendungskontext mit einem mehr oder weniger hohen Automatisierungsgrad in den verschiedenen Bearbeitungsschritten.

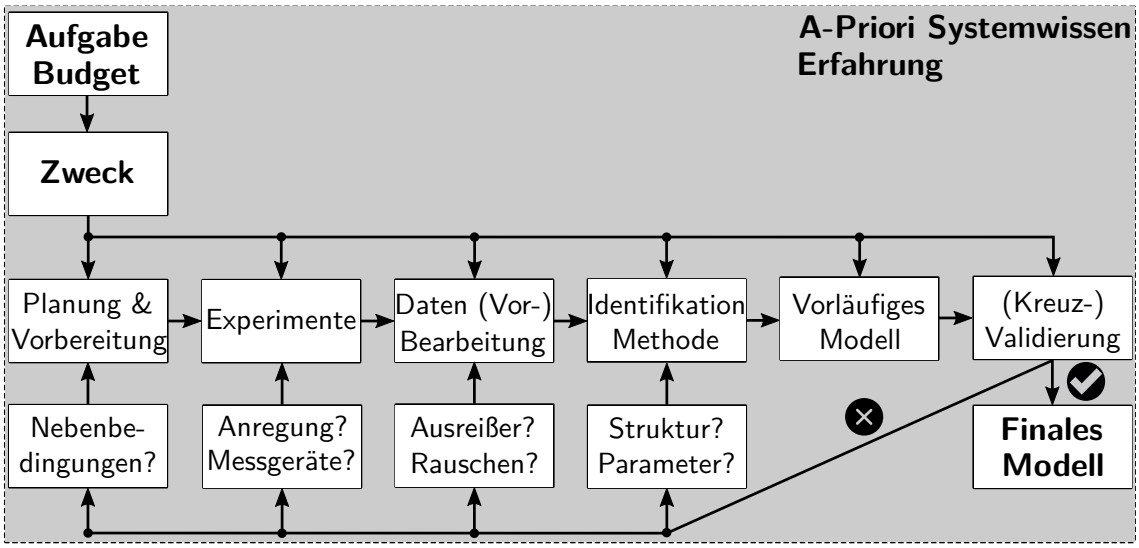


Abb. 1.4: Sequentielles Vorgehen zur Lösung von Identifikationsproblemen (vgl. [IM11])

1.2 Grundlegende Begriffsdefinitionen

Im Folgenden werden einige grundlegende Begriffsdefinitionen und -abgrenzungen vorgenommen, auf welche in den darauffolgenden Kapiteln Bezug genommen wird. Begonnen wird hierbei mit der Differenzierung der Begriffe *System* und *Prozess*:

Definition 1.1: System

Unter einem System versteht man eine begrenzte Anordnung von gegenseitig betroffenen Entitäten (DIN 66201). Im Folgenden sind diese Entitäten (technische) Prozesse. Jedes System wird durch seine räumlichen und zeitlichen Grenzen beschränkt, durch seine Umgebung beeinflusst, durch seine Struktur und seinen Zweck beschrieben und in seiner Funktionsweise ausgedrückt.

Definition 1.2: Prozess

Ein Prozess ist eine Gesamtheit von aufeinander einwirkenden Vorgängen in einem System, durch die Materie, Energie oder Information umgeformt oder gespeichert werden (ISO 9001:2015). Ein technischer Prozess ist ein Prozess, dessen physikalische Größen mit technischen Mitteln erfasst und beeinflusst werden.

Dabei unterscheidet man typischerweise zwischen einzelnen (Teil-)Prozessen und dem Gesamtprozess. Ein einzelner (Teil-)Prozess kann beispielsweise die Erzeugung von mechanischer Energie aus elektrischer Energie sein, während der Gesamtprozess ein elektrischer Generator sein kann. Aufgrund der engen Verknüpfung zwischen diesen beiden Begrifflichkeiten werden die Ausdrücke *Systemidentifikation* und *Prozessidentifikation* in der Literatur häufig synonym verwendet, obwohl streng genommen hiermit nicht der gleiche Sachverhalt adressiert wird.

Weiterhin gilt es zu unterscheiden, ob ein *parametrisches* oder ein *parameterfreies Modell* zu identifizieren ist:

Definition 1.3: Parametrisches Modell

Parametrische Modelle sind Gleichungen mit definierter Struktur und finiter Anzahl von Parametern, die explizit die Prozessparameter enthalten. Beispiele sind Differentialgleichungen oder Übertragungsfunktionen, die als algebraischer Ausdruck angegeben werden.

Definition 1.4: Parameterfreies Modell

Parameterfreie Modelle stellen eine Beziehung zwischen einem bestimmten Eingang und dem entsprechenden Ausgang über eine Tabelle oder eine charakteristische Kennlinie her. Sie haben eine infinite Anzahl von Parametern, aber keine spezifische Struktur. Beispiele sind Impulsantworten, Sprungantworten oder Frequenzgänge in tabellarischer oder grafischer Form. Sie enthalten die Systemparameter daher nur implizit.

Zudem gilt es verschiedene wichtige Systemeigenschaften zu differenzieren, welche maßgeblichen Einfluss auf den Identifikationsprozess und die Wahl der richtigen Identifikationswerkzeuge haben:

Definition 1.5: Lineare bzw. nichtlineare Modelle

Wenn alle Operatoren in einem mathematischen Modell die Eigenschaften der Additivität und Homogenität aufweisen, ist das Modell linear. Andernfalls ist dieses nichtlinear.

Definition 1.6: Diskrete bzw. kontinuierliche Modelle

Ein diskretes Modell behandelt diskrete Objekte, wie die Partikel in einem molekularen Modell oder die Zustände in einem statistischen Modell. Demgegenüber adressieren kontinuierliche Modelle Objekte, wie z. B. das Geschwindigkeitsfeld von Flüssigkeiten oder elektrische Felder, die sich kontinuierlich ausbreiten. Das Gleiche gilt auch bezogen auf die Zeitskala eines gegebenen Modells.

Definition 1.7: Statische bzw. dynamische Modelle

Ein dynamisches Modell berücksichtigt die zeitabhängigen Zustandsänderungen, sodass das Systemverhalten nicht nur von den aktuellen, sondern auch von dem vergangenen Verlauf der Eingangsgrößen abhängt. Demgegenüber beschreibt ein statisches (oder stationäres) Modell das System im Gleichgewicht – hier besteht kein Zusammenhang zu vorherigen Systemzuständen.

Definition 1.8: Deterministische bzw. stochastische Modelle

Ein deterministisches Modell ist durch einen Satz von Zuständen, Parametern und ggf. deren vergangenen zeitlichen Verlauf eindeutig bestimmt; daher verhält sich ein deterministisches Modell für einen gegebenen Inertialzustand sowie für eine gegebene externe Anregung stets gleich. Umgekehrt ist in einem stochastischen Modell Zufälligkeit vorhanden, sodass Zustände nicht durch eindeutige Werte beschrieben, sondern durch Wahrscheinlichkeitsaussagen abgebildet werden.

Definition 1.9: Modelle mit konzentrierten bzw. verteilten Parametern

Modelle mit konzentrierten Parametern sind ortsunabhängig und werden typischerweise mittels gewöhnlicher Differentialgleichungen beschrieben (z. B. elektrische Ersatzschaltbilder). Modelle mit verteilten Parametern sind hingegen ortsabhängig und werden typischerweise durch partielle Differentialgleichungen abgebildet (z. B. elektromagnetische Wellenausbreitung).

Weiterhin sei auf die grundsätzlichen, unterschiedlichen Eigenschaften zur Realisierung von Identifikationsmethoden hingewiesen:

Definition 1.10: Offline- bzw. Online-Identifikation

Für die Offline-Identifikation werden die gemessenen Daten zunächst gespeichert, später an den für die Datenauswertung genutzten Rechner übertragen und dann verarbeitet. Die Online-Identifikation erfolgt parallel zum Experiment. Der Computer ist mit dem Prozess gekoppelt und die Daten werden verarbeitet, sobald diese verfügbar sind.

Definition 1.11: Rekursive bzw. nicht-rekursive Identifikation

Nicht-rekursive Methoden bestimmen das Modell aus den vorherigen Messungen und sind das bevorzugte Mittel für die Offline-Identifikation. Demgegenüber aktualisieren die rekursiven Methoden das Modell, sobald eine neue Messung verfügbar wird. Die neue Messung wird genutzt, um das zuvor abgeleitete Modell zu verbessern. Die alten Messungen müssen nicht gespeichert werden. Dies ist der typische Ansatz für die Online-Identifikation.

Definition 1.12: Direkte bzw. iterative Identifikation

Die direkte Identifikation bestimmt das Modell in einem Durchgang. Die iterative Verarbeitung bestimmt das Modell schrittweise. So entstehen Iterationszyklen, durch die die Daten mehrfach verarbeitet werden müssen.

2 Mathematische Grundlagen

In diesem Kapitel werden die systemtheoretischen Grundlagen für die wesentlichen Modellklassen, welche diese Veranstaltung behandelt, rekapituliert. Dies dient insbesondere dem Zweck, eine geschlossene Nomenklatur sicherzustellen. Unabhängig davon wird für die darauffolgenden Kapitel ein gefestigtes Basiswissen im Bereich der ingenieurwissenschaftlichen Systemtheorie, Stochastik sowie Regelungstechnik vorausgesetzt, welches über die Inhalte dieses Kapitels hinaus geht. Insofern wird eine Auffrischung der Grundlagen durch Studium einschlägiger Literatur (z. B.[1] [Gra13][Kug17] für Systemtheorie, [Hen13] für Stochastik, [Föl13][Lun13][LW14] für Regelungstechnik) empfohlen, sofern hier Wissenslücken bestehen.

2.1 Zeitkontinuierliche, deterministische und dynamische Modelle

Zunächst sollen zeit- und wertkontinuierliche Modelle behandelt werden.[2] Hier gilt folgende Basisdefinition:

Definition 2.1: Deterministisch-dynamisches Modell

Deterministische und dynamische Modelle sein gegeben durch

$$\begin{aligned} \dot{\boldsymbol{x}}(t) &= \boldsymbol{f}\left(\boldsymbol{x}, \boldsymbol{u}, t\right) , \quad \boldsymbol{x}(t_0) = \boldsymbol{x}_0 , \\ \boldsymbol{y}(t) &= \boldsymbol{g}\left(\boldsymbol{x}, \boldsymbol{u}, t\right) \end{aligned} \tag{2.1}$$

mit dem Zustand $\boldsymbol{x} \in \mathbb{R}^n$, dem Eingang $\boldsymbol{u} \in \mathbb{R}^m$, dem Ausgang $\boldsymbol{y} \in \mathbb{R}^p$ und der Zeit t. Ferner seien $\boldsymbol{x}_0 \in \mathbb{R}^n$ der Anfangszustand, $\boldsymbol{f} : \mathbb{R}^n \to \mathbb{R}^n$ eine Funktion zur Beschreibung der Zustandsänderung sowie $\boldsymbol{g} : \mathbb{R}^n \to \mathbb{R}^p$ die Ausgangsfunktion.[3]

Kann auf das Modell (2.1) das Superpositionsprinzip (Additivität und Homogenität) angewandt werden, so ist dieses linear:

[1] Gemäß DIN 5008 ist bei Abkürzungen, die aus mehreren Buchstaben bestehen und die jeweils einen Punkt haben und für ein Wort stehen, zwischen den abgekürzten Wörtern ein geschütztes Leerzeichen zu setzen.

[2] Wertdiskrete Modelle bzw. Signale werden in dieser Veranstaltung nicht behandelt. Es wird angenommen, dass eine hinreichend genaue Wertdiskretisierung (Sensorik + Analog-Digital-Konverter) sowie digitale Signalverarbeitung im jeweiligen Prozess stattfindet, sodass die Berechnungen quasi-wertkontinuierlich erfolgen können.

[3] Fette Schriftzeichen beschreiben im Folgenden Vektoren und Matrizen, während nicht-fette Symbole skalare Größen repräsentieren.

Definition 2.2: Lineares Modell

Das Modell (2.1) *wird als linear bezeichnet, wenn zu dem Startzeitpunkt* $t_0 > 0$ *und für alle (zulässigen) Anfangszustände* $\boldsymbol{x}_0$ *sowie Eingangsfolgen* $\boldsymbol{u}(t)$ *die folgenden Bedingungen für* $t > t_0$ *und* $\alpha_1, \alpha_2, \beta_1, \beta_2 \in \mathbb{R}$ *erfüllt sind:*

$$\begin{aligned} \boldsymbol{y}(0, \alpha_1\boldsymbol{u}_1 + \alpha_2\boldsymbol{u}_2, t) &= \alpha_1\boldsymbol{y}(0, \boldsymbol{u}_1, t) + \alpha_2\boldsymbol{y}(0, \boldsymbol{u}_2, t)\,, \\ \boldsymbol{y}(\beta_1\boldsymbol{x}_{0,1} + \beta_2\boldsymbol{x}_{0,2}, 0, t) &= \beta_1\boldsymbol{y}(\boldsymbol{x}_{0,1}, 0, t) + \beta_2\boldsymbol{y}(\boldsymbol{x}_{0,2}, 0, t)\,, \\ \boldsymbol{y}(\boldsymbol{x_0}, \boldsymbol{u}, t) &= \boldsymbol{y}(0, \boldsymbol{u}, t) + \boldsymbol{y}(\boldsymbol{x_0}, 0, t)\,.^{1} \end{aligned} \tag{2.2}$$

Weiterhin kann gezeigt werden, dass (2.2) angewandt auf (2.1) zur klassischen Zustandsraumbeschreibung führt:

Satz 2.1: Lineares Modell im Zustandsraum

Das Modell (2.1) *ist linear, wenn es sich in die Form*

$$\begin{aligned} \dot{\boldsymbol{x}}(t) &= \boldsymbol{A}(t)\boldsymbol{x}(t) + \boldsymbol{B}(t)\boldsymbol{u}(t)\,, \\ \boldsymbol{y}(t) &= \boldsymbol{C}(t)\boldsymbol{x}(t) + \boldsymbol{D}(t)\boldsymbol{u}(t) \end{aligned} \tag{2.3}$$

überführen lässt.[2] *Hier ist* $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ *die Systemmatrix,* $\boldsymbol{B} \in \mathbb{R}^{n\times m}$ *die Eingangsmatrix,* $\boldsymbol{C} \in \mathbb{R}^{p\times n}$ *die Ausgangsmatrix und* $\boldsymbol{D} \in \mathbb{R}^{p\times m}$ *die Durchgangsmatrix, deren Einträge lediglich von der Zeit* t *abhängen.*

Als wichtiger Spezialfall eines linearen Systems lässt sich die Eigenschaft der *Zeitinvarianz* nennen, welche zur bekannten Systemklasse der LTI-Systeme (*linear time-invariant*) führt:

Definition 2.3: Linear-zeitinvariantes Modell

Das Modell (2.1) *wird als zeitinvariant bezeichnet, wenn zu dem Startzeitpunkt* $t_0 > 0$ *und für alle (zulässigen) Anfangszustände* $\boldsymbol{x}_0$ *sowie Eingangsfolgen* $\boldsymbol{u}(t)$ *die folgenden Bedingung für* $T \in \{\mathbb{R}|0 < T\}$ *erfüllt ist:*

$$\boldsymbol{y}(\boldsymbol{x_0}, \boldsymbol{u}(t), t) = \boldsymbol{y}(\boldsymbol{x_0}, \boldsymbol{u}(t-T), t-T). \tag{2.4}$$

Für die Darstellung im Zustandsraum lässt sich zeigen:

[1] Gemäß DIN 1338 sind Formeln, die Bestandteile eines Satzes sind, auch wenn sie frei stehen, hinsichtlich der Satzzeichen wie ein Satzteil zu behandeln. Einzige Ausnahme hiervon sind tabellarisch aufgelistete Formeln, wie sie z. B. innerhalb einer längeren mathematischen Umformungskette, auftreten.

[2] Die Umkehrung der Aussage trifft i. A. nicht zu, d. h. es gibt lineare Systeme, welche eine andere Form aufweisen. Beispielsweise sei hier $\dot{x}(t) = x_0$ genannt, denn die Lösung im Zeitbereich, $x(t) = (1+t)x_0$, ist homogen und additiv.

Satz 2.2: Linear-zeitinvariantes Modell im Zustandsraum

Das Modell (2.1) *ist genau dann linear und zeitinvariant, wenn es sich in die Form*

$$\begin{aligned} \dot{\boldsymbol{x}}(t) &= \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{B}\boldsymbol{u}(t)\,, \\ \boldsymbol{y}(t) &= \boldsymbol{C}\boldsymbol{x}(t) + \boldsymbol{D}\boldsymbol{u}(t) \end{aligned} \tag{2.5}$$

überführen lässt.

Reale, technische Systeme zeichnen sich demgegenüber häufig durch ein nichtlineares Verhalten aus. Sollen derartige Systeme durch lineare Modelle abgebildet werden, um z. B. einfache Methoden der linearen Regelungstechnik oder Systemidentifikation anwenden zu können, ist eine Linearisierung notwendig. Im einfachsten Fall findet diese Linearisierung um den jeweiligen Arbeitspunkt statt:

Satz 2.3: Linearisierung um den Arbeitspunkt

Sei $(\boldsymbol{x}_0, \boldsymbol{u}_0)$ *ein konstanter Arbeitspunkt (oder auch Ruhelage) des Systems* (2.1) *mit dem zugehörigen Ausgang* $\boldsymbol{y}_0$*. Die Abweichungen* $(\Delta\boldsymbol{x}(t), \Delta\boldsymbol{y}(t))$ *von* $(\boldsymbol{x}_0, \boldsymbol{y}_0)$ *bei hinreichend kleiner Änderung* $\Delta\boldsymbol{u}(t)$ *um* $\boldsymbol{u}_0$ *werden durch das lineare, zeitinvariante System*

$$\begin{aligned} \Delta\dot{\boldsymbol{x}}(t) &= \boldsymbol{A}\Delta\boldsymbol{x}(t) + \boldsymbol{B}\Delta\boldsymbol{u}(t)\,, \\ \Delta\boldsymbol{y}(t) &= \boldsymbol{C}\Delta\boldsymbol{x}(t) + \boldsymbol{D}\Delta\boldsymbol{u}(t) \end{aligned} \tag{2.6}$$

mit

$$\boldsymbol{A} = \left.\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}}\right|_{\substack{\boldsymbol{x}_0 \\ \boldsymbol{u}_0}}, \quad \boldsymbol{B} = \left.\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{u}}\right|_{\substack{\boldsymbol{x}_0 \\ \boldsymbol{u}_0}}, \quad \boldsymbol{C} = \left.\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{x}}\right|_{\substack{\boldsymbol{x}_0 \\ \boldsymbol{u}_0}}, \quad \boldsymbol{D} = \left.\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{u}}\right|_{\substack{\boldsymbol{x}_0 \\ \boldsymbol{u}_0}} \tag{2.7}$$

abgebildet.

Weiterhin kann die Linearisierung auch um die Lösungskurve (oder auch zulässige Trajektorie) von (2.1) gegeben durch $\boldsymbol{x}^*(t), \boldsymbol{u}^*(t), \boldsymbol{y}^*(t)$ erfolgen. In diesem Fall gilt:

Satz 2.4: Linearisierung um eine zulässige Trajektorie

Sei $(\boldsymbol{x}^*(t), \boldsymbol{u}^*(t))$ *eine zulässige Lösungskurve des Systems* (2.1) *mit dem zugehörigen Ausgang* $\boldsymbol{y}^*(t)$ *bei gegebenem Anfangswert* $\boldsymbol{x}_0$*. Die Abweichungen* $(\Delta\boldsymbol{x}(t), \Delta\boldsymbol{y}(t))$ *von* $(\boldsymbol{x}^*(t), \boldsymbol{y}^*(t))$ *bei hinreichend kleiner Änderung* $\Delta\boldsymbol{u}(t)$ *um* $\boldsymbol{u}^*(t)$ *werden durch das lineare System*

$$\begin{aligned} \Delta\dot{\boldsymbol{x}}(t) &= \boldsymbol{A}(t)\Delta\boldsymbol{x}(t) + \boldsymbol{B}(t)\Delta\boldsymbol{u}(t)\,, \\ \Delta\boldsymbol{y}(t) &= \boldsymbol{C}(t)\Delta\boldsymbol{x}(t) + \boldsymbol{D}(t)\Delta\boldsymbol{u}(t) \end{aligned} \tag{2.8}$$

mit

$$\boldsymbol{A}(t) = \left.\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}}\right|_{\substack{\boldsymbol{x}^*(t) \\ \boldsymbol{u}^*(t)}}, \quad \boldsymbol{B}(t) = \left.\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{u}}\right|_{\substack{\boldsymbol{x}^*(t) \\ \boldsymbol{u}^*(t)}}, \quad \boldsymbol{C}(t) = \left.\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{x}}\right|_{\substack{\boldsymbol{x}^*(t) \\ \boldsymbol{u}^*(t)}}, \quad \boldsymbol{D}(t) = \left.\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{u}}\right|_{\substack{\boldsymbol{x}^*(t) \\ \boldsymbol{u}^*(t)}} \tag{2.9}$$

abgebildet.

Ferner besteht die Möglichkeit lineare und zeitinvariante Systeme mittels *Übertragungsfunktionen* im Frequenz- bzw. Laplace-Bildbereich darzustellen. Hierzu werden die entsprechenden

Modellgleichungen mittels Laplace-Transformation in den Frequenzbereich überführt. Es wird zunächst angenommen, dass das System jeweils nur einen Ausgang $\boldsymbol{y}(t) = y(t)$ sowie eine Eingangsgröße $\boldsymbol{u}(t) = u(t)$ (SISO-System – *single-input single-output*) aufweise:

Definition 2.4: Laplace-Transformation

Für lineare, zeitinvariante und kausale[1] *Systeme bzw. Signale ist die Laplace-Transformation definiert als*

$$\begin{aligned} y(s) &= \mathcal{L}(y(t)) = \int_0^\infty y(t) e^{-st} \mathrm{d}t \,, \\ u(s) &= \mathcal{L}(u(t)) = \int_0^\infty u(t) e^{-st} \mathrm{d}t \end{aligned} \tag{2.10}$$

mit s als komplexer Frequenzparameter der Laplace-Transformierten.

Die Übertragungsfunktion stellt dann den Quotienten von Ausgangs- zur Eingangsgröße dar:

Satz 2.5: Übertragungsfunktion für SISO-Systeme

Für lineare, zeitinvariante und kausale SISO-Systeme wird die Übertragungsfunktion $G(s)$ unter der Annahme $\boldsymbol{x}_0 = 0$ wie folgt gebildet:

$$G(s) = \frac{y(s)}{u(s)} = \boldsymbol{c}^T \left(s\boldsymbol{I} - \boldsymbol{A}\right)^{-1} \boldsymbol{b} + d = \frac{b_n s^n + ... + b_1 s + b_0}{s^n + a_{n-1} s^{n-1} + ... + a_1 s + a_0} \tag{2.11}$$

mit $\boldsymbol{c}^T \in \mathbb{R}^{1\times n}$ als Ausgangsvektor, $\boldsymbol{b} \in \mathbb{R}^{n\times 1}$ als Eingangsvektor, $d \in \mathbb{R}$ als Durchgangsfaktor und $\boldsymbol{I}$ als Einheitsmatrix. Ferner sind

Gegenüber der Systembeschreibung im Zustandsraum findet bei der Übertragungsfunktion eine unmittelbare Kopplung zwischen Ein- und Ausgangsgrößen statt – Aussagen über die internen Zustände des Systems sind nicht mehr möglich. In diesem Kontext wird das Zustandsraummodell (2.5) auch als *Realisierung* von (2.11) bezeichnet. Für jede Übertragungsfunktion existiert daher eine unendliche Anzahl an Realisierungen im Zustandsraum.

[1] Alle technisch-physikalisch realisierbaren Systeme sind kausale Systeme. Dies bedeutet, dass der Ausgangswert eines Systems nur von dem aktuellen und den vergangenen Eingangswerten abhängt, jedoch nicht von zukünftigen Eingangswerten. Anschaulich ausgedrückt erfolgt eine Wirkung frühestens zum Zeitpunkt der Ursache, aber nicht früher. Folglich ist in Definition 2.4 die Verwendung der einseitigen Laplace-Transformation für positive Zeitwerte ausreichend.

Eine spezielle Realisierung des LTI-Modells gemäß (2.5) ist die sog. *Beobachtungsnormalform*:

$$\boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix}, \quad \boldsymbol{A} = \begin{bmatrix} 0 & 0 & \cdots & 0 & -a_0 \\ 1 & 0 & \ddots & 0 & -a_1 \\ 0 & 1 & \ddots & 0 & \vdots \\ \vdots & \ddots & \ddots & 0 & -a_{n-2} \\ 0 & 0 & \cdots & 1 & -a_{n-1} \end{bmatrix}, \quad \boldsymbol{b} = \begin{bmatrix} b_0 - b_n a_o \\ b_1 - b_n a_1 \\ \vdots \\ b_{n-2} - b_n a_{n-2} \\ b_{n-1} - b_n a_{n-1} \end{bmatrix}, \tag{2.12}$$

$$\boldsymbol{c}^T = \begin{bmatrix} 0 & 0 & \cdots & 0 & 1 \end{bmatrix}, \quad d = b_n\,.$$

Hierbei entspricht $\boldsymbol{x}$ einem Pseudo-Zustandsvektor, der i. A. keinerlei Rückschlüsse über die internen physikalisch-technischen Vorgänge eines Systems zulässt. Ein mittels (2.12) realisiertes System ist stets vollständig beobachtbar:

Definition 2.5: Beobachtbarkeit

Das System (2.5) heißt (vollständig) beobachtbar, wenn der Anfangszustand $\boldsymbol{x}_0$ aus Kenntnis der Ein- und Ausgangsgrößen $\boldsymbol{u}(t)$ bzw. $\boldsymbol{y}(t)$ auf einem Intervall $0 \leq t \leq T$ bestimmt werden kann.

Die Beobachtbarkeit (nach Kalman) kann anhand der Beobachtbarkeitsmatrix geprüft werden:

Satz 2.6: Beobachtbarkeit nach Kalman

Das System (2.5) ist genau dann vollständig beobachtbar, wenn die Beobachtbarkeitsmatrix $\boldsymbol{Q}_B$

$$\boldsymbol{Q}_B = \begin{bmatrix} \boldsymbol{c}^T \\ \boldsymbol{c}^T\boldsymbol{A} \\ \vdots \\ \boldsymbol{c}^T\boldsymbol{A}^{n-1} \end{bmatrix} \quad \text{bzw.} \quad \boldsymbol{Q}_B = \begin{bmatrix} \boldsymbol{C} \\ \boldsymbol{C}\boldsymbol{A} \\ \vdots \\ \boldsymbol{C}\boldsymbol{A}^{n-1} \end{bmatrix} \tag{2.13}$$

vollen Rang

$$\operatorname{rang}(\boldsymbol{Q}_B) = n \tag{2.14}$$

aufweist.

Alternativ zur Realisierung in der Beobachtungsnormalform kann der analoge Weg über die

sog. *Regelungsnormalform* beschritten werden:

$$\boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix}, \quad \boldsymbol{A} = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \ddots & 0 & \ddots & \vdots \\ 0 & 0 & \ddots & \ddots & 1 \\ -a_0 & -a_1 & -a_2 & \cdots & -a_{n-1} \end{bmatrix}, \quad \boldsymbol{b} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}, \tag{2.15}$$

$$\boldsymbol{c}^T = \begin{bmatrix} b_0 - b_n a_o & b_1 - b_n a_1 & \cdots & b_{n-2} - b_{bn} a_{n-2} & b_{n-1} - b_n a_{n-1} \end{bmatrix}, \quad d = b_n \,.$$

Ein mittels (2.15) realisiertes System ist stets vollständig steuerbar:

Definition 2.6: Steuerbarkeit

Das System (2.5) heißt (vollständig) steuerbar, wenn es in endlicher Zeit $0 \leq t \leq T$ von jedem Anfangszustand $\boldsymbol{x}_0$ durch einen zulässigen Stellgrößenverlauf $\boldsymbol{u}(t)$ in den Endzustand $\boldsymbol{x}(T)$ überführt werden kann.

Die Steuerbarkeit (nach Kalman) kann anhand der Steuerbarkeitsmatrix geprüft werden:

Satz 2.7: Steuerbarkeit nach Kalman

Das System (2.5) ist genau dann vollständig steuerbar, wenn die Steuerbarkeitsmatrix $\boldsymbol{Q}_S$

$$\boldsymbol{Q}_S = \begin{bmatrix} \boldsymbol{b} & \boldsymbol{Ab} & \ldots & \boldsymbol{A}^{n-1}\boldsymbol{b} \end{bmatrix} \quad \text{bzw.} \quad \boldsymbol{Q}_S = \begin{bmatrix} \boldsymbol{B} & \boldsymbol{AB} & \ldots & \boldsymbol{A}^{n-1}\boldsymbol{B} \end{bmatrix} \tag{2.16}$$

vollen Rang

$$\operatorname{rang}(\boldsymbol{Q}_S) = n \tag{2.17}$$

aufweist.

In diesem Kontext sei noch auf folgende Definition hingewiesen:

Definition 2.7: Minimale Realisierung

Das System $(\boldsymbol{A}, \boldsymbol{b}, \boldsymbol{c}^T, d)$ ist eine minimale Realisierung von $G(s)$, falls es keine andere Realisierung mit geringerer Dimension gibt. Dies bedeutet, dass das betrachtete Ein-/Ausgangsverhalten mit der minimalen Anzahl erforderlicher Systemzustände abgebildet wird.

Ob eine Realisierung minimal ist, kann mittels folgendem Satz geprüft werden:

Satz 2.8: Minimale Realisierung

Das System $(\boldsymbol{A}, \boldsymbol{b}, \boldsymbol{c}^T, d)$ ist eine minimale Realisierung von $G(s)$, falls es vollständig beobachtbar und steuerbar ist.

Im Fall von Systemen mit mehr als einer Eingangs- bzw. Ausgangsgröße, sog. MIMO-Systeme

(*multiple-input multiple-output*), kann die Beschreibung des Ein-/Ausgangsverhaltens ebenfalls im Frequenzbereich mittels Übertragungsfunktionen erfolgen:

$$\boldsymbol{y}(s) = \boldsymbol{G}(s)\boldsymbol{u}(s). \tag{2.18}$$

Hier ist $\boldsymbol{G}(s)$ die Übertragungsmatrix

$$\begin{bmatrix} y_1(s) \\ y_2(s) \\ \vdots \\ y_i(s) \end{bmatrix} = \begin{bmatrix} G_{11}(s) & G_{12}(s) & \dots & G_{1j}(s) \\ G_{21}(s) & G_{22}(s) & \dots & G_{2j}(s) \\ \vdots & \vdots & & \vdots \\ G_{i1}(s) & G_{i2}(s) & \dots & G_{ij}(s) \end{bmatrix} \begin{bmatrix} u_1(s) \\ u_2(s) \\ \vdots \\ u_j(s) \end{bmatrix}. \tag{2.19}$$

deren Elemente $G_{ij}(s)$ die Übertragungsfunktionen zwischen den verschiedenen Ausgangs- und Eingängsgrößen $(y_i(s), u_j(s))$ beinhalten. Die Übertragungsmatrix kann wie folgt berechnet werden:

Satz 2.9: Übertragungsmatrix für MIMO-Systeme

Für lineare, zeitinvariante und kausale MIMO-Systeme kann die Übertragungsmatrix

$$\boldsymbol{G}(s) = \boldsymbol{C}\left(s\boldsymbol{I} - \boldsymbol{A}\right)^{-1}\boldsymbol{B} + \boldsymbol{D} \tag{2.20}$$

unter der Annahme $\boldsymbol{x}_0 = 0$ unmittelbar aus den Matrizen der Zustandsraumdarstellung (2.5) *gewonnen werden.*

Das Auffinden einer Realisierung für MIMO-Systeme ist gegenüber der Regelungs- bzw. Beobachtungsnormalform für SISO-Systeme aufwendiger. Eine Möglichkeit besteht in der sog. *Gilbert-Realisierung*, hier sei aber auf die weitere Literatur verwiesen (z. B. [Lun16]).

2.2 Zeitdiskrete, deterministische und dynamische Modelle

Heutzutage sind die meisten Automatisierungs- und Regelungsprozesse digitalisiert, d. h. das insbesondere eine zeitlich diskret abgetastete Messwertefolge einem digitalen Regler zugeführt wird, der hieraus wiederum eine zeitdiskrete Steuerfolge berechnet, siehe hierzu Abb. 2.1. Dies bedingt, dass Regelungs- sowie Identifikationsverfahren auf einem diskreten Zeitraster mit Lauf-

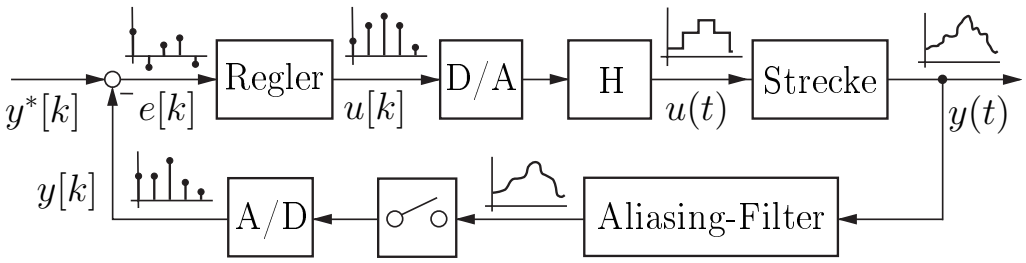


Abb. 2.1: Digitaler Regelkreis

index $k \in \{\mathbb{Z}|0 \leq k \leq \infty\}$ zu implementieren sind. Hierfür ist die zeitkontinuierliche Systembeschreibung (2.1) in einer Differenzengleichung zu überführen:

Definition 2.8: Zeitdiskretes, deterministisch-dynamisches Modell

Die zeitdiskrete Modellbeschreibung analog zu (2.1) *ist gegeben durch*

$$\begin{aligned} \boldsymbol{x}[k+1] &= \boldsymbol{f}_d\left(\boldsymbol{x}[k], \boldsymbol{x}[k-1], ..., \boldsymbol{u}[k], \boldsymbol{u}[k-1], ...\right) , \quad \boldsymbol{x}[k=0] = \boldsymbol{x}_0 , \\ \boldsymbol{y}[k] &= \boldsymbol{g}\left(\boldsymbol{x}[k], \boldsymbol{u}[k]\right) . \end{aligned} \tag{2.21}$$

Hierbei ist $\boldsymbol{f}_d(\,\cdot\,)$ eine zu berechnende, zeitdiskrete Version von $\boldsymbol{f}(\,\cdot\,)$. Da die Ausgangsfunktion $\boldsymbol{g}(\,\cdot\,)$ lediglich einen statischen Zusammenhang beschreibt, ist hier keine zeitliche Diskretisierung notwendig.

In (2.21) wird die Differenzengleichung in *expliziter Form* dargestellt, d. h. der Zustandsvektor $\boldsymbol{x}[k+1]$ ist nur von Vergangenheitswerten $\{\boldsymbol{x}[k], \boldsymbol{x}[k-1], ..., \boldsymbol{u}[k], \boldsymbol{u}[k-1], ...\}$ abhängig. Folglich kann die Differenzengleichung durch ein einfaches Vorwärts-Rechnen gelöst werden. Die einfachste Form der zeitlichen Diskretisierung von (2.1) erfolgt durch das explizite Euler-Verfahren (auch Euler-Vorwärts genannt):

Definition 2.9: Explizites Euler-Verfahren

Das Modell (2.1) *lässt sich für jeden diskreten Rechenschritt k durch folgenden Vorwärts-Differenzenquotienten approximieren und in eine Differenzengleichung der Form* (2.21) *überführen:*

$$\begin{aligned} \boldsymbol{x}[k+1] &= \boldsymbol{x}[k] + T_a \boldsymbol{f}\left(\boldsymbol{x}[k], \boldsymbol{u}[k]\right) , \\ \boldsymbol{y}[k] &= \boldsymbol{g}\left(\boldsymbol{x}[k], \boldsymbol{u}[k]\right) . \end{aligned} \tag{2.22}$$

Hier ist $T_a \in \{\mathbb{R}|0 < T_a\}$ die Abtastzeit zwischen k und $k+1$.

Gegenüber der allgemeinen Notation in (2.21) verwendet das explizite Euler-Verfahren jeweils nur den letzten Abtastschritt, es zählt daher zur Gruppe der numerischen *Einschrittverfahren*[1]. Die Verallgemeinerung von Definition 2.9 ist das *Runge-Kutta-Verfahren*. Dieses lautet in expliziter Form:

[1] Mehrschrittverfahren, wie die Adams-Bashforth-Methode, nutzen die Information aus den zuvor bereits errechneten Stützpunkten. Hierdurch steigt sowohl die numerische Genauigkeit als auch der Berechnungsaufwand. Für weitere Informationen hierzu sei auf die Literatur verwiesen (z. B. [SWP12]).

Definition 2.10: Explizites Runge-Kutta-Verfahren

Das Modell (2.1) *lässt sich für jeden diskreten Rechenschritt* k *in eine Differenzengleichung der Form* (2.21) *überführen:*

$$\begin{aligned} \boldsymbol{x}[k+1] &= \boldsymbol{x}[k] + T_a \sum_{j=1}^{s} b_j \boldsymbol{h}_j \,, \\ \boldsymbol{y}[k] &= \boldsymbol{g}\left(\boldsymbol{x}[k], \boldsymbol{u}[k]\right) \,, \\ \boldsymbol{h}_j &= \boldsymbol{f}\left(\boldsymbol{x}(t) + T_a \sum_{l=1}^{s} \alpha_{j,l} \cdot \boldsymbol{h}_l,\ \boldsymbol{u}(t + c_j T_a)\right) \,. \end{aligned} \tag{2.23}$$

Hierbei sind $\alpha_{j,l}$, b_j *und* c_j *die charakteristischen Koeffizienten des* s*-stufigen Runge-Kutta-Verfahrens, welche derart gewählt werden, dass die Fehlerordnung des Verfahrens maximal wird.*

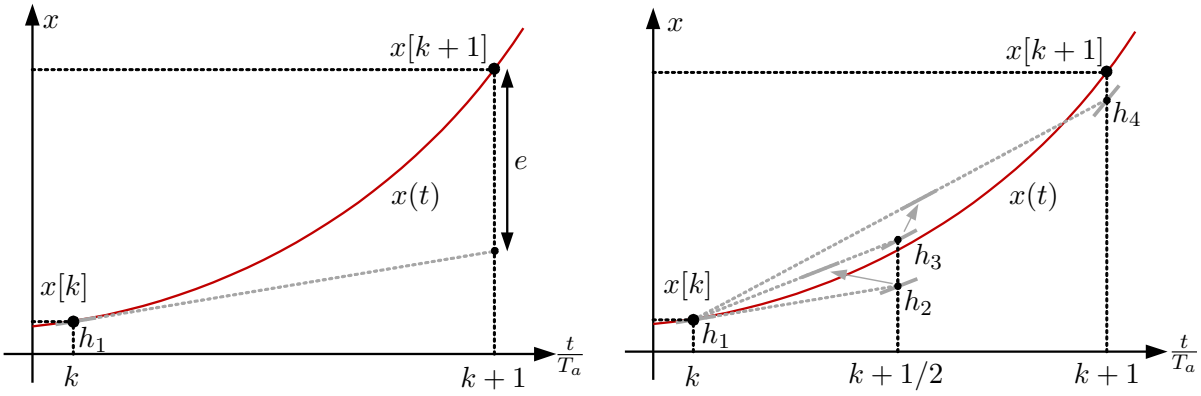
Je nach gewählter Stufe des Runge-Kutta-Verfahrens können die charakteristischen Koeffizienten der einschlägigen Literatur entnommen werden (z. B. [SWP12]). Als klassisch wird das Verfahren der 4. Stufe bezeichnet:

$$\begin{aligned} \boldsymbol{x}[k+1] &= \boldsymbol{x}[k] + \frac{T_a}{6}\left(\boldsymbol{h}_1 + 2\boldsymbol{h}_2 + 2\boldsymbol{h}_3 + \boldsymbol{h}_4\right) \,, \\ \boldsymbol{h}_1 &= \boldsymbol{f}(x(kT_a), \boldsymbol{u}(kT_a)) \,, \\ \boldsymbol{h}_2 &= \boldsymbol{f}(x(kT_a) + \frac{1}{2}T_a\boldsymbol{h}_1, \boldsymbol{u}(T_a(k+\frac{1}{2}))) \,, \\ \boldsymbol{h}_3 &= \boldsymbol{f}(x(kT_a) + \frac{1}{2}T_a\boldsymbol{h}_2, \boldsymbol{u}(T_a(k+\frac{1}{2}))) \,, \\ \boldsymbol{h}_4 &= \boldsymbol{f}(x(kT_a) + T_a\boldsymbol{h}_3, \boldsymbol{u}(T_a(k+1))) \,. \end{aligned} \tag{2.24}$$

Ein exemplarischer Vergleich des Runge-Kutta-Verfahrens 1. Stufe (explizites Euler-Verfahren) sowie 4. Stufe ist in Abb. 2.2 dargestellt. Es ist ersichtlich, dass dort grundsätzlich ein Kompromiss zwischen numerischer Genauigkeit der zeitdiskreten Approximation und der resultierenden Rechenlast zu treffen ist. Dieser Kompromiss ist stets anwendungsspezifisch in Abhängigkeit der jeweiligen Systemdynamik, Abtastzeit[1] und zur Verfügung stehenden Rechenkapazitäten zu bewerten. Auch sei darauf hingewiesen, dass eine ungünstige Wahl des Diskretisierungsverfahrens in Kombination mit der Abtastzeit zur numerischen Instabilität führen kann, sodass der Approximationsfehler divergiert – weitergehende Informationen zur numerischen Stabilität können u. a. [SWP12] entnommen werden.

Liegt ein LTI entsprechend (2.5) vor, kann dieses auch unmittelbar im Zustandsraum diskretisiert werden:

[1] Die Abtastzeit T_a ist bei Regelungsaufgaben häufig konstant und wird aufgrund der verwendeten Sensoren, Analog-zu-Digital-Konvertern sowie digitalen Rechnerplattform festgelegt. In Simulationsaufgaben ist die Abtastzeit hingegen i. A. flexibel und kann mittels einer Schrittweitensteuerung zur Laufzeit angepasst werden. So kombiniert der aus *MATLAB/Simulink* bekannte Algorithmus *ode45* ein Runge-Kutta-Verfahren vierter Stufe mit einer Schrittweitensteuerung fünfter Stufe.



(a) Explizites Euler-Verfahren (b) Klassisches Runge-Kutta-Verfahren

Abb. 2.2: Grundprinzip der expliziten Runge-Kutta-Diskretisierung (vgl. [JGD06])

Satz 2.10: Exakte Diskretisierung im Zustandsraum

Die exakte Diskretisierung linearer, zeit-invarianter Systeme entsprechend (2.5) *lautet*

$$\begin{aligned} \boldsymbol{x}[k+1] &= \boldsymbol{\Phi}\boldsymbol{x}[k] + \boldsymbol{H}\boldsymbol{u}[k]\,, \\ \boldsymbol{y}[k] &= \boldsymbol{C}\boldsymbol{x}[k] + \boldsymbol{D}\boldsymbol{u}[k] \end{aligned} \tag{2.25}$$

mit $\boldsymbol{\Phi} \in \mathbb{R}^{n\times n}$ *als zeitdiskrete Transitionsmatrix und* $\boldsymbol{H} \in \mathbb{R}^{n\times m}$ *als zeitdiskrete Eingangsmatrix:*

$$\begin{aligned} \boldsymbol{\Phi} &= e^{\boldsymbol{A}T_a} = \sum_{n=0}^{\infty} \frac{(\boldsymbol{A}T_a)^n}{n!}\,, \\ \boldsymbol{H} &= \int_0^{T_a} e^{\boldsymbol{A}\tau}\mathrm{d}\tau\boldsymbol{B} = \boldsymbol{A}^{-1}(e^{\boldsymbol{A}T_a} - \boldsymbol{I})\boldsymbol{B}\,. \end{aligned} \tag{2.26}$$

Für die Berechnung der zeitdiskreten Transitions- bzw. Eingangsmatrix ist ein Matrixexponential sowie ein darauf aufbauendes Integral zu berechnen, welches insbesondere für Systeme größerer Ordnung einen hohen Berechnungsaufwand nach sich zieht. Daher kann es sinnvoll sein, das Matrixexponential in (2.26) durch die entsprechende Taylor-Reihenentwicklung lediglich anzunähern. Für den Fall, dass diese nach dem 1. Glied abgebrochen wird, resultiert das explizite Euler-Verfahren im Zustandsraum:

$$\boldsymbol{\Phi} \approx \tilde{\boldsymbol{\Phi}} = \sum_{n=0}^{1} \frac{(\boldsymbol{A}T_a)^n}{n!} = \boldsymbol{I} + T_a\boldsymbol{A}, \qquad \boldsymbol{H} \approx \tilde{\boldsymbol{H}} = T_a\boldsymbol{B}. \tag{2.27}$$

Dieses lässt sich vergleichsweise einfach umsetzen, da sich die approximierte, zeitdiskrete Systemmatrix $\tilde{\boldsymbol{\Phi}}$ bzw. Eingangsmatrix $\tilde{\boldsymbol{H}}$ durch einfache Multiplikation und Addition aus den zeitkontinuierlichen Gegenstücken berechnen lassen. Das explizite Euler-Verfahren bietet sich daher auch zur Diskretisierung von linear-parametervarianten Systemen an:

Satz 2.11: Linear-parametervariantes Modell im Zustandsraum

Das Modell (2.1) *ist genau dann linear-parametervariant (linear parameter-varying – LPV), wenn es sich in die Form*

$$\begin{aligned} \dot{\boldsymbol{x}}(t) &= \boldsymbol{A}(\boldsymbol{\theta})\boldsymbol{x}(t) + \boldsymbol{B}(\boldsymbol{\theta})\boldsymbol{u}(t)\,, \\ \boldsymbol{y}(t) &= \boldsymbol{C}(\boldsymbol{\theta})\boldsymbol{x}(t) + \boldsymbol{D}(\boldsymbol{\theta})\boldsymbol{u}(t) \end{aligned} \tag{2.28}$$

überführen lässt. Hier ist $\boldsymbol{\theta} \in \mathbb{R}^{p\times 1}$ *ein Parametervektor.*

Werden Variationen in $\boldsymbol{\theta}$ zur Laufzeit, z. B. mittels Beobachterverfahren oder Messung, detektiert und nachgeführt, ist eine fortschreitende Adaption von (2.27) auch auf Rechenplattformen mit begrenzter Leistungsfähigkeit möglich[1]:

$$\tilde{\boldsymbol{\Phi}}(\boldsymbol{\theta}[k]) = \sum_{n=0}^{1} \frac{(\boldsymbol{A}(\boldsymbol{\theta}[k])T_a)^n}{n!} = \boldsymbol{I} + T_a\boldsymbol{A}(\boldsymbol{\theta}[k]), \quad \tilde{\boldsymbol{H}}(\boldsymbol{\theta}[k]) = T_a\boldsymbol{B}(\boldsymbol{\theta}[k]). \tag{2.29}$$

Zur Abgrenzung der bisher ausschließlich explizit erfolgten Diskretisierung, sei der Vollständigkeit halber noch die *implizite Diskretisierung* vorgestellt:

$$\begin{aligned} \boldsymbol{x}[k+1] &= \boldsymbol{f}_d\left(\boldsymbol{x}[k+1], \boldsymbol{x}[k], ..., \boldsymbol{u}[k], \boldsymbol{u}[k-1], ...\right)\,, \quad \boldsymbol{x}[k=0] = \boldsymbol{x}_0\,, \\ \boldsymbol{y}[k] &= \boldsymbol{g}\left(\boldsymbol{x}[k], \boldsymbol{u}[k]\right)\,. \end{aligned} \tag{2.30}$$

Gegenüber der expliziten Diskretisierung muss hier in jedem Abtastschritt eine algebraische, ggf. nichtlineare, Gleichung gelöst werden, da der gesuchte Zustandsvektor $\boldsymbol{x}[k+1]$ nun auf beiden Seiten des Gleichheitszeichen steht. Dies kann beispielsweise numerisch mittels Newton-Raphson-Verfahren oder Levenberg-Marquardt-Algorithmus erfolgen. Verglichen mit der einfachen Vorwärtsrechnung der expliziten Verfahren besteht somit ein erhöhter Rechenbedarf – auf der anderen Seite kann gezeigt werden, dass implizite Diskretisierungverfahren *A-stabil* sind, d. h. ihr numerisches Stabilitätsgebiet enthält also die komplette linke Halbebene der komplexen Zahlenebene [SWP12]. Es gibt somit für implizite Verfahren keine Einschränkungen an das Abtastintervall aufgrund von Stabilitätseinschränkungen. Der bekannteste Vertreter der impliziten Verfahren ist erneut das Euler-Verfahren (auch Rückwärts-Euler-Verfahren genannt):

Definition 2.11: Implizites Euler-Verfahren

Das Modell (2.1) *lässt sich für jeden diskreten Rechenschritt* k *durch folgenden Rückwärts-Differenzenquotient approximieren und in eine Differenzengleichung der Form* (2.30) *überführen:*

$$\begin{aligned} \boldsymbol{x}[k+1] &= \boldsymbol{x}[k] + T_a\boldsymbol{f}\left(\boldsymbol{x}[k+1], \boldsymbol{u}[k]\right)\,, \\ \boldsymbol{y}[k] &= \boldsymbol{g}\left(\boldsymbol{x}[k], \boldsymbol{u}[k]\right)\,. \end{aligned} \tag{2.31}$$

Auch können abgetastete, zeitdiskrete Systeme bzw. Signale in den Frequenzbereich überführt werden. Hierfür kann die *z-Transformation* herangezogen werden:

[1] Annahme: Die zeitliche Parametervariation ist deutlich langsamer als die Systemdynamik bzw. als die Abtastzeit.

Definition 2.12: z-Transformation

Für lineare, kausale und abgetastete Systeme bzw. Signale ist die z-Transformation definiert als

$$\begin{aligned} y(z) &= \mathcal{Z}(y[k]) = \sum_{k=0}^{\infty} y[k] z^{-k}, \\ u(z) &= \mathcal{Z}(u[k]) = \sum_{k=0}^{\infty} u[k] z^{-k} \end{aligned} \tag{2.32}$$

mit z als komplexer Frequenzparameter der z-Transformierten.

Ist das betrachtete System zudem zeitinvariant, kann die z-Übertragungsfunktion zur Beschreibung des zeitdiskreten Ein-/Ausgangsverhaltens gebildet werden. Für den SISO-Fall folgt:

Satz 2.12: z-Übertragungsfunktion für SISO-Systeme

Für lineare, zeitinvariante, kausale und abgetastete SISO-Systeme entsprechend (2.25) wird die Übertragungsfunktion $G(z)$ unter der Annahme $\boldsymbol{x}_0 = 0$ wie folgt gebildet:

$$G(z) = \frac{y(z)}{u(z)} = \boldsymbol{c}^T \left(z\boldsymbol{I} - \boldsymbol{\Phi}\right)^{-1} \boldsymbol{h} + d = \frac{b_n z^n + \ldots + b_1 z + b_0}{z^n + a_{n-1} z^{n-1} + \ldots + a_1 z + a_0}. \tag{2.33}$$

Hier ist $\boldsymbol{h} \in \mathbb{R}^{n\times 1}$ der zeitdiskrete Eingangsvektor, $a_i \in \mathbb{R}$ und $b_i \in \mathbb{R}$ sind konstante Koeffizienten.

Durch Anwendung des Rechts-Verschiebesatzes der z-Transformation

$$f[k-d] \quad \circ\!\!-\!\!\bullet \quad z^{-d}\mathcal{Z}(f[k])$$

auf (2.33) kann dann unmittelbar auf die korrespondierende Differenzengleichung geschlossen werden:

$$\begin{aligned} y[k] = &- a_{n-1} y[k-1] - \ldots - a_1 y[k-n+1] - a_0 y[k-n] \\ &+ b_n u[k] + \ldots + b_1 u[k-n+1] + b_0 u[k-n]. \end{aligned} \tag{2.34}$$

Aus dieser Umschreibung wird bereits deutlich, dass aus der zeitdiskreten Übertragungsfunktion durch weitere Umformungen ebenfalls eine zeitdiskrete Realisierung im Zustandsraum gefunden werden kann. Die in Kap. 2.1 getätigten Aussagen hinsichtlich der (minimalen) Realisierung, Beobachtbarkeit sowie Steuerbarkeit bzw. Zustandsraumdarstellung in der Regelungs- sowie Beobachtungsnormalform gelten für zeitdiskrete Systeme vollständig analog – diese werden daher hier nicht erneut ausgeführt. Wer eine explizite Darstellung wünscht, sei z. B. auf [Lun16] verwiesen.

Auch im Fall von zeitdiskreten MIMO-Systemen kann die Beschreibung des Ein-/Ausgangsverhaltens ebenfalls im Frequenzbereich mittels Übertragungsfunktionen erfolgen:

$$\boldsymbol{y}(z) = \boldsymbol{G}(z)\boldsymbol{u}(z). \tag{2.35}$$

Hier ist $\boldsymbol{G}(z)$

$$\begin{bmatrix} y_1(z) \\ y_2(z) \\ \vdots \\ y_i(z) \end{bmatrix} = \begin{bmatrix} G_{11}(z) & G_{12}(z) & \dots & G_{1j}(z) \\ G_{21}(z) & G_{22}(z) & \dots & G_{2j}(z) \\ \vdots & \vdots & & \vdots \\ G_{i1}(z) & G_{i2}(z) & \dots & G_{ij}(z) \end{bmatrix} \begin{bmatrix} u_1(z) \\ u_2(z) \\ \vdots \\ u_j(z) \end{bmatrix} \tag{2.36}$$

die zeitdiskrete Übertragungsmatrix, deren Elemente $G_{ij}(z)$ die Übertragungsfunktionen zwischen den verschiedenen Ausgangs- und Eingängsgrößen $(y_i(z), u_j(z))$ beinhalten. Die zeitdiskrete Übertragungsmatrix kann wie folgt berechnet werden:

Satz 2.13: Zeitdiskrete Übertragungsmatrix für MIMO-Systeme

Für lineare, zeitinvariante, kausale und abgetastete MIMO-Systeme entsprechend (2.25) *kann die zeitdiskrete Übertragungsmatrix*

$$\boldsymbol{G}(z) = \boldsymbol{C}\left(z\boldsymbol{I} - \boldsymbol{\Phi}\right)^{-1}\boldsymbol{H} + \boldsymbol{D} \tag{2.37}$$

unter der Annahme $\boldsymbol{x}_0 = 0$ unmittelbar aus den Matrizen der Zustandsraumdarstellung gewonnen werden.

2.3 Eine kleine Einführung in die Stochastik

Die Abbildung stochastischer Prozesse spielt in der ingenieurwissenschaftlichen Systemidentifikation eine ebenfalls große Rolle z. B. zur Beschreibung von Mess- oder Modellunsicherheiten. Der Verlauf eines stochastischen Signals ist zufällig und kann daher nicht genau charakterisiert werden. Mit Hilfe stochastischer Methoden lassen sich jedoch die Eigenschaften dieser stochastischen Signale beschreiben. Messbare stochastische Signale sind in der Regel nicht völlig zufällig, sondern folgen einigen internen Zusammenhängen, die in mathematische Signalmodelle überführt werden können. Hierzu sei folgende Definition von Wahrscheinlichkeiten bzw. allgemeiner eines Wahrscheinlichkeitsraum erwähnt:

Definition 2.13: Wahrscheinlichkeitsraum

Ein Wahrscheinlichkeitsraum ist ein Maßraum $(\Omega, \mathcal{F}, P)$, dessen Maß P ein Wahrscheinlichkeitsmaß ist und somit ein mathematisches Modell eines realen Prozesses (Experiment) darstellt. Hierbei sind:

- Ω: *Die Ergebnismenge, welche alle möglichen Resultate ω des betrachteten Experiments umfasst.*
- $\mathcal{F}$: *Das Ereignissystem, welches alle möglichen Kombinationen der Resultate aus der Ergebnismenge umfasst.*
- P: *Die Wahrscheinlichkeit, welche allen Ereignissen aus $\mathcal{F}$ eine reelle Größe im Intervall $[0, 1]$ zuordnet ($P(\omega) : \mathcal{F} \to [0, 1]$ mit $P(\Omega) = 1$).*

Die Begrifflichkeiten aus Definition 2.13 werden in Abb. 2.3 zur besseren Anschauung illustriert. Im Folgenden werden die wichtigsten Begriffe und Definitionen der Stochastik vorgestellt.

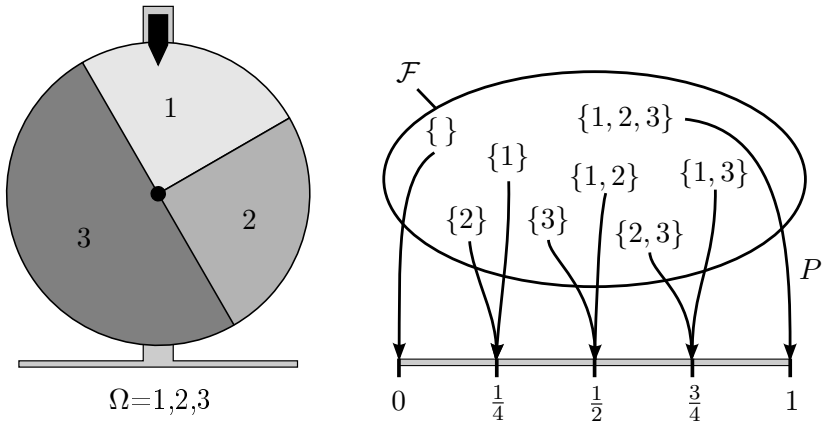
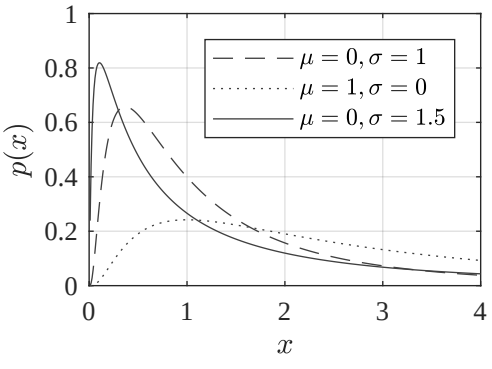


Abb. 2.3: Illustration des Wahrscheinlichkeitsraums anhand eines Glücksrads (vgl. [Wik18d])

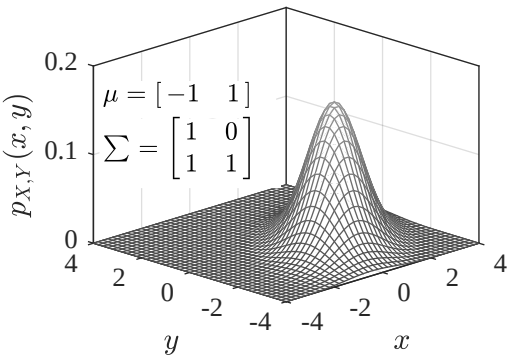
2.3.1 Stetige Zufallsvariablen

Eine Zufallsvariable X bildet den Wahrscheinlichkeitsraum $(\Omega, \mathcal{F}, P)$ eines stochastischen Experiments auf einen quantitativen Wert ab, typischerweise aus den reellen Zahlen. Ein spezifischer Wert $x \in \mathbb{R}$ der Zufallsvariable X wird typischerweise mit einem Kleinbuchstaben bezeichnet, während die Zufallsvariable selbst durch einen Großbuchstaben dargestellt wird. Die Zufallsvariable X selbst ist keine reelle Zahl, sondern nimmt nur die Werte in $\mathbb{R}$ an: $X(\omega) : \Omega \to \mathbb{R}$ bzw. $\boldsymbol{X}(\boldsymbol{\omega}) : \boldsymbol{\Omega} \to \mathbb{R}^n$. Der Einfachheit halber wird im Folgenden $X \in \mathbb{R}$ bzw. $\boldsymbol{X} \in \mathbb{R}^n$ verwendetet, um anzuzeigen, dass die Realisierung der Zufallsvariable skalare oder vektorielle Werte annimmt. Die Wahrscheinlichkeit, dass ein bestimmtes Ereignis A eintritt, wird durch $P(A)$ bezeichnet und $P(A)$ ist eine reelle Zahl im Intervall $[0, 1]$. Das Ereignis A ist typischerweise durch eine Bedingung definiert, die die Zufallsvariable erfüllt. Beispielsweise wird die Wahrscheinlichkeit, dass der Wert einer Zufallsvariablen X größer als eine Konstante a ist, durch $P(X > a)$ angegeben.

Ferner wird der Begriff der *Wahrscheinlichkeitsdichtefunktion* benötigt, welche in Abb. 2.4 anhand zweier Beispiele illustriert wird:



(a) Logarithmische Normalverteilungen $\mathcal{LN}(\mu, \sigma)$ mit Parametern $\mu \in \mathbb{R}$ und $\sigma \in \{\mathbb{R} | \sigma \geq 0\}$



(b) Bivariate Normalverteilung $\mathcal{N}_2(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ mit Erwartungsvektor $\boldsymbol{\mu}$ und Kovarianzmatrix $\boldsymbol{\Sigma}$

Abb. 2.4: Beispielhafte uni- und bivariate Wahrscheinlichkeitsdichtefunktionen

Definition 2.14: Wahrscheinlichkeitsdichtefunktion

Gegeben sei eine Wahrscheinlichkeitsverteilung P sowie eine reellwertige Zufallsvariable $\boldsymbol{X} \in \mathbb{R}^n$. Existiert eine reelle Funktion $p : \mathbb{R}^n \to [0, \infty)$, sodass für das n-dimensionale Intervall $I = [a_1, b_1] \times \cdots \times [a_n, b_n]$

$$P(\boldsymbol{X} \in I) = \int_{a_n}^{b_n} \cdots \int_{a_1}^{b_1} p_{\boldsymbol{X}}(x_1, \ldots, x_n) \mathrm{d}x_1 \cdots \mathrm{d}x_n \tag{2.38}$$

gilt, so heißt p die Wahrscheinlichkeitsdichtefunktion von P bzw. $\boldsymbol{X}$.

Für eindimensionale Zufallsvariablen folgt die Vereinfachung:

$$P(a \leq X \leq b) = \int_a^b p(x) \mathrm{d}x. \tag{2.39}$$

Zwei Zufallsvariablen X, Y werden *unabhängig* genannt, falls die gemeinsame Dichtefunktion dem Produkt der individuellen Dichtefunktionen entspricht:

$$p_{X,Y}(x, y) = p_X(x) p_Y(y) \tag{2.40}$$

mit $p_{X_i}(x_i)$ als der sogenannten *Randdichte*:

$$p_{X_i}(x_i) = \int_{a_n}^{b_n} \cdots \int_{a_{i+1}}^{b_{i+1}} \int_{a_{i-1}}^{b_{i-1}} \cdots \int_{a_1}^{b_1} p_{\boldsymbol{X}}(x_1, \ldots, x_{i-1}, x_i, x_{i+1}, \ldots, x_n) \mathrm{d}x_1 \cdots \mathrm{d}x_{i-1} \mathrm{d}x_{i+1} \cdots \mathrm{d}x_n.$$

Der *Erwartungswert* (oder auch Schwerpunkt genannt) ist ein weiterer Grundbegriff der Stochastik. Der Erwartungswert einer Zufallsvariablen beschreibt den Wert, den die Zufallsvariable im Mittel annimmt. Er ergibt sich bei unbegrenzter Wiederholung des zugrunde liegenden Experiments als Durchschnitt der Ergebnisse. Dieser ist folgendermaßen definiert:

Definition 2.15: Erwartungswert

Gegeben sei eine gemeinsame Wahrscheinlichkeitsdichtefunktion $p_{\boldsymbol{X}}(\boldsymbol{x})$ des Zufallsvektors $\boldsymbol{X} \in \mathbb{R}^n$. Der Erwartungswert berechnet sich dann zu

$$\mathrm{E}(\boldsymbol{X}) = \boldsymbol{\mu}(\boldsymbol{X}) = \int_{\mathbb{R}^n} \boldsymbol{x}\, p_{\boldsymbol{X}}(\boldsymbol{x}) \mathrm{d}^n \boldsymbol{x} = \Big[\mathrm{E}(X_1), \ldots, \mathrm{E}(X_n)\Big] \tag{2.41}$$

mit

$$\mathrm{E}(X_i) = \mu(X_i) = \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} x_i\, p_{\boldsymbol{X}}(\boldsymbol{x}) \mathrm{d}x_1 \cdots \mathrm{d}x_n \tag{2.42}$$

als Erwartungswert von X_i.

Liegt lediglich eine skalare Wahrscheinlichkeitsdichtefunktion $p_X(x)$ vor, vereinfacht sich der

Ausdruck zu:

$$\mathrm{E}(X) = \mu(X) = \int_{-\infty}^{\infty} x\, p_X(x)\mathrm{d}x. \tag{2.43}$$

Abgeleitet von obigen Basisdefinitionen können verschiedene stochastische *Momente* definiert werden, welche die Zufallsvariable bezüglich ihrer wesentlichen Eigenschaften beschreiben[1]. Klassischerweise werden die *zentralen* Momente hierbei um den jeweiligen Mittelwert $\boldsymbol{\mu}$ zentriert. Das erste zentrale (absolute) Moment ist:

Definition 2.16: Mittlere absolute Abweichung

Gegeben sei eine Wahrscheinlichkeitsdichtefunktion $p_{\boldsymbol{X}}(\boldsymbol{x})$ mit Erwartungswertvektor $\boldsymbol{\mu}$ der Zufallsvariable $\boldsymbol{X} \in \mathbb{R}^n$. Die mittlere absolute Abweichung (MAE – mean absolute error) ist dann gegeben durch:

$$\mathrm{MAE}(\boldsymbol{X}) = \mathrm{E}(|\boldsymbol{X} - \boldsymbol{\mu}|) = \int_{\mathbb{R}^n} |(\boldsymbol{x} - \boldsymbol{\mu})\, p_{\boldsymbol{X}}(\boldsymbol{x})|\, \mathrm{d}^n\boldsymbol{x}. \tag{2.44}$$

Weiterhin ist die *Varianz* als zweites zentrales Moment zu nennen. Diese ist ein Maß für die Streuung der Wahrscheinlichkeitsdichte um ihren Erwartungswert und sei zunächst für eine skalare Zufallsvariable X definiert als:

Definition 2.17: Varianz

Gegeben sei eine Wahrscheinlichkeitsdichtefunktion $p_X(x)$ der Zufallsvariable $X \in \mathbb{R}$. Die Varianz von X ergibt sich dann als Erwartungswert der Zufallsvariablen $\widetilde{X} = (X - \mu)^2$:

$$\mathrm{Var}(X) = \mathrm{E}((X - \mu)^2) = \sigma^2(X) = \int_{-\infty}^{\infty} (x - \mu)^2\, p_X(x)\mathrm{d}x. \tag{2.45}$$

Im Falle eines reellen Zufallsvektors $\boldsymbol{X} = [X_1, \ldots, X_n]^T$ mit dem dazugehörigen Erwartungswertvektor $\boldsymbol{\mu} = [\mu_1, \ldots, \mu_n]^T$ verallgemeinert sich die Varianz beziehungsweise Kovarianz zu der symmetrischen Varianz-Kovarianz-Matrix (oder einfach *Kovarianzmatrix*) des Zufallsvektors:

[1] Im Zuge dieser Kurzzusammenfassung der Grundlagen der Stochastik werden nur die ersten beiden zentralen Momente behandelt, da diese für die nachfolgenden Vorlesungsinhalte von besonderer Wichtigkeit sind. Für die weiteren Momente wie *Schiefe* oder *Wölbung* sei auf die weiterführende Literatur verwiesen.

Definition 2.18: Kovarianz

Gegeben sei eine gemeinsame Wahrscheinlichkeitsdichtefunktion $p_{\boldsymbol{X}}(\boldsymbol{x})$ des Zufallsvektors $\boldsymbol{X} \in \mathbb{R}^n$ mit dem Erwartungswertvektor $\boldsymbol{\mu} \in \mathbb{R}^n$. Eine Kovarianzmatrix des Zufallsvektors $\boldsymbol{X}$ lässt sich wie folgt definieren:

$$\begin{aligned}\mathrm{Cov}(\boldsymbol{X}) = \Sigma(\boldsymbol{X}) &= \mathrm{E}\left((\boldsymbol{X}-\boldsymbol{\mu})(\boldsymbol{X}-\boldsymbol{\mu})^T\right) \\ &= \begin{bmatrix} \mathrm{Var}(X_1) & \mathrm{Cov}(X_1,X_2) & \cdots & \mathrm{Cov}(X_1,X_n) \\ \mathrm{Cov}(X_2,X_1) & \mathrm{Var}(X_2) & \cdots & \mathrm{Cov}(X_2,X_n) \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{Cov}(X_n,X_1) & \mathrm{Cov}(X_n,X_2) & \cdots & \mathrm{Var}(X_n) \end{bmatrix} = \begin{bmatrix} \sigma_1^2 & \sigma_{12}^2 & \cdots & \sigma_{1n}^2 \\ \sigma_{21}^2 & \sigma_2^2 & \cdots & \sigma_{2n}^2 \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{n1}^2 & \sigma_{n2}^2 & \cdots & \sigma_n^2 \end{bmatrix}\end{aligned} \tag{2.46}$$

mit der Kovarianz zweier Zufallsvariablen (X_i, X_j)

$$\begin{aligned}\mathrm{Cov}(X_i,X_j) &= \mathrm{E}\left[(X_i - \mathrm{E}(X_i)(X_j - \mathrm{E}(X_j))\right] \\ &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} (x_i - \mathrm{E}(X_i))(x_j - \mathrm{E}(X_j))\, p_{X_i,X_j}(x_i,x_j)\mathrm{d}x_i\mathrm{d}x_j.\end{aligned} \tag{2.47}$$

Gilt für zwei Zufallsvariablen

$$\mathrm{Cov}(X_i,X_j) = 0,$$

so heißen diese *unkorreliert*[1]. Die Kovarianz ist ein nichtstandardisiertes Zusammenhangsmaß von Zufallsvariablen mit gemeinsamer Wahrscheinlichkeitsverteilung. Der Wert dieser Kenngröße macht tendenzielle Aussagen darüber, ob hohe Werte der einen Zufallsvariablen eher mit hohen oder eher mit niedrigen Werten der anderen Zufallsvariablen einhergehen. Die Kovarianz gibt zwar somit die Richtung einer Beziehung zwischen zwei Zufallsvariablen an, über die Stärke des Zusammenhangs wird aber keine Aussage getroffen. Hierfür muss die Kovarianz normiert werden z. B. in Form des Korrelationskoeffizienten.

Definition 2.19: Korrelationskoeffizient

Gegeben seien zwei Zufallsvariablen $(X_i, X_j) \in \mathbb{R}$. Der Korrelationskoeffizient von X_i und $X_j$ist definiert als

$$\rho(X_i,X_j) = \rho_{ij} = \frac{\mathrm{Cov}(X_i,X_j)}{\sqrt{\mathrm{Var}(X_i)}\sqrt{\mathrm{Var}(X_j)}}^2. \tag{2.48}$$

[1] Dies ist vom Begriff der *statistischen Unabhängigkeit* zu unterscheiden, siehe hierzu (2.40). Unabhängigkeit ist eine stärkere Eigenschaft als Unkorreliertheit, d. h. zwei unabhängige Zufallsvariablen sind immer unkorreliert, während die Umkehrung aber nicht gilt.

[2] Hier ist $\sqrt{\mathrm{Var}(X_i)} = \sigma(X_i)$ die sog. *Standardabweichung*. Dieses Maß wird in der Literatur auch gerne genutzt, da die jeweilige Einheit der Zufallsvariable nicht quadriert wird und so ein gewisse, bessere Anschaulichkeit gegeben ist.

Die Korrelationskoeffizienten können zudem in der Korrelationsmatrix

$$\begin{aligned}\mathrm{Corr}(\boldsymbol{X}) &= \mathrm{diag}\left(\mathrm{Cov}(\boldsymbol{X})\right)^{-\frac{1}{2}} \mathrm{Cov}(\boldsymbol{X}) \mathrm{diag}\left(\mathrm{Cov}(\boldsymbol{X})\right)^{-\frac{1}{2}} \\ &= \begin{bmatrix} 1 & \rho_{12} & \cdots & \rho_{1n} \\ \rho_{21} & 1 & \cdots & \rho_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \rho_{n1} & \rho_{n2} & \cdots & 1 \end{bmatrix}\end{aligned} \tag{2.49}$$

zusammengefasst werden. Der Korrelationskoeffizient kann Werte im Intervall $[-1, 1]$ annehmen. Die Grenzwerte des Intervalls repräsentieren einen vollständig negativen bzw. positiven linearen Zusammenhang zwischen den betrachteten Merkmalen. Wenn der Korrelationskoeffizient den Wert 0 aufweist, hängen die beiden Zufallsvariablen überhaupt nicht linear voneinander ab. Allerdings können diese ungeachtet dessen in nichtlinearer Weise voneinander abhängen.

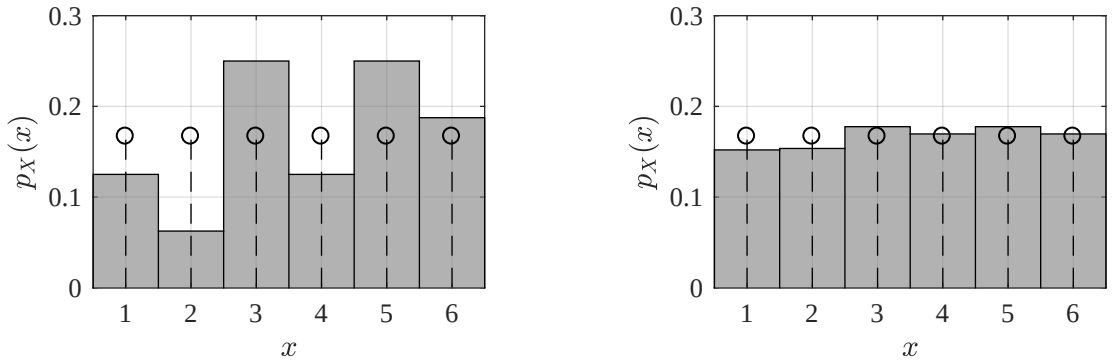
2.3.2 Diskrete Zufallsvariablen

Neben den zuvor behandelten stetigen Zufallsvariablen spielen *diskrete Zufallsvariablen* in ingenieurtechnischen Anwendungen eine wichtige Rolle:

Definition 2.20: Diskrete Zufallsvariable

Eine Zufallsvariable X heißt diskret, falls sie nur endlich oder abzählbar unendlich viele Werte $\mathcal{T} = \{x_1, x_2, \ldots\}$ annehmen kann. Die Menge $\mathcal{T}$ der möglichen Ausprägungen von X, also alle $x_i \in \mathbb{R}$ mit $p_X(x_i) > 0$, heißt der Träger von X.

Typische Beispiele sind die Anzahl der Messwerte eines Sensors, welche in einem gegebenen Werteintervall liegen oder auch die Menge der Schaltzustände in elektronischen Schaltungen. Auch für diskrete Zufallsvariablen kann eine Dichtefunktion definiert werden:



(a) Stichprobenanzahl $N = 16$ (b) Stichprobenanzahl $N = 625$

Abb. 2.5: Vergleich von theoretischer und empirischer Verteilung einer diskreten Zufallsvariablen zum klassischen Würfel-Experiment (Grau: Empirisches Histogramm mit N Stichproben, Schwarz: Theoretische Gleichverteilung)

Definition 2.21: Diskrete Wahrscheinlichkeitsdichtefunktion

Gegeben sei eine diskrete Wahrscheinlichkeitsverteilung P sowie eine diskrete und reellwertige Zufallsvariable $\boldsymbol{X} \in \mathbb{R}^n$ mit dem Träger $\boldsymbol{\mathcal{T}}$. Existiert eine reelle Funktion $p : \mathbb{R}^n \to [0, \infty)$, sodass für das n-dimensionale Intervall $I = [a_1, b_1] \times \cdots \times [a_n, b_n]$

$$P(\boldsymbol{X} \in I) = \sum_{a_n}^{b_n} \cdots \sum_{a_1}^{b_1} p_{\boldsymbol{X}}(x_1, \ldots, x_n) \tag{2.50}$$

gilt, so heißt p die diskrete Wahrscheinlichkeitsdichtefunktion von P bzw. $\boldsymbol{X}$.

Hierbei gilt es zwischen *empirischen Verteilungen* (Häufigkeitsverteilungen) und den *theoretischen Verteilungen* von diskreten Zufallsvariablen (Wahrscheinlichkeitsverteilungen) zu differenzieren: Empirische Verteilungen basieren auf Daten, während theoretische Verteilungen Modelle sind, mit denen man die Realität näherungsweise abzubilden versucht. Empirische Verteilungen bzw. Daten können daher herangezogen werden, um mittels entsprechender Schätzer auf die zugrundeliegende stetige oder diskrete Verteilung zu schließen. Eine sehr einfache Möglichkeit stellt hierbei das *Histogramm* dar, ein potenteres Verfahren der sog. *Kerndichteschätzer*.

Obige Problematik wird am einfachen Würfel-Experiment in Abb. 2.5 demonstriert: Theoretisch liegt hier eine diskrete Gleichverteilung vor, wobei jedes Würfelergebnis eine Eintrittswahrscheinlichkeit von 1/6 besitzt. Bei einer empirischen Stichprobe zu diesem Experiment mit begrenzter Stichprobenanzahl N kommt es zu einer Verzerrung der empirischen Häufigkeitsverteilung im Vergleich zum erwarteten theoretischen Ergebnis. Mit steigendem Stichprobenumfang nähert sich allerdings die empirische Verteilung der theoretischen an. Hierzu sei auf folgenden Satz verwiesen:

Satz 2.14: Konvergenzeigenschaft von empirischen Verteilungen

Sei $P(\boldsymbol{X})$ eine diskrete Verteilung und $P_n(\boldsymbol{X})$ eine empirische Stichprobe dieser mit N Beobachtungen. Das starke Gesetz der großen Zahlen sichert in diesem Fall zu, dass die empirische Stichprobe $P_n(\boldsymbol{X})$ fast sicher für jeden Zufallsvektor $\boldsymbol{X}$ gegen die wahre Verteilungsfunktion $P(\boldsymbol{X})$ konvergiert:

$$P_n(\boldsymbol{X}) \xrightarrow{N \to \infty} P(\boldsymbol{X}). \tag{2.51}$$

Somit ist festzuhalten, dass i. A. für die Rekonstruktion von Verteilungsfunktionen aus empirischen Daten auf einen ausreichend großen Stichprobenumfang zu achten ist. Die obigen Definitionen hinsichtlich zu Randverteilungen, Erwartungswert, Varianz, etc. für stetige Zufallsvariablen lassen sich analog auf diskrete Zufallsvariablen transferieren, indem die Integrationsvorgänge gegen entsprechende Summationen ausgetauscht werden. Für weitere Details sei auf die entsprechende Fachliteratur zur Stochastik verwiesen (z. B. [Hen13]).

2.3.3 Stochastische Prozesse

Ein stochastischer Prozess ist ein mathematisches Modell für einen realen Vorgang, der zufällig ist und von einem Parameter abhängt. Typische Beispiele sind die Populationsentwicklung in der Biologie oder Aktienkurse. Meist ist der besagte Parameter die Zeit, welche im Folgenden

ausschließlich betrachtet wird[1]. Bei einer Zufallsvariablen X wurde bisher jedem Ergebnis ω eines gegebenen Zufallsexperiments eine reelle Zahl über die Abbildung $X(\omega) : \Omega \to \mathbb{R}$ zugeordnet. Ein stochastisches Signal bzw. ein stochastischer Prozess $x(t)$ wird analog definiert, wobei jedem Ergebnis ω ein Signal $x(t)$ über eine Abbildung $X(\omega, t) : \mathbb{T} \times \Omega \to \mathbb{R}$ zugeordnet wird:

Definition 2.22: Stochastischer Prozess

Sei $\mathbb{T}$ eine nichtleere Teilmenge von $\mathbb{R}$, welche einen Zeitraum darstellt, d. h.

$$\textit{zeitdiskreter Fall}: \quad \mathbb{T} = \{0, t_0, t_1, \ldots\}\,, \tag{2.52}$$

$$\textit{zeitkontinuierlicher Fall}: \quad \mathbb{T} = [0, T]\,. \tag{2.53}$$

Eine Funktion $\boldsymbol{X}(\boldsymbol{\omega}, t) : \mathbb{T} \times \boldsymbol{\Omega} \to \mathbb{R}^n$, wobei $\boldsymbol{X}(t, \cdot)$ für jedes $t \in \mathbb{T}$ bzw. $\boldsymbol{X}[k, \cdot]$ für jedes $k \in \mathbb{T}$ eine Zufallsvariable ist, heißt stochastischer Prozess.

Die Menge aller möglichen Signale $x(t)$ wird Schar genannt und die einzelnen Signale der Schar heißen Realisierungen. Jeder Abtastwert $x(t_0)$ eines stochastischen Prozesses $x(t)$ zum Zeitpunkt t_0 stellt somit eine Zufallsvariable der Form $X(\omega, t_0) = X(\omega)$ dar – siehe hierzu auch Abb. 2.6. Folglich kann ein stochastischer Prozess auf zwei Weisen interpretiert werden:

1. Als Ensemble von Signalen (Realisierungen), aus der ein Signalverlauf im Zuge eines Zufallsexperiments ausgewählt wurde oder
2. als Menge von (zeitlich geordneten) Zufallsvariablen.

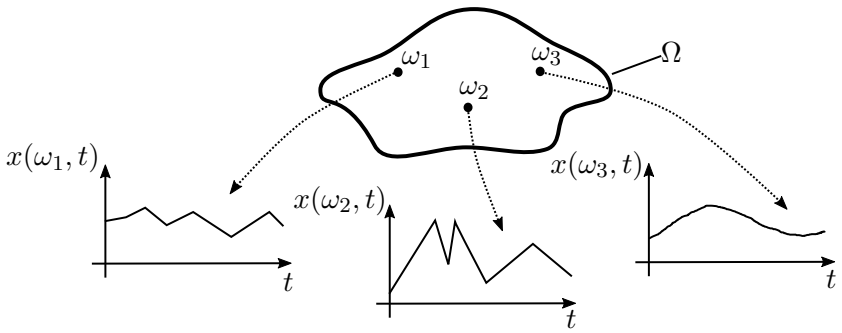


Abb. 2.6: Realisierung eines stochastischen Prozesses

In ingenieurtechnischen Anwendungen werden stochastische Prozesse i. d. R. durch empirisch gewonnene Daten auf einem zeitdiskreten Abtastungsraster $k = 0, 1, \ldots, N$ erfasst. In diesem Sinne soll ein gegebener stochastischer Prozess bzw. eine gegebene (empirische) Zeitreihe in Form eines Datenvektors $\boldsymbol{\xi}$ für die skalare Größe $x \in \mathbb{R}$

$$\boldsymbol{\xi} = \begin{bmatrix} x[1] & x[2] & \ldots & x[k] \end{bmatrix}^T \tag{2.54}$$

[1] In einigen Problemstellungen ist eine gemeinsame Abhängigkeit von Ort und Zeit, wie beispielsweise zur Beschreibung einer Wellenausbreitung, anzutreffen. Der Einfachheit halber werden ausschließlich räumlich konzentrierte physikalische Größen betrachtet, wie punktförmig wirkende Kräfte oder ortsunabhängige elektrische Ströme.

bzw. Datenmatrix $\boldsymbol{\Xi}$ für einen Größenvektor $\boldsymbol{x} \in \mathbb{R}^n$

$$\boldsymbol{\Xi} = \begin{bmatrix} \boldsymbol{\xi}_1^T & \boldsymbol{\xi}_2^T & \dots & \boldsymbol{\xi}_n^T \end{bmatrix}^T = \begin{bmatrix} x_1[1] & x_1[2] & \dots & x_1[k] \\ x_2[1] & x_2[2] & \dots & x_2[k] \\ \vdots & \vdots & & \vdots \\ x_n[1] & x_n[2] & \dots & x_n[k] \end{bmatrix} \tag{2.55}$$

repräsentiert werden. Im Folgenden werden einige wichtige Kenngrößen zur Beschreibung stochastischer Prozesse eingeführt, angefangen mit der Autokovarianz:

Definition 2.23: Autokovarianz

Gegeben sei ein stationärer[1], stochastischer Prozess $X(t)$. Dann ist die Autokovarianz für eine gegebene Zeitdifferenz τ definiert als

$$C_{xx}(\tau) = \mathrm{E}[(X(t) - \mu)(X(t+\tau) - \mu)]. \tag{2.56}$$

Liegt eine empirische Stichprobe des Prozesses als abgetastetes Signal $x[k]$ vor, so kann die Autokovarianz approximiert werden mit

$$C_{xx}(\tau) \approx \frac{1}{N-\tau} \sum_{k=1}^{N-\tau} (x[k] - \overline{x})(x[k+\tau] - \overline{x}), \quad \tau = \{0, 1, \dots\} \tag{2.57}$$

wobei $\overline{x}$ der Stichprobenmittelwert des Signals ist.

Obige Näherung stellt die (unverzerrte azyklische) Stichprobenautokovarianz dar, sofern nur endlich viele Abtastungen N vorliegen. Sie beschreibt die Korrelation eines stochastischen Prozesses an unterschiedlichen Zeitpunkten $[k, k+\tau]$ und stellt somit ein Maß für den Einfluss vergangener Werte auf den jeweils aktuellen Wert dar.

Demgegenüber stellt die Autokorrelation[2] eine normalisierte Variante der Autokovarianz dar:

[1] Ein stationärer Prozess ist invariant gegenüber Zeitverschiebungen, d. h. insbesondere, dass dieser zu allen Zeitpunkten den gleichen Erwartungswert und die gleiche Varianz aufweist.

[2] In der Stochastik, in der Zeitreihenanalyse und im Ingenieurswesen existieren durchaus verschiedene Varianten der empirischen Autokorrelation. Die in Definition 2.24 angegebene Variante entspricht der üblichen Darstellung in der Zeitreihenanalyse. Sie besitzt insbesondere den Vorteil dimensionslos zu sein. Eine zu (2.58) formal stimmigere empirische Autokorrelation ist $R_{xx}(\tau) \approx \frac{1}{N-\tau} \sum_{k=1}^{N-\tau} (x[k]x[k+\tau]), \quad \tau = \{0, 1, \dots\}$. Diese hängt aber von der Größenordnung und den Einheiten des zugrundeliegenden Signals ab, sodass der Vergleich unterschiedlicher Signale (z. B. im Kontext einer Genauigkeitsanalyse) nicht intuitiv möglich ist.

Definition 2.24: Autokorrelation

Gegeben sei ein stationärer, stochastischer Prozess $X(t)$. Dann ist die Autokorrelation für eine gegebene Zeitdifferenz τ definiert als

$$R_{xx}(\tau) = \mathrm{E}[X(t)X(t+\tau)]. \tag{2.58}$$

Liegt eine empirische Stichprobe des Prozesses als abgetastetes Signal $x[k]$ vor, so kann die (normalisierte) Autokorrelation approximiert werden mit

$$R_{xx}(\tau) \approx \frac{\frac{1}{N-\tau}\sum_{k=1}^{N-\tau}(x[k]-\overline{x})(x[k+\tau]-\overline{x})}{\frac{1}{N}\sum_{k=1}^{N}(x[k]-\overline{x})^2}, \quad \tau = \{0,1,\ldots\}. \tag{2.59}$$

wobei $\overline{x}$ der Stichprobenmittelwert des Signals ist.

Weiterhin kann die Autokorrelation herangezogen werden, um zu bestimmen, ob ein Signal weißes Rauschverhalten aufzeigt:

Definition 2.25: Weißes Rauschen

Gegeben sei ein stochastisches, zeitdiskretes Signal der Form $x[k]$. Dieses wird als weißes Rauschen bezeichnet, falls die abgetasteten Signalwerte untereinander stochastisch unabhängig sind. Genau dann gilt:

$$R_{xx}(\tau) = \sigma_x^2\delta(\tau) \quad \textit{mit dem Kronecker-Delta:} \quad \delta(\tau) = \begin{cases} 1 & \textit{für } \tau = 0 \\ 0 & \textit{für } \tau \neq 0 \end{cases}. \tag{2.60}$$

Entspricht die Verteilung von $x[k] \sim \mathcal{N}(\mu, \sigma^2)$ zudem noch der Normalverteilung, dann spricht man von *Gaußschem weißem Rauschen.*

Sollen demgegenüber nun unterschiedliche Signale bzw. stochastische Prozesse miteinander verglichen werden, führt dies zur Kreuzkovarianz:

Definition 2.26: Kreuzkovarianz

Gegeben seien zwei stationäre, stochastische Prozesse $X(t)$ und $Y(t)$. Dann ist die Kreuzkovarianz für eine gegebene Zeitdifferenz τ definiert als

$$C_{xy}(\tau) = \mathrm{E}[(X(t)-\mu_x)(Y(t+\tau)-\mu_y)]. \tag{2.61}$$

Liegen empirische Stichproben der Prozesses als abgetastete Signale $x[k]$ und $y[k]$ vor, so kann die Kreuzkovarianz approximiert werden mit

$$C_{xy}(\tau) \approx \frac{1}{N-\tau}\sum_{k=1}^{N-\tau}(x[k]-\overline{x})(y[k+\tau]-\overline{y}), \quad \tau = \{0,1,\ldots\}. \tag{2.62}$$

Das normierte Gegenstück hierzu ist die Kreuzkorrelation:

Definition 2.27: Kreuzkorrelation

Gegeben seien zwei stationäre, stochastische Prozesse $X(t)$ und $Y(t)$. Dann ist die Kreuzkorrelation für eine gegebene Zeitdifferenz τ definiert als

$$R_{xy}(\tau) = \mathrm{E}[X(t)Y(t+\tau)]. \tag{2.63}$$

Liegen empirische Stichproben der Prozesse als abgetastete Signale $x[k]$ und $y[k]$ vor, so kann die (normalisierte) Kreuzkorrelation[1] approximiert werden mit

$$R_{xy}(\tau) \approx \frac{\frac{1}{N-\tau}\sum_{k=1}^{N-\tau}(x[k]-\overline{x})(y[k+\tau]-\overline{y})}{\sqrt{\frac{1}{N}\sum_{k=1}^{N}(x[k]-\overline{x})^2}\sqrt{\frac{1}{N}\sum_{k=1}^{N}(y[k]-\overline{y})^2}}, \quad \tau = \{0,1,\ldots\}. \tag{2.64}$$

Im Allgemeinen ist die Autokorrelation bzw. Kreuzkorrelation der Autokovarianz bzw. Kreuzkovarianz vorzuziehen, da erstere durch entsprechende Normierung auch einen Vergleich von verschiedenen Größen unterschiedlicher Skalierung bzw. Einheiten ermöglichen.

2.3.4 Beispiele für lineare, stochastische Prozesse

Abschließend werden noch die bekanntesten linearen Differenzengleichungen als Beispiele für lineare, stochastische Prozesse wiedergegeben, welche im Kontext der Zeitreihenanalyse häufig Verwendung finden. Hier findet eine Beschränkung auf SISO-Systeme statt, eine Erweiterung auf MIMO-Systeme ist i. A. möglich und kann der weiterführenden Literatur entnommen werden. Grundsätzlich sind lineare Zeitreihenmodelle in zwei Kategorien einzuordnen: rein stochastische Differenzengleichungen und deterministische Differenzengleichungen mit stochastischer Störung. In die erste Kategorie fallen sog. *ARMA-Modelle (autoregressive-moving average model)*:

Definition 2.28: ARMA-Modell

Ein lineares, zeitdiskretes Modell des Signals $x[k]$ der Form

$$x[k] = \sum_{i=1}^{p} a_i x[k-i] + \sum_{j=1}^{q} b_j \nu[k-j+1] + c \tag{2.65}$$

mit den Koeffizienten $\{a_i, b_j, c\} \in \mathbb{R}$ sowie dem weißen Rauschterm $\nu[k]$ wird ARMA(q,p)-Modell genannt, wobei q und p die Ordnung des Modells darstellen.

Der aktuelle Signalwert $x[k]$ setzt sich somit linear sowohl aus den Einflüssen der vergangenen Rauschterme als auch aus aus den Einflüssen der vergangenen Zeitreihenwerte selbst zusammen. Hieraus lassen sich zwei Spezialfälle ableiten: Für $a_i = 0 \;\forall\; i = 1,\ldots,p$ entspricht (2.65) einer gewichteten, gleitenden Mittelwertbildung der Rauschterme $\nu[k-j+1] \;\forall\; j = 1,\ldots,q$:

$$x[k] = \sum_{j=1}^{q} b_j \nu[k-j+1] + c. \tag{2.66}$$

[1] Analog zur Autokorrelation existieren in den verschiedenen Fachdisziplinen ebenfalls unterschiedliche Definitionen der Kreuzkorrelation. Basierend auf der exakt-stochastischen Defintion (2.63) lautet die in der Stochastik übliche Definition der empirischen Kreuzkorrelation $R_{xy}(\tau) \approx \frac{1}{N-\tau}\sum_{k=1}^{N-\tau}(x[k]y[k+\tau]), \quad \tau = \{0,1,\ldots\}$.

Dieser Modelltyp wird daher *MA-Modell (moving average model)* genannt. Hingegen für $b_1 = 1 \wedge b_j = 0 \; \forall \; j = 2, \ldots, q$ entspricht (2.65) einer gewichteten, gleitenden Summenbildung der Signalterme $x[k-i] \; \forall \; i = 1, \ldots, p$ mit einem zusätzlichen Rauschanteil:

$$x[k] = \sum_{i=1}^{p} a_i x[k-i] + \nu[k] + c. \tag{2.67}$$

Dieser Modelltyp wird *AR-Modell (autoregressive model)* genannt. Wiederum als Spezialfall der AR-Modelle sei die sog. *Irrfahrt (random walk)* erwähnt, welche sich für $p = 1$ und $a_i = 1$ ergibt:

$$x[k] = x[k-1] + \nu[k] + c, \tag{2.68}$$

wobei der Fall $c \neq 0$ einen sog. Drift bezeichnet.

Gegenüber obigen stochastischen Differenzengleichungen, welche ausschließlich ein Rauschsignal als Eingangsgröße aufweisen, überlagern deterministische Differenzengleichungen mit stochastischer Störung ein exogenes und bekanntes Eingangssignal $u[k]$ mit einer zusätzlichen stochastischen Anregung $\nu[k]$. In diese Modellklasse fallen die sog. *ARMAX-Modelle (autoregressive-moving average model with exogenous inputs)*[1]:

Definition 2.29: ARMAX-Modell

Ein lineares, zeitdiskretes Modell des Signals $x[k]$ der Form

$$x[k] = \sum_{i=1}^{p} a_i x[k-i] + \sum_{j=1}^{q} b_j \nu[k-j+1] + \sum_{l=1}^{r} c_l u[k-l+1] + d \tag{2.69}$$

mit den Koeffizienten $\{a_i, b_j, c_l, d\} \in \mathbb{R}$ sowie dem weißen Rauschterm $\nu[k]$ und einem deterministischen, exogenen Anregungssignal $u[k]$ wird $ARMAX(q,p,r)$-Modell genannt, wobei p, q und r die Ordnung des Modells darstellen.

Vorangegangene Modelle lassen sich zudem in eine Darstellung mittels Übertragungsfunktionen überführen. Zur übersichtlicheren Darstellbarkeit wird hierfür der Verschiebeoperator (*shift operator*) eingeführt[2]:

Definition 2.30: Verschiebeoperator

Gegeben sei ein zeitdiskretes Signal $x[k]$, dann wird eine zeitliche Verschiebung mittels Verschiebeoperator ϱ definiert als

$$\varrho^{-1} x[k] = x[k-1], \quad \varrho x[k] = x[k+1]. \tag{2.70}$$

Hiermit lassen sich insbesondere obige Signaldefinitionen umschreiben zu:

$$x[k] = \sum_{i=1}^{N} a_i x[k-i] = \left[\sum_{i=1}^{N} a_i \varrho^{-i} \right] x[k] = A(\varrho) x[k]. \tag{2.71}$$

[1] Das 'X' steht hier für *eXogenous*.

[2] In der weiteren Literatur ist die Verwendung der Formelbuchstabens q für den Verschiebeoperator ebenfalls üblich. Um demgegeüber Verwechslungen mit der Ordnung der obigen stochastischen Modelle zu vermeiden, wird daher ϱ verwendet.

Hierbei wird

$$A(\varrho) = \sum_{i=1}^{N} a_i \varrho^{-i} \tag{2.72}$$

als sog. *Transferoperator* genutzt, der neben der zeitlichen Verschiebung noch eine polynomiale Gewichtung mit den Koeffizienten a_i vornimmt. Mit dieser Schreibweise können obige Modellansätze hin zu

$$A(\varrho)x[k] = \frac{B(\varrho)}{C(\varrho)}u[k] + \frac{D(\varrho)}{E(\varrho)}\nu[k] \tag{2.73}$$

verallgemeinert werden. Je nachdem welche Gewichtungspolynome in (2.73) genutzt werden, ergeben sich hieraus bis zu 32 verschiedene Modellklassen. in Tab. 2.1 ist daher eine ausgewählte tabellarische Übersicht zu bekannten Modelltypen aufgeführt.

Genutzte Transferoperatoren	Resultierende Modellklasse
B	FIR (*finite impulse response*)
C	MA
D	AR
AD	ARMA
ABD	ARMAX
BC	OE (*output error*)
BCDE	BJ (*Box-Jenkins*)

Tab. 2.1: Auswahl verschiedener Prozessmodelle abgeleitet aus (2.73)

Abschließend sind in Abb. 2.7, Abb. 2.8 und Abb. 2.9 drei beispielhafte AR-Modelle der Form

$$x[k] = 0,95x[k-1] + \nu[k] \tag{2.74}$$

für zwei unterschiedliche Abtastlängen N sowie drei unterschiedlichen Zufallsgeneratoren für $\nu[k] \sim \mathcal{N}(\sigma = 1, \mu = 0)$ dargestellt. Es wird deutlich, dass obwohl die Modelle für alle Signalverläufe identisch sind, der Zufallseinfluss maßgeblich für den konkreten Signalverlauf ist. Der Verlauf der Autokorrelation ist zudem stark durch die konkrete Realisierung geprägt und durch die limitierte Anzahl an Abtastungen signifikant vom theoretisch zu erwarten Verlauf bei weißen Rauschen entfernt (siehe Definition 2.25). Ferner gilt es, bei der Interpretation der Signalverläufe zu beachten, dass die verwendeten Rauschterme auf Basis synthetischer Signalgeneratoren erzeugt wurden und hierdurch weitere numerische Abweichungen zwischen Theorie und Empirie hervorgerufen werden.

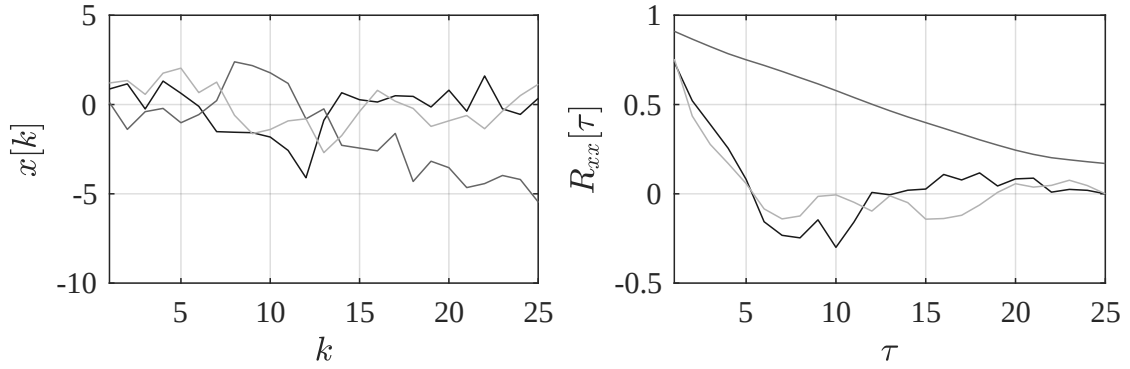


Abb. 2.7: Signalverläufe für MA-Modell (2.74) mit $N = 25$ Abtastungen und unterschiedlichen Zufallsgeneratoren $\nu[k]$

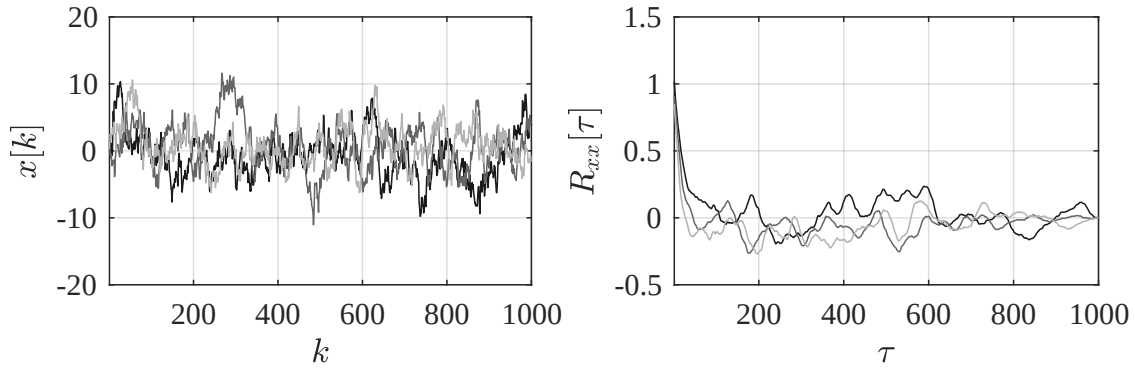


Abb. 2.8: Signalverläufe für MA-Modell (2.74) mit $N = 1.000$ Abtastungen und unterschiedlichen Zufallsgeneratoren $\nu[k]$

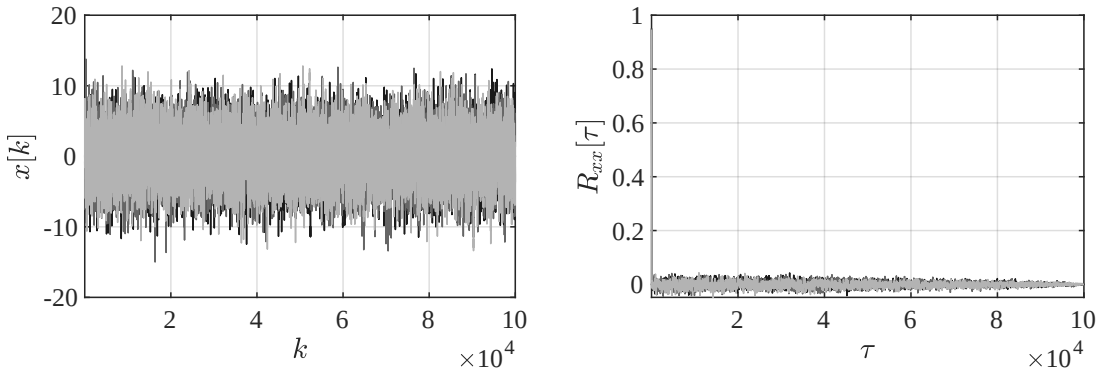


Abb. 2.9: Signalverläufe für MA-Modell (2.74) mit $N = 100.000$ Abtastungen und unterschiedlichen Zufallsgeneratoren $\nu[k]$

3 Identifikation statischer Modelle

In diesem Kapitel werden die Grundlagen zur Identifikation von statischen Modellen (siehe Definition 1.7) behandelt. Die Identifikation wird hierbei zunächst für lineare Problemstellungen diskutiert, wobei am Ende des Kapitels noch ein Exkurs zu nichtlinearen Aufgaben folgt. Das grundsätzliche Identifikationsvorgehen ist in Abb. 3.1 für ein MISO-System zusammengefasst: Der Ausgang $y[k]$ eines statischen Prozesses mit optionaler Anregung $\boldsymbol{u}[k]$ wird messtechnisch erfasst, sodass die Messungen $\psi[k]$ vorliegen. Zudem kann ausgangsseitig das Rauschen $\nu[k]$ einwirken. Der Messwert wird dem Schätzwert des Modellausgangs $\hat{y}[k]$ gegenübergestellt und die entsprechenden Residuen $e[k]$ gebildet. Diese werden dann einer Kostenfunktion zugeführt, welche einen skalaren Kostenwert J ausgibt, der einem Optimierer als Eingangsgröße zur eigentlichen Parameteridentifikation dient:

$$\boldsymbol{\theta}^* = \arg\min J(\boldsymbol{\theta}). \tag{3.1}$$

Zur besseren Anschauung sei auf folgendes Beispiel verwiesen: Gegeben sei ein Elektrofahrzeug, dessen Fahreigenschaften zu bestimmen sind. Konkret sollen die Systemparameter zur Beschreibung des lateralen Fahrwiderstands ermittelt werden. Die laterale Widerstandskraft F_w ist gegeben durch:

$$F_w = mgc_r \cos(\gamma) + mg \sin(\gamma) + \eta_v v + 1/2 c_w A_f \rho_l v^2. \tag{3.2}$$

Hier ist m die Fahrzeugmasse, g die Erdbeschleunigung, c_r der Rollreibungsbeiwert, γ die Fahrbahnsteigung, η_v der Koeffizient für viskose Reibung in Lagern und Getrieben, v die Fahrzeuggeschwindigkeit, c_w der Luftwiderstandsbeiwert, A_f die laterale Fahrzeugfläche und ρ_l die Luftdichte. Der Einfachheit halber werden die Versuche zur Bestimmung der Systemparameter in der ebenen Fläche ($\gamma = 0$) durchgeführt, sodass folgt:

$$F_w = mgc_r + \eta_v v + \frac{1}{2} c_w A_f \rho_l v^2. \tag{3.3}$$

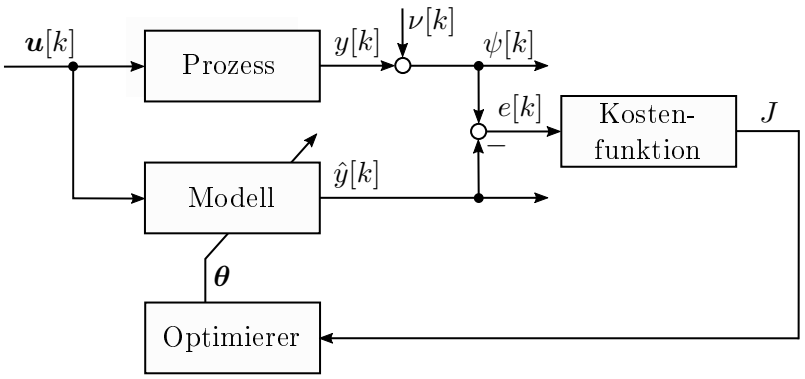


Abb. 3.1: Schematische Darstellung der Parameteridentifikation in einem statischen Prozess

Die Parameter m, g, A_f sowie ρ_l sind Konstanten und können typischerweise über einfache Messungen oder Datenblätter ermittelt werden – diese werden daher als bekannt vorausgesetzt. Die Geschwindigkeit v sowie die am Fahrzeug angreifende Kraft F_w werden messtechnisch erfasst[1], sodass der gesuchte Parametervektor

$$\boldsymbol{\theta} = \begin{bmatrix} c_r & \eta_v & c_w \end{bmatrix}^T \tag{3.4}$$

ist. Diese Modellgleichung kann in vektorielle Form

$$\underbrace{F_w}_{\psi} = \underbrace{\begin{bmatrix} mg & v & \frac{1}{2}A_f\rho_l v^2 \end{bmatrix}}_{\boldsymbol{\xi}^T} \underbrace{\begin{bmatrix} c_r \\ \eta_v \\ c_w \end{bmatrix}}_{\boldsymbol{\theta}} \tag{3.5}$$

gebracht werden, wobei ψ die sog. Zielgröße (beinhaltet die abhängigen Variablen) ist und $\boldsymbol{\xi}$ repräsentiert die sog. Regressoren (unabhängigen Variablen). In (3.5) wird deutlich, dass der gesuchte Parametervektor $\boldsymbol{\theta}$ linear in die Modellgleichung eingeht und somit ein lineares Identifkationsproblem vorliegt. Der zunächst augenscheinlich nichtlineare Einfluss der Geschwindigkeit v ist unerheblich, da diese als Regressor bekannt ist. Für einen Messpunkt $\psi[k]$ gilt dann:

$$F_w[k] = \begin{bmatrix} mg & v[k] & \frac{1}{2}A_f\rho_l v^2[k] \end{bmatrix} \begin{bmatrix} c_r \\ \eta_v \\ c_w \end{bmatrix} + e[k]. \tag{3.6}$$

Hierbei ist $e[k]$ ein Residuenterm, welcher Messrauschen sowie Modellierungsfehler abbildet. Liegen mehrere Messungen $k = 1, \ldots, n$ vor, ergibt sich das folgende Gleichungssystem:

$$\underbrace{\begin{bmatrix} F_w[1] \\ \vdots \\ F_w[n] \end{bmatrix}}_{\boldsymbol{\psi}} = \underbrace{\begin{bmatrix} mg & v[1] & \frac{1}{2}A_f\rho_l v^2[1] \\ \vdots & \vdots & \vdots \\ mg & v[n] & \frac{1}{2}A_f\rho_l v^2[n] \end{bmatrix}}_{\boldsymbol{\Xi}} \underbrace{\begin{bmatrix} c_r \\ \eta_v \\ c_w \end{bmatrix}}_{\boldsymbol{\theta}} + \underbrace{\begin{bmatrix} e[1] \\ \vdots \\ e[n] \end{bmatrix}}_{\boldsymbol{e}} \tag{3.7}$$

mit $\boldsymbol{\psi}$ als Messvektor, $\boldsymbol{e}$ als Residuenvektor und $\boldsymbol{\Xi}$ als Regressormatrix. Hinsichtlich der Lösung dieses Gleichungssystems sei auf folgenden Satz verwiesen:

[1] Im stationären Zustand entspricht die Fahrtwiderstandskraft gerade der vom Elektromotor aufgebrachten Antriebskraft (Annahme: schlupffreies Reifenverhalten), welche auf Basis des an der Motorwelle angreifenden Drehmoments berechnet werden kann. Daher muss das Drehmoment entweder messtechnisch erfasst oder mittels Motormodellen aus bekannten Messgrößen berechnet werden. Spannende Details hierzu gibt es u. a. in den Lehrveranstaltungen *Geregelte Drehstromantriebe* sowie *Antriebe für umweltfreundliche Fahrzeugantriebe*. Für dieses Beispiel sei angenommen, dass diese Größe messtechnisch ermittelt werden kann.

Satz 3.1: Lösbarkeit linearer Gleichungssysteme

Gegeben sei ein lineares Gleichungssystem der Form

$$\boldsymbol{\psi} = \boldsymbol{\Xi}\boldsymbol{\theta} \tag{3.8}$$

mit dem gesuchten Vektor $\boldsymbol{\theta} \in \mathbb{R}^m$ *sowie dem gegebenen Vektor* $\boldsymbol{\psi} \in \mathbb{R}^n$ *bzw. der Matrix* $\boldsymbol{\Xi} \in \mathbb{R}^{n \times m}$. *Dann folgt hinsichtlich der Lösbarkeit des Gleichungssystems:*

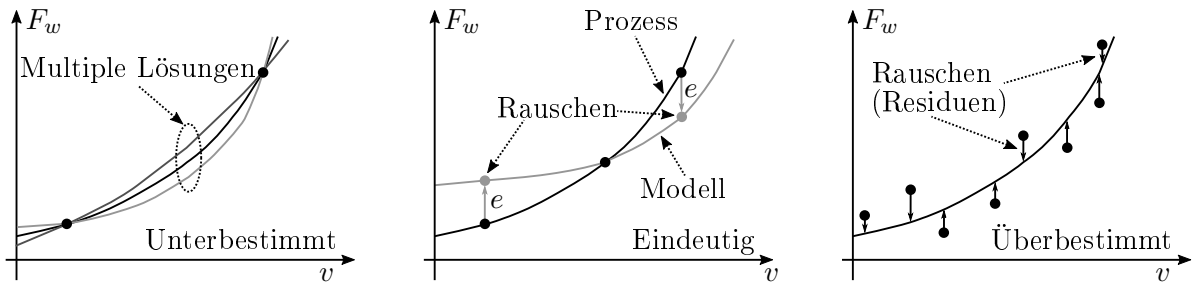
- $m > n$: *lediglich parametrische Lösung möglich* $\boldsymbol{\theta} = \boldsymbol{\theta}(\theta_1, \ldots)$ *(unterbestimmt),*
- $m = n$: *eine exakte Lösung* $\boldsymbol{\theta} = \boldsymbol{\Xi}^{-1}\boldsymbol{\psi}$ *(falls* $\operatorname{rang}(\boldsymbol{\Xi}) = n$*),*
- $m < n$: *keine Lösung, welche alle Gleichungen exakt erfüllt (überbestimmt).*

Im Fall eines unterbestimmten Gleichungssystems müssen demnach Annahmen für einen Teil der gesuchten Parameter getroffen werden, um eine Lösung zu erhalten. Für das Beispiel des Elektrofahrzeugs wäre dies sicherlich ein unbefriedigendes Ergebnis (siehe Abb. 3.2a), sodass hier mindestens die drei notwendigen Messpunkte aufgenommen werden, um die gesuchten Fahrzeugparameter zu identifizieren. Hier sei allerdings auf (3.7) verwiesen, konkret auf den Residuenterm $\boldsymbol{e}$. Im gegebenen Beispiel modelliert dieser Messfehler hinsichtlich der Kraft F_w. Wird angenommen, dass $\boldsymbol{e}[k]$ die zeitliche Realisierung einer Zufallsvariablen (stochastischer Prozess) mit

$$\mathrm{E}(\boldsymbol{e}) = 0, \quad \mathrm{Var}(\boldsymbol{e}) = \sigma^2$$

sei, dann kann das Rauschen die zu identifizierenden Parameterwerte maßgeblich verfälschen. Genau dieser Zusammenhang ist bildlich in Abb. 3.2b dargestellt, in der eine signifikante Abweichung zwischen tatsächlichem Prozess (schwarze Kennlinie) und dem auf Basis der Messdaten identifizierten Modell (graue Kennlinie) zu erkennen ist.

Um diesen systematischen Identifikationsfehler zu vermeiden, können weitere Messpunkte aufgenommen werden, um dass in Abb. 3.2c dargestellte überbestimmte Gleichungssystem zu erzeugen. Das resultierende Problem, einen Parametervektor $\boldsymbol{\theta}$ zu finden, welcher die Abweichungen zwischen Modell und Messung systematisch minimiert, wird im Folgenden behandelt. Der bekannteste Lösungsansatz hierzu ist die Methode der kleinsten Quadrate (*least squares* – LS).



(a) $m > n$ (b) $m = n$ (c) $m < n$

Abb. 3.2: Beispielhafte Identifikation lateraler Fahrzeugparameter anhand der Widerstandskraft für unterschiedliche Anzahl an Messpunkten

3.1 Methode der kleinsten Quadrate

Wie der Name der Methode bereits vermuten lässt, hat diese zum Ziel, die quadratischen Abweichungen zwischen Modell und Messung

$$\boldsymbol{e} = \boldsymbol{\psi} - \boldsymbol{\Xi}\boldsymbol{\theta} \tag{3.9}$$

zu minimieren. Im SISO-Fall lautet die Kostenfunktion:

$$J(\boldsymbol{\theta}) = \sum_{k=1}^{N}(e[k])^2 = \sum_{k=1}^{N}(\psi[k] - \boldsymbol{\xi}^T[k]\boldsymbol{\theta})^2 = (\boldsymbol{\psi} - \boldsymbol{\Xi}\boldsymbol{\theta})^T(\boldsymbol{\psi} - \boldsymbol{\Xi}\boldsymbol{\theta}). \tag{3.10}$$

Hierzu wird die folgende Problemdefinition formuliert:

Definition 3.1: LS-Problem für lineare, statische Systeme

Gegeben sei eine Regressionsgleichung der Form

$$\boldsymbol{e} = \boldsymbol{\psi} - \boldsymbol{\Xi}\boldsymbol{\theta} \tag{3.11}$$

mit dem Parametervektor $\boldsymbol{\theta} \in \mathbb{R}^m$, dem Messdatenvektor $\boldsymbol{\psi} \in \mathbb{R}^n$, der Regressormatrix $\boldsymbol{\Xi} \in \mathbb{R}^{n\times m}$ sowie dem Residuenvektor $\boldsymbol{e} \in \mathbb{R}^n$. Es gelte $m < n$. Der Messdatenvektor $\boldsymbol{\psi}$ weise ein additives Messrauschen ν mit $\mathrm{E}(\nu) = 0$ und $\mathrm{Cov}(\nu) = \sigma^2$ auf, während $\boldsymbol{\Xi}$ exakt bekannt sei. Das Auffinden des Parametervektors $\boldsymbol{\theta}$ mittels Minimierung der quadratischen Kostenfunktion (3.10) entsprechend

$$\boldsymbol{\theta}^* = \arg\min J(\boldsymbol{\theta}) \tag{3.12}$$

wird als LS-Problem für lineare, statische Systeme bezeichnet.

Nachfolgend wird der Lösungsweg für das LS-Problem skizziert. Umschreiben von (3.10) liefert:

$$\begin{aligned} J &= (\boldsymbol{\psi} - \boldsymbol{\Xi}\boldsymbol{\theta})^T(\boldsymbol{\psi} - \boldsymbol{\Xi}\boldsymbol{\theta}) \\ &= (\boldsymbol{\psi}^T - \boldsymbol{\theta}^T\boldsymbol{\Xi}^T)(\boldsymbol{\psi} - \boldsymbol{\Xi}\boldsymbol{\theta}) \\ &= \boldsymbol{\psi}^T\boldsymbol{\psi} - \boldsymbol{\theta}^T\boldsymbol{\Xi}^T\boldsymbol{\psi} - (\boldsymbol{\Xi}^T\boldsymbol{\psi})^T\boldsymbol{\theta} + \boldsymbol{\theta}^T\boldsymbol{\Xi}^T\boldsymbol{\Xi}\boldsymbol{\theta}. \end{aligned} \tag{3.13}$$

Zur Minimierung von J werden die partiellen Ableitung nach $\boldsymbol{\theta}$ zu Null gesetzt,

$$\frac{\partial J}{\partial \boldsymbol{\theta}} = \nabla J(\boldsymbol{\theta}) = \begin{bmatrix} \frac{\partial J}{\partial \theta_1} & \frac{\partial J}{\partial \theta_2} & \dots & \frac{\partial J}{\partial \theta_m} \end{bmatrix}^T = \begin{bmatrix} 0 & 0 & \dots & 0 \end{bmatrix}^T = \mathbf{0}, \tag{3.14}$$

d. h., die notwendige Bedingung ist erfüllt, wenn alle partiellen Ableitungen von J bezüglich der Komponenten von $\boldsymbol{\theta}$ null sind. Unter Berücksichtigung der in Anhang A.1 und Anhang A.2 zusammengefassten Rechenregeln werden die Summanden in (3.13) zunächst einzeln behandelt

$$\frac{\partial}{\partial \boldsymbol{\theta}}\left(\boldsymbol{\psi}^T\boldsymbol{\psi}\right) = \mathbf{0}\,, \tag{3.15a}$$

$$\frac{\partial}{\partial \boldsymbol{\theta}}\left(\boldsymbol{\theta}^T\boldsymbol{\Xi}^T\boldsymbol{\psi}\right) = \left(\boldsymbol{\Xi}^T\boldsymbol{\psi}\right)^T\,, \tag{3.15b}$$

$$\frac{\partial}{\partial \boldsymbol{\theta}}\left(\left(\boldsymbol{\Xi}^T\boldsymbol{\psi}\right)^T\boldsymbol{\theta}\right) = \left(\boldsymbol{\Xi}^T\boldsymbol{\psi}\right)^T\,, \tag{3.15c}$$

$$\frac{\partial}{\partial \boldsymbol{\theta}} \left(\boldsymbol{\theta}^T \boldsymbol{\Xi}^T \boldsymbol{\Xi} \boldsymbol{\theta} \right) = 2 \boldsymbol{\theta}^T \boldsymbol{\Xi}^T \boldsymbol{\Xi} \tag{3.15d}$$

und das Einsetzen in (3.14) liefert dann:

$$\frac{\partial J}{\partial \boldsymbol{\theta}} = -2 \left(\boldsymbol{\Xi}^T \boldsymbol{\psi} \right)^T + 2 \boldsymbol{\theta}^T \boldsymbol{\Xi}^T \boldsymbol{\Xi} = -2 \left(\boldsymbol{\psi}^T - \boldsymbol{\theta}^T \boldsymbol{\Xi}^T \right) \boldsymbol{\Xi} = \mathbf{0}. \tag{3.16}$$

Auflösen nach $\boldsymbol{\theta}^T$ ergibt den gesuchten Parametervektor

$$\boldsymbol{\theta}^T = \boldsymbol{\psi}^T \boldsymbol{\Xi} (\boldsymbol{\Xi}^T \boldsymbol{\Xi})^{-1}, \quad \text{falls} \quad \det(\boldsymbol{\Xi}^T \boldsymbol{\Xi}) \neq 0 \tag{3.17}$$

bzw.

$$\boldsymbol{\theta} = (\boldsymbol{\Xi}^T \boldsymbol{\Xi})^{-1} \boldsymbol{\Xi}^T \boldsymbol{\psi}, \quad \text{falls} \quad \det(\boldsymbol{\Xi}^T \boldsymbol{\Xi}) \neq 0. \tag{3.18}$$

Weiterhin gilt zu prüfen, ob das obige Ergebnis auch tatsächlich das Minimum der Kostenfunktion darstellt. Hierfür muss die Hesse-Matrix der Kostenfunktion

$$\frac{\partial^2 J}{\partial \boldsymbol{\theta}^2} = (\nabla^2 J)(\boldsymbol{\theta}) = \begin{bmatrix} \frac{\partial^2 J}{\partial \theta_1 \partial \theta_1}(\boldsymbol{\theta}) & \frac{\partial^2 J}{\partial \theta_1 \partial \theta_2}(\boldsymbol{\theta}) & \cdots & \frac{\partial^2 J}{\partial \theta_1 \partial \theta_n}(\boldsymbol{\theta}) \\ \frac{\partial^2 J}{\partial \theta_2 \partial \theta_1}(\boldsymbol{\theta}) & \frac{\partial^2 J}{\partial \theta_2 \partial \theta_2}(\boldsymbol{\theta}) & \cdots & \frac{\partial^2 J}{\partial \theta_2 \partial \theta_n}(\boldsymbol{\theta}) \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 J}{\partial \theta_n \partial \theta_1}(\boldsymbol{\theta}) & \frac{\partial^2 J}{\partial \theta_n \partial \theta_2}(\boldsymbol{\theta}) & \cdots & \frac{\partial^2 J}{\partial \theta_n \partial \theta_n}(\boldsymbol{\theta}) \end{bmatrix} \tag{3.19}$$

positiv definit sein[1]. Die Hesse-Matrix heißt positiv definit, falls die Eigenwerte der Hesse-Matrix λ_i echt größer Null sind [BSMM05]:

$$\lambda_i > 0 \quad i = [1, \ldots, m] \quad \text{für} \quad \det \left(\frac{\partial^2 J}{\partial \boldsymbol{\theta}^2} - \boldsymbol{\lambda} \boldsymbol{I} \right) = 0 \tag{3.20}$$

mit

$$\boldsymbol{\lambda} = \begin{bmatrix} \lambda_1 & \ldots & \lambda_m \end{bmatrix}^T. \tag{3.21}$$

Im vorliegenden Fall ergibt sich die Hesse-Matrix zu:

$$\frac{\partial^2 J}{\partial \boldsymbol{\theta}^2} = 2 \boldsymbol{\Xi}^T \boldsymbol{\Xi} \tag{3.22}$$

Somit lässt sich die Lösung für das gegebene LS-Problem wie folgt zusammenfassen:

[1] Ist die Hesse-Matrix negativ definit (alle Eigenwerte echt kleiner Null) handelt es sich um ein Maximum der Kostenfunktion. Ist sie indefinit (postive und negative Eigenwerte) so liegt ein Sattelpunkt vor.

Satz 3.2: Lösung des LS-Problems für lineare, statische Systeme

Die Lösung des LS-Problems (3.14) *lautet*

$$\boldsymbol{\theta}^* = \left(\boldsymbol{\Xi}^T\boldsymbol{\Xi}\right)^{-1}\boldsymbol{\Xi}^T\boldsymbol{\psi}, \tag{3.23}$$

sofern die Produktsummenmatrix $(\boldsymbol{\Xi}^T\boldsymbol{\Xi})$ *invertierbar*

$$\det(\boldsymbol{\Xi}^T\boldsymbol{\Xi}) \neq 0 \tag{3.24}$$

und die Hesse-Matrix $\frac{\partial^2 J}{\partial \boldsymbol{\theta}^2}$ *positiv definit ist, d. h. für Eigenwerte* λ_i *gilt*

$$\lambda_i > 0 \quad i = [1, \ldots, m]\,. \tag{3.25}$$

Die Kostenfunktion (3.13) *ergibt sich dann zu:*

$$J(\boldsymbol{\theta}^*) = \boldsymbol{\psi}^T\boldsymbol{\psi} - (\boldsymbol{\theta}^*)^T\boldsymbol{\Xi}^T\boldsymbol{\psi} = \left(\boldsymbol{\psi}^T - \hat{\boldsymbol{y}}^T\right)\boldsymbol{\psi}. \tag{3.26}$$

Entsprechend Abb. 3.1 gilt es allerdings zu berücksichtigen, dass die Messung nicht exakt den Prozessausgang darstellt, sondern durch einen Rauschterm $\boldsymbol{\psi} = \boldsymbol{y} + \boldsymbol{\nu}$ gestört wird. Umstellen von (3.23) führt dann zu

$$\begin{aligned}\boldsymbol{\theta}^* &= \left(\boldsymbol{\Xi}^T\boldsymbol{\Xi}\right)^{-1}\boldsymbol{\Xi}^T(\boldsymbol{y} + \boldsymbol{\nu}) \\ &= \boldsymbol{\theta} + \left(\boldsymbol{\Xi}^T\boldsymbol{\Xi}\right)^{-1}\boldsymbol{\Xi}^T\boldsymbol{\nu}.\end{aligned} \tag{3.27}$$

Sind $\boldsymbol{\nu}$ und $\boldsymbol{\Xi}$ unkorreliert und es gelte $\mathrm{E}(\boldsymbol{\nu}) = 0$, dann folgt für die Schätzung $\boldsymbol{\theta}^*$:

$$\mathrm{E}(\boldsymbol{\theta}^*)\Big|_{n\to\infty} = \boldsymbol{\theta} + \mathrm{E}(\left(\boldsymbol{\Xi}^T\boldsymbol{\Xi}\right)^{-1}\boldsymbol{\Xi}^T)\mathrm{E}(\boldsymbol{\nu}) = \boldsymbol{\theta}. \tag{3.28}$$

Parameter	Wert	Parameter	Wert
Masse	$m = 1500$ kg	Erdbeschleunigung	$g = 9,81\ \frac{\mathrm{m}}{\mathrm{s}}$
Rollwiderstandsbeiwert	$c_r = 0,015$	Lagerreibungskoeffizient	$\eta_v = 4,5\ \frac{\mathrm{kg}}{\mathrm{s}}$
Luftwiderstandsbeiwert	$c_w = 0,35$	Fahrzeugfläche	$A_f = 2,0$ m
Luftdichte	$\rho_l = 1,29\ \frac{\mathrm{kg}}{\mathrm{m}^3}$	Fahrbahnsteigung	$\gamma = 0\ ^\circ$

Tab. 3.1: Fahrzeugkennwerte für laterale Widerstandskraftberechnung

Demnach müssen unendlich viele Messpunkte aufgenommen werden, damit der Erwartungswert des Parametervektors gegen den wahren Wert konvergiert. Da dies in realen technischen Anwendungen allerdings nicht praktikabel bzw. realisierbar ist, resultieren Parameterabweichungen. Dieser Umstand wird in Abb. 3.3 sowie Abb. 3.4 für das einleitende Beispiel des Elektrofahrzeugs verdeutlicht. In den Abbildungen wird die Methode der kleinsten Quadrate für die drei gesuchten Parameter durchgeführt, wobei die wahren Parameter sowie die weiteren unabhängigen Variablen in Tab. 3.1 zusammengefasst sind. Obwohl in beiden Versuchsreihen das gleiche Messrauschen modelliert wurde, sind signifikante Unterschiede zwischen dem geschätzten Parametervektor $\hat{\boldsymbol{\theta}} = \boldsymbol{\theta}^*$ und dem entsprechenden Modell zu den tatsächlichen Prozesswerten

festzustellen. Aus Sicht der Identifikationsgenauigkeit ist somit ein möglichst hoher Stichprobenumfang wünschenswert.

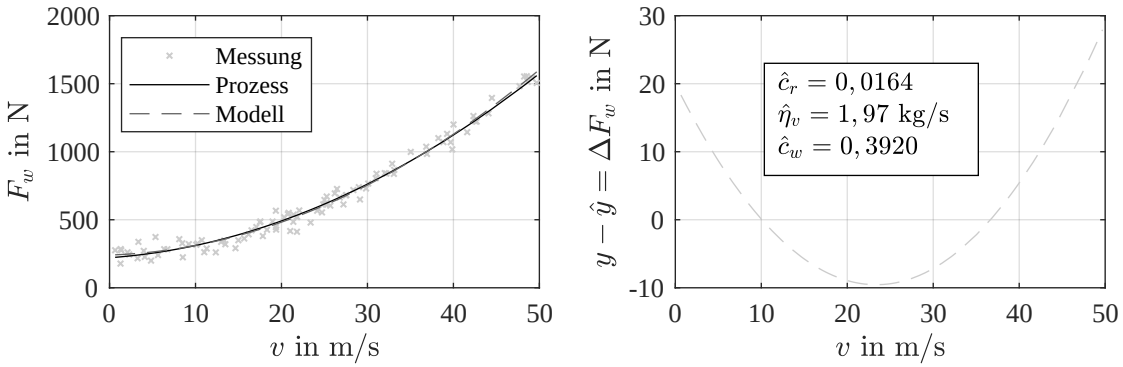


Abb. 3.3: Ergebnis des LS-Problems aus Beispiel (3.7) für $n = 100$ Messspunkte und einem Messrauschen $\nu \sim \mathcal{N}(0, \sigma = 50\ \mathrm{N})$

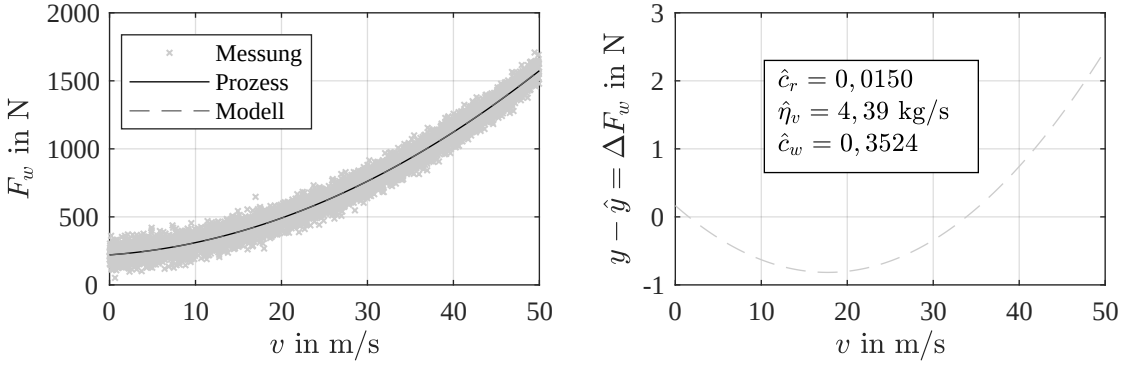


Abb. 3.4: Ergebnis des LS-Problems aus Beispiel (3.7) für $n = 100.000$ Messspunkte und einem Messrauschen $\nu \sim \mathcal{N}(0, \sigma = 50\ \mathrm{N})$

Während das vorangegangene Beispiel ein SISO-System darstellte (Geschwindigkeit v als Eingang, Kraft F_w als Ausgang), kann die Methode der kleinsten Quadrate auch direkt auf MISO-Systeme angewandt werden[1]. Da der Eingangsvektor $\boldsymbol{u}$ als bekannt vorausgesetzt wird, findet eine direkte Überführung in die Regressor-Matrix statt, sodass es unerheblich ist, ob eine oder mehrere Eingangsgröße betrachtet werden. Dies soll am Prozess

$$y(u_1, u_2, \boldsymbol{\theta}) = u_1\theta_1 + u_1^2\theta_2 + \sin\left(0,05 \cdot (u_1^2 + u_2^2)\right)\theta_3 + u_2\theta_4 + e^{u_2}\theta_5 \tag{3.29}$$

verdeutlicht werden[2]. Dieser Prozess ist erneut nichtlinear in den Eingangsgrößen u_1 und u_2,

[1] Bei MIMO-Systemen wird i. d. R. ein sequentielles Vorgehen gewählt, d. h., das Problem wird in mehrere SISO- bzw. MISO-Systeme unterteilt und dann durch wiederholtes Anwenden der LS-Methode gelöst.

[2] Das Prozessmodell wurde [Hol18] entnommen.

aber linear im gesuchten Parametervektor $\boldsymbol{\theta}$. Die Regressionsgleichung

$$\begin{aligned}\boldsymbol{e} &= \boldsymbol{\psi} - \boldsymbol{\Xi}\boldsymbol{\theta} \\ &= \begin{bmatrix}\psi_1 \\ \vdots \\ \psi_n\end{bmatrix} - \begin{bmatrix}u_{1,1} & u_{1,1}^2 & \sin(0,05(u_{1,1}^2+u_{2,1}^2)) & u_{2,1} & e^{u_{2,1}} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ u_{1,n} & u_{1,n}^2 & \sin(0,05(u_{1,n}^2+u_{2,n}^2)) & u_{2,n} & e^{u_{2,n}}\end{bmatrix}\begin{bmatrix}\theta_1 \\ \vdots \\ \theta_5\end{bmatrix}\end{aligned} \tag{3.30}$$

kann somit direkt aufgestellt werden. Die Lösung erfolgt wiederum mit (3.23), sofern die Messpunkte derart gewählt wurden, dass (3.24) gilt. Das resultierende Ergebnis ist in Tab. 3.2 sowie Abb. 3.5 dargestellt. Analog zum vorangegangenen Beispiel werden die Parameter aufgrund des modellierten Rauschens nur mit einer gewissen Abweichung geschätzt. Nichtsdestotrotz kann der Prozess über das identifizierte Modell mit guter Genauigkeit abgebildet werden.

	θ_1	θ_2	θ_3	θ_4	θ_5
Wahre Werte θ_i	5	0,1	25	-5	0,00001
Identifizierte Werte $\hat{\theta}_i$	4,953	0,1026	25,5413	-4,9557	0,0000095

Tab. 3.2: Parameterwerte für Prozess (3.29) mit $n = 225$ Messpunkten

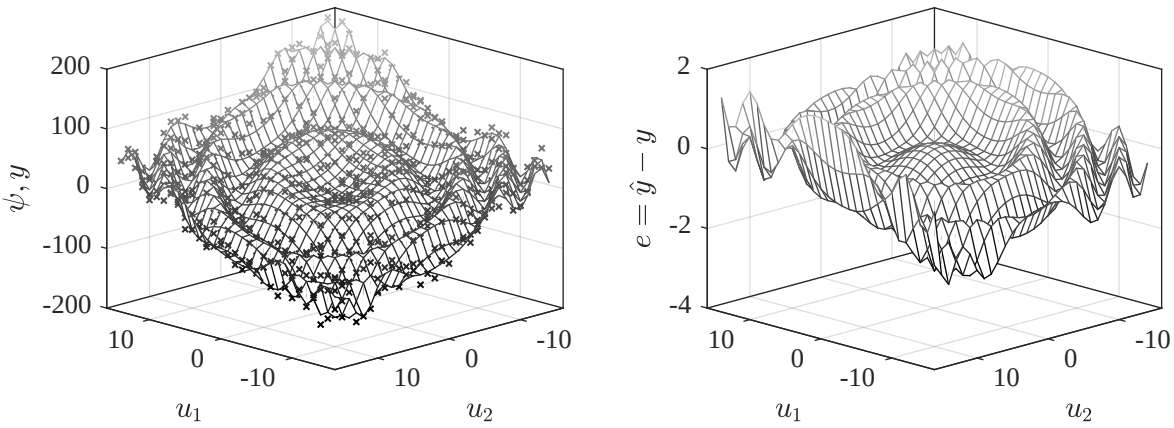


Abb. 3.5: Ergebnis des LS-Problems für den Prozess (3.29) mit $n = 225$ Messpunkten und einem Messrauschen $\nu \sim \mathcal{N}(0, \sigma = 10)$

3.1.1 Eigenschaften des LS-Schätzers

Im Folgenden werden einige wichtige Eigenschaften des zuvor eingeführten LS-Schätzers diskutiert. In diesem Kontext werden auch die wesentlichen Annahmen des LS-Verfahren beleuchtet.

Erwartungswert / Bias
Damit der Erwartungswert des LS-Schätzers überhaupt berechnet werden kann, muss folgende Annahme erfüllt sein (siehe auch Satz 3.2):

A1 : Die Produktsummenmatrix ist invertierbar, d. h. $\det(\boldsymbol{\Xi}^T\boldsymbol{\Xi}) \neq 0$.

Für den Erwartungswert des geschätzten Parametervektors gilt dann:

$$\begin{aligned} \mathrm{E}(\hat{\boldsymbol{\theta}}) &= \mathrm{E}\left(\left(\boldsymbol{\Xi}^T\boldsymbol{\Xi}\right)^{-1}\boldsymbol{\Xi}^T\boldsymbol{\psi}\right) \\ &= \mathrm{E}\left(\left(\boldsymbol{\Xi}^T\boldsymbol{\Xi}\right)^{-1}\boldsymbol{\Xi}^T(\boldsymbol{y}+\boldsymbol{\nu})\right) \\ &= \boldsymbol{\theta} + \mathrm{E}\left(\left(\boldsymbol{\Xi}^T\boldsymbol{\Xi}\right)^{-1}\boldsymbol{\Xi}^T\boldsymbol{\nu}\right). \end{aligned} \tag{3.31}$$

Hierauf wird folgende Annahme getroffen:

A2 : Das Rauschen $\boldsymbol{\nu}$ und die Regressionsmatrix $\boldsymbol{\Xi}$ seien unkorreliert.

Dann kann der Erwartungswert in (3.31) weiter aufgespalten werden:

$$\mathrm{E}(\hat{\boldsymbol{\theta}}) = \boldsymbol{\theta} + \mathrm{E}\left(\left(\boldsymbol{\Xi}^T\boldsymbol{\Xi}\right)^{-1}\boldsymbol{\Xi}^T\right)\mathrm{E}\left(\boldsymbol{\nu}\right). \tag{3.32}$$

Des Weiteren sei die in Definition 3.1 bereits eingeführte Annahme hinsichtlich $\boldsymbol{\nu}$ wiederholt:

A3 : Das Messrauschen sei mittelwertfrei, d. h. $\mathrm{E}\left(\boldsymbol{\nu}\right) = 0$.

Dann folgt unmittelbar durch Einsetzen in (3.32):

Satz 3.3: Bias-Freiheit des LS-Schätzers

Für das LS-Problem entsprechend Definition 3.1 liefert die Lösung (3.23) *ein biasfreies Ergebnis*

$$\mathrm{E}(\hat{\boldsymbol{\theta}}) = \boldsymbol{\theta}, \tag{3.33}$$

falls die Annahmen A1, A2 und A3 gelten.

Demgegenüber sei betont, dass jeder Mittelwert in $\boldsymbol{\nu}$ entsprechend (3.32) zu einem Abweichungsfehler (auch Offset genannt) hinsichtlich des gesuchten Parametervektors führt.

Varianz

Die Kovarianzmatrix des geschätzten Parametervektor ist definiert als

$$\begin{aligned} \mathrm{Cov}(\hat{\boldsymbol{\theta}},\hat{\boldsymbol{\theta}}) &= \mathrm{E}\left(\left(\hat{\boldsymbol{\theta}} - \mathrm{E}(\hat{\boldsymbol{\theta}})\right)\left(\hat{\boldsymbol{\theta}} - \mathrm{E}(\hat{\boldsymbol{\theta}})\right)^T\right) \\ &= \mathrm{E}\left(\left(\left(\boldsymbol{\Xi}^T\boldsymbol{\Xi}\right)^{-1}\boldsymbol{\Xi}^T\boldsymbol{\psi} - \mathrm{E}(\hat{\boldsymbol{\theta}})\right)\left(\left(\boldsymbol{\Xi}^T\boldsymbol{\Xi}\right)^{-1}\boldsymbol{\Xi}^T\boldsymbol{\psi} - \mathrm{E}(\hat{\boldsymbol{\theta}})\right)^T\right). \end{aligned} \tag{3.34}$$

Unter Verwendung der Annahmen A2 und A3 folgt:

$$\begin{aligned} \mathrm{Cov}(\hat{\boldsymbol{\theta}},\hat{\boldsymbol{\theta}}) &= \mathrm{E}\left(\left(\left(\boldsymbol{\Xi}^T\boldsymbol{\Xi}\right)^{-1}\boldsymbol{\Xi}^T(\boldsymbol{y}+\boldsymbol{\nu}) - \boldsymbol{\theta}\right)\left(\left(\boldsymbol{\Xi}^T\boldsymbol{\Xi}\right)^{-1}\boldsymbol{\Xi}^T(\boldsymbol{y}+\boldsymbol{\nu}) - \boldsymbol{\theta}\right)^T\right) \\ &= \mathrm{E}\left(\left(\boldsymbol{\theta} + \left(\boldsymbol{\Xi}^T\boldsymbol{\Xi}\right)^{-1}\boldsymbol{\Xi}^T\boldsymbol{\nu} - \boldsymbol{\theta}\right)\left(\boldsymbol{\theta} + \left(\boldsymbol{\Xi}^T\boldsymbol{\Xi}\right)^{-1}\boldsymbol{\Xi}^T\boldsymbol{\nu} - \boldsymbol{\theta}\right)^T\right) \\ &= \mathrm{E}\left(\left(\left(\boldsymbol{\Xi}^T\boldsymbol{\Xi}\right)^{-1}\boldsymbol{\Xi}^T\boldsymbol{\nu}\right)\left(\left(\boldsymbol{\Xi}^T\boldsymbol{\Xi}\right)^{-1}\boldsymbol{\Xi}^T\boldsymbol{\nu}\right)^T\right) \\ &= \mathrm{E}\left(\left(\boldsymbol{\Xi}^T\boldsymbol{\Xi}\right)^{-1}\boldsymbol{\Xi}^T\boldsymbol{\nu}\boldsymbol{\nu}^T\boldsymbol{\Xi}\left(\left(\boldsymbol{\Xi}^T\boldsymbol{\Xi}\right)^{-1}\right)^T\right). \end{aligned} \tag{3.35}$$

Da die Regressionsmatrix keine Zufallsvariable darstellt, sondern bekannte (Mess-)Werte beinhaltet, kann diese aus Sicht der Berechnung des Erwartungswerts als Konstante behandelt werden. Ferner wurde in A2 angenommen, dass die Regressionsmatrix und das Messrauschen unkorreliert seien. Daher gilt für (3.35):

$$\mathrm{Cov}(\hat{\boldsymbol{\theta}}, \hat{\boldsymbol{\theta}}) = \left(\boldsymbol{\Xi}^T\boldsymbol{\Xi}\right)^{-1}\boldsymbol{\Xi}^T\mathrm{E}\left(\boldsymbol{\nu}\boldsymbol{\nu}^T\right)\boldsymbol{\Xi}\left(\left(\boldsymbol{\Xi}^T\boldsymbol{\Xi}\right)^{-1}\right)^T. \tag{3.36}$$

Des Weiteren sei eine weitere Annahme aus Definition 3.1 hinsichtlich $\boldsymbol{\nu}$ wiederholt:

A4 : Das Messrauschen besitzt eine konstante Varianz: $\mathrm{Cov}(\nu) = \sigma_\nu^2$.

Dies in (3.36) eingesetzt ergibt:

Satz 3.4: Varianz des LS-Schätzers

Für das LS-Problem entsprechend Definition 3.1 liefert die Lösung (3.23) *ein Ergebnis mit der Varianz*

$$\begin{aligned}\mathrm{Cov}(\hat{\boldsymbol{\theta}}, \hat{\boldsymbol{\theta}}) &= \sigma_\nu^2\left(\boldsymbol{\Xi}^T\boldsymbol{\Xi}\right)^{-1}\boldsymbol{\Xi}^T\boldsymbol{\Xi}\left(\left(\boldsymbol{\Xi}^T\boldsymbol{\Xi}\right)^{-1}\right)^T \\ &= \sigma_\nu^2\left(\left(\boldsymbol{\Xi}^T\boldsymbol{\Xi}\right)^{-1}\right)^T \\ &= \sigma_\nu^2\left(\boldsymbol{\Xi}^T\boldsymbol{\Xi}\right)^{-1},\end{aligned} \tag{3.37}$$

falls die Annahmen A1, A2, A3 und A4 gelten.

Effizienz

Unter einem linearen Schätzer wird ein Schätzer der Form

$$\hat{\boldsymbol{\theta}} = \boldsymbol{M}\boldsymbol{\psi} \tag{3.38}$$

verstanden, d. h., der Schätzwert hängt linear von den Ausgangsdaten ab. Ohne Einschränkung der Allgemeinheit kann ein allgemeiner, linearer Schätzer daher durch

$$\hat{\boldsymbol{\theta}} = \boldsymbol{M}\boldsymbol{\psi} = \left(\left(\boldsymbol{\Xi}^T\boldsymbol{\Xi}\right)^{-1}\boldsymbol{\Xi}^T + \widetilde{\boldsymbol{\Xi}}\right)\boldsymbol{\psi} \tag{3.39}$$

dargestellt werden, wobei $\widetilde{\boldsymbol{\Xi}}$ ein frei wählbarer Verschiebungsterm ist, der als Auslegungsparameter des Schätzers interpretiert werden kann. Für den Erwartungswert dieses allgemeinen, linearen Schätzer folgt unter Berücksichtigung der Annahmen A2 und A3:

$$\begin{aligned}\mathrm{E}(\hat{\boldsymbol{\theta}}) &= \mathrm{E}\left(\left(\left(\boldsymbol{\Xi}^T\boldsymbol{\Xi}\right)^{-1}\boldsymbol{\Xi}^T + \widetilde{\boldsymbol{\Xi}}\right)\boldsymbol{\psi}\right) \\ &= \mathrm{E}\left(\left(\left(\boldsymbol{\Xi}^T\boldsymbol{\Xi}\right)^{-1}\boldsymbol{\Xi}^T + \widetilde{\boldsymbol{\Xi}}\right)(\boldsymbol{y} + \boldsymbol{\nu})\right) \\ &= \left(\left(\boldsymbol{\Xi}^T\boldsymbol{\Xi}\right)^{-1}\boldsymbol{\Xi}^T\boldsymbol{\Xi} + \widetilde{\boldsymbol{\Xi}}\boldsymbol{\Xi}\right)\boldsymbol{\theta} \\ &= \left(\boldsymbol{I} + \widetilde{\boldsymbol{\Xi}}\boldsymbol{\Xi}\right)\boldsymbol{\theta}.\end{aligned} \tag{3.40}$$

Soll der allgemeine, lineare Schätzer ebenfalls biasfrei sein, muss $\widetilde{\boldsymbol{\Xi}}\boldsymbol{\Xi} = \mathbf{0}$ gelten. Einsetzen von

(3.39) in (3.35) führt zu:

$$\begin{aligned}\mathrm{Cov}(\hat{\boldsymbol{\theta}},\hat{\boldsymbol{\theta}}) &= \mathrm{E}\left(\left(\left(\left(\boldsymbol{\Xi}^T\boldsymbol{\Xi}\right)^{-1}\boldsymbol{\Xi}^T+\widetilde{\boldsymbol{\Xi}}\right)(\boldsymbol{y}+\boldsymbol{\nu})-\boldsymbol{\theta}\right)\left(\left(\left(\boldsymbol{\Xi}^T\boldsymbol{\Xi}\right)^{-1}\boldsymbol{\Xi}^T+\widetilde{\boldsymbol{\Xi}}\right)(\boldsymbol{y}+\boldsymbol{\nu})-\boldsymbol{\theta}\right)^T\right)\\ &= \mathrm{E}\left(\left(\boldsymbol{\theta}+\widetilde{\boldsymbol{\Xi}}\boldsymbol{y}+\left(\left(\boldsymbol{\Xi}^T\boldsymbol{\Xi}\right)^{-1}\boldsymbol{\Xi}^T+\widetilde{\boldsymbol{\Xi}}\right)\boldsymbol{\nu}-\boldsymbol{\theta}\right)\left(\boldsymbol{\theta}+\widetilde{\boldsymbol{\Xi}}\boldsymbol{y}+\left(\left(\boldsymbol{\Xi}^T\boldsymbol{\Xi}\right)^{-1}\boldsymbol{\Xi}^T+\widetilde{\boldsymbol{\Xi}}\right)\boldsymbol{\nu}-\boldsymbol{\theta}\right)^T\right)\\ &= \mathrm{E}\left(\left(\widetilde{\boldsymbol{\Xi}}\left(\boldsymbol{\Xi}\boldsymbol{\theta}\right)+\left(\left(\boldsymbol{\Xi}^T\boldsymbol{\Xi}\right)^{-1}\boldsymbol{\Xi}^T+\widetilde{\boldsymbol{\Xi}}\right)\boldsymbol{\nu}\right)\left(\widetilde{\boldsymbol{\Xi}}\left(\boldsymbol{\Xi}\boldsymbol{\theta}\right)+\left(\left(\boldsymbol{\Xi}^T\boldsymbol{\Xi}\right)^{-1}\boldsymbol{\Xi}^T+\widetilde{\boldsymbol{\Xi}}\right)\boldsymbol{\nu}\right)^T\right).\end{aligned}$$

Unter Verwendung von $\widetilde{\boldsymbol{\Xi}}\boldsymbol{\Xi}=\mathbf{0}$ folgt:

$$\mathrm{Cov}(\hat{\boldsymbol{\theta}},\hat{\boldsymbol{\theta}}) = \mathrm{E}\left(\left(\left(\left(\boldsymbol{\Xi}^T\boldsymbol{\Xi}\right)^{-1}\boldsymbol{\Xi}^T+\widetilde{\boldsymbol{\Xi}}\right)\boldsymbol{\nu}\right)\left(\left(\left(\boldsymbol{\Xi}^T\boldsymbol{\Xi}\right)^{-1}\boldsymbol{\Xi}^T+\widetilde{\boldsymbol{\Xi}}\right)\boldsymbol{\nu}\right)^T\right). \tag{3.41}$$

Mittels der Annahme A4 ergibt sich dann:

$$\begin{aligned}\mathrm{Cov}(\hat{\boldsymbol{\theta}},\hat{\boldsymbol{\theta}}) &= \left(\left(\boldsymbol{\Xi}^T\boldsymbol{\Xi}\right)^{-1}\boldsymbol{\Xi}^T+\widetilde{\boldsymbol{\Xi}}\right)\mathrm{E}\left(\boldsymbol{\nu}\boldsymbol{\nu}^T\right)\left(\left(\boldsymbol{\Xi}^T\boldsymbol{\Xi}\right)^{-1}\boldsymbol{\Xi}^T+\widetilde{\boldsymbol{\Xi}}\right)^T\\ &= \sigma_\nu^2\left(\left(\boldsymbol{\Xi}^T\boldsymbol{\Xi}\right)^{-1}\boldsymbol{\Xi}^T+\widetilde{\boldsymbol{\Xi}}\right)\left(\boldsymbol{\Xi}\left(\boldsymbol{\Xi}^T\boldsymbol{\Xi}\right)^{-1}+\widetilde{\boldsymbol{\Xi}}^T\right).\end{aligned} \tag{3.42}$$

Ausmultiplizieren liefert schlussendlich folgenden Ausdruck:

$$\begin{aligned}\mathrm{Cov}(\hat{\boldsymbol{\theta}},\hat{\boldsymbol{\theta}}) &= \sigma_\nu^2\left(\left(\boldsymbol{\Xi}^T\boldsymbol{\Xi}\right)^{-1}+\underbrace{\widetilde{\boldsymbol{\Xi}}\boldsymbol{\Xi}}_{0}\left(\boldsymbol{\Xi}^T\boldsymbol{\Xi}\right)^{-1}+\left(\boldsymbol{\Xi}^T\boldsymbol{\Xi}\right)^{-1}\left(\underbrace{\widetilde{\boldsymbol{\Xi}}\boldsymbol{\Xi}}_{0}\right)^T+\widetilde{\boldsymbol{\Xi}}\widetilde{\boldsymbol{\Xi}}^T\right)\\ &= \sigma_\nu^2\left(\left(\boldsymbol{\Xi}^T\boldsymbol{\Xi}\right)^{-1}+\widetilde{\boldsymbol{\Xi}}\widetilde{\boldsymbol{\Xi}}^T\right).\end{aligned} \tag{3.43}$$

Es wird demnach deutlich, dass die Varianz des allgemeinen, linearen Schätzer für jedes $\widetilde{\boldsymbol{\Xi}}\neq\mathbf{0}$ stets größer ist als die Lösung des LS-Problems entsprechend Satz 3.2:

Satz 3.5: Effizienz des LS-Schätzers

Für das LS-Problem entsprechend Definition 3.1 liefert die Lösung (3.23) *ein Ergebnis mit minimaler Varianz*

$$\mathrm{Cov}(\hat{\boldsymbol{\theta}},\hat{\boldsymbol{\theta}}) = \sigma_\nu^2\left(\boldsymbol{\Xi}^T\boldsymbol{\Xi}\right)^{-1}, \tag{3.44}$$

falls die Annahmen A1 bis A4 gelten. Es gibt demnach keinen anderen linearen, biasfreien Schätzers, welcher eine geringere Parametervarianz aufweist.

Konsistenz

Abschließend soll die Varianz der Parameterschätzung in Bezug auf die Anzahl der genutzten Messpunkte n untersucht werden. Die Varianz des LS-Schätzers ist gegeben durch:

$$\mathrm{Cov}(\hat{\boldsymbol{\theta}},\hat{\boldsymbol{\theta}}) = \sigma_\nu^2\left(\boldsymbol{\Xi}^T\boldsymbol{\Xi}\right)^{-1}. \tag{3.45}$$

Nachfolgende Manipulation verändert die Gleichung nicht:

$$\mathrm{Cov}(\hat{\boldsymbol{\theta}},\hat{\boldsymbol{\theta}}) = \frac{1}{n}\sigma_\nu^2\left(\frac{1}{n}\boldsymbol{\Xi}^T\boldsymbol{\Xi}\right)^{-1}. \tag{3.46}$$

Grenzwertbildung liefert:

$$\lim_{n\to\infty} \mathrm{Cov}(\hat{\boldsymbol{\theta}}, \hat{\boldsymbol{\theta}}) = \lim_{n\to\infty} \left[\frac{1}{n}\sigma_\nu^2 \left(\frac{1}{n}\boldsymbol{\Xi}^T\boldsymbol{\Xi} \right)^{-1} \right]. \tag{3.47}$$

Der mittlere Term hierbei entspricht gerade:

$$\frac{1}{n}\boldsymbol{\Xi}^T\boldsymbol{\Xi} = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{\xi}_i\boldsymbol{\xi}_i^T. \tag{3.48}$$

Wird $\boldsymbol{X}_i = \boldsymbol{\xi}_i\boldsymbol{\xi}_i^T$ als Zufallsvariable interpretiert, kann folgender Satz herangezogen werden:

Satz 3.6: Schwaches Gesetz der großen Zahlen

Seien $\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots$ eine Folge unabhängiger und identisch verteilter Zufallsvariablen mit Erwartungswert $\boldsymbol{\mu}$ und Varianz $\boldsymbol{\sigma}^2$. Ferner sei

$$\overline{\boldsymbol{X}} = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{X}_i, \tag{3.49}$$

dann gilt für jedes $\epsilon > 0$:

$$\lim_{n\to\infty} P\left(|\overline{\boldsymbol{X}} - \boldsymbol{\mu}| > \epsilon \right) = 0. \tag{3.50}$$

Die Wahrscheinlichkeit, dass der Mittelwert der n Werte um weniger als einen beliebigen Wert ϵ vom Erwartungswert der Verteilung abweicht beträgt Null, d. h., der Mittelwert konvergiert in Wahrscheinlichkeit gegen den Erwartungswert.

Angewandt auf (3.48) bedeutet dies insbesondere:

$$\lim_{n\to\infty} \frac{1}{n}\boldsymbol{\Xi}^T\boldsymbol{\Xi} \to \text{begrenzt}. \tag{3.51}$$

Dieses wiederum führt zu:

$$\lim_{n\to\infty} \left[\frac{1}{n}\sigma_\nu^2 \left(\frac{1}{n}\boldsymbol{\Xi}^T\boldsymbol{\Xi} \right)^{-1} \right] = 0. \tag{3.52}$$

Somit konnte gezeigt werden:

Satz 3.7: Konsistenz des LS-Schätzers

Für das LS-Problem entsprechend Definition 3.1 liefert die Lösung (3.23) einen konsistenten Schätzer, da dieser biasfrei ist und dessen Varianz für $n \to \infty$ gegen Null konvergiert. Es folgt

$$\lim_{n\to\infty} \hat{\boldsymbol{\theta}}(n) = \boldsymbol{\theta}, \tag{3.53}$$

falls die Annahmen A1 bis A4 gelten.

Somit konnte eine Reihe interessanter Eigenschaften des LS-Schätzers bewiesen werden. Diese werden in der Literatur auch unter dem Term BLUE (*best linear unbiased estimator*) zusammengefasst.

Allerdings werden die Annahmen A1 bis A4 in realen Anwendungen häufig nicht eingehalten. Hier sei insbesondere das Messrauschen zu nennen, welches durch parasitäre Einflüsse und Be-

schränkungen der verfügbaren Messmittel i. d. R. weder biasfrei ist noch eine konstante Varianz aufweist. Insofern müssen obige Eigenschaften des LS-Schätzers in realen Applikationen kritisch hinterfragt werden. Folglich kommt der Genauigkeitsanalyse nach erfolgter Identifikation eine wichtige Rolle zu, um ein gefundenes Identifikationsergebnis bzw. mehrere Identifikationsvorgänge in Relation zueinander praxisorientiert bewerten zu können. Siehe hierzu das noch folgende Kap. 3.1.3.

3.1.2 Multikollinearität

In Satz 3.2 wurde hervorgehoben, dass eine Lösung des LS-Problems grundsätzlich dann gefunden werden kann, wenn die Produktsummenmatrix ($\boldsymbol{\Xi}^T\boldsymbol{\Xi}$) invertierbar ist. Darüber hinaus stellt sich allerdings die Frage, ob die verwendete Regressormatrix eine ausreichende bzw. gute Datenbasis abbildet. Ein häufiges Problem in diesem Kontext ist die sog. Multikollinearität. Diese liegt vor, wenn zwei oder mehr Regressoren eine sehr starke Korrelation miteinander haben. Zum einen wird mit zunehmender Multikollinearität das Verfahren zur Parameteridentifikation zunehmend ungenau und zum anderen wird das LS-Problem ggf. nicht mehr lösbar, sofern die Multikollinearität derart stark ausgeprägt ist, dass ($\boldsymbol{\Xi}^T\boldsymbol{\Xi}$) nicht mehr invertierbar ist.

Beispiel:
Gegeben sei ein einfaches lineares Regressionsmodell

$$y = a_1 x_1 + a_2 x_2 \tag{3.54}$$

mit der Ausgangsgröße y, den unbekannten Parametern a_1 und a_2 sowie den Regressoren x_1 und x_2. Wird während der Messwertaufnahme ungünstigerweise ein linearer Zusammenhang zwischen den Regressoren hergestellt, also

$$\begin{aligned} x_2 &= b_0 + b_1 x_1 \quad \text{bzw.} \\ x_1 &= \frac{1}{b_1} x_2 - \frac{b_0}{b_1}, \end{aligned} \tag{3.55}$$

dann folgt für das Regressionsmodell:

$$y = a_2 b_0 + (a_1 + a_2 b_1)\, x_1 \quad \text{bzw.} \tag{3.56a}$$

$$y = -\frac{a_1 b_0}{b_1} + \left(a_2 + \frac{a_1}{b_1}\right) x_2. \tag{3.56b}$$

In (3.56a) hängt y dann nur noch von x_1 ab und in (3.56b) nur noch von x_2. In diesem Fall der perfekten Multikollinearität ist eine eindeutige Identifikation der Modellparameter daher nicht mehr möglich. Unabhängig von diesem einfachen Beispiel sollte bei einem Vorliegen signifikanter Multikollinearität die weitere Identifikation unterbrochen und zunächst Schritte zur Reduktion der Multikollinearität unternommen werden, z. B.

- die Veränderung der zu identifizierenden Modellstruktur und/oder
- die Anpassung der Datenbasis (z. B. ergänzende Messungen).

Identifikation von Multikollinearität:
Weil empirische Daten immer einen gewissen Grad an Multikollinearität aufweisen, gilt es Kenn-

zahlen heranzuziehen, die Hinweise auf Multikollinearität liefern. Ein einfacher Zugang hierzu ist die Korrelationsmatrix der Regressormatrix

$$\mathrm{Corr}(\boldsymbol{\Xi}^T\boldsymbol{\Xi}) = \begin{bmatrix} 1 & \rho_{12} & \cdots & \rho_{1m} \\ \rho_{21} & 1 & \cdots & \rho_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ \rho_{m1} & \rho_{m2} & \cdots & 1 \end{bmatrix}$$

entsprechend (2.49). Sehr hohe positive oder negative Korrelationskoeffizienten zeigen einen starken Zusammenhang zwischen den Regressoren und damit Multikollinearität an. Wenn es lineare Abhängigkeiten zwischen mehreren Variablen gibt, ist dies in der einfachen Korrelationsmatrix nicht mehr unbedingt erkennbar, sodass betragsmäßig kleine Korrelationskoeffizienten nicht zwangsläufig eine signifikante Multikollinearität ausschließen. Als ergänzende Kennzahl kann daher der Konditionsindex herangezogen werden:

Definition 3.2: Konditionsindex

Ist die Produktsummenmatrix $(\boldsymbol{\Xi}^T\boldsymbol{\Xi})$ *positiv definit, d. h. alle Eigenwerte* λ_i *der Matrix sind positiv, ist das LS-Problem grundsätzlich lösbar und der Konditionsindex berechnet sich wie folgt:*

$$\mathrm{KI}_j = \sqrt{\frac{\lambda_j}{\min_i \lambda_i}}. \tag{3.57}$$

In der Literatur findet sich häufig die Angabe, dass ein Konditionsindex von $\mathrm{KI}_j > 30$ ebenfalls auf eine starke Multikollinearität hinweise – dennoch ist auch dies nur eine grobe Faustformel. Im Zuge der Messwertaufnahme bzw. Modellstrukturierung sollten Maßnahmen genutzt werden, um den Konditionsindex möglichst gering zu halten. Sofern $(\boldsymbol{\Xi}^T\boldsymbol{\Xi})$ eine normale Matrix[1] darstellt, dann entspricht das Quadrat des größten Konditionsindex auch unmittelbar der Kondition κ der Produktsummenmatrix:

$$\kappa = \left\|(\boldsymbol{\Xi}^T\boldsymbol{\Xi})\right\|_2 = \left(\max_j \{\mathrm{KI}_j\}\right)^2 = \left|\frac{\max_i \lambda_i}{\min_i \lambda_i}\right|. \tag{3.58}$$

Wenn die Kondition der Matrix nahe Eins ist, ist die Matrix gut konditioniert, was bedeutet, dass ihre Inverse mit guter Genauigkeit numerisch berechnet werden kann. Wenn die Kondition der Matrix hingegen sehr groß ist, dann gilt die Matrix als schlecht konditioniert. Praktisch ist eine solche Matrix fast singulär, d. h., die Berechnung ihrer Inversen ist anfällig für große numerische Fehler. Eine Matrix, die nicht invertierbar ist, hat eine Kondition, die gleich Unendlich ist. Einige praktische Hinweise zur Lösung des LS-Problems bei schlecht konditionierter Regressormatrix werden noch in Kap. 3.1.4 gegeben.

3.1.3 Genauigkeitsbetrachtung

Nach erfolgter Identifikation steht häufig die Frage im Raum, mit welcher Genauigkeit die gesuchten Parameter berechnet wurden. Hierzu können verschiedene Kenngrößen herangezogen werden, welche im Folgenden in verschiedene Kategorien eingeteilt werden.

[1] Eine reelle Matrix $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ wird normal genannt, wenn gilt: $A^TA = AA^T$.

Varianz-basierte Kenngrößen:
Häufig wird die Kovarianzmatrix des identifizierten Parametervektors

$$\mathrm{Cov}(\hat{\boldsymbol{\theta}}) = \mathrm{E}\left((\hat{\boldsymbol{\theta}} - \boldsymbol{\theta})(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta})^T\right) = \begin{bmatrix} \sigma_1^2 & \sigma_{12}^2 & \cdots & \sigma_{1m}^2 \\ \sigma_{21}^2 & \sigma_2^2 & \cdots & \sigma_{2m}^2 \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{m1}^2 & \sigma_{m2}^2 & \cdots & \sigma_m^2 \end{bmatrix} \tag{3.59}$$

zur Genauigkeitsbetrachtung herangezogen. Auf der Hauptdiagonalen von (3.59) stehen die Parametervarianzen, d. h. je kleiner diese ausfallen, desto sicherer bzw. genauer wurde der jeweilige Parameter θ_i ermittelt. Ferner können auf Basis der Kovarianzmatrix die Korrelationskoeffizienten berechnet werden:

$$\rho(\hat{\theta}_i, \hat{\theta}_j) = \frac{\sigma_{ij}^2}{\sigma_i \sigma_j}. \tag{3.60}$$

Wie in Kap. 2.3.1 erläutert, kann ρ als normierte Größe Werte zwischen $[-1, 1]$ annehmen. Absolute Werte nahe Eins bedeuten, dass die betrachteten Parameter stark miteinander verknüpft sind. Sie beschreiben im Prozessmodell ähnliche Effekte und können daher nur unzureichend separiert werden, sodass eine genaue und unabhängige Identifizierung unwahrscheinlich ist.

Allerdings stellt sich die Frage, wie (3.59) zu berechnen ist, da die wahren Werte für $\boldsymbol{\theta}$ offensichtlich unbekannt sind. Hierzu wird (3.59) mittels

$$\begin{aligned} \boldsymbol{\theta} &= \left(\boldsymbol{\Xi}^T \boldsymbol{\Xi}\right)^{-1} \boldsymbol{\Xi}^T \boldsymbol{y}, \\ \hat{\boldsymbol{\theta}} &= \left(\boldsymbol{\Xi}^T \boldsymbol{\Xi}\right)^{-1} \boldsymbol{\Xi}^T \boldsymbol{\psi} \end{aligned} \tag{3.61}$$

zunächst umgeschrieben zu:

$$\begin{aligned} \mathrm{Cov}(\hat{\boldsymbol{\theta}}) &= \mathrm{E}\left(\left(\left(\boldsymbol{\Xi}^T \boldsymbol{\Xi}\right)^{-1} \boldsymbol{\Xi}^T (\boldsymbol{\psi} - \boldsymbol{y})\right)\left(\left(\boldsymbol{\Xi}^T \boldsymbol{\Xi}\right)^{-1} \boldsymbol{\Xi}^T (\boldsymbol{\psi} - \boldsymbol{y})\right)^T\right) \\ &= \mathrm{E}\left(\left(\boldsymbol{\Xi}^T \boldsymbol{\Xi}\right)^{-1} \boldsymbol{\Xi}^T (\boldsymbol{\psi} - \boldsymbol{y})(\boldsymbol{\psi} - \boldsymbol{y})^T \boldsymbol{\Xi} \left(\boldsymbol{\Xi}^T \boldsymbol{\Xi}\right)^{-1}\right). \end{aligned} \tag{3.62}$$

Da die Regressionsmatrix $\boldsymbol{\Xi}$ als unabhängige Variable eine bekannte Konstante darstellt, folgt:

$$\mathrm{Cov}(\hat{\boldsymbol{\theta}}) = \left(\boldsymbol{\Xi}^T \boldsymbol{\Xi}\right)^{-1} \boldsymbol{\Xi}^T \mathrm{E}\left(\boldsymbol{e}\boldsymbol{e}^T\right) \boldsymbol{\Xi} \left(\boldsymbol{\Xi}^T \boldsymbol{\Xi}\right)^{-1} \tag{3.63}$$

mit $\boldsymbol{e} = \boldsymbol{\psi} - \boldsymbol{y}$ als Residuum. Weiterhin entspricht $\mathrm{E}\left(\boldsymbol{e}\boldsymbol{e}^T\right)$ gerade der Varianz des additiven Messrauschens, welches in Definition 3.1 als konstant angenommen wurde:

$$\mathrm{E}\left(\boldsymbol{e}\boldsymbol{e}^T\right) = \mathrm{Cov}\left(\nu\right) \boldsymbol{I} = \sigma^2 \boldsymbol{I}. \tag{3.64}$$

Einsetzen in (3.63) und ausmultiplizieren führt zu:

$$\mathrm{Cov}(\hat{\boldsymbol{\theta}}) = \sigma^2 \left(\boldsymbol{\Xi}^T \boldsymbol{\Xi}\right)^{-1}. \tag{3.65}$$

In realen Anwendungen ist der wahre Wert σ^2 zur Charakterisierung des Messrauschens i. A.

nicht bekannt. Daher muss dieser ebenfalls geschätzt werden:

$$\sigma^2 \approx \hat{\sigma}^2 = \frac{1}{n-m} \sum_{k=1}^{n} \left(\psi[k] - \hat{y}[k]\right)^2 . \tag{3.66}$$

Für das Fahrzeugbeispiel in (3.7) ergeben sich folgende Abschätzungen:

$$\hat{\sigma}\big|_{n=10^2} = 46,64 \text{ N}, \quad \hat{\sigma}\big|_{n=10^5} = 49,77 \text{ N}. \tag{3.67}$$

Darauf aufbauend können folgende Kovarianzmatrizen

$$\begin{aligned} \text{Cov}(\hat{\boldsymbol{\theta}})\big|_{n=10^2} &= \begin{bmatrix} 8,85 \cdot 10^{-7} & -0,001 \text{ kg/s} & 1,33 \cdot 10^{-5} \\ -0,001 \text{ kg/s} & 1,57 \text{ (kg/s)}^2 & -0,023 \text{ kg/s} \\ 1,33 \cdot 10^{-5} & -0,023 \text{ kg/s} & 3,67 \cdot 10^{-4} \end{bmatrix} \\ \text{Cov}(\hat{\boldsymbol{\theta}})\big|_{n=10^5} &= \begin{bmatrix} 9,89 \cdot 10^{-9} & -1,19 \cdot 10^{-5} \text{ kg/s} & 1,53 \cdot 10^{-7} \\ -1,19 \cdot 10^{-5} \text{ kg/s} & 0,019 \text{ (kg/s)}^2 & -2,78 \cdot 10^{-4} \text{ kg/s} \\ 1,53 \cdot 10^{-7} & -2,78 \cdot 10^{-4} \text{ kg/s} & 4,25 \cdot 10^{-6} \end{bmatrix} \end{aligned} \tag{3.68}$$

gewonnen werden. Auch hier wird schnell deutlich, dass der größere Stichprobenumfang die Genauigkeit hinsichtlich der Abschätzung der Rauschvarianz sowie der Kovarianzmatrix der Parameter signifikant erhöht. Um die Varianzen der Parameter besser vergleichen zu können, ist eine Normalisierung zielführend: Auch hier sticht heraus, dass ein großer Stichprobenumfang

	c_r	η_v	c_w
$\frac{\hat{\sigma}_i}{\theta_i}\big\|_{n=10^2}$ in %	6,27	31,3	6,39
$\frac{\hat{\sigma}_i}{\theta_i}\big\|_{n=10^5}$ in %	0,67	3,43	0,69

Tab. 3.3: Normalisierte Varianzen der geschätzten Parameter

die Unsicherheit hinsichtlich der Identifikationsgenauigkeit maßgeblich reduziert. Ferner scheint η_v mit der vergleichsweise größten Unsicherheit behaftet zu sein. Um dies näher zu betrachten, wird die Korrelationsmatrix

$$\begin{aligned} \text{Corr}(\hat{\boldsymbol{\theta}})\big|_{n=10^2} &= \begin{bmatrix} 1 & -0,87 & 0,74 \\ -0,87 & 1 & -0,96 \\ 0,74 & -0,96 & 1 \end{bmatrix}, \\ \text{Corr}(\hat{\boldsymbol{\theta}})\big|_{n=10^5} &= \begin{bmatrix} 1 & -0,86 & 0,74 \\ -0,86 & 1 & -0,97 \\ 0,74 & -0,97 & 1 \end{bmatrix} \end{aligned} \tag{3.69}$$

entsprechend (2.49) gebildet. Die Korrelationskoeffizienten sind im gegebenen Beispiel vergleichsweise robust gegenüber dem Stichprobenumfang. Auch wird bestätigt, dass η_v sehr stark mit den anderen Parametern (in einem linearen Kontext) verbunden ist, was die korrekte Identifikation erschwert.

Summe der Abweichungsquadrate:
Die Summe der Abweichungsquadrate, auch schlicht Summe der Quadrate (*sum of squares*) ge-

nannt, ist die Summe der quadratischen Abweichungen der Messwerte von ihrem arithmetischen Mittel, auch Gesamtabweichungsquadratsumme genannt:

$$\begin{aligned} \mathrm{SQT} &= \sum_{k=1}^{n} (\psi[k] - \overline{\psi})^2 \qquad \text{mit} \quad \overline{\psi} = \frac{1}{n} \sum_{k=1}^{n} \psi[k] \\ &= \sum_{k=1}^{n} \psi^2[k] - 2\overline{\psi} \sum_{k=1}^{n} \psi[k] + \overline{\psi}^2 \sum_{k=1}^{n} 1 = \boldsymbol{\psi}^T \boldsymbol{\psi} - 2n\overline{\psi}^2 + n\overline{\psi}^2 \\ &= \boldsymbol{\psi}^T \boldsymbol{\psi} - n\overline{\psi}^2. \end{aligned} \tag{3.70}$$

Diese bildet die empirische Varianz des Prozesses ab. Dem gegenüber modelliert die erklärte Abweichungsquadratsumme (*explained sum of squares*) den empirischen Varianzanteil, der durch das Modell abgebildet wird:

$$\begin{aligned} \mathrm{SQE} &= \sum_{k=1}^{n} (\hat{y}[k] - \overline{\psi})^2 \\ &= \sum_{k=1}^{n} \hat{y}^2[k] - 2\overline{\psi} \sum_{k=1}^{n} \hat{y}[k] + \overline{\psi}^2 \sum_{k=1}^{n} 1 \\ &= (\boldsymbol{\Xi}\boldsymbol{\theta})^T \boldsymbol{\Xi}\boldsymbol{\theta} - 2\overline{\psi} \sum_{k=1}^{n} (\psi[k] - \nu[k]) + n\overline{\psi}^2 \\ &= \boldsymbol{\theta}^T \boldsymbol{\Xi}^T \boldsymbol{\Xi}\boldsymbol{\theta} - 2n\overline{\psi}^2 + n\overline{\psi}^2 \\ &= \hat{\boldsymbol{y}}^T \hat{\boldsymbol{y}} - n\overline{\psi}^2. \end{aligned} \tag{3.71}$$

In obiger Umformung wird hinsichtlich des Messrauschens ν entsprechend Definition 3.1 angenommen, dass dieses mittelwertfrei ist, sodass der Mittelwert des Modellausgangs $\hat{y}$ dem Mittelwert des Prozessausgangs ψ entspricht. Ferner wird noch die Residuenquadratsumme (*residual sum of squares*) eingeführt, welche die empirische Varianz der Modellabweichungen abbildet und somit ein Maß für die Modellzulänglichkeiten ist:

$$\begin{aligned} \mathrm{SQR} &= \sum_{k=1}^{n} (\psi[k] - \hat{y}[k])^2 = J(\boldsymbol{\theta}^*) \\ &= \boldsymbol{\psi}^T \boldsymbol{\psi} - \boldsymbol{\theta}^T \boldsymbol{\Xi}^T \boldsymbol{\psi} \\ &= \left(\boldsymbol{\psi}^T - \hat{\boldsymbol{y}}^T\right) \boldsymbol{\psi}. \end{aligned} \tag{3.72}$$

Die Residuenquadratsumme kann somit unmittelbar aus (3.26) berechnet werden. Vergleicht man die drei vorherigen Kennzahlen, so lässt sich folgender Satz durch einfaches Einsetzen beweisen:

Satz 3.8: Quadratsummenzerlegung

Die Quadratsummenzerlegung, auch Zerlegung der Summe der Abweichungsquadrate genannt, beschreibt die Zerlegung der gesamten Abweichungsquadratsumme in die erklärte Abweichungsquadratsumme und die Residuenquadratsumme:

$$\mathrm{SQT} = \mathrm{SQE} + \mathrm{SQR}. \tag{3.73}$$

Die Quadratsummenzerlegung kann daher genutzt werden, um die Güte des Modells im Sinne

der Prozessabdeckung zu beschreiben. Insbesondere an den SQR kann a-priori eine Zielmarke definiert werden, welche es zu erreichen gilt, um ein zielführendes Identifikationsergebnis zu beschreiben. Nachteilig an der Quadratsummenzerlegung ist allerdings, dass diese einheitenbehaftet ist und der zu erwartende Wertebereich vorab ggf. schwer abschätzbar ist. Auch sind obige Kennzahlen direkt abhängig von der Anzahl der Messpunkte. Daher erscheint es zielführend, eine normalisierte Größe hierfür einzuführen:

Definition 3.3: Bestimmtheitsmaß

Das (empirische) Bestimmtheitsmaß ist eine dimensionslose Maßzahl, die den Anteil der Variabilität in den Messwerten der Regressoren (abhängigen Variablen) ausdrückt, der durch das Modell „erklärt" wird. Das Bestimmtheitsmaß

$$R^2 = \frac{\mathrm{SQE}}{\mathrm{SQT}} = 1 - \frac{\mathrm{SQR}}{\mathrm{SQT}} \tag{3.74}$$

kann daher als relative Modellabdeckung interpretiert werden. Es gilt $R^2 \in [0,1]$, wobei ein Wert von Eins ein perfektes Fitting des Modells an die gegebene Datengrundlage darstellt.

In der Literatur finden sich häufig Angaben, demzufolge ein identifiziertes Modell mit $R^2 > 0,9$ als zielführend bezeichnet wird. Diese Faustregel muss dennoch nicht auf alle Identifikationsprobleme zutreffen, sodass entsprechend Abb. 1.4 zu Beginn der jeweiligen Identifikation aufgabenspezifische Zielkorridore definiert werden sollten.

Der Vollständigkeit halber seien für das Fahrzeugproblem (3.7) noch die verschiedenen Kenngrößen auf Basis der Abweichungsquadrate angegeben. Für das gegebene Beispiel nimmt R^2 vergleichsweise hohe Werte nahe Eins an, da es sich hier um eine White-Box Identifikation handelt – das Prozessmodell war zu Beginn der Identifikation bereits exakt bekannt. Demgegenüber bietet sich das Bestimmtheitsmaß insbesondere zum Vergleich verschiedener Modelltypen bei einer Black-Box Identifikation an.

	SQT	SQE	SQR	R^2
$n = 10^2$	$1,36 \cdot 10^7$ N^2	$1,34 \cdot 10^7$ N^2	$2,11 \cdot 10^5$ N^2	0,984
$n = 10^5$	$1,65 \cdot 10^9$ N^2	$1,62 \cdot 10^9$ N^2	$2,48 \cdot 10^7$ N^2	0,985

Tab. 3.4: Kennziffern auf Basis der Abweichungsquadrate für das Fahrzeugproblem (3.7)

Analyse auf Basis der Residuen:
Die Abweichungen zwischen Modell und Prozess

$$e[k] = \psi[k] - \hat{y}[k]$$

können ebenfalls zur Genauigkeitsbetrachtung herangezogen werden. Hierbei bietet es sich insbesondere an zu prüfen, ob die Residuenverteilungen unkorreliert zueinander sind und somit einem weißen Rauschsignal entspricht. Hierfür kann die Autokorrelation entsprechend Definition 2.24 herangezogen werden, wobei für idealtypisches weißes Rauschen entsprechend Definition 2.25 der folgende Zusammenhang gelte:

$$R_{ee}(\tau = 0) \neq 0, \quad R_{ee}(\tau \neq 0) = 0.$$

Dieser Zusammenhang ist allerdings in praktischen Anwendungen nicht zielführend, da aufgrund von Messunsicherheiten, parasitären Effekten sowie einem begrenzten Stichprobenumfang $R_{ee}(\tau \neq 0)$ nie exakt Null sein wird. Praktikabler erscheint es daher, ein Konfidenzintervall anzugeben, innerhalb dessen sich die Autokorrelation bewegen sollte, um in guter Näherung als weißes Rauschen aufgefasst zu werden. Hierzu sei zunächst auf folgende formale Definition verwiesen:

Definition 3.4: Konfidenzintervall

Es seien unabhängige und identisch verteilte Zufallsvariablen $X_1, \ldots, X_n$ mit unbekanntem reellen Verteilungsparameter ϑ gegeben. Wenn sich Stichprobenfunktionen U und V angeben lassen, so dass gilt:

$$P(U < \vartheta < V) \geq \gamma \tag{3.75}$$

mit $\gamma \in [0, 1]$, dann bezeichnet das (stochastische) Intervall $[U, V]$ ein Konfidenzintervall für ϑ zum Konfidenzniveau γ (auch: γ-Konfidenzintervall).

Da die Realisationen u und v der U und V keine Zufallsvariablen sind und ϑ ein fixer Wert ist, kann man nicht sagen, dass das Schätzintervall $[u, v]$ mit einer Wahrscheinlichkeit von γ den unbekannten Parameter ϑ enthält. Es bedeutet vielmehr, dass im Mittel ein Anteil von γ aller so berechneten Schätzintervalle den unbekannten Parameter überdeckt.

Für den vorliegenden Fall der Autokorrelation kann das Konfidenzintervall zu

$$\left[-z_{(1-\frac{\alpha}{2})}\frac{R_{ee}(\tau=0)}{\sqrt{n}}, z_{(1-\frac{\alpha}{2})}\frac{R_{ee}(\tau=0)}{\sqrt{n}}\right] \tag{3.76}$$

gewählt werden. Hierbei ist $z_{(1-\frac{\alpha}{2})}$ das entsprechende $(1-\alpha/2)$-Quantil der zugrundeliegenden (oder ggf. angenommenen) Verteilungsfunktion. Ferner wird $R_{xx}(\tau = 0)/\sqrt{n}$ auch Standardfehler genannt. Für die häufig anzutreffende Normalverteilung ergibt sich die in Tab. 3.5 dargestellte Zuordnung von Konfidenzniveaus und Quantilen[1].

Konfidenzniveau γ	Quantil z
50 %	0,675
90 %	1,645
95 %	1,960
99 %	2,576

Tab. 3.5: Zuordnung von Konfidenzniveaus und Quantilen für die Normalverteilung

In der Statistik sowie in der Identifikation wird das Konfidenzniveau häufig zu 95 % gewählt, sodass das folgende Konfidenzintervall

$$R_{ee}(\tau \neq 0) \in \left[-1,96\frac{R_{ee}(\tau=0)}{\sqrt{n}}, 1,96\frac{R_{ee}(\tau=0)}{\sqrt{n}}\right] \tag{3.77}$$

[1] Für einen geringen Stichprobenumfang ($n < 30$) wird statt der Normalverteilung die Student-t-Verteilung herangezogen. Für ein gegebenes Konfidenzniveau sind die derart berechneten Quantile größer als die der Normalverteilung, sodass hier die Unsicherheit in Folge der geringen Stichprobe Berücksichtigung findet. Quantiltabellen für die verschiedenen Wahrscheinlichkeitsverteilungen können der einschlägigen Literatur entnommen werden.

für die Autokorrelation resultiert. Dieser Zusammenhang ist beispielhaft für das bereits bekannte Fahrzeugproblem in Abb. 3.6 und Abb. 3.7 dargestellt. Es ist ersichtlich, dass unabhängig vom Stichprobenumfang das 95 % Konfidenzintervall (nahezu) vollständig eingehalten wird, was nicht weiter verwunderlich ist, da das (synthetisch erzeugte) Messrauschen in dieser White-Box Identifikationsaufgabe gerade dem weißen Rauschen entsprach.

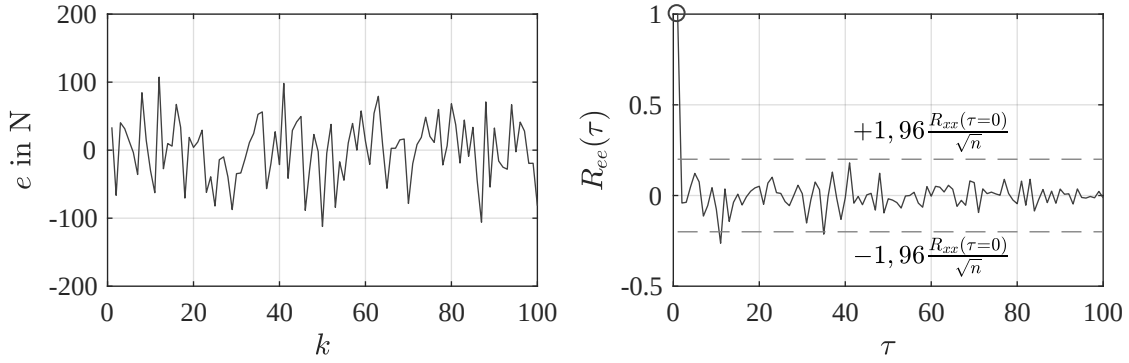


Abb. 3.6: Residuenanalyse des LS-Problems aus Beispiel (3.7) für $n = 100$ Messspunkte

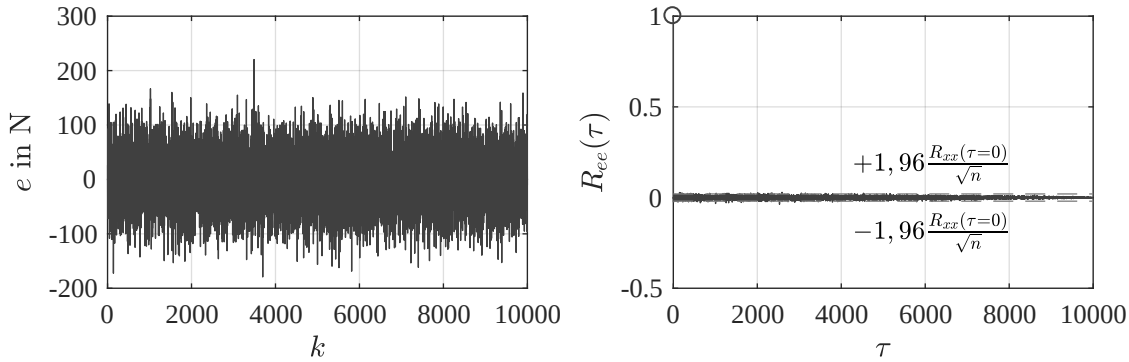


Abb. 3.7: Residuenanalyse des LS-Problems aus Beispiel (3.7) für $n = 100.000$ Messspunkte

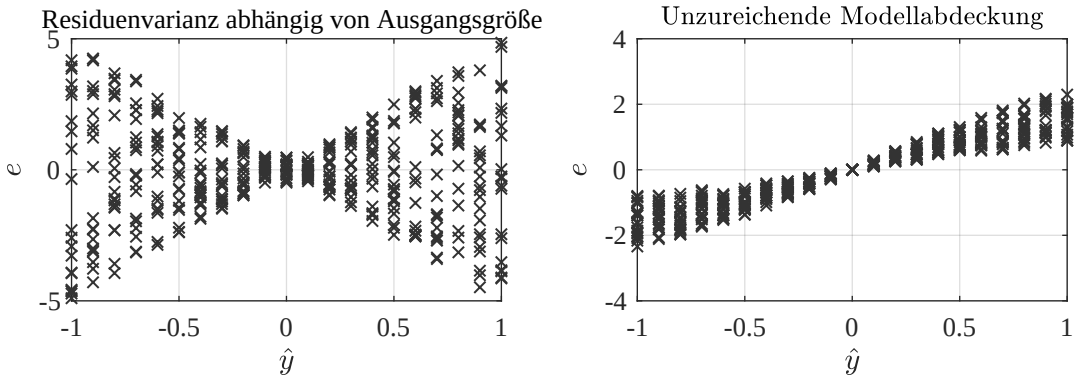


Abb. 3.8: Beispielhafte Residuenverläufe, welche auf Modellunzulänglichkeiten hinweisen

Neben der Verwendung der Autokorrelation können auch die Residuenverläufe selbst betrachtet werden. Das vorangegangene Beispiel aus Abb. 3.6 und Abb. 3.7 lässt bereits vermuten, dass die betrachteten Residuen ein unkorreliertes weißes Rauschen aufweisen und somit ein Indiz für ein zielführendes Identifikationsergebnis darstellen. Demgegenüber sind in Abb. 3.8 zwei beispielhafte Residuenverläufe für Identifikationsergebnisse mit signifikanten Modellunzulänglichkeiten dargestellt.

3.1.4 Hinweise zur numerischen Berechnung

Die Lösung des LS-Problems (3.23) wurde derart präsentiert, dass zum Auffinden des gesuchten Parametervektors eine explizite Matrix-Inversion vorzunehmen sei:

$$\boldsymbol{\theta}^* = \left(\boldsymbol{\Xi}^T\boldsymbol{\Xi}\right)^{-1}\boldsymbol{\Xi}^T\boldsymbol{\psi}.$$

Demgegenüber kann das Auffinden einer Lösung der *Gaußschen Normalengleichung*

$$\underbrace{\left(\boldsymbol{\Xi}^T\boldsymbol{\Xi}\right)}_{\boldsymbol{A}}\underbrace{\boldsymbol{\theta}}_{\boldsymbol{x}} = \underbrace{\boldsymbol{\Xi}^T\boldsymbol{\psi}}_{\boldsymbol{b}} \tag{3.78}$$

auch über alternative Methoden der numerischen Matrix-Zerlegung stattfinden. Nachfolgend werden drei beispielhafte Implementierungsvariante auf Basis von *MATLAB* diskutiert:

- Explizite Matrix-Inversion
- LR-Zerlegung (Gaußsches Eliminationsverfahren)
- Cholesky-Zerlegung
- QR-Zerlegung

Obwohl der erste Ansatz sich mathematisch anzubieten scheint, ist hiervon in der Praxis jedoch abzuraten, da dieser für schlecht konditionierte Matrizen $\boldsymbol{\Xi}$ zu sehr ungenauen Identifikationsergebnissen führen kann. Die *LR-Zerlegung* ist hingegen bei positiv definiten Produktsummenmatrizen (hinreiche Bedingung für Minimum der LS-Kostenfunktion) numerisch stabil. Die LR-Zerlegung stellt die algorithmische Umsetzung des Gaußschen Eliminationsverfahren dar und wird auf dem abgewandelteten Problem

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}\,, \tag{3.79}$$

$$\boldsymbol{PA} = \boldsymbol{LR}\,, \tag{3.80}$$

$$\boldsymbol{y} = \boldsymbol{Rx}\,, \tag{3.81}$$

$$\hat{\boldsymbol{b}} = \boldsymbol{Pb} \tag{3.82}$$

durchgeführt. Hier ist $\boldsymbol{P} \in \mathbb{R}^{m\times m}$ eine aus der Einheitsmatrix abgeleitet Permutationsmatrix, $\boldsymbol{L} \in \mathbb{R}^{m\times m}$ eine untere Dreiecksmatrix und $\boldsymbol{R} \in \mathbb{R}^{m\times m}$ eine obere Dreiecksmatrix. Ferner sind $\boldsymbol{y} \in \mathbb{R}^m$ und $\hat{\boldsymbol{b}} \in \mathbb{R}^m$ Hilfsvektoren. Nach erfolgter LR-Zerlegung kann das ursprüngliche Gleichungssystem gelöst werden durch:

1. Vorwärtseinsetzen: Durch lösen von $\boldsymbol{Ly} = \boldsymbol{b}$,
2. Rückwärtseinsetzen: Durch lösen von $\boldsymbol{R}^T\boldsymbol{x} = \boldsymbol{y}$.

Eine numerisch effizientere Variante des Gaußschen Eliminationsverfahren stellt die *Cholesky-Zerlegung* dar. Diese ist numerisch ebenfalls stabil, wenn $\boldsymbol{A} = \boldsymbol{\Xi}^T\boldsymbol{\Xi}$ eine symmetrische und positiv definite Matrix ist. Ziel der Zerlegung ist es, $\boldsymbol{A}$ in eine untere Dreiecksmatrix $\boldsymbol{L} \in \mathbb{R}^{m\times m}$ sowie eine Diagonalmatrix $\boldsymbol{D} \in \mathbb{R}^{m\times m}$ mit positiven Einträgen zu zerlegen:

$$\boldsymbol{A} = \boldsymbol{LDL}^T \quad \text{mit} \quad \boldsymbol{D} = \boldsymbol{D}^{\frac{1}{2}}\boldsymbol{D}^{\frac{1}{2}},\ \boldsymbol{G} = \boldsymbol{LD}^{\frac{1}{2}}. \tag{3.83}$$

Kann die Cholesky-Zerlegung durchgeführt werden, so lässt sich das Gleichungssystem $\boldsymbol{Ax} = \boldsymbol{b}$ effizient lösen:

1. Vorwärtseinsetzen: Durch lösen von $\boldsymbol{Gy} = \boldsymbol{b}$,
2. Rückwärtseinsetzen: Durch lösen von $\boldsymbol{G}^T\boldsymbol{x} = \boldsymbol{y}$.

Die *QR-Zerlegung* hingegen basiert nicht auf dem Gaußschen Eliminationsverfahren, sondern es wird eine orthogonale Zerlegung der Form

$$\boldsymbol{A} = \boldsymbol{QR} \tag{3.84}$$

gesucht. Hier ist $\boldsymbol{Q} \in \mathbb{R}^{m\times m}$ eine orthogonale Matrix ($\boldsymbol{Q}^T\boldsymbol{Q} = \boldsymbol{I}$) und $\boldsymbol{R} \in \mathbb{R}^{m\times m}$ eine obere Dreiecksmatrix. Kann eine QR-Zerlegung gefunden werden, folgt die Lösung des LS-Problems durch:

1. Berechnung: $\boldsymbol{y} = \boldsymbol{Q}^T\boldsymbol{b}$,
2. Rückwärtseinsetzen: Durch Lösen von $\boldsymbol{Rx} = \boldsymbol{y}$.

Eine kleine Gegenüberstellung des Aufwands und der Stabilität der Verfahren ist in Tab. 3.6 zusammengefasst. Für weitere mathematische Details zur numerischen Stabilität und Genauigkeit sowie der algorithmischen Implementierung der unterschiedlichen Zerlegungen sei auf die entsprechende Fachliteratur (z. B. [SK06]) verwiesen.

	LR-Zerlegung	Cholesky-Zerlegung	QR-Zerlegung
Rechenoperationen	$\frac{2}{3}m^3 + \mathcal{O}(m^2)$	$\frac{1}{3}m^3 + \mathcal{O}(m^2)$	$\frac{4}{3}m^3 + \mathcal{O}(m^2)$
Stabilität	stabil	stabil	sehr stabil

Tab. 3.6: Eigenschaften der numerischen Lösungsverfahren für $\boldsymbol{Ax} = \boldsymbol{b}$ mit $\boldsymbol{A} \in \mathbb{R}^{m\times m}$ als positiv definite Matrix. Hierbei ist $\mathcal{O}(x)$ die Landau-Notation, d. h., der Berechnungsaufwand wächst nicht wesentlich schneller als x.

Die Implementierungsunterschiede drei der vier Lösungen sind im nachfolgenden MATLAB-Code dargestellt. Die potentiell signifikanten Unterschiede bezüglich der gefundenen Lösungen mittels der obigen Ansätze sollen nachfolgend lediglich an einem Beispiel verdeutlicht werden.

Quellcode 3.1: Verschiedene MATLAB-Implementierungen der LS-Lösung

```
function theta = LS(Xi, psi)

    %% Explizite Matrix-Inversion
    theta = inv(Xi.' * Xi) * (Xi.' * psi); % Numerisch problematisch

    %% Cholesky-Zerlegung
    theta = (Xi.' * Xi) \ (Xi.' * psi);
    % Matlab erkennt, dass (Xi.' * Xi) positiv-definit ist und wendet
    % in diesem Fall die Cholesky-Zerlegung automatisch an.

    %% QR-Zerlegung
    theta = Xi \ psi;
    % Matlab erkennt das LS-Problem und wendet die QR-Zerlegung
    % automatisch an.
```

Beispiel[1]:
Gegeben sei die Funktion

$$y[k] = \theta_1 + \theta_2 u[k] + \theta_3 u^2[k] + \theta_4 u^3[k] + \theta_5 u^4[k] + \theta_6 u^5[k] \tag{3.85}$$

mit dem Parametervektor (dessen Einträge einheitlich den Wert zwei aufweisen)

$$\boldsymbol{\theta} = \begin{bmatrix} \theta_1 & \theta_2 & \theta_3 & \theta_4 & \theta_5 & \theta_6 \end{bmatrix}^T = \begin{bmatrix} 2 & 2 & 2 & 2 & 2 & 2 \end{bmatrix}^T . \tag{3.86}$$

Es wird angenommen, dass im Sinne der LS-Aufgabe die zuvor unbekannten Parameter aus ungestörten Messwerten bestimmt werden sollen, d. h., die Messwerte $\psi[k] = y[k]$ enthalten nur die Rundungsfehler aufgrund der endlichen Rechengenauigkeit. Der Regressorvektor pro Messpunkt ergibt sich zu:

$$\boldsymbol{\xi}^T[k] = \begin{bmatrix} 1 & u[k] & u^2[k] & u^3[k] & u^4[k] & u^5[k] \end{bmatrix} . \tag{3.87}$$

Zur Identifikation werden je $n = 100$ gleichmäßig verteilte Messpunkte genutzt, wobei im Folgenden zwei unterschiedliche Messintervalle betrachtet werden:

$$I_1 = [-1, 1], \quad I_2 = [10, 12].$$

Der Funktionsverlauf beider Messintervalle ist in Abb. 3.9 dargestellt. Aufgrund der unterschiedlichen Messintervalle sind die beiden Regressormatrizen stark unterschiedlich konditioniert:

$$\kappa(\boldsymbol{\Xi})\big|_{I_1} = 41,92 , \quad \kappa(\boldsymbol{\Xi})\big|_{I_2} = 7,53 \cdot 10^{11} . \tag{3.88}$$

Hier ist $\kappa = \|\cdot\|_2$ die Spektralnorm. Für das Intervall I_1 sind die Identifikationsergebnisse in Tab. 3.7 und für das Intervall I_2 in Tab. 3.8 zusammengefasst. Für den Fall einer gut konditio-

[1] Das Beispiel wurde [Len17] entnommen.

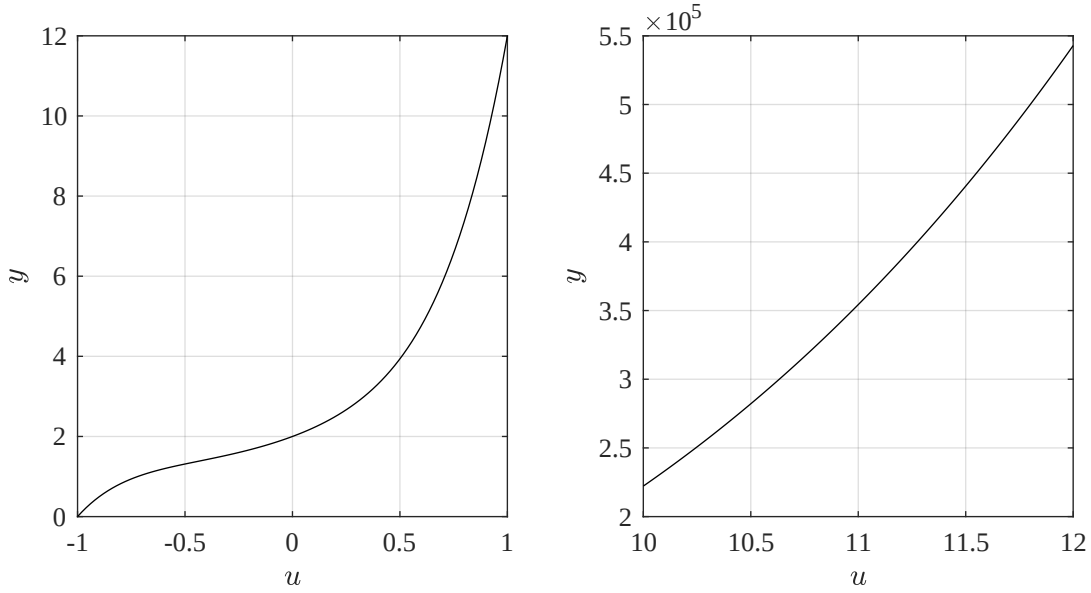


Abb. 3.9: Funktionsverlauf für unterschiedliche Eingangsgrößenintervalle

nierten Regressormatrix $\boldsymbol{\Xi}$ zeigen die drei Lösungsansätze nahezu keine Unterschiede auf – die verbleibenden Abweichungen hinsichtlich der

$$\text{Norm. Parameterabweichung:} \frac{||\Delta\hat{\boldsymbol{\theta}}||_2}{||\boldsymbol{\theta}||_2} \quad \text{und} \quad \text{Quadr. Residuenabweichung:} ||\boldsymbol{\Xi}\hat{\boldsymbol{\theta}} - \boldsymbol{\psi}||_2$$

sind vernachlässigbar gering. Demgegenüber sieht es bei der Identifikation auf dem Intervall I_2 vollkommen anders aus: Sowohl über die Cholesky-Zerlegung als auch über die explizite Inversion ist der gefundene Parametervektor als unbrauchbar zu beschreiben. Zwar liefert die Cholesky-Zerlegung noch halbwegs kleine Modellierungsresiduen, dies ist aber zweitrangig, da es bei der Systemidentifikation aber letztlich nicht darauf ankommt, nur die vorliegenden Messwerte gut abzubilden, sondern auch neue, unbekannte Messwerte erklären zu können, ist die Güte der geschätzten Parameter das gewichtigere Kriterium.

Die QR-Zerlegung liefert hingegen auch auf I_2 weiterhin ein genaues und in diesem Fallbeispiel ebenfalls zufriedenstellendes Identifikationsergebnis ab. Dennoch ist das Ergebnis, verglichen mit der Identifikation auf I_1, auch bei der QR-Zerlegung um mehrere Größenordnungen ungenauer, was die Zusammenstellung sinnvoller Messpunkte motiviert.

	Explizite Inversion	Cholesky-Zerlegung	QR-Zerlegung
$\hat{\theta_1}$	2,0000	1,9999	2,0000
$\hat{\theta_2}$	2,0000	2,0000	1,9999
$\hat{\theta_3}$	1,9999	2,0000	2,0000
$\hat{\theta_4}$	2,0000	2,0000	1,9999
$\hat{\theta_5}$	2,0000	1,9999	2,0000
$\hat{\theta_6}$	1,9999	2,0000	1,9999
$\frac{\|\Delta\hat{\boldsymbol{\theta}}\|_2}{\|\boldsymbol{\theta}\|_2}$	$1,1404 \cdot 10^{-27}$	$0,96507 \cdot 10^{-27}$	$5,3002 \cdot 10^{-30}$
$\|\|\boldsymbol{\Xi}\hat{\boldsymbol{\theta}} - \boldsymbol{\psi}\|\|_2$	$2,0962 \cdot 10^{-25}$	$1,7347 \cdot 10^{-27}$	$9,2506 \cdot 10^{-29}$

Tab. 3.7: Lösung des gut konditionierten LS-Problems auf dem Intervall I_1

	Explizite Inversion	Cholesky-Zerlegung	QR-Zerlegung
$\hat{\theta_1}$	$-3,39 \cdot 10^3$	$-2,63 \cdot 10^4$	2,0000
$\hat{\theta_2}$	$2,43 \cdot 10^3$	$1,19 \cdot 10^4$	1,9999
$\hat{\theta_3}$	$-1,61 \cdot 10^3$	$-2,18 \cdot 10^3$	2,0000
$\hat{\theta_4}$	0,1844	200,89	1,9999
$\hat{\theta_5}$	0,4573	-7,047	1,9999
$\hat{\theta_6}$	1,9371	2,1645	2,0000
$\frac{\|\Delta\hat{\boldsymbol{\theta}}\|_2}{\|\boldsymbol{\theta}\|_2}$	$8,3325 \cdot 10^5$	$3,4925 \cdot 10^7$	$1,2009 \cdot 10^{-12}$
$\|\|\boldsymbol{\Xi}\hat{\boldsymbol{\theta}} - \boldsymbol{\psi}\|\|_2$	$4,4088 \cdot 10^{12}$	$4,4431 \cdot 10^{-3}$	$9,4848 \cdot 10^{-20}$

Tab. 3.8: Lösung des schlecht konditionierten LS-Problems auf dem Intervall I_2

3.2 Weitere Varianten der Methode der kleinsten Quadrate

Nachfolgend werden einige, ausgewählte Varianten der Methode der kleinsten Quadrate vorgestellt. Diese und viele weitere Methoden im Kontext der linearen Regression können etablierten Softwarepaketen entnommen werden, z. B.

- Matlab Statistics and Machine Learning Toolbox (kostenpflichtig)
  `https://de.mathworks.com/help/stats/index.html`
- SciPy (Python, frei zugänglich)
  `https://www.scipy.org/`
- scikit-learn (Python, frei zugänglich)
  `https://scikit-learn.org/`

3.2.1 Methode der gewichteten kleinsten Quadrate

Die Methode der gewichteten kleinsten Quadrate (*weighted least squares* – WLS) stellt eine Erweiterung des vorherigen LS-Schätzers[1] dar. Wie der Name bereits vermuten lässt, wird beim WLS eine Gewichtung innerhalb der Kostenfunktion eingeführt:

$$J_{WLS}(\boldsymbol{\theta}) = (\boldsymbol{\psi} - \boldsymbol{\Xi}\boldsymbol{\theta})^T \boldsymbol{W} (\boldsymbol{\psi} - \boldsymbol{\Xi}\boldsymbol{\theta}) \tag{3.89}$$

mit der Gewichtungsmatrix $\boldsymbol{W} \in \mathbb{R}^{n \times n}$. Analog zur Herleitung des LS-Schätzers folgt für die Lösung dieser Schätzaufgabe:

Satz 3.9: Lösung des WLS-Problems für lineare, statische Systeme

Die Lösung des LS-Problems mit gewichteter Kostenfunktion (3.89) *lautet*

$$\boldsymbol{\theta}^* = \left(\boldsymbol{\Xi}^T \boldsymbol{W} \boldsymbol{\Xi}\right)^{-1} \boldsymbol{\Xi}^T \boldsymbol{W} \boldsymbol{\psi}, \tag{3.90}$$

sofern die gewichtete Produktsummenmatrix $(\boldsymbol{\Xi}^T \boldsymbol{W} \boldsymbol{\Xi})$ *invertierbar ist und die Hesse-Matrix* $\frac{\partial^2 J}{\partial \boldsymbol{\theta}^2}$ *positiv definit ist, d. h.*

$$\det(\boldsymbol{\Xi}^T \boldsymbol{W} \boldsymbol{\Xi}) \neq 0 \quad \wedge \quad \mathrm{Eig}(\boldsymbol{\Xi}^T \boldsymbol{W} \boldsymbol{\Xi}) > 0. \tag{3.91}$$

Analog zum LS-Schätzer kann gezeigt werden, dass unter Verwendung der Annahmen A1...A3 auch der WLS-Schätzer biasfrei ist

$$\mathrm{E}(\hat{\boldsymbol{\theta}})_{WLS} = \boldsymbol{\theta} \tag{3.92}$$

und die Kovarianz der Parameterschätzung ergibt sich zu:

$$\mathrm{Cov}(\hat{\boldsymbol{\theta}}, \hat{\boldsymbol{\theta}})_{WLS} = \left(\boldsymbol{\Xi}^T \boldsymbol{W} \boldsymbol{\Xi}\right)^{-1} \boldsymbol{\Xi}^T \boldsymbol{W} \mathrm{E}\left(\boldsymbol{\nu}\boldsymbol{\nu}^T\right) \boldsymbol{W} \boldsymbol{\Xi} \left(\boldsymbol{\Xi}^T \boldsymbol{W} \boldsymbol{\Xi}\right)^{-1}. \tag{3.93}$$

Im Folgenden soll das ursprüngliche LS-Problem erweitert werden und die Gewichtung $\boldsymbol{W}$ derart gewählt werden, dass der resultierende WLS-Schätzer effizient ist, d. h., es gibt keinen anderen linearen, biasfreien Schätzer, welcher eine geringere Parametervarianz aufweist. Zur Erinnerung: Für das OLS-Verfahren entsprechend Definition 3.1 wurde angenommen, dass das additive Messrauschen ν biasfrei, unkorreliert und eine konstante Varianz aufweise, d. h.

$$\mathrm{E}(\nu)_{OLS} = 0 \quad \text{und} \quad \mathrm{Cov}(\nu)_{OLS} = \sigma^2.$$

Demgegenüber wird für das WLS Folgendes angenommen:

$$\mathrm{E}(\nu)_{WLS} = 0 \quad \text{und} \quad \mathrm{Cov}(\nu)_{WLS} = \boldsymbol{R} \tag{3.94}$$

mit $\boldsymbol{R} \in \mathbb{R}^{n \times n}$ als positiv-definite Messrauschmatrix. Diese Erweiterung deckt folgende Szenarien ab:

- Korreliertes Messrauschen: Falls das Messrauschen mit sich selbst korreliert ist, sind die Nebendiagonaleinträge in $\boldsymbol{R}$ ungleich Null.
- Variierende Varianz: Falls das Messrauschen nicht konstante Varianz aufzeigt, sind die Hauptdiagonaleinträge in $\boldsymbol{R}$ nicht gleich.

[1] Dieser wird im Englischen daher auch häufig als *ordinary least squares* (OLS) *estimator* bezeichnet.

In diesem Fall kann analog zum Beweis von Satz 3.5 gezeigt werden:

Satz 3.10: Optimale Gewichtung des WLS-Problems

Für das WLS-Problem gemäß (3.89) *unter der Annahme* (3.94) *führt die Lösung* (3.90) *zu einem biasfreien, konsistent und effizienten Schätzer mit minimaler Varianz, sofern eine Gewichtung invers zur Messrauschmatrix vorgenommen wird:*

$$\boldsymbol{W} = \boldsymbol{R}^{-1}. \tag{3.95}$$

Der WLS-Schätzer hat unter den gegebenen Randbedingungen daher BLUE-Eigenschaften.

Die optimale Gewichtung lässt sich z. B. anhand eines unkorrelierten Messrauschens verdeutlichen – hier weist jede Messung eine andere Varianz $\sigma^2[k]$ auf. In diesem Fall hat die Messrauschmatrix Diagonalform:

$$\boldsymbol{R} = \operatorname{diag}\left(\begin{bmatrix}\sigma^2[1] & \sigma^2[2] & \dots & \sigma^2[n]\end{bmatrix}\right).$$

Dann ergeben sich die optimalen Gewichte gerade als die inversen Varianzen:

$$\boldsymbol{W} = \operatorname{diag}\left(\begin{bmatrix}\frac{1}{\sigma^2[1]} & \frac{1}{\sigma^2[2]} & \dots & \frac{1}{\sigma^2[n]}\end{bmatrix}\right).$$

Die Messungen werden demnach umgekehrt zu ihrer Unsicherheit gewichtet, was eine sehr anschauliche Verdeutlichung der Wirkungsweise des WLS-Ansatzes darstellt.

Zur weiteren Veranschaulichung wurde in Abb. 3.10 das Eingangsbeispiel zur Identifikation der Fahrzeugparameter dahingehend angepasst, dass das Messrauschen nicht mehr konstant ist, sondern mit steigender Geschwindigkeit ebenfalls zunimmt – demnach hat die Messrauschmatrix Diagonalform und Messpunkte bei hoher Fahrzeuggeschwindigkeit sind besonders unsicher. Dem Ergebnis in Abb. 3.10 bzw. Tab. 3.9 ist zu entnehmen, dass der OLS mit dieser nicht konstanten Varianz Probleme hat und sowohl die Residuen zwischen Modell und realem Prozess als auch die identifizierten Parameter deutlich stärker vom Idealergebnis abweichen, als dies beim WLS der Fall ist.

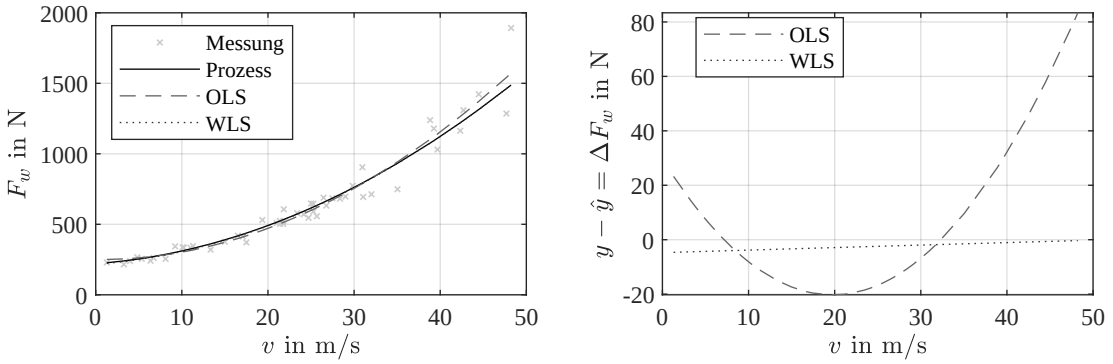


Abb. 3.10: Ergebnis des WLS-Problems aus Beispiel (3.89) für $n = 50$ Messspunkte und einem linear ansteigenden Messrauschen in Abhängigkeit der Geschwindigkeit

Gegenüber obigem Beispiel, bei dem die Messrauschmatrix als bekannt vorausgesetzt wurde, stehen der Anwendung des WLS-Ansatzes in realen Anwendungen einige Hürden entgegen:

	c_r	η_v	c_w
$\boldsymbol{\theta}$	$0,0150$	$4,00\ \frac{\text{kg}}{\text{s}}$	$0,350$
$\hat{\boldsymbol{\theta}}_{OLS}$	$0,1700$	$-0,55\ \frac{\text{kg}}{\text{s}}$	$0,449$
$\hat{\boldsymbol{\theta}}_{WLS}$	$0,0147$	$4,59\ \frac{\text{kg}}{\text{s}}$	$0,350$

Tab. 3.9: Identifizierte Modellparameter entsprechend Resultaten aus Abb. 3.10

- Typischerweise ist $\boldsymbol{R}$ nicht (exakt) bekannt, sodass sich die Ermittlung einer optimalen Gewichtung $\boldsymbol{W}$ als schwierig bzw. unsicher erweist.
- Die Matrizen $\boldsymbol{R}$ und $\boldsymbol{W}$ sind quadratisch und die Anzahl ihrer Zeilen bzw. Spalten hängt von der Anzahl der Messpunkte ab. Für Identifikationsaufgaben mit vielen Messpunkten ist die Berechnung der WLS-Lösung (3.90) daher numerisch deutlich aufwendiger als die OLS-Lösung (3.23).

Demnach ist für den konkreten Anwendungsfall abzuwägen, ob das WLS praktikabel ist oder ob etwaige Abweichungen von den OLS-Annahmen entsprechend Definition 3.1 aufgrund der einfacheren Anwendbarkeit akzeptiert werden können.

3.2.2 Rekursiver Ansatz

Bei Betrachtung der OLS-Lösung (3.23) wird unmittelbar deutlich, dass der numerische Berechnungsaufwand mit steigender Anzahl der Messpunkte massiv zunimmt. Für umfangreiche Identifikationsvorhaben kann es daher angezeigt sein, die direkte LS-Berechnung in eine rekursive Form zu überführen. Bei der rekursiven Berechnung bleibt auch bei Hinzukommen neuer Daten in jedem Schritt der Rechenaufwand gleich, da das vorherige Ergebnis als Ausgangspunkt genutzt wird. Ausgangspunkt für den rekursiven LS-Ansatz (*recursive least square* – RLS) ist erneut die Modellgleichung (3.8):

$$\boldsymbol{\psi}[k] = \boldsymbol{\Xi}[k]\hat{\boldsymbol{\theta}}[k]. \tag{3.96}$$

Hier ist k der k-te Rekursionsschritt. Analog gilt für den gesuchten Parametervektor:

$$\hat{\boldsymbol{\theta}}[k] = \left(\boldsymbol{\Xi}[k]^T\boldsymbol{\Xi}[k]\right)^{-1}\boldsymbol{\Xi}[k]^T\boldsymbol{\psi}[k]. \tag{3.97}$$

Für $k+1$ erweitert sich obige Gleichungen zu:

$$\boldsymbol{\psi}[k+1] = \boldsymbol{\Xi}[k+1]\boldsymbol{\theta}[k+1]\,, \tag{3.98}$$

$$\hat{\boldsymbol{\theta}}[k+1] = \left(\boldsymbol{\Xi}[k+1]^T\boldsymbol{\Xi}[k+1]\right)^{-1}\boldsymbol{\Xi}[k+1]^T\boldsymbol{\psi}[k+1]. \tag{3.99}$$

Für den Schritt $[k+1]$ lassen sich der Messvektor als auch die Regressormatrix umschreiben zu:

$$\boldsymbol{\psi}[k+1] = \begin{bmatrix}\boldsymbol{\psi}[k]\\ \psi[k+1]\end{bmatrix}, \quad \boldsymbol{\Xi}[k+1] = \begin{bmatrix}\boldsymbol{\Xi}[k]\\ \boldsymbol{\xi}^T[k+1]\end{bmatrix}. \tag{3.100}$$

Einsetzen in (3.99) führt zu:

$$\hat{\boldsymbol{\theta}}[k+1] = \left(\begin{bmatrix}\boldsymbol{\Xi}^T[k] & \boldsymbol{\xi}[k+1]\end{bmatrix}\begin{bmatrix}\boldsymbol{\Xi}[k]\\ \boldsymbol{\xi}^T[k+1]\end{bmatrix}\right)^{-1}\begin{bmatrix}\boldsymbol{\Xi}^T[k] & \boldsymbol{\xi}[k+1]\end{bmatrix}\begin{bmatrix}\boldsymbol{\psi}[k]\\ \psi[k+1]\end{bmatrix}. \tag{3.101}$$

Ausmultiplizieren unter Verwendung von (3.8) ergibt dann:

$$\hat{\boldsymbol{\theta}}[k+1] = \left(\boldsymbol{\Xi}^T[k]\boldsymbol{\Xi}[k] + \boldsymbol{\xi}[k+1]\boldsymbol{\xi}^T[k+1]\right)^{-1}\left(\boldsymbol{\Xi}^T[k]\boldsymbol{\Xi}[k]\hat{\boldsymbol{\theta}}[k] + \boldsymbol{\xi}[k+1]\psi[k+1]\right). \quad (3.102)$$

Ferner wird die Abkürzung

$$\boldsymbol{P}^{-1} = \boldsymbol{\Xi}^T\boldsymbol{\Xi} \quad (3.103)$$

für die Produktsummenmatrix eingeführt. Einsetzen in (3.102) liefert:

$$\hat{\boldsymbol{\theta}}[k+1] = \left(\boldsymbol{P}^{-1}[k] + \boldsymbol{\xi}[k+1]\boldsymbol{\xi}^T[k+1]\right)^{-1}\left(\boldsymbol{P}^{-1}[k]\hat{\boldsymbol{\theta}}[k] + \boldsymbol{\xi}[k+1]\psi[k+1]\right). \quad (3.104)$$

Mit Blick auf Satz 3.4 fällt auf, dass $\boldsymbol{P}$ mit der Kovarianz des geschätzten Parametervektors zusammenhängt:

$$\boldsymbol{P} = \frac{1}{\sigma_\nu^2}\mathrm{Cov}(\hat{\boldsymbol{\theta}}, \hat{\boldsymbol{\theta}}). \quad (3.105)$$

Unter Verwendung der Sherman-Morrison-Woodbury-Formel[1] zur Matrixinversion folgt:

$$\hat{\boldsymbol{\theta}}[k+1] = \left(\boldsymbol{P}[k] - \frac{\boldsymbol{P}[k]\boldsymbol{\xi}[k+1]\boldsymbol{\xi}^T[k+1]\boldsymbol{P}[k]}{1+\boldsymbol{\xi}^T[k+1]\boldsymbol{P}[k]\boldsymbol{\xi}[k+1]}\right)\left(\boldsymbol{P}^{-1}[k]\hat{\boldsymbol{\theta}}[k] + \boldsymbol{\xi}[k+1]\psi[k+1]\right). \quad (3.106)$$

Umschreiben ergibt dann:

$$\hat{\boldsymbol{\theta}}[k+1] = \hat{\boldsymbol{\theta}}[k] + \underbrace{\frac{\boldsymbol{P}[k]\boldsymbol{\xi}[k+1]}{1+\boldsymbol{\xi}^T[k+1]\boldsymbol{P}[k]\boldsymbol{\xi}[k+1]}}_{\text{Verstärkungsterm: } \boldsymbol{\gamma}[k]} \underbrace{\left(\psi[k+1] - \boldsymbol{\xi}^T[k+1]\hat{\boldsymbol{\theta}}[k]\right)}_{\text{Korrekturterm}}. \quad (3.107)$$

Demnach wird die neue Parameterschätzung $\hat{\boldsymbol{\theta}}[k+1]$ anhand der letzten Schätzung $\hat{\boldsymbol{\theta}}[k]$ plus einer mittels $\boldsymbol{\gamma}[k]$ gewichteten Korrektur gebildet. Der Korrekturterm

$$\psi[k+1] - \boldsymbol{\xi}^T[k+1]\hat{\boldsymbol{\theta}}[k]$$

basiert hierbei auf dem Prädiktionsfehler, der sich als Differenz zwischen dem neusten Messwert $\psi[k+1]$ und der Modellvorhersage $\boldsymbol{\xi}^T[k+1]\hat{\boldsymbol{\theta}}[k]$ auf Basis der vorherigen $[1,\ldots,k]$ Messpunkte ergibt. Analog zu obigem Vorgehen kann zudem gezeigt werden, dass die Aktualisierung von $\boldsymbol{P}$ nicht als Inversion der Produktsummenmatrix erfolgen muss, sondern auch hier eine rekursive Berechnung möglich ist:

$$\boldsymbol{P}[k+1] = \left(\boldsymbol{I} - \boldsymbol{\gamma}[k]\boldsymbol{\xi}^T[k+1]\right)\boldsymbol{P}[k]. \quad (3.108)$$

Somit kann der gesamte RLS-Algorithmus ohne jegliche Matrixinversion zusammengefasst werden mit:

[1] Gegeben seien zwei Vektoren $\{\boldsymbol{u}, \boldsymbol{v}\} \in \mathbb{R}^n$ und eine reguläre Matrix $\boldsymbol{A} \in \mathbb{R}^{n\times n}$. Unter der Voraussetzung $\boldsymbol{v}^T\boldsymbol{A}^{-1}\boldsymbol{u} \neq 1$ gilt: $\left(\boldsymbol{A} - \boldsymbol{u}\boldsymbol{v}^T\right)^{-1} = \boldsymbol{A}^{-1} + \frac{\boldsymbol{A}^{-1}\boldsymbol{u}\boldsymbol{v}^T\boldsymbol{A}^{-1}}{1-\boldsymbol{v}^T\boldsymbol{A}^{-1}\boldsymbol{u}}$

Satz 3.11: Rekursive LS-Berechnungsvorschrift

Für das LS-Problem entsprechend Definition 3.1 lautet die rekursive Berechnungsvorschrift:

$$\begin{aligned} \boldsymbol{\gamma}[k] &= \frac{\boldsymbol{P}[k]\boldsymbol{\xi}[k+1]}{1+\boldsymbol{\xi}^T[k+1]\boldsymbol{P}[k]\boldsymbol{\xi}[k+1]}\,, \\ \hat{\boldsymbol{\theta}}[k+1] &= \hat{\boldsymbol{\theta}}[k] + \boldsymbol{\gamma}[k]\left(\psi[k+1] - \boldsymbol{\xi}^T[k+1]\hat{\boldsymbol{\theta}}[k]\right)\,, \\ \boldsymbol{P}[k+1] &= \left(\boldsymbol{I} - \boldsymbol{\gamma}[k]\boldsymbol{\xi}^T[k+1]\right)\boldsymbol{P}[k]. \end{aligned} \tag{3.109}$$

Hierbei repräsentiert k den k-ten Rekursionsschritt (welcher entsprechend k Messpunkte umfasst).

Wie jeder rekursive Algorithmus braucht auch der RLS-Ansatz eine Initialisierung, konkret hinsichtlich $\boldsymbol{P}[k=0]$ und $\boldsymbol{\theta}[k=0]$. Im einfachsten Fall kann das direkte LS-Verfahren genutzt werden, d. h., es werden ausreichend viele Messpunkte aufgenommen, um $\boldsymbol{P}[0]$ und $\boldsymbol{\theta}[0]$ numerisch sicher besetzen zu können – der RLS-Algorithmus würde dann letztendlich nur genutzt werden, um weitere Messpunkte einzugliedern und die Schätzunsicherheit zu reduzieren. Wie eingangs aber bereits diskutiert wird das RLS-Verfahren insbesondere dann genutzt, wenn aus Gründen des Berechnungs- bzw. Speicherbedarfs das direkte LS-Verfahren überhaupt nicht anwendbar ist, sodass in diesem Fall eine andere Initialisierung genutzt werden muss.

Hinsichtlich $\boldsymbol{\theta}[0]$ liegt ggf. Vorwissen über den betrachteten Prozess vor, sodass der jeweilige Anwender eine zielführende Ersteinschätzung vornehmen kann. Ist dies nicht möglich bleibt noch die triviale Wahl

$$\boldsymbol{\theta}[0] = 0.$$

Gemäß (3.105) ist $\boldsymbol{P}[0]$ mit der Varianz des Parametervektors verknüpft, d. h. hierüber kann die Unsicherheit hinsichtlich der Initialisierung $\boldsymbol{\theta}[0]$ bewertet werden. Geht man im einfachsten Fall davon aus, dass die Parameter untereinander unkorreliert sind, kann eine Diagonalmatrix angesetzt werden:

$$\boldsymbol{P}[0] = \operatorname{diag}\left([p_1[0] \quad \dots \quad p_m[0]]\right) = \operatorname{diag}\left(\left[\frac{\sigma_{\theta_1}^2}{\sigma_\nu^2} \quad \dots \quad \frac{\sigma_{\theta_m}^2}{\sigma_\nu^2}\right]\right).$$

Demnach kann für jeden Parameter individuell ein Unsicherheitsmaß bezüglich dessen Initialisierung angegeben werden. Sollte $\boldsymbol{\theta}[0]$ hingegen gänzlich unbekannt sein, bietet sich

$$\boldsymbol{P}[0] = \boldsymbol{I}\alpha \quad \text{mit} \quad \alpha \gg 1$$

an. Allerdings gilt es zu beachten, dass je größer die Werte in $\boldsymbol{P}[0]$ gewählt werden, desto mehr Messpunkte werden benötigt, bis der Algorithmus konvergiert[1].

[1] Für die ungünstige Wahl $\boldsymbol{P}[0] = \boldsymbol{0}$ konvergiert der RLS-Algorithmus allerdings nicht, da dann $\boldsymbol{\theta}[k] = \boldsymbol{\theta}[0]$ für alle $k = 1, \dots, n$ folgt, d. h. eingehende Messungen führen zu keiner Schätzveränderung mehr.

3.2.3 Exponentielles Vergessen

Beim LS-Verfahren mit exponentiellen Vergessen handelt es sich um einen Spezialfall des WLS. Hier wird die Gewichtungsmatrix zu

$$\boldsymbol{W} = \operatorname{diag}\left(\begin{bmatrix}\lambda^{n-1} & \lambda^{n-2} & \dots & \lambda & 1\end{bmatrix}\right) \quad \text{mit} \quad 0 < \lambda < 1 \tag{3.110}$$

gewählt. Damit werden die Schätzfehler der weiter zurückliegenden Messungen geringer gewichtet als die der neueren Messungen, sofern der Messvektor in aufsteigender Reihenfolge der Messzeitpunkte sortiert ist. Beispielhafte Gewichtungsverläufe sind in Abb. 3.11 dargestellt. Das

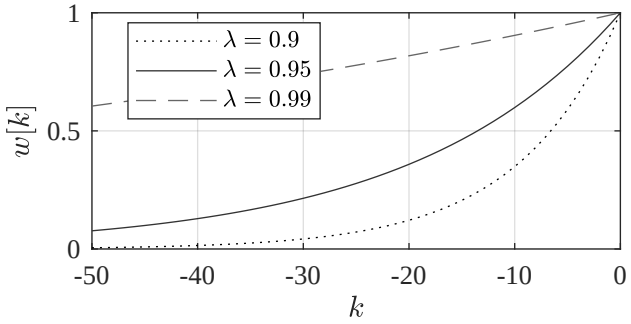


Abb. 3.11: Gewichtungen bei beispielhaften exponentiellen Vergessensfaktoren (der aktuelle Messwert entspricht $k = 0$)

exponentielle Vergessen findet bei der direkten LS-Implementierung nur sehr selten Anwendung, es sei denn, dass das Messrauschen mit jedem weiteren Messpunkt exponentiell abnimmt. Demgegenüber kann eine Verwendung mit dem RLS-Algorithmus gewinnbringend sein, z. B. wenn die anfängliche Initialisierung sehr unsicher ist oder insbesondere auch dann, wenn sich die gesuchten Parameter (langsam) mit der Zeit ändern. Als einfaches Beispiel sei hier ein ohmscher Widerstand genannt, dessen Widerstandswert sich aufgrund von Erwärmung durch fortwährende Bestromung verändert. Analog zur Herleitung in Kap. 3.2.2 ergibt sich der RLS-Algorithmus mit exponentiellem Vergessen zu:

Satz 3.12: RLS-Algorithmus mit exponentiellem Vergessen

Für das WLS-Problem entsprechend (3.89) *unter Annahme einer exponentiellen Gewichtung gemäß* (3.110) *lautet die rekursive Berechnungsvorschrift:*

$$\begin{aligned}
\boldsymbol{\gamma}[k] &= \frac{\boldsymbol{P}[k]\boldsymbol{\xi}[k+1]}{\lambda[k+1] + \boldsymbol{\xi}^T[k+1]\boldsymbol{P}[k]\boldsymbol{\xi}[k+1]}\,, \\
\hat{\boldsymbol{\theta}}[k+1] &= \hat{\boldsymbol{\theta}}[k] + \boldsymbol{\gamma}[k]\left(\psi[k+1] - \boldsymbol{\xi}^T[k+1]\hat{\boldsymbol{\theta}}[k]\right)\,, \\
\boldsymbol{P}[k+1] &= \left(\boldsymbol{I} - \boldsymbol{\gamma}[k]\boldsymbol{\xi}^T[k+1]\right)\boldsymbol{P}[k]\frac{1}{\lambda[k+1]}.
\end{aligned} \tag{3.111}$$

Hierbei repräsentiert $\lambda \in \{\mathbb{R}|0 < \lambda < 1\}$ *den Vergessensfaktor, der bei Bedarf auch zur Rechenzeit angepasst werden kann.*

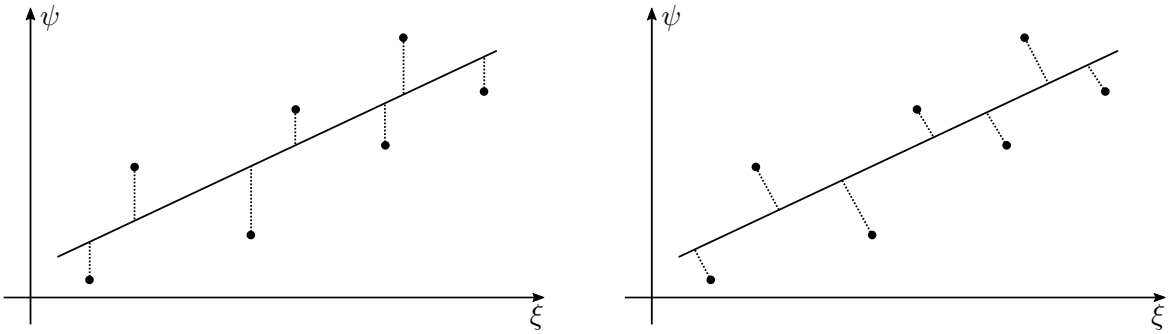
In obiger Darstellung ist erkenntlich, dass durch die gewählte Gewichtung der Term

$$\boldsymbol{P}[k]\frac{1}{\lambda[k+1]}$$

gegenüber dem regulären RLS-Ansatzes künstlich erhöht wird. Da $\boldsymbol{P}$ proportional zur Varianzmatrix des Parametervektors ist, beutetet dies, dass zurückliegende Schätzwerte als unsicherer deklariert werden, um so den Einfluss neuer Messwerte in der Mittelwertbildung zu erhöhen.

3.2.4 Orthogonale Regression

Die bisherigen LS-Ansätze nahmen an, dass die unabhängigen Variablen (Regressormatrix $\boldsymbol{\Xi}$) exakt bekannt seien. Diese Annahme ist typischerweise nur bedingt zulässig, da die unabhängigen Variablen i. d. R. messtechnisch aufgezeichnet werden und daher ebenfalls mit einem Rauschen behaftet sind. In diesem Fall kann die alternative orthogonale Regression (*total least squares* – TLS) angewandt werden, welche nicht die Abstände zwischen den Messungen und der zu identifizierende Funktion in ψ-Richtung (OLS-Ansatz), sondern die orthogonalen Distanzen der Messpunkte zur Funktion minimiert (siehe Abb. 3.12).



(a) OLS (b) TLS

Abb. 3.12: Gegenüberstellung des OLS und dem TLS Vorgehen

Im TLS-Fall wird sowohl ein Messrauschen $\boldsymbol{e} \in \mathbb{R}^n$ als auch ein Rauschen in der Regressormatrix $\boldsymbol{\Pi} \in \mathbb{R}^{n\times m}$ angenommen:

$$\boldsymbol{\psi} + \boldsymbol{e} = (\boldsymbol{\Xi} + \boldsymbol{\Pi})\,\boldsymbol{\theta}. \tag{3.112}$$

Auf eine Indexierung wird im Folgenden aus Gründen der Lesbarkeit verzichtet. Umstellen ergibt dann

$$\begin{aligned}\mathbf{0} &= (\boldsymbol{\Xi} + \boldsymbol{\Pi})\,\boldsymbol{\theta} - \boldsymbol{\psi} - \boldsymbol{e} \\ &= \underbrace{\begin{bmatrix}\boldsymbol{\Xi} & \boldsymbol{\psi}\end{bmatrix}}_{\boldsymbol{\Xi}_{TLS}}\begin{bmatrix}\boldsymbol{\theta}\\ -1\end{bmatrix} + \underbrace{\begin{bmatrix}\boldsymbol{\Pi} & \boldsymbol{e}\end{bmatrix}}_{\boldsymbol{\Pi}_{TLS}}\begin{bmatrix}\boldsymbol{\theta}\\ -1\end{bmatrix} = (\boldsymbol{\Xi}_{TLS} + \boldsymbol{\Pi}_{TLS})\begin{bmatrix}\boldsymbol{\theta}\\ -1\end{bmatrix}\end{aligned} \tag{3.113}$$

mit der erweiterten Regressormatrix

$$\boldsymbol{\Xi}_{TLS} = \begin{bmatrix}\boldsymbol{\xi}_{TLS}^T[1]\\ \vdots \\ \boldsymbol{\xi}_{TLS}^T[n]\end{bmatrix} = \begin{bmatrix}\xi_1[1] & \xi_2[1] & \cdots & \xi_m[1] & \psi[1]\\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \xi_1[n] & \xi_2[n] & \cdots & \xi_m[n] & \psi[n]\end{bmatrix} \tag{3.114}$$

sowie der zusammengefassten Rauschmatrix

$$\boldsymbol{\Pi}_{TLS} = \begin{bmatrix} \boldsymbol{\pi}_{TLS}^T[1] \\ \vdots \\ \boldsymbol{\pi}_{TLS}^T[n] \end{bmatrix} = \begin{bmatrix} \pi_1[1] & \pi_2[1] & \cdots & \pi_m[1] & e[1] \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \pi_1[n] & \pi_2[n] & \cdots & \pi_m[n] & e[n] \end{bmatrix}. \tag{3.115}$$

Die Kostenfunktion ergibt sich dann als die quadratische Summe über $e[k]$ in $\boldsymbol{\psi}$ sowie $\boldsymbol{\pi}^T[k]$ in $\boldsymbol{\Xi}$ über $k = 1, \ldots, n$ Messpunkte

$$J = \sum_{k=1}^{n} \left(e^2[k] + \sum_{j=1}^{m} \pi_j^2[k] \right) = \sum_{k=1}^{n} \sum_{j=1}^{m} \pi_{TLS,j}^2[k] = \|\boldsymbol{\Pi}_{TLS}\|_2^2 \tag{3.116}$$

mit $\|\cdot\|_2^2$ als Frobeniusnorm der zusammengefassten Rauschmatrix. In (3.116) entspricht

$$\sqrt{e^2[k] + \sum_{j=1}^{m} \pi_j^2[k]}$$

gerade dem euklidischen Abstand des Messpunkts zur Ausgleichungsfunktion im $(m+1)$-dimensionalen Raum, welcher durch das Ausgleichsproblem aufgespannt wird. Das TLS-Problem ergibt sich dann zu:

Definition 3.5: TLS-Problem für lineare, statische Systeme

Gegeben sei eine Regressionsgleichung der Form

$$\boldsymbol{\psi} + \boldsymbol{e} = (\boldsymbol{\Xi} + \boldsymbol{\Pi})\,\boldsymbol{\theta} \tag{3.117}$$

mit dem Parametervektor $\boldsymbol{\theta} \in \mathbb{R}^m$, dem Messdatenvektor $\boldsymbol{\psi} \in \mathbb{R}^n$, der Regressormatrix $\boldsymbol{\Xi} \in \mathbb{R}^{n \times m}$ sowie dem Rauschen $\boldsymbol{e} \in \mathbb{R}^n$ für die Messungen bzw. dem Rauschen $\boldsymbol{\Pi} \in \mathbb{R}^{n \times m}$ in den unabhängigen Variablen . Es gelte $m < n$. Das Auffinden des Parametervektors $\boldsymbol{\theta}$ mittels Minimierung der quadratischen Kostenfunktion (3.116) *entsprechend*

$$\boldsymbol{\theta}^* = \arg\min J(\boldsymbol{\theta}) \tag{3.118a}$$

$$\text{u. d. Nb.} \quad \left(\begin{bmatrix} \boldsymbol{\Xi} & \boldsymbol{\psi} \end{bmatrix} + \begin{bmatrix} \boldsymbol{\Pi} & \boldsymbol{e} \end{bmatrix} \right) \begin{bmatrix} \boldsymbol{\theta} \\ -1 \end{bmatrix} = \mathbf{0} \tag{3.118b}$$

wird als TLS-Problem für lineare Systeme bezeichnet.

Gegenüber dem OLS-Optimierungsproblem gemäß Definition 3.1 resultiert im TLS-Fall eine quadratische Kostenfunktion mit Gleichungsnebenbedingung. Diese Optimierungsaufgabe kann grundsätzlich unter Verwendung eines numerischen Verfahren aus Kap. 4.3 gelöst werden. Allerdings ist auch eine Lösung mittels Matrixalgebra, konkret durch eine Singulärwertzerlegung (*singular value decomposition* – SVD), möglich. Die umfängliche Herleitung und Erläuterung zur SVD sind für interessierte Leser in Anhang B zusammengefasst.

Mittels SVD kann eine gegebene Matrix als Produkt dreier spezieller Matrizen dargestellt werden. Singulärwertzerlegungen existieren für jede Matrix – auch für nichtquadratische Matrizen.

Im Fall des TLS-Verfahrens wird eine SVD von $\boldsymbol{\Xi}_{TLS}$ benötigt:

$$\boldsymbol{\Xi}_{TLS} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T = \boldsymbol{U} \begin{bmatrix} \sigma_1^2 & 0 & & \\ 0 & \ddots & & 0 \\ & & 0 & \sigma_{(m+1)}^2 \\ 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix} \begin{bmatrix} \boldsymbol{V}_{pp} & \boldsymbol{v}_{pq} \\ \boldsymbol{v}_{qp} & v_{qq} \end{bmatrix}^T . \tag{3.119}$$

Hier ist $\boldsymbol{U} \in \mathbb{R}^{n\times n}$ eine unitäre Matrix, $\boldsymbol{V} \in \mathbb{R}^{m\times m}$ ist die Adjungierte einer unitären Matrix $\boldsymbol{V}$ und $\boldsymbol{\Sigma} \in \mathbb{R}^{n\times m}$ hat obige Gestalt. Weiterhin ist $\boldsymbol{V}_{pp} \in \mathbb{R}^{m\times m}$ eine Matrix, $\boldsymbol{v}_{pq} \in \mathbb{R}^m$ ist ein Spaltenvektor, $\boldsymbol{v}_{qp} \in \mathbb{R}^m$ ist ein Zeilenvektor und $v_{qq} \in \mathbb{R}$ ist ein Skalar. Auf dieser Basis kann die TLS-Lösung wie folgt angegeben werden:

Satz 3.13: Lösung des TLS-Problems für lineare Systeme

Die Lösung des TLS-Problems (3.118) *lautet*

$$\boldsymbol{\theta}^* = -\frac{1}{v_{qq}}\boldsymbol{v}_{pq} \tag{3.120}$$

sofern eine SVD von $\boldsymbol{\Xi}_{TLS}$ *gemäß* (3.119) *existiert und* $v_{qq} \neq 0$ *gilt.*

Die entsprechende MATAB-Implementierung lautet:

Quellcode 3.2: MATLAB-Implementierungen der TLS-Lösung

```
1 function theta = TLS(Xi, psi)
2
3         [U, S, V] = svd([Xi psi]); %Singulaerwertzerlegung
4         theta = -1/V(end, end)*V(1:end-1,end); %TLS-Loesung
```

Zur Verwendung des TLS-Verfahrens seien folgende Hinweise gegeben:

- Die verwendete SVD teilt das Rauschen zwischen der Ausgangsmessung und den Regressoren auf, wobei ein näherungsweise gleichstarker Rauscheinfluss auf beiden Größen angenommen wird.
- Sollte lediglich ein Messrauschen vorhanden sein, wird der TLS-Ansatz fälschlicherweise einen Teil des Rauschens den Regressoren zuordnen und i. A. schlechtere Resultate als der OLS-Ansatz liefern.
- Der TLS-Algorithmus kann je nach Modell und Datenbasis numerisch instabil bzw. problematisch werden, insbesondere wenn v_{qq} sehr kleine Werte annimmt.

Ein weiteres wesentliches Problem tritt auf, falls die Messgröße und die Regressoren unterschiedliche Einheiten bzw. stark unterschiedliche Größenordnungen aufweisen. Aus physikalischer Sicht stellt sich dann auch die Frage, wie überhaupt der orthogonale Abstand zwischen Variablen unterschiedlicher Einheiten sinnvoll gemessen werden soll. In diesem Fall ist eine Normalisierung der unterschiedlichen Daten notwendig, um die Identifikation auf Basis dimensionsloser Größen durchführen zu können. Es gibt jedoch verschiedene Möglichkeiten (z. B. durch Mittelwertbereinigung und Standardabweichung) der Normalisierung und diese führen

zu angepassten Modellen, die nicht gleichwertig sind. Auch gilt es zu bedenken, dass durch die Normalisierung eine mögliche physikalische Interpretierbarkeit der gefundenen Systemparameter verloren geht.

Zur Verdeutlichung der numerischen Problematik beim TLS wird das Beispiel (3.7) derart modifiziert, dass neben dem bekannten Messrauschen auf F_w zudem noch ein Rauschen hinsichtlich der Fahrzeuggeschwindigkeit v modelliert wird:

$$\tilde{v}[k] = v[k] + \pi_v[k] \quad \text{mit} \quad \pi_v[k] \sim \mathcal{N}(\mu = 0, \sigma = 1\,\frac{\text{m}}{\text{s}}).$$

Hier sei $v[k]$ die tatsächliche und $\tilde{v}[k]$ die gemessene Fahrzeuggeschwindigkeit. Die resultierenden Ergebnisse der Identifikation mit TLS und OLS sind in Abb. 3.13 dargestellt, wobei die identifizierten Parameter bzw. das Bestimmtheitsmaß sich wie folgt ergeben:

	c_r	η_v	c_w	R^2
Wahrer Wert	$0,0150$	$4,50\,\frac{\text{kg}}{\text{s}}$	$0,350$	–
OLS	$0,0151$	$3,848\,\frac{\text{kg}}{\text{s}}$	$0,352$	$98,12\,\%$
TLS	$-0,012$	$43,95\,\frac{\text{kg}}{\text{s}}$	$-0,226$	$82,49\,\%$

Tab. 3.10: Identifikationsergebnise für TLS und OLS

Während das OLS-Verfahren trotz zusätzlichem Regressorauschen eine weiterhin passable Schätzung ermöglicht, liefert die TLS-Schätzung bizarre Werte. Dieses Beispiel soll daher verdeutlichen, dass der OLS-Ansatz trotz Abweichung von den maßgeblichen Grundannahmen weiterhin zielführende Ergebnisse erbringen kann während das TLS-Verfahren mittels SVD numerisch problematisch sein kann. Eine kritische Überprüfung der Ergebnisse gemäß Kap. 3.1.3 ist stets angezeigt.

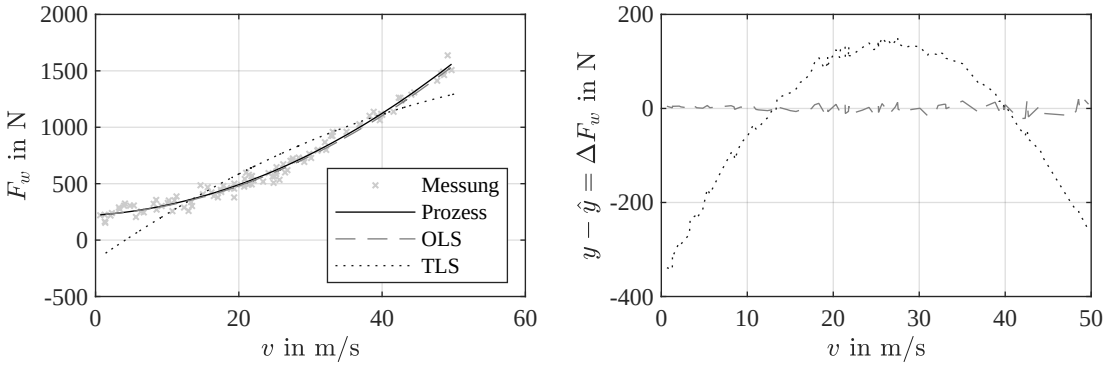


Abb. 3.13: Ergebnis des LS-Problems aus Beispiel (3.7) für $n = 100$ Messspunkte und einem Rauschen $\nu \sim \mathcal{N}(0, \sigma = 50\text{ N})$ auf F_w sowie $\pi_v[k] \sim \mathcal{N}(\mu = 0, \sigma = 1\,\frac{\text{m}}{\text{s}})$ auf v

3.2.5 Regularisierung durch Ridge- und LASSO-Regression

Die bisher behandelten Regressionsmethoden stellen unbeschränkte Optimierungsprobleme dar, sodass die gesuchten Parameter $\boldsymbol{\theta}$ beliebig große Werte annehmen können. Dies kann insbesondere bei der Identifikation von Black-Box Modellen kritisch sein, bei denen eine technisch-

physikalische Interpretation der Ergebnisse schwierig ist. Daher kann es sinnvoll sein, dass ursprüngliche Problem um eine *Regularisierung* zu erweitern. Diese beschreibt der Vorgang des Hinzufügens von Informationen, um ein schlecht gestelltes bzw. schlecht konditioniertes Problem zu lösen oder eine Überanpassung (Overfitting, siehe Abb. 3.16) zu verhindern. Zwei bekannte Regularisierungsmethoden sind die Ridge- und LASSO-Regression[1]:

Definition 3.6: Ridge- und LASSO-Regression

Gegeben sei die Regressionsgleichung entsprechend Definition 3.1. Das Auffinden des Parametervektors $\boldsymbol{\theta}$ mittels Minimierung der quadratischen Kostenfunktion

$$\textit{Ridge:}\quad J(\boldsymbol{\theta}) = \sum_{k=1}^{N}(e[k])^2 + \lambda \sum_{j=1}^{m} \theta_j^2 = (\boldsymbol{\psi} - \boldsymbol{\Xi}\boldsymbol{\theta})^T(\boldsymbol{\psi} - \boldsymbol{\Xi}\boldsymbol{\theta}) + \lambda \left\|\boldsymbol{\theta}\right\|_2 , \tag{3.121}$$

$$\textit{LASSO:}\quad J(\boldsymbol{\theta}) = \sum_{k=1}^{N}(e[k])^2 + \lambda \sum_{j=1}^{m} |\theta_j| = (\boldsymbol{\psi} - \boldsymbol{\Xi}\boldsymbol{\theta})^T(\boldsymbol{\psi} - \boldsymbol{\Xi}\boldsymbol{\theta}) + \lambda \left\|\boldsymbol{\theta}\right\|_1 \tag{3.122}$$

wird als Ridge- bzw. LASSO-Regression für lineare, statische Systeme bezeichnet.

Die beiden LS-Varianten führen demnach einen quadratischen bzw. absoluten Strafterm bezüglich der identifizierten Parameter ein. Damit dieser Strafterm sinnvoll genutzt werden kann, insbesondere wenn Regressoren und Modellausgang unterschiedliche Einheiten aufweisen bzw. Wertebereiche abdecken, ist eine Standardisierung[2] von $\boldsymbol{\Xi}$ und $\boldsymbol{\psi}$ angezeigt. Die quadratische Bestrafung im Ridge-Ansatz führt dazu, dass besonders große Werte innerhalb von $\boldsymbol{\theta}$ vermieden werden, sodass ein möglichst homogener Parametervektor resultiert. Der absolute Strafterm in der LASSO-Methodik führt tendenziell zu einer dünn-besetzten Lösung $\boldsymbol{\theta}$, d. h. unwichtige Parameter werden zu Null bzw. vernachlässigbar klein gewählt[3].

Bei der Lösung der obigen Regressionsprobleme muss differenziert werden: Die Ridge-Regression (3.121) hat nach wie vor eine quadratische Kostenfunktion und der eingeführte Strafterm ist stetig-differenzierbar. Analog zum OLS-Lösungsweg aus Kap. 3.1 ist daher eine geschlossene Lösung auffindbar:

$$\boldsymbol{\theta}^*_{Ridge} = \left(\boldsymbol{\Xi}^T\boldsymbol{\Xi} + \lambda \boldsymbol{I}\right)^{-1}\boldsymbol{\Xi}^T\boldsymbol{\psi}. \tag{3.123}$$

Demgegenüber hat das LASSO-Problem keine geschlossene Lösung. Hier muss eine Lösung iterativ mittels numerischer Optimierungsmethoden (siehe Kap. 4) erzielt werden. In beiden Varianten stellt die Gewichtung λ einen freien (übergeordneten) Parameter dar. Dessen Wahl ist anwendungsspezifisch und im Vorhinein nicht einfach eingrenzbar. Ansätze mittels Versuch-und-Irrtum und anschließender Überprüfung der resultierenden Fitting-Ergebnisse (siehe Kap. 3.1.3) in Kombination mit einer Kreuzvalidierung (sieh Kap. 3.3.2) sind üblich. Weiterführende Details zur Ridge- und LASSO-Regression können beispielsweise [HTF17] entnommen werden.

---

[1] Die Abkürzungen stehen für *least absolute shrinkage and selection operator* (LASSO) sowie *ridge* (eng. Bergrücken), d. h. es soll versinnbildlicht werden, dass eine schlecht konditionierte Kostenlandschaft deart modifiziert wird, um das „Kostental" eindeutiger identifizieren zu können.

[2] Sei $\boldsymbol{x} \in \mathbb{R}$ ein Datenvektor mit empirischem Mittelwert μ und empirischer Standardabweichung σ. Die Standardisierung ergibt sich dann zu $\boldsymbol{z} = \frac{\boldsymbol{x}-\mu}{\sigma}$.

[3] Die LASSO-Regression wird daher in Black-Box-Problemen (beispielsweise Polynom-Fitting) auch gerne zur Wahl der Modellordnung herangezogen, da vernachlässigbar kleine Parametergewichte das Verwerfen des jeweiligen Regressors motivieren.

3.3 Bestimmung der Modellordnung

Bisher wurde die zu identifizierende Modellstruktur als bekannt bzw. gegeben angenommen. Dies ist, insbesondere in abstrakten Black-Box Identifikationen, aber häufig nicht der Fall, sodass neben den Parametern auch die Modellstruktur selbst bestimmt werden muss. Hierzu wird der Parametervektor für die nachfolgenden Betrachtungen aufgespalten:

$$\boldsymbol{\theta}^T = \begin{bmatrix} \boldsymbol{\theta}_0^T & \boldsymbol{\theta}_+^T \end{bmatrix}. \tag{3.124}$$

Zur Bestimmung der Modellordnung wird folgende Untersuchungsfrage formuliert:

Beeinflusst die Teilmenge $\boldsymbol{\theta}_+$ das Modell maßgeblich?

Sprich, es muss eine Testmetrik herangezogen werden, um zu bewerten, ob die Modellabdeckung für den Fall $\boldsymbol{\theta}_+ = 0$ signifikant reduziert wird. Hierzu bietet sich das bereits in (3.72) eingeführte SQR-Maß an. Dieses soll im Folgenden genutzt werden, um folgenden Hypothesentest zu bewerten:

H_0: Die Berücksichtigung von $\boldsymbol{\theta}_+$ führt lediglich zu einer zufälligen bzw. nicht signifikanten Erhöhung der Modellabdeckung. Daher ist $\boldsymbol{\theta}_+ = 0$ zu wählen.

H_1: Die Berücksichtigung von $\boldsymbol{\theta}_+$ führt zu einer Verbesserung der Modellabdeckung, welche über das Maß eines zufälligen Einflusses hinaus geht. Daher ist $\boldsymbol{\theta}_+ \neq 0$ zu wählen.

Hierzu werden für das reduzierte (H_0) sowie für das erweiterte Modell (H_1) entsprechende LS-Identifikationen durchgeführt und danach die SQR-Werte ermittelt:

$$\mathrm{SQR}_0 = \sum_{k=1}^{n} (\psi[k] - \hat{y}_0[k])^2, \quad \mathrm{SQR}_1 = \sum_{k=1}^{n} (\psi[k] - \hat{y}_1[k])^2. \tag{3.125}$$

Anschließend wird die relative Verbesserung der Modellabdeckung berechnet:

$$f = \frac{\frac{\mathrm{SQR}_0 - \mathrm{SQR}_1}{d_0 - d_1}}{\frac{\mathrm{SQR}_1}{d_1}} = \left(\frac{\mathrm{SQR}_0 - \mathrm{SQR}_1}{d_0 - d_1} \right) \left(\frac{d_1}{\mathrm{SQR}_1} \right). \tag{3.126}$$

Hier entspricht d der Anzahl der Modellfreiheitsgrade (*degrees of freedom*), d. h., der Differenz zwischen der Anzahl der Messpunkte n und der Anzahl der zu bestimmenden Modellparameter m[1]:

$$d = n - m. \tag{3.127}$$

Weiterhin kann die Modellgleichung (3.11) partitioniert werden zu:

$$\boldsymbol{\psi} = \boldsymbol{\Xi}\boldsymbol{\theta} + \boldsymbol{\nu} = \boldsymbol{\Xi}_0\boldsymbol{\theta}_0 + \tilde{\boldsymbol{\Xi}}\tilde{\boldsymbol{\theta}} + \boldsymbol{\nu}. \tag{3.128}$$

Mittels der LS-Lösungen

$$\hat{\boldsymbol{\theta}}_0 = \left(\boldsymbol{\Xi}_0^T \boldsymbol{\Xi}_0\right)^{-1} \boldsymbol{\Xi}_0^T \boldsymbol{\psi}, \quad \hat{\boldsymbol{\theta}} = \left(\boldsymbol{\Xi}^T \boldsymbol{\Xi}\right)^{-1} \boldsymbol{\Xi}^T \boldsymbol{\psi} \tag{3.129}$$

[1] Beachte, dass die Anzahl der Freiheitsgrade des reduzierten Modells d_0 aufgrund der geringeren Parameteranzahl bei gleicher Messpunktanzahl größer ist als die des vollständigen Modells d_1.

folgt dann für den SQR:

$$\mathrm{SQR}_0 = \boldsymbol{\psi}^T\boldsymbol{\psi} - \boldsymbol{\theta}_0^T\boldsymbol{\Xi}_0^T\boldsymbol{\psi}, \quad \mathrm{SQR}_1 = \boldsymbol{\psi}^T\boldsymbol{\psi} - \boldsymbol{\theta}^T\boldsymbol{\Xi}^T\boldsymbol{\psi}. \tag{3.130}$$

Wird neben den in Kap. 3.1.1 gemachten Annahmen zum LS-Verfahren weiterhin angenommen, dass das Rauschen $\boldsymbol{\nu}$ normalverteilt ist, folgt, dass das SQR-Maß der *Chi-Quadrat-Verteilung*[1] entspricht.

Die Testmetrik (3.126) wurde hingegen als Quotient der SQR-Werte definiert, sodass der Hypothesen-Test anhand der F-Verteilung (oder auch *Fisher-Verteilung*) vorzunehmen ist: Eine F-verteilte Zufallsvariable ergibt sich als Quotient zweier jeweils durch die zugehörige Anzahl der Freiheitsgrade geteilter Chi-Quadrat-verteilter Zufallsvariablen: $F(a, b)$. Hierbei entspricht a den Freiheitsgraden im Zähler und b denen im Nenner. Beispielhafte Verläufe der Wahrscheinlichkeitsdichte der F-Verteilung sind in Abb. 3.14 abgebildet.

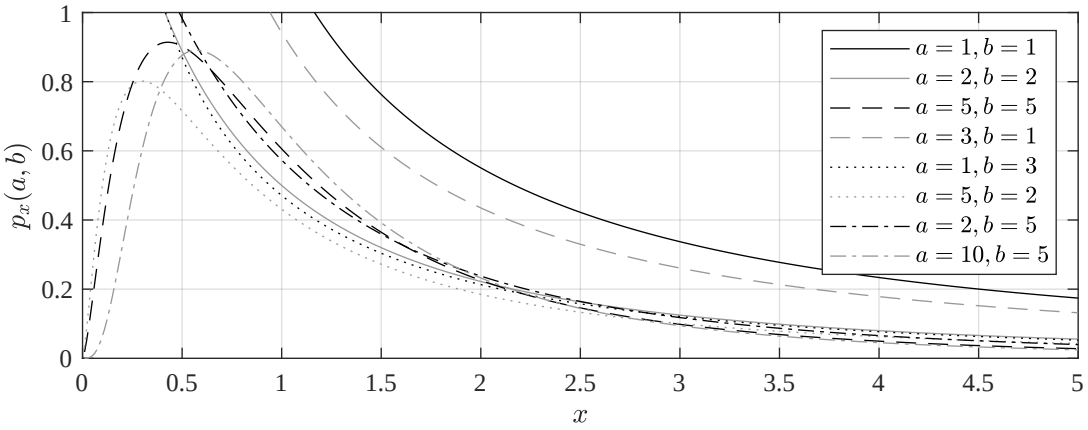


Abb. 3.14: Wahrscheinlichkeitsdichte der F-Verteilung für beispielhafte $\{a, b\}$

Um die Hypothese zu überprüfen, wird die Testmetrik (3.126) gegen ein Konfidenzniveau α bei gegebenen Freiheitsgraden $\{a, b\}$ der F-Verteilung verglichen:

$$F_{1-\alpha} = F_{1-\alpha}(a, b) = F_{1-\alpha}(d_0 - d_1, d_1) = F_{1-\alpha}(m_{\boldsymbol{\theta}} - m_{\boldsymbol{\theta}_0}, n - m_{\boldsymbol{\theta}}). \tag{3.131}$$

Hier ist $F_{1-\alpha}$ das entsprechende Quantil der F-Verteilung. Ist beispielsweise gefordert, dass im Mittel 95 % der untersuchten Modellveränderungen durch $\boldsymbol{\theta}_+$ systematischer und nicht zufälliger Natur waren, wird $\alpha = 0,95$ angesetzt. Die entsprechenden Quantilswerte können Tabellenbüchern entnommen oder numerisch berechnet werden. Somit ergibt sich zusammengefasst für den Hypothesentest:

[1] Die Chi-Quadrat-Verteilung (χ^2-Verteilung) ist eine Wahrscheinlichkeitsverteilung über der Menge der nicht negativen reellen Zahlen. Sie ist eine der Verteilungen, die aus der Normalverteilung abgeleitet werden kann: Hat man n Zufallsvariablen X_i, die unabhängig und standardnormalverteilt sind, so ist die Chi-Quadrat-Verteilung mit n Freiheitsgraden definiert als die Verteilung der Summe der quadrierten Zufallsvariablen $X_1^2 + \ldots + X_n^2$. Ihr einziger Parameter ist somit die Anzahl der Freiheitsgrade n.

Satz 3.14: Bestimmung der Modellordnung mittels Hypothesentest

Gegeben ist die Teilmenge $\boldsymbol{\theta}_+$ des Parametervektors $\boldsymbol{\theta}$. Diese trägt maßgeblich zur Modellabdeckung bei, falls

$$f > F_{1-\alpha} : H_0 \text{ verwerfen} \rightarrow \boldsymbol{\theta}_+\text{: signifikant.} \tag{3.132}$$

Andernfalls kann der Einfluss von $\boldsymbol{\theta}_+$ als zufällig eingestuft werden:

$$f \leq F_{1-\alpha} : H_1 \text{ verwerfen} \rightarrow \boldsymbol{\theta}_+\text{: irrelevant.} \tag{3.133}$$

Hier ist f eine Testmetrik entsprechend (3.126) *und $F_{1-\alpha}$ ein zu wählendes Quantil der F-Verteilung zum Konfidenzniveau α entsprechend* (3.131).

Es sei angemerkt, dass obiges Vorgehen auf Basis der SQR-Metrik lediglich ein möglicher Weg zur Überprüfung der Modellordnung ist. Analog zu diesem Vorgehen können andere Metriken gewählt und ein Hypothesentest auf der resultierende Zufallsverteilung durchgeführt werden. Beispielsweise führt die Metrik

$$t = \frac{\theta_+}{\hat{\sigma}_{\theta_+}}, \tag{3.134}$$

also der auf die empirische Standardabweichung $\hat{\sigma}_{\theta_+}$ normierte Parameter θ_+, zu einem Test auf Basis der Student-t-Verteilung. Je nach Modell und Datengrundlage (insb. bei praxisnahen Datensätzen, welche die LS-Annahmen typischerweise nicht erfüllen) kann es für die verschiedenen Metriken zudem zu unterschiedlichen Ergebnissen hinsichtlich der Signifikanz-Aussage bezüglich eines Parameters kommen. Daher kann es zielführend sein, mehrere Testmetriken heranzuziehen und die Einzelergebnisse (gewichtet) zu Mitteln.

Über den Hypothesentest können somit systematisch einzelne oder mehrere Parameter hinsichtlich ihrer Signifikanz im Gesamtmodell geprüft werden. Bezüglich der Modellstrukturgestaltung kann obiger Ansatz genutzt werden, um entweder manuelle Strukturveränderungen einer/eines Anwenderin/Anwenders zu prüfen oder um automatisiert eine geeignete Modellstruktur zu finden. Auf automatisiertem Wege bieten sich folgende, einfache Verfahren an:

- Vorwärtsselektion:
  Auf Basis einer leeren Modellstruktur werden anhand einer zuvor definierten Parametermenge so lange neue Parameter ergänzt, bis kein neuer Parameter mehr gefunden werden kann, der durch den Hypothesentest bestätigt wird.
- Rückwärtsselektion:
  Startpunkt ist ein stark überparametriertes Modell. Auf Basis der f-Metrik werden die Parameter mit der geringsten Aussagekraft für das Modell entfernt bis ein Parameterset übrig bleibt, welches durch den Hypothesentest bestätigt wird.

Diese Verfahrensweisen haben allerdings einen Einbahnstraßencharakter – ist ein Parameter einmal entfernt oder hinzugenommen, besteht nicht mehr die Möglichkeit diese Entscheidung zu revidieren. Dies ist insofern kritisch zu bewerten, da sich die f-Metrik und somit das Ergebnis des Hypothesentests nach jeder Modellstrukturänderung ebenfalls ändert. Demnach scheint eine Kombination aus beiden Verfahren, also ein schrittweises Hinzufügen von Parametern mit der Möglichkeit ihrer späteren Entfernung, zielführender. Diese Herangehensweise wird auch schrittweise Regression (*stepwise regression*) genannt – die algorithmische Umsetzung wird u. a. in [SL03] erörtert.

3.3.1 Bias-Varianz-Dilemma

Zur weiteren Diskussion der Modellordnung sei noch auf das Bias-Varianz-Dilemma hingewiesen. Hierzu soll folgende Identifikationsaufgabe betrachtet werden:

- Die wahre Modellstruktur sei bekannt.
- Die Varianz der identifizierten Parameter ist groß, da der Informationsgehalt in den Messdaten gering ist.

In diesem Fall kann das Entfernen von Parametern mit hoher Unsicherheit (Varianz) trotz des damit einhergehenden systematischen Modellierungsfehlers (Bias) zu einer insgesamt besseren Schätzgüte führen, da die verbleibenden Parameter genauer identifiziert werden können. Dieser Zusammenhang ist vereinfacht in Abb. 3.15 dargestellt.

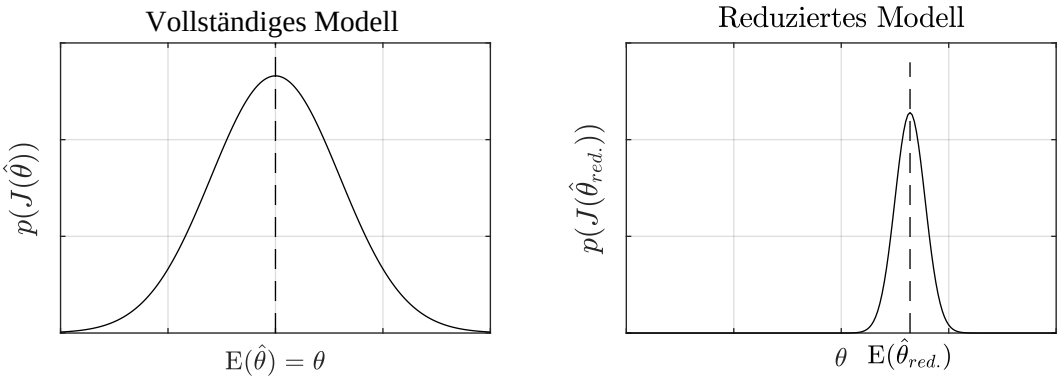


Abb. 3.15: Qualitative Wahrscheinlichkeitsdichte der quadratischen Kostenfunktion für zwei unterschiedliche Modellordnungen

Diese qualitative Betrachtung kann ebenfalls mathematisch untermauert werden, wenn die Kostenfunktion des LS-Schätzers (3.10) als Zufallsvariable aufgefasst wird. Der Erwartungswert dieser entspricht dann gerade den quadratischen Kosten (MSE - *mean squared error*):

$$\mathrm{MSE} = \mathrm{E}(J(\hat{\boldsymbol{\theta}})) = \mathrm{E}\left((\boldsymbol{\psi} - \boldsymbol{\Xi}\hat{\boldsymbol{\theta}})^T(\boldsymbol{\psi} - \boldsymbol{\Xi}\hat{\boldsymbol{\theta}})\right) = \mathrm{E}\left(\boldsymbol{\psi}^2 - 2\boldsymbol{\psi}\boldsymbol{\Xi}\hat{\boldsymbol{\theta}} + (\boldsymbol{\Xi}\hat{\boldsymbol{\theta}})^2\right). \tag{3.135}$$

Durch Anwendung des Verschiebungssatzes[1], $\hat{\boldsymbol{\psi}} = \boldsymbol{\Xi}\hat{\boldsymbol{\theta}}$ und der Zusammenhänge (siehe Definitionen in Abb. 3.1)

$$\begin{aligned} \boldsymbol{\psi} &= \boldsymbol{y} + \boldsymbol{\nu}\,, \\ \hat{\boldsymbol{y}} &= \boldsymbol{\Xi}\hat{\boldsymbol{\theta}} \end{aligned} \tag{3.136}$$

folgt:

$$\begin{aligned} \mathrm{MSE} &= \mathrm{E}\left((\boldsymbol{y} + \boldsymbol{\nu})^2 - 2(\boldsymbol{y} + \boldsymbol{\nu})\hat{\boldsymbol{y}} + \hat{\boldsymbol{y}}^2\right) \\ &= \mathrm{E}\left(\boldsymbol{y}^2 + 2\boldsymbol{y}\boldsymbol{\nu} + \boldsymbol{\nu}^2 - 2\boldsymbol{y}\hat{\boldsymbol{y}} - 2\boldsymbol{\nu}\hat{\boldsymbol{y}} + \hat{\boldsymbol{y}}^2\right) \\ &= \mathrm{E}(\boldsymbol{y}^2) + \mathrm{E}(2\boldsymbol{y}\boldsymbol{\nu}) + \mathrm{E}(\boldsymbol{\nu}^2) - 2\mathrm{E}(\boldsymbol{y}\hat{\boldsymbol{y}}) - 2\mathrm{E}(\boldsymbol{\nu}\hat{\boldsymbol{y}}) + \mathrm{E}(\hat{\boldsymbol{y}}^2). \end{aligned} \tag{3.137}$$

[1] Sei X eine Zufallsvariable. Dann gilt $\mathrm{E}(X^2) = \mathrm{Var}(X) + \mathrm{E}(X)^2$.

Weiterhin gilt es zu beachten, dass der tatsächliche Prozessausgang $\boldsymbol{y}$ keine Zufallsvariable ist, da der Prozess selbst deterministisch ist:

$$\mathrm{MSE} = \boldsymbol{y}^2 + 2\boldsymbol{y}\underbrace{\mathrm{E}(\boldsymbol{\nu})}_{0} + \mathrm{E}(\boldsymbol{\nu}^2) - 2\boldsymbol{y}\mathrm{E}(\hat{\boldsymbol{y}}) - 2\mathrm{E}(\boldsymbol{\nu}\hat{\boldsymbol{y}}) + \mathrm{E}(\hat{\boldsymbol{y}}^2). \tag{3.138}$$

Unter erneuter Verwendung des Verschiebungssatzes folgt:

$$\mathrm{MSE} = \boldsymbol{y}^2 + \underbrace{\mathrm{Var}(\boldsymbol{\nu})}_{\sigma_\nu^2} + \underbrace{\mathrm{E}(\boldsymbol{\nu})^2}_{0} - 2\boldsymbol{y}\mathrm{E}(\hat{\boldsymbol{y}}) - 2\mathrm{E}(\boldsymbol{\nu}\hat{\boldsymbol{y}}) + \mathrm{Var}(\hat{\boldsymbol{y}}) + \mathrm{E}(\hat{\boldsymbol{y}})^2. \tag{3.139}$$

Da das Rauschen $\boldsymbol{\nu}$ mit dem geschätzten Modellausgang $\hat{\boldsymbol{y}}$ unkorreliert ist, folgt zudem:

$$\mathrm{E}(\boldsymbol{\nu}\hat{\boldsymbol{y}}) = 0. \tag{3.140}$$

Umschreiben von (3.139) ergibt dann:

$$\mathrm{MSE} = \underbrace{(\boldsymbol{y} - \mathrm{E}(\hat{\boldsymbol{y}}))^2}_{\text{Bias}} + \mathrm{Var}(\hat{\boldsymbol{y}}) + \sigma_\nu^2. \tag{3.141}$$

Demnach setzt sich der Erwartungswert der Kostenfunktion aus folgenden Termen zusammen:

- Bias:
  Der systematische Fehler resultierend durch Strukturabweichungen zwischen Modell und tatsächlichem Prozess.
- Varianz des Modells:
  Die Unsicherheit, welche durch das Schätzverfahren eingebracht wird.
- Varianz des Rauschens:
  Der unvermeidbare bzw. nicht beeinflussbare Fehleranteil.

Dies unterstreicht, wie im Beispiel aus Abb. 3.15 bereits angedeutet, dass das Akzeptieren eines systematischen Modellfehlers die Schätzgüte im Sinne der Kostenfunktion dennoch steigern kann. Diese Beobachtung ist als *Bias-Varianz-Dilemma* bekannt, welches in Abb. 3.16 nochmals in verallgemeinerter Form verdeutlicht wird: Mit steigender Modellkomplexität bzw. Modellfreiheitsgraden findet ein *Overfitting* statt, d. h., das gegebene Modell lernt die Messdaten auswendig. Die Modellvarianz steigt übermäßig stark an. Werden hingegen zu wenig Modellfreiheitsgrade bereitgestellt, kann das Systemverhalten nur unzureichend modelliert werden und systematische Schätzfehler (Bias) sind die Folge. Dieses wird als sog. *Underfitting* bezeichnet.

3.3.2 Kreuzvalidierung

Die Kreuzvalidierung ist eine Modellvalidierungstechnik zur Beurteilung der *Generalisierungsfähigkeit* eines zu identifizierenden Modells. Hierzu wird der zur Verfügung stehende Datensatz aufgespalten: ein oder mehrere Teile werden als *Trainingsdatensätze* zur Identifikation der Modellparameter herangezogen, während ein Teil des Datensatzes mit unbekannten Daten zurückgehalten wird, um auf dessen Basis die Modellgüte zu bewerten (*Validierungs-* oder *Testdatensatz*). Hierdurch soll insbesondere das zuvor beschriebene Overfitting-Problem, also die Überanpassung eines Modells auf den Trainingsdatensatz, erkannt werden. Die Kreuzvalidierung eignet sich daher als:

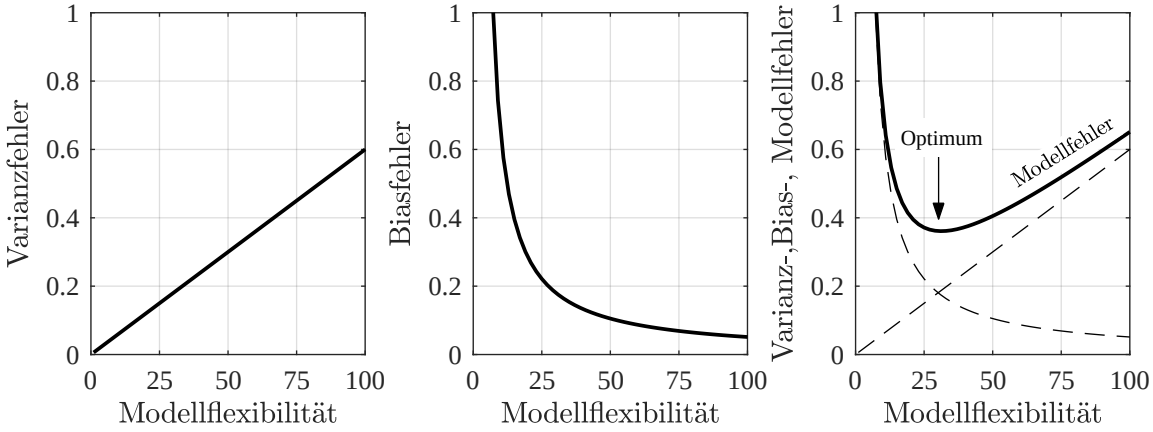


Abb. 3.16: Prinzipdarstellung des Bias-Varianz-Dilemmas

- **Allgemeine Modellvalidierungstechnik**
  Wie gut reagiert ein identifiziertes Modell auf ungesehene Daten (Robustheit)?

- **Modellselektion**
  Welche Modellstruktur bzw. Regressoren führen zu einer besonders hohen Schätzgüte?

Neben verschiedenen Varianten ist die sog. k-fach Kreuzvalidierung (*k-fold cross-validation*) die am häufigst anzutreffenden Umsetzung [1]. Die Vorgehensweise bei dieser Variante unterscheidet sich je nachdem, ob eine einfache Modellvalidierung oder eine Modellselektion durchgeführt werden soll. Für den ersten Fall ist die Umsetzung wie folgt:

1. Aufteilung des Datensatzes $\boldsymbol{D} = \{\boldsymbol{\Xi}, \boldsymbol{\psi}\}$ in k separate Teile $\boldsymbol{D} = \{\boldsymbol{D}_1, \boldsymbol{D}_2, \ldots, \boldsymbol{D}_k\}$ (ungefähr) gleicher Größe.
2. Identifikation der Modelle $\hat{f}_i$ für $i = 1, \ldots, k$ unter Nutzung aller Trainingsdatensätze außer $\boldsymbol{D}_i$.
3. Bewertung der Modellgüte J_i von $\hat{f}_i$ für $i = 1, \ldots, k$ anhand des zuvor ausgeblendeten Datensatzes $\boldsymbol{D}_i$, welcher somit als Testdatensatz dient.
4. Aggregierung der $i = 1, \ldots, k$ Bewertungsgrößen J_i und abschließende Diskussion.

Es werden demnach k-Modelle gleicher Struktur unabhängig voneinander identifiziert, welche sich lediglich innerhalb der Parameterwerte unterscheiden. Die abschließende Aggregierung der Bewertungsgrößen J_i kann beispielsweise anhand der Streuung und des Mittelwerts des MSEs geschehen, um die Modellperformanz auf ihre Robustheit hin zu bewerten.

Wird die Kreuz-Validierung demgegenüber genutzt, um verschiedene Modelle gegeneinander zu vergleichen (Modellselektion), ist obiger Ablauf geringfügig zu erweitern, da hier $j = 1, 2, \ldots$ verschiedene Modellstrukturen $\hat{f}^j$ betrachtet werden:

1. Aufteilung des gesamten Datensatzes $\boldsymbol{D} = \{\boldsymbol{\Xi}, \boldsymbol{\psi}\}$ in einen Trainingsteil $\boldsymbol{D}_t$ sowie Validierungsteil $\boldsymbol{D}_v$.

[1] Weitere Modellvalidierungs- und Selektionsmethoden, wie z. B. die vollständie Kreuzvalidierung oder Bootstrapping-Verfahren, können u. a. [HTF17][Ras18] entnommen werden.

2. Aufteilung des Trainingsdatensatzes in k separate Teile $\boldsymbol{D}_t = \{\boldsymbol{D}_{t,1}, \boldsymbol{D}_{t,2}, \ldots, \boldsymbol{D}_{t,k}\}$ (ungefähr) gleicher Größe.
3. Identifikation der Modelle $\hat{f}_i^j$ für $i = 1, \ldots, k$ unter Nutzung aller Trainingsdatensätze außer $\boldsymbol{D}_{t,i}$.
4. Bewertung der Modellgüte von $\hat{f}_i^j$ für $i = 1, \ldots, k$ anhand des zuvor ausgeblendeten Datensatzes $\boldsymbol{D}_{t,i}$.
5. Aggregierung der $i = 1, \ldots, k$ Bewertungsgrößen zur Beschreibung der Performanz des Modells $\hat{f}_i^j$.
6. Wiederholung der Schritte 3-5 für alle in Frage kommenden $j = 1, 2, \ldots$ Modelle.
7. Auswahl der optimalen Modellstruktur $\hat{f}^*$ und abschließende Überprüfung auf dem zurückgehaltenen Datensatz $\boldsymbol{D}_v$.

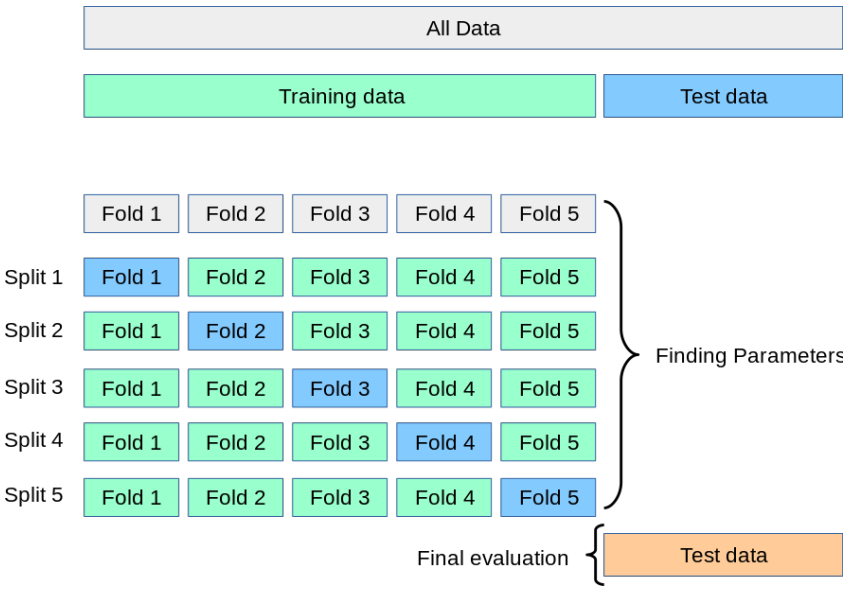


Abb. 3.17: Darstellung einer 5-fach Kreuzvalidierung (Quelle: https://scikit-learn.org, abgerufen: 26.08.2019)

Dieser Vorgang ist in Abb. 3.17 zudem graphisch dargestellt. Bei der Durchführung der Kreuzvalidierung bestehen demnach verschiedene Freiheitsgrade:

- **Stückelung**
  Grundsätzlich kann der Parameter k zur Stückelung des Datensatzes frei gewählt werden. Gegen eine zu feine Stückelung, also ein großer k-Wert spricht, dass die resultierenden Datenteilsätze zu klein und somit ggf. nicht mehr repräsentativ sind (siehe Kap. 3.1.3) sowie das eine hohe Anzahl an Identifikationsvorgängen mit entsprechend großem Berechnungsaufwand notwendig sind. Ein zu kleiner k-Wert hingegen führt ggf. zu einer nicht ausreichenden Diversifikation des Datensatzes, sodass die Modellbewertung wenig belastbar ist. In der Praxis ist die Wahl $k = \{3, 5, 10\}$ häufig anzutreffen.
- **Bewertungsmetrik**
  Die Kenngröße zur Bewertung der Modellperformanz ist bei der Kreuzvalidierung nicht

vorgegeben und sollte je nach Anwendung (Regression, Klassifikation, etc.) gewählt werden. Häufig wird allerdings der MSE herangezogen, da hier statistische Auswertung im Zuge der Aggregierung besonders intuitiv ist.

- **Datenaufteilung**
  Klassischerweise werden die Daten zufällig auf die k-Teilmengen aufgeteilt. Bei kleinen Datensets kann dies ggf. zu Schieflagen führen, da die Datensets zu heterogen sind. In diesem Fall kann eine (teilweise) manuelle Zuteilung sinnvoll sein[1]. Darüber hinaus besteht bei der Modellselektion der Freiheitsgrad den Testdatensatz zu bestimmen: auch hier kann entweder eine beliebiger zufälliger Teil vom Gesamtdatensatz oder manuell ein für die Anwendung besonders repräsentativer Teil ausgewählt werden.

Die Kreuzvalidierung ist in der Praxis ein beliebtes Werkzeug, um ein Overfitting zu vermeiden und somit die Robustheit eines identifizierten Modells zu überprüfen. Nichtsdestotrotz ist diese Methode unmittelbar abhängig von den zur Verfügung stehenden Daten, sodass beispielsweise ein schlecht konditionierter Datensatz oder eine nicht-repräsentative Abdeckung des zu identifizierenden Systems durch die Kreuzvalidierung nicht behoben werden können. Sie motiviert vielmehr, dass eine ausreichend große Menge informativer Datenpunkte im Vorfeld der Identifikation gesammelt werden müssen.

3.4 Nichtlineare Problemstellungen

Die bisherigen Betrachtungen gingen von einer linearen Identifikationsaufgabe aus, d. h., es galt ein lineares Gleichungssystem der Form (3.78) zu lösen. Hierzu wurden in Kap. 3.1.4 explizite und numerische Verfahren vorgestellt, welche jeweils eindeutige Lösungen hervorbrachten. Allerdings existieren zahlreiche Identifikationsprobleme, bei denen die gesuchten Parameter nichtlinear in die Modellgleichung eingehen. Als einfaches Beispiel sei hier die Shockley-Gleichung genannt, welche die Strom-Spannungs-Kennlinie einer Halbleiterdiode in Durchlassrichtung beschreibt:

$$I_D = I_S(T)\left(e^{\frac{U_F}{\kappa \cdot U_T(T)}} - 1\right). \tag{3.142}$$

Hier ist I_D der Diodenstrom, I_S der von der absoluten Temperatur T abhängige Sättigungssperrstrom, U_F die Flussspannung, κ der Emissionskoeffizient sowie U_T die Temperaturspannung, welche ebenfalls temperaturabhängig ist

$$U_T(T) = \frac{k_B T}{q} \tag{3.143}$$

und die Boltzmannkonstante k_B sowie die Elementarladung q beinhaltet. Angenommen es soll eine neue Diode vermessen werden, um wichtige Kennwerte für ein entsprechendes Datenblatt zu gewinnen – konkret sind der Emissionskoeffizient κ und der Sättigungssperrstrom I_S zu ermitteln. Für eine Messreihe werde die Diodentemperatur konstant gehalten, sodass I_S eine konstante Unbekannte und U_T eine bekannte Konstante ist. Der Diodenstrom I_D sei die Messgröße, welche durch das Anlegen verschiedener, konstanter Spannungen U_F an der Diode

[1] Dies ist auch bekannt als *stratifizierte Kreuzvalidierung*, bei der durch manuelles Eingreifen oder automatisierte Verfahren eine möglichst gleiche Verteilung zwischen den k Teilmengen angestrebt wird.

angeregt wird. Es gilt somit:

$$\boldsymbol{\theta} = \begin{bmatrix} I_S & \kappa \end{bmatrix}^T, \quad \boldsymbol{\psi} = \begin{bmatrix} I_D[1] & \dots & I_D[n] \end{bmatrix}^T, \quad \boldsymbol{u} = \begin{bmatrix} U_F[1] & \dots & U_F[n] \end{bmatrix}^T. \tag{3.144}$$

Eine beispielhafte Diodenkennlinie ist hierzu in Abb. 3.18 skizziert. Die Residuengleichung für

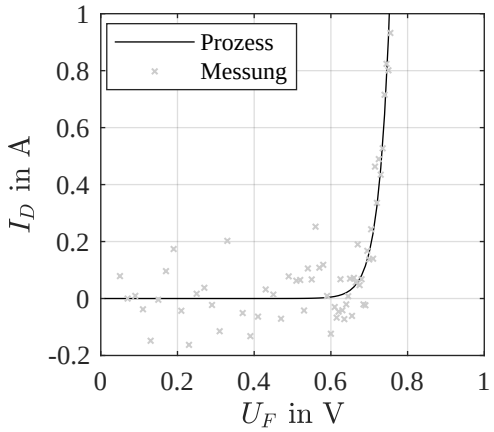


Abb. 3.18: Beispielhafte Diodenkennlinie mit einem additiven und unkorrelierten Messrauschen der Form $\nu \sim \mathcal{N}(0, \sigma = 0,1\ \mathrm{A})$

einen Messpunkt k ergibt sich somit zu

$$e[k] = \psi[k] - I_S \left(e^{\frac{U_F[k]}{\kappa \cdot U_T}} - 1 \right) = \psi[k] - f(u[k], \boldsymbol{\theta}). \tag{3.145}$$

Die Kostenfunktion im Sinne eines LS-Problems bleibt strukturell unverändert:

$$J(\boldsymbol{\theta}) = \sum_{k=1}^{n} (e[k])^2 = \sum_{k=1}^{n} \left(\psi[k] - f(u[k], \boldsymbol{\theta}) \right)^2. \tag{3.146}$$

Somit kann das nichtlineare LS-Problem wie folgt zusammengefasst werden:

Definition 3.7: LS-Problem für nichtlineare, statische Systeme

Gegeben sei eine Regressionsgleichung der Form

$$\boldsymbol{e} = \boldsymbol{\psi} - \boldsymbol{f}(\boldsymbol{u}, \boldsymbol{\theta}) \tag{3.147}$$

mit dem Parametervektor $\boldsymbol{\theta} \in \mathbb{R}^m$, dem Messdatenvektor $\boldsymbol{\psi} \in \mathbb{R}^n$, einer nichtlinearen Modellfunktion $\boldsymbol{f} : \mathbb{R}^m \to \mathbb{R}^n$ sowie dem Residuenvektor $\boldsymbol{e} \in \mathbb{R}^n$. Es gelte $m < n$. Die Messdaten $\boldsymbol{\psi}$ weisen ein additives Messrauschen ν mit $\mathrm{E}(\nu) = 0$ und $\mathrm{Cov}(\nu) = \sigma^2$ auf. Das Auffinden des Parametervektors $\boldsymbol{\theta}$ mittels Minimierung der quadratischen Kostenfunktion (3.146) entsprechend

$$\boldsymbol{\theta}^* = \arg\min J(\boldsymbol{\theta}) \tag{3.148}$$

wird als LS-Problem für nichtlineare, statische Systeme bezeichnet.

Da im Gegensatz zum linearen LS-Problem (3.9) der Residuenterm (3.145) aufgrund des nichtlinearen Zusammenhangs zwischen Messgröße und Parametervektor $f(u[k], \boldsymbol{\theta})$ ebenfalls nichtlinear ist, kann auf Basis der Kostenfunktion (3.146) keine geschlossene Lösung gefunden werden.

Der Blick auf die partiellen Ableitungen von (3.146)

$$\frac{\partial J}{\partial \boldsymbol{\theta}} = \nabla J(\boldsymbol{\theta}) = 2 \left[\sum_{k=1}^{n} e[k] \frac{\partial e[k]}{\theta_1} \quad \sum_{k=1}^{n} e[k] \frac{\partial e[k]}{\theta_2} \quad \dots \quad \sum_{k=1}^{n} e[k] \frac{\partial e[k]}{\theta_m} \right] \tag{3.149}$$

bzw. auf das gegebene Beispiel der Diode

$$\begin{aligned} e[k] \frac{\partial e[k]}{\partial I_s} &= \left(\psi[k] - I_S \left(e^{\frac{U_F[k]}{\kappa \cdot U_T}} - 1 \right) \right) \left(1 - e^{\frac{U_F[k]}{\kappa \cdot U_T}} \right), \\ e[k] \frac{\partial e[k]}{\partial \kappa} &= \left(\psi[k] - I_S \left(e^{\frac{U_F[k]}{\kappa \cdot U_T}} - 1 \right) \right) \left(I_s \frac{U_F[k]}{\kappa^2 \cdot U_T} e^{\frac{U_F[k]}{\kappa \cdot U_T}} \right) \end{aligned} \tag{3.150}$$

verrät, dass innerhalb der Ableitungen sowohl die gesuchten Parameter untereinander als auch die Parameter mit den Regressoren nichtlinear gekoppelt sind und somit eine analytische Umformung zur Auffindung der Nullstelle des Gradienten i. d. R. nicht möglich ist. Um das Identifikationsproblem trotzdem zu lösen, bestehen prinzipiell folgende Möglichkeiten:

Transformation auf ein lineares Problem
In manchen Spezialfällen ist es möglich, eine nichtlineare Modellfunktion in eine Lineare zu überführen. Als akademisches Beispiel sei

$$f(u[k], \boldsymbol{\theta}) = \theta_1 e^{\theta_2 u[k]} \tag{3.151}$$

genannt. Durch Anwendung des Logarithmus auf beiden Seiten ergibt sich dann der lineare Zusammenhang:

$$\ln\left(f(u[k], \boldsymbol{\theta})\right) = \ln\left(\theta_1\right) + \theta_2 u[k]. \tag{3.152}$$

Um eine quadratische Kostenfunktion in Analogie zum linearen LS-Problem zu erhalten, folgt:

$$\begin{aligned} J(\boldsymbol{\theta}) &= \sum_{k=1}^{n} (e[k])^2 = \sum_{k=1}^{n} \left(\ln\left(\psi[k]\right) - \ln\left(f(u[k], \boldsymbol{\theta})\right) \right)^2 \\ &= \sum_{k=1}^{n} \left(\ln\left(\psi[k]\right) - \ln\left(\theta_1\right) - \theta_2 u[k] \right)^2. \end{aligned} \tag{3.153}$$

Augenscheinlich kann der lineare Lösungsansatz (3.23) auf das obige, transformierte Problem angewandt werden, sodass prinzipiell eine geschlossene Lösung existiert. Allerdings werden die Messungen $\psi[k]$ ebenfalls nichtlinear transformiert, sodass dies unmittelbar auch auf das Rauschen $\nu[k]$ zutrifft. Wird analog zu Definition 3.1 ein mittelwertfreies, additives Rauschen mit konstanter Varianz angenommen, folgt:

$$\ln\left(\psi[k]\right) = \ln\left(y[k] + \nu[k]\right) = \underbrace{\ln\left(y[k]\right)}_{\text{Transf. Messung: } \tilde{y}} + \underbrace{\ln\left(1 + \frac{\nu[k]}{y[k]}\right)}_{\text{Transf. Rauschen: } \tilde{\nu}}. \tag{3.154}$$

Der transformierte Rauschterm $\tilde{\nu}$ korreliert dann mit dem Prozessausgang y, sodass eine der zentralen Annahmen für die Lösung des linearen LS-Problems verletzt wird. Dies bedeutet, dass die in Kap. 3.1.1 diskutierten Eigenschaften des LS-Schätzers nicht mehr gelten. Folglich kann der gefundene Parametervektor systematische Schätzfehler aufweisen. Insofern ist die Anwendung einer linearisierenden Transformation i. d. R. als kritisch zu bewerten, sofern das reale Prozessrauschen $\nu[k]$ durch die Transformation nicht auf ein mittelwertfreies, additives Rau-

schen mit konstanter Varianz überführt werden kann. In obigem Beispiel wäre dies der Fall, sofern $\nu[k]$ der Normalverteilung folgt und multiplikativ-exponentiell auf den Prozessausgang wirke:

$$\ln\left(\psi[k]\right) = \ln\left(y[k]e^{\nu[k]}\right) = \ln\left(y[k]\right) + \nu[k]. \tag{3.155}$$

Dies ist natürlich ein rein mathematisch-abstraktes Beispiel ohne nennenswerten Anwendungsbezug.

Numerische Optimierung

Zum Auffinden der Nullstelle in (3.149) kann der Parametervektor iterativ angepasst werden:

$$\boldsymbol{\theta}[k+1] = \boldsymbol{\theta}[k] + \Delta\boldsymbol{\theta}. \tag{3.156}$$

Hier ist k der entsprechende Iterationsschritt. Es gilt demnach Strategien zu finden, um $\Delta\boldsymbol{\theta}$ möglichst zielführend zwischen den verschiedenen Iterationsschritten anzupassen bzw. Abbruchbedingungen zu definieren, welche eine erfolgreiche Identifikation repräsentieren. Diese Fragestellungen werden im nachfolgenden Kapitel zur numerischen Optimierung eingehend behandelt.

4 Numerische Optimierungsverfahren

Unter Optimierung versteht man i. A. die Suche nach einem im Sinne einer bestimmten Zielsetzung bestmöglichen Punkt (optimale Lösung) in einem Entscheidungsraum, wobei hinsichtlich dieser Suche meist Nebenbedingungen zu berücksichtigen sind. Im Folgenden werden einige Grundlagen zur Lösung von Optimierungsproblemen mittels numerischer Methoden zusammengefasst. Hierbei besteht jedoch ausdrücklich nicht der Anspruch, einen vollständigen Überblick über dieses Wissenschaftsfeld zu geben, sondern vielmehr ist dieses Kapitel als Einstieg in diese Thematik zu verstehen. Interessierten Leserinnen und Lesern sei darüber hinaus der Blick in die weiterführende Literatur, z. B. in [NW06][PLB12][Ste18], empfohlen.

4.1 Grundlagen

Zunächst sollen Klassen von Optimierungsproblemen (hinsichtlich ihrer wichtigen Eigenschaften) charakterisiert und voneinander abgegrenzt werden. Den Anfang machen:

Definition 4.1: Statische Optimierungsprobleme

Die Minimierung einer Funktion mit Optimierungsvariablen, die Elemente eines finitdimensionalen Raumes (z. B. dem Euklidischen Raum) sind, werden statische Optimierungsprobleme genannt. Die Standardformulierung eines statischen Optimierungsproblems lautet

$$\min_{\boldsymbol{x}} \quad J(\boldsymbol{x})\,, \tag{4.1}$$

$$\text{u. d. Nb.} \quad \boldsymbol{g}(\boldsymbol{x}) = \boldsymbol{0}\,, \tag{4.2}$$

$$\boldsymbol{h}(\boldsymbol{x}) \leq \boldsymbol{0}. \tag{4.3}$$

Hier ist $J : \mathbb{R}^n \to \mathbb{R}$ eine skalare Kostenfunktion, $\boldsymbol{x} \in \mathbb{R}^n$ ein Vektor mit den Optimierungsparametern, $\boldsymbol{g} : \mathbb{R}^n \to \mathbb{R}^p$ eine Funktion zur Beschreibung von Gleichungsbedingungen und $\boldsymbol{h} : \mathbb{R}^n \to \mathbb{R}^q$ eine Funktion zur Beschreibung von Ungleichungsbedingungen.

Entfallen die Nebenbedingungen (4.2) sowie (4.3), so liegt ein *unbeschränktes Optimierungsproblem* vor, andernfalls handelt es sich um ein *beschränktes Optimierungsproblem*. Die Menge $\mathcal{X} \in \mathbb{R}^n$, welche (4.2) und (4.3) erfüllt,

$$\mathcal{X} = \{\boldsymbol{x} \in \mathbb{R}^n | g_i(\boldsymbol{x}) = 0, i = 1, \ldots, p, \quad h_j(\boldsymbol{x}) \leq 0, j = 1, \ldots, q\} \tag{4.4}$$

wird *zulässiges Gebiet* genannt. Ferner wurde die Formulierung (4.1) derart gewählt, dass eine Optimierung stets ein Minimierungsproblem darstelle. Dies stellt insofern keine Einschränkung, sondern lediglich eine (stilistische) Definitionsfrage dar, da jedes Maximierungsproblem in eine

Minimierungsaufgabe überführt werden kann:

$$\max_{\boldsymbol{x}} \; J(\boldsymbol{x}) = \min_{\boldsymbol{x}} \; -J(\boldsymbol{x}). \tag{4.5}$$

Auf Basis dieser zunächst sehr allgemeinen Problemformulierung, können verschiedene Klassen von statischen Optimierungsaufgaben differenziert werden:

Definition 4.2: Typische Klassen statischer Optimierungsprobleme

Bei statischen Optimierungsproblemen werden häufig folgende Klassen unterschieden:

- *Lineare Optimierung: Die Kostenfunktion und die Beschränkungen sind linear.*
- *Quadratische Optimierung: Die Kostenfunktion ist quadratisch, während die Beschränkungen linear sind.*
- *Konvexe Optimierung [1]: Die notwendigen Optimalitätsbedingungen erster Ordnung sind gleichzeitig hinreichende Bedingungen für ein globales Optimum, d. h., dass jedes lokale Optimum auch ein globales ist.*
- *Nichtlineare Optimierung: Die Kostenfunktion und/oder mindestens eine Beschränkung sind nichtlinear.*
- *Ganzzahlige Optimierung: Alle Optimierungsvariablen nehmen diskrete Werte an.*
- *Gemischt-Ganzzahlige Optimierung: Es treten diskrete und kontinuierliche Optimierungsvariablen auf.*

Je nach zugrundeliegendem Optimierungsproblem sind verschiedene Optima bzw. Minima zu differenzieren:

Definition 4.3: Globale und lokale Minima

Die Kostenfunktion $J(\boldsymbol{x})$ besitzt in $\mathcal{X}$ an der Stelle $\boldsymbol{x}^*$

(a) ein lokales Minimum, falls für eine Norm $||\cdot||$ ein $\varepsilon > 0$ existiert, sodass $\{J(\boldsymbol{x}^*) \leq J(\boldsymbol{x})\}$ für alle $\boldsymbol{x} \in \{\boldsymbol{x} \in \mathcal{X}, ||\boldsymbol{x} - \boldsymbol{x}^*|| \leq \varepsilon\}$ gilt,

(b) ein striktes lokales Minimum, falls für eine Norm $||\cdot||$ ein $\varepsilon > 0$ existiert, sodass $J(\boldsymbol{x}^*) < J(\boldsymbol{x})$ für alle $\boldsymbol{x} \in \{\boldsymbol{x} \in \mathcal{X}, ||\boldsymbol{x} - \boldsymbol{x}^*|| < \varepsilon, \boldsymbol{x} \neq \boldsymbol{x}^*\}$ gilt,

(c) ein globales Minimum, falls $J(\boldsymbol{x}^*) \leq J(\boldsymbol{x})$ für alle $\boldsymbol{x} \in \mathcal{X}$ gilt, und

(d) ein striktes globales Minimum, falls $J(\boldsymbol{x}^*) < J(\boldsymbol{x})$ für alle $\boldsymbol{x} \in \{\boldsymbol{x} \in \mathcal{X}, \boldsymbol{x} \neq \boldsymbol{x}^*\}$ gilt.

Die unterschiedlichen Minima-Arten sind in Abb. 4.1 illustriert. Bereits in dieser einfachen Skizze wird deutlich, dass die Minima-Klassifikation von besonderer Bedeutung ist, da es das Ziel jeder Optimierung ist, in das (strikt) globale Optimum zu konvergieren. Insofern stellt sich im Zuge eines gegebenen Optimierungsvorgangs die Frage, ob ein gefundenes Minimum globaler

[1] Die Klassen der linearen und quadratischen Optimierung zählen ebenfalls zur konvexen Optimierung. Ein weiteres Beispiel für die konvexe Optimierung ist z. B. ein sog. *Geometrisches Programm*, bei dem die Kostenfunktion und Nebenbedingungen in polynomialer Form vorliegen.

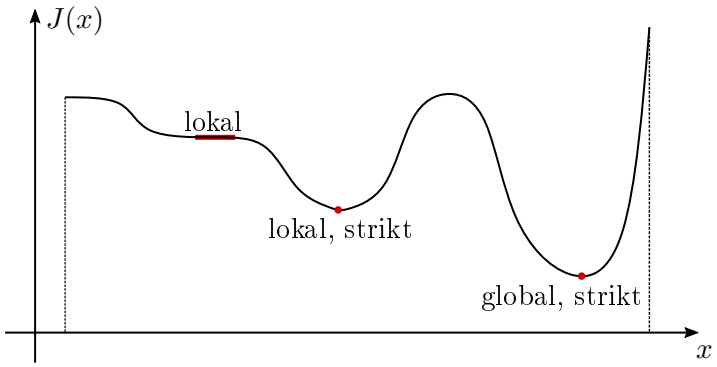


Abb. 4.1: Schematische Darstellung der verschiedenen Minima für eine Funktion $J(x)$ mit $x \in \mathbb{R}$

oder lokaler Natur ist. Hier hat die Klasse der *konvexen Optimierungsprobleme* die besondere Eigenschaft, dass jedes lokale Optimum auch ein globales Optimum ist, sodass im Sinne der Optimierungsaufgabe das bestmögliche Ergebnis sicher gefunden werden kann. Daher werden im folgenden Teilkapitel kurz die Eigenschaft der Konvexität diskutiert.

Zuvor findet aber noch die Abgrenzung der bisher betrachteten statischen Optimierungsprobleme zur dynamischen Variante statt:

Definition 4.4: Dynamische Optimierungsprobleme

Die Minimierung eines Funktionals von Optimierungsvariablen, die Elemente eines unendlich-dimensionalen Raumes sind (z. B. Zeit), werden dynamische Optimierungsprobleme genannt. Ihre Standardformulierung lautet

$$\min_{\boldsymbol{u}} \quad J(\boldsymbol{u}) = E(t_1, \boldsymbol{x}(t_1)) + \int_{t_0}^{t_1} L(t, \boldsymbol{x}(t), \boldsymbol{u}(t))\mathrm{d}t\,, \tag{4.6}$$

$$\text{u. d. Nb.} \quad \dot{\boldsymbol{x}} = \boldsymbol{f}(t, \boldsymbol{x}(t), \boldsymbol{u}(t)), \quad \boldsymbol{x}(t_0) = \boldsymbol{x}_0\,, \tag{4.7}$$

$$\boldsymbol{h}(\boldsymbol{x}(t), \boldsymbol{u}(t)) \leq \boldsymbol{0}. \tag{4.8}$$

Hier ist $\boldsymbol{J} : \mathbb{R}^n \to \mathbb{R}$ ein Kostenfunktional, welches sich aus einem Strafterm für den Endzustand E sowie für die Wegtrajektorie L zusammensetzt. Ferner ist $\boldsymbol{x}(t)$ der Systemzustand, $\boldsymbol{u}(t)$ die gesuchte Stellgrößentrajektorie, $\boldsymbol{f}$ die Systemfunktion und $\boldsymbol{h}$ eine Funktion zur Beschreibung von Ungleichungsbedingungen.

Neben der Bezeichnung dynamische Optimierung werden häufig auch die Begriffe *unendlich-dimensionale Optimierung*, *Optimalsteuerungsproblem* oder *dynamische Programmierung* verwendet. Wird das dynamische Optimierungsproblem auf einem rollierenden (fortlaufenden) Zeithorizont $[t_0, t_1]$ gelöst, spricht man auch von einer *modellprädiktiven Regelung*. Da dynamische Optimierungsprobleme allerdings eine eigenständige, komplexe Klasse innerhalb der mathematischen Optimierung darstellen, welche eine umfängliche Behandlung über die zulässigen Grenzen dieser Lehrveranstaltung hinaus erfordern, wird im Folgenden auf diese nicht weiter eingegangen. Somit liegt der Fokus im Weiteren auf statischen Optimierungsproblemen.

4.1.1 Konvexität

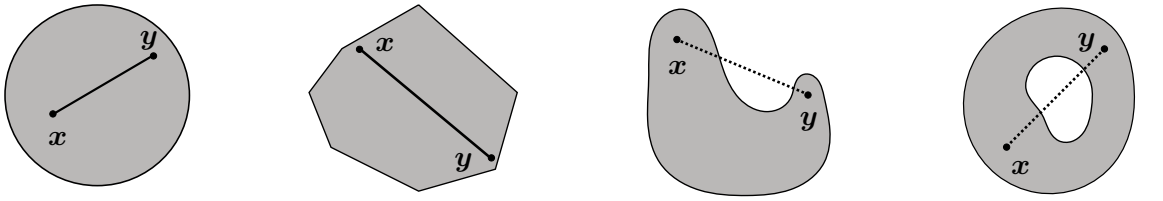
Der Begriff der *Konvexität* kann u. a. auf Mengen angewandt werden:

Definition 4.5: Konvexe Menge

Eine Menge $\mathcal{X} \in \mathbb{R}^n$ nennt man konvex, falls für $\{\boldsymbol{x}, \boldsymbol{y}\} \in \mathcal{X}$ und $\alpha \in \{\mathbb{R} | 0 < \alpha < 1\}$ gilt:

$$(\alpha \boldsymbol{x} + (1 - \alpha)\boldsymbol{y}) \in \mathcal{X}. \tag{4.9}$$

Die geometrische Interpretation dieser Definition ist, dass eine Menge $\mathcal{X} \in \mathbb{R}^n$ genau dann konvex ist, wenn die Verbindungslinie zwischen zwei beliebigen Punkten $\{\boldsymbol{x}, \boldsymbol{y}\} \in \mathcal{X}$ komplett in $\mathcal{X}$ enthalten ist. Dies wird beispielhaft in Abb. 4.2 illustriert.



(a) konvex (b) konvex (c) nicht-konvex (d) nicht-konvex

Abb. 4.2: Beispiele von konvexen und nicht-konvexen Mengen in $\mathbb{R}^2$

Im Kontext der Optimierung lassen sich einige interessante Eigenschaften konvexer Mengen zusammenfassen:

Satz 4.1: Ausgewählte Eigenschaften konvexer Menge

Seien $\{\mathcal{X}, \mathcal{Y}\} \in \mathbb{R}^n$ konvexe Mengen. Dann gelten die folgenden Aussagen:

(a) Die Schnittmenge $\mathcal{X} \cap \mathcal{Y}$ ist ebenfalls konvex.

(b) Die Menge $\{\boldsymbol{x} + \boldsymbol{y} | \boldsymbol{x} \in \mathcal{X}, \boldsymbol{x} \in \mathcal{Y}\}$ (Minkowski-Summe[1]) ist ebenfalls konvex.

(c) Für eine Matrix $\boldsymbol{A} \in \mathbb{R}^{n \times m}$ und einen Vektor $\boldsymbol{b} \in \mathbb{R}^n$ ist die transformierte Menge $\{\boldsymbol{A}\boldsymbol{x} + \boldsymbol{b} | \boldsymbol{x} \in \mathcal{X}\}$ ebenfalls konvex.

Obige Eigenschaften sind u. a. hilfreich, um das zulässige Gebiet $\mathcal{X}$ (Beschränkungen) einer Optimierungsaufgabe zu charakterisieren. Sie können genutzt werden, um z. B. gezielt eine konvexe Menge als zulässiges Gebiet zu definieren bzw. die Beschränkungen $\boldsymbol{g}$ und $\boldsymbol{h}$ entsprechend zu manipulieren. Typische konvexe Menge sind:

- Strecken und Geraden
- Regelmäßige Polygonflächen (z. B. Dreiecksflächen oder Hexagon)
- Kreisscheiben und Kugeln
- Würfel

[1] Die Minkowski-Summe zweier Teilmengen A und B eines Vektorraums ist die Menge, deren Elemente Summen von je einem Element aus A und einem Element aus B sind.

Auch findet die Konvexität Anwendung bei der Charakterisierung von Funktionen:

Definition 4.6: Konvexe Funktionen

Sei $\mathcal{X} \in \mathbb{R}^n$ eine konvexe Menge. Eine Funktion $f : \mathcal{X} \to \mathbb{R}$ wird konvex auf $\mathcal{X}$ genannt, falls für alle $\{\boldsymbol{x}, \boldsymbol{y}\} \in \mathcal{X}$ und $\alpha \in \{\mathbb{R}|0 \leq \alpha \leq 1\}$ gilt:

$$f(\alpha\boldsymbol{x} + (1-\alpha)\boldsymbol{y}) \leq \alpha f(\boldsymbol{x}) + (1-\alpha)f(\boldsymbol{y}). \tag{4.10}$$

Die Funktion f wird darüber hinaus strikt konvex genannt, falls für $\alpha \in \{\mathbb{R}|0 < \alpha < 1\}$ und $\boldsymbol{x} \neq \boldsymbol{y}$ gilt:

$$f(\alpha\boldsymbol{x} + (1-\alpha)\boldsymbol{y}) < \alpha f(\boldsymbol{x}) + (1-\alpha)f(\boldsymbol{y}). \tag{4.11}$$

Man nennt die Funktion f demgegenüber (strikt) konkav, falls $-f$ (strikt) konvex ist.

Die geometrische Interpretation dieser Definition lautet wie folgt: Eine Funktion f ist genau dann konvex, wenn für alle $\{\boldsymbol{x}, \boldsymbol{y}\} \in \mathcal{X}$ und $\alpha \in \{\mathbb{R}|0 < \alpha < 1\}$ alle Funktionswerte von $f(\alpha\boldsymbol{x} + (1-\alpha)\boldsymbol{y})$ unterhalb der Verbindungslinie zwischen $f(\boldsymbol{x})$ und $f(\boldsymbol{y})$ liegen. Beispielhafte Funktionsverläufe für den eindimensionalen Fall sind in Abb. 4.3 dargestellt.

Auch lassen sich einige interessante Eigenschaften konvexer Funktionen im Kontext der Optimierung zusammenfassen:

Satz 4.2: Ausgewählte Eigenschaften konvexer Funktionen

Konvexe Funktionen besitzen folgende Eigenschaften:

(a) *Die Summe*

$$f(\boldsymbol{x}) = \sum_{i=1}^{n} \alpha_i f_i(\boldsymbol{x})$$

von auf der konvexen Menge $\mathcal{X}$ konvexen Funktionen $f_i(\boldsymbol{x})$ mit $\alpha \in \{\mathbb{R}|\alpha \geq 0\}$ ist auf $\mathcal{X}$ ebenfalls konvex.

(b) *Ist die Funktion $f(\boldsymbol{x})$ auf der konvexen Menge $\mathcal{X}$ konvex, so ist auch die Menge*

$$\{\boldsymbol{x} \in \mathcal{X} | f(\boldsymbol{x} \leq c)\}$$

für alle $c \in \mathbb{R}$ ebenfalls konvex.

(c) *Eine stetig differenzierbare Funktion f ist genau dann konvex auf der konvexen Menge $\mathcal{X}$, wenn für alle $\{\boldsymbol{x}, \boldsymbol{y}\} \in \mathcal{X}$ die Ungleichung*

$$f(\boldsymbol{y}) \geq f(\boldsymbol{x}) + (\boldsymbol{y} - \boldsymbol{x})^T (\nabla f)(\boldsymbol{x})$$

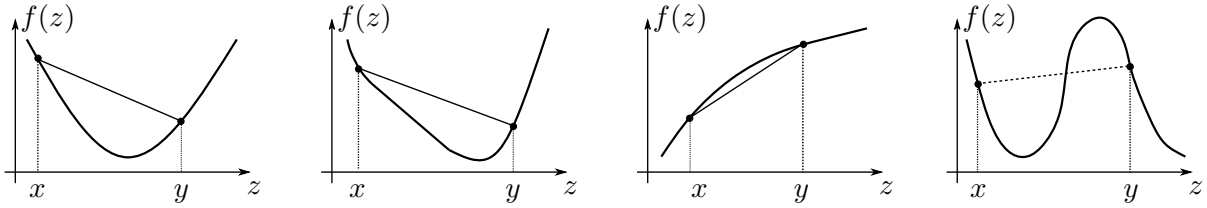
erfüllt ist.[1]

(d) *Eine zweifach stetig differenzierbare Funktion f ist genau dann konvex auf der konvexen Menge $\mathcal{X}$, wenn für alle $\boldsymbol{x} \in \mathcal{X}$ die Hesse-Matrix $(\nabla^2 f)(\boldsymbol{x})$ positiv semidefinit ist. Ist die Hesse-Matrix positiv definit, so ist f strikt konvex auf $\mathcal{X}$.*

[1] Die geometrische Interpretation ist, dass an jedem Punkt x einer konvexen Funktion $f(\boldsymbol{x})$ eine sogenannte stützende Hyperebene (skalarer Fall: stützende Tangente) existieren muss, oberhalb oder auf der $f(\boldsymbol{x})$ verläuft.

Obige Eigenschaften und Definitionen können gezielt herangezogen werden, um konvexe Kostenfunktion zu bilden[1]. Typische Beispiele sind:

- Die quadratische Funktion $f(x) = x^2$ ist strikt konvex auf $\mathbb{R}$.
- Die Exponentialfunktion $f(x) = e^x$ ist strikt konvex auf $\mathbb{R}$.
- Der negierte natürliche Logarithmus $f(x) = -\ln(x)$ ist strikt konvex auf $x \in \{\mathbb{R}|x > 0\}$.



(a) strikt konvex (b) konvex (c) strikt konkav (d) weder noch

Abb. 4.3: Beispiele von konvexen und konkaven Funktionen in $\mathbb{R}$

4.1.2 Berechnung von Ableitungen

Bereits in Kap. 3 wurde deutlich, dass der Berechnung von Ableitungen im Kontext der Optimierung eine wichtige Rolle zukommt. Zur Berechnung von Ableitungen stehen verschiedene Verfahren zur Verfügung:

Analytisches Differenzen
Ist eine gegebene Funktion $f(\boldsymbol{x})$ als analytischer Ausdruck verfügbar, so können die Ableitungen direkt mittels analytischer Differenzierung berechnet werden. Dies kann im Vorfeld der jeweiligen Optimierung durchgeführt werden, sodass der (numerische) Berechnungsaufwand des Optimierungsprogramms gegenüber den nachfolgend alternativen Methode i. A. deutlich geringer ausfällt.

Ableitung durch Differenzenquotienten
Die Ableitung mittels Differenzenquotienten[2] geht auf die Definition der Ableitung über den Differenzialquotienten zurück. Sei die skalare Funktion $f(x)$ eine stetig differenzierbare Funktion, dann gilt für die Ableitung an der Stelle x_0:

$$\frac{\partial f}{\partial x}(x_0) = \lim_{h \to 0} \frac{f(x_0 + h) - f(x_0)}{h}. \tag{4.12}$$

Hier ist h die Schrittweite. Die Ableitung bedeutet geometrisch die Steigung der Tangente im Punkt $f(x_0)$. Diese Tangentensteigung erhält man, indem man die Sekante durch die Funktionswerte an den Stellen x_0 und $x_0 + h$ aufstellt, die Sekantensteigung

$$\frac{f(x_0 + h) - f(x_0)}{(x_0 + h) - x_0} \tag{4.13}$$

[1] Bei Identifikationsaufgaben hat die zu identifizierende Modellklasse unmittelbar Auswirkungen auf die Eigenschaften der Kostenfunktion. Daher kann es in manchen Fällen sinnvoll sein, die Modellstruktur derart zu wählen, dass das resultierende Optimierungsproblem konvex ist.

[2] Genauer gesagt: Die Approximation der Funktionsableitung mittels Differenzenquotienten.

bestimmt und den Grenzübergang $h \to 0$ bildet. Da der Grenzübergang $h \to 0$ numerisch nicht durchgeführt werden kann, muss eine endliche Schrittweite h gewählt werden. Verschiedene Varianten des resultierenden Differenzenquotienten sind für eine hinreichend oft differenzierbare Funktion $f(\boldsymbol{x})$ mit $\boldsymbol{x} \in \mathbb{R}^n$ in Tab. 4.1 zusammengefasst:

Ableitung, Richtung	Formel	Abschneide-fehler	Rundungs-fehler
1. Ableitung, vorwärts	$(\nabla f)(\boldsymbol{x}) \approx \left[\frac{f(\boldsymbol{x}+h\boldsymbol{e}_i)-f(\boldsymbol{x})}{h}\right]_{i=1,\dots,n}$	$\mathcal{O}(h)$	$\mathcal{O}(e_r h^{-1})$
1. Ableitung, rückwärts	$(\nabla f)(\boldsymbol{x}) \approx \left[\frac{f(\boldsymbol{x})-f(\boldsymbol{x}-h\boldsymbol{e}_i)}{h}\right]_{i=1,\dots,n}$	$\mathcal{O}(h)$	$\mathcal{O}(e_r h^{-1})$
1. Ableitung, zentral	$(\nabla f)(\boldsymbol{x}) \approx \left[\frac{f(\boldsymbol{x}+h\boldsymbol{e}_i)-f(\boldsymbol{x}-h\boldsymbol{e}_i)}{2h}\right]_{i=1,\dots,n}$	$\mathcal{O}(h^2)$	$\mathcal{O}(e_r h^{-1})$
2. Ableitung, vorwärts	$(\nabla^2 f)(\boldsymbol{x}) \approx \frac{1}{h^2}\left[f(\boldsymbol{x}+h\boldsymbol{e}_i+h\boldsymbol{e}_j)-f(\boldsymbol{x}+h\boldsymbol{e}_i)-f(\boldsymbol{x}+h\boldsymbol{e}_j)+f(\boldsymbol{x})\right]_{\{i,j\}=1,\dots,n}$	$\mathcal{O}(h)$	$\mathcal{O}(e_r h^{-2})$
2. Ableitung, zentral	$(\nabla^2 f)(\boldsymbol{x}) \approx \frac{1}{4h^2}\left[f(\boldsymbol{x}+h\boldsymbol{e}_i+h\boldsymbol{e}_j)-f(\boldsymbol{x}+h\boldsymbol{e}_i-h\boldsymbol{e}_j)-f(\boldsymbol{x}-h\boldsymbol{e}_i+h\boldsymbol{e}_j)-f(\boldsymbol{x}+h\boldsymbol{e}_j)+f(\boldsymbol{x})\right]_{\{i,j\}=1,\dots,n}$	$\mathcal{O}(h^2)$	$\mathcal{O}(e_r h^{-2})$

Tab. 4.1: Zusammenstellung einiger Differenzenquotienten und ihrer Fehlerordnung (vgl. [Ste18])

Hierbei ist h die Schrittweite und $\boldsymbol{e}_i$ der Einheitsvektor mit lediglich einem Eintrag an der Stelle i. Die Tabelle enthält zudem die Ordnungen der Fehler, welche bei dieser näherungsweisen Ableitungsberechnung entstehen können. Das sind *Abschneidefehler* und *Rundungsfehler*, wobei e_r der maximale relative Fehler als Folge von Rundungsoperationen bei Gleitkommaarithmetik ist. Die Herleitung von Differenzenquotienten am Punkt $\boldsymbol{x}$ basiert auf der Taylor-Reihenentwicklung:

Definition 4.7: Mittelwersatz (Satz von Taylor)

Sei $f(\boldsymbol{x})$ eine stetig differenzierbare Funktion auf $\mathcal{X}$, welche das Segment $[\boldsymbol{x}_1, \boldsymbol{x}_2]$ beinhaltet. Dann existiert ein $\alpha \in \{\mathbb{R} | 0 \leq \alpha \leq 1\}$, sodass

$$f(\boldsymbol{x}_2) = f(\boldsymbol{x}_1) + (\boldsymbol{x}_2 - \boldsymbol{x}_1)^T (\nabla f)(\alpha \boldsymbol{x}_1 + (1-\alpha)\boldsymbol{x_2}) \tag{4.14}$$

gilt. Ist die Funktion $f(\boldsymbol{x})$ zudem zweifach stetig differenzierbar, dann existiert darüber hinaus ein α, sodass

$$\begin{aligned} f(\boldsymbol{x}_2) = f(\boldsymbol{x}_1) &+ (\boldsymbol{x}_2 - \boldsymbol{x}_1)^T (\nabla f)(\boldsymbol{x}_1) \\ &+ \frac{1}{2}(\boldsymbol{x}_2 - \boldsymbol{x}_1)^T (\nabla^2 f)(\alpha \boldsymbol{x}_1 + (1-\alpha)\boldsymbol{x_2})(\boldsymbol{x}_2 - \boldsymbol{x}_1) \end{aligned} \tag{4.15}$$

gilt.

Der Abschneidefehler resultiert, da die Taylor-Reihe bei einer bestimmten Ordnung abgebrochen wird. Dieser ist also der Methode der Differenzenquotienten geschuldet und nicht auf die limitierte arithmetische Berechnungsgenauigkeit des verwendeten Rechners zurückzuführen.

Die beiden Fehlerarten sollen nachfolgend am Beispiel der 1. Ableitung mittels Vorwärtsdifferenzenquotient verdeutlicht werden. Die Taylor-Reihenentwicklung entsprechend Definition 4.7 liefert für eine Richtung $\boldsymbol{e}_i$ und Schrittweite h:

$$f(\boldsymbol{x}+h\boldsymbol{e}_i) = f(\boldsymbol{x}) + h\frac{\partial f}{\partial x_i}\bigg|_{\boldsymbol{x}} + \frac{h^2}{2}\frac{\partial^2 f}{\partial x_i^2}\bigg|_{\boldsymbol{x}+\alpha h\boldsymbol{e}_i}. \tag{4.16}$$

Umstellen liefert dann:

$$\frac{\partial f}{\partial x_i}\bigg|_{\boldsymbol{x}} = \underbrace{\frac{f(\boldsymbol{x}+h\boldsymbol{e}_i)-f(\boldsymbol{x})}{h}}_{\text{Differenzenquotient}} \underbrace{-\frac{h}{2}\frac{\partial^2 f}{\partial x_i^2}\bigg|_{\boldsymbol{x}+\alpha h\boldsymbol{e}_i}}_{\text{Abschneidefehler, } \mathcal{O}(h)}. \tag{4.17}$$

Geht man ferner davon aus, dass e_r der maximale numerische Fehler der verarbeitenden Rechnerarchitektur ist (z. B. Rundungsfehler der Gleitkommaarithmetik), so ergibt sich im ungünstigsten Fall der berechnete Wert an den Stellen $f(\boldsymbol{x})$ und $f(\boldsymbol{x}+h\boldsymbol{e}_i)$ zu:

$$\frac{f(\boldsymbol{x}+h\boldsymbol{e}_i)(1+e_r)-f(\boldsymbol{x})(1-e_r)}{h} = \frac{f(\boldsymbol{x}+h\boldsymbol{e}_i)-f(\boldsymbol{x})}{h} \underbrace{+\frac{(f(\boldsymbol{x}+h\boldsymbol{e}_i)+f(\boldsymbol{x}))\,e_r}{h}}_{\text{Rundungsfehler, } \mathcal{O}(e_r h^{-1})}. \tag{4.18}$$

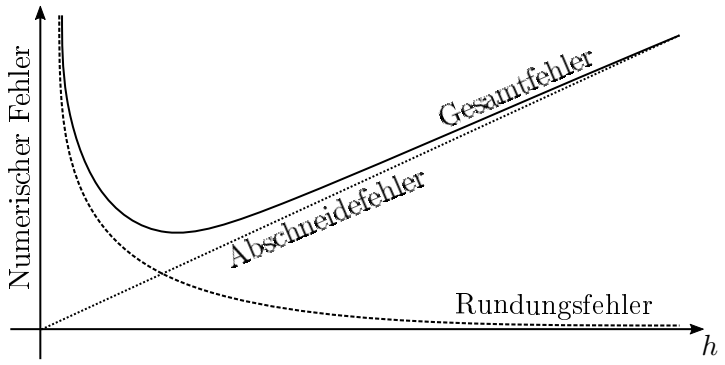


Abb. 4.4: Qualitative Fehlerentwicklung bei der 1. Ableitung mittels Vorwärtsdifferenzenquotient in Abhängigkeit der Schrittweite h

Zusammenfassend ist somit für dieses Beispiel festzustellen, dass der Rundfehler bei unendlich kleiner Schrittweite unendlich groß wird, während beim Abschneidefehler ein linearer Zusammenhang zur Schrittweite besteht. Folglich existiert für den resultierenden Gesamtfehler eine optimale Schrittweite, um diesen zu minimieren. Dies wird in Abb. 4.4 verdeutlicht.

Ferner ist der Rechenaufwand zur Ermittlung der Ableitungen mittels Differenzenquotient zu thematisieren. Entsprechend Tab. 4.1 muss die gegebene Funktion $f(x)$ an n verschiedenen Stellen ausgewertet werden:

- 1. Ableitung (vorwärts/rückwärts): $(n+1)$
- 1. Ableitung (zentral): $2n$

- 2. Ableitung (vorwärts/rückwärts): $n^2 + 1$
- 2. Ableitung (zentral): $(2n)^2$

Die erste Ableitung skaliert demnach linear $\mathcal{O}(n)$ und die zweite Ableitung quadratisch $\mathcal{O}(n^2)$ mit der Anzahl der Funktionsrichtungen n. Dies kann je nach Funktionsverlauf zu einem erheblichen numerischen Berechnungsaufwand führen, welcher i. A. deutlich größer ist verglichen mit einem einfachen Einsetzen in bereits analytisch vorliegende Ableitungsfunktionen.

Algorithmisches Differenzieren
Das algorithmische Differenzieren (oder auch automatisches Differenzieren genannt) kann genutzt werden, wenn eine zu betrachtende Funktion durch eine definierende Gleichung beschrieben werden kann. Zwar wäre es in diesem Fall theoretisch auch möglich, die Ableitungen explizit, d. h. in diesem Fall durch den Anwender, zu berechnen – dies kann bei stark nichtlinearen Funktionen mit vielen Veränderlichen allerdings auch einen sehr hohen Aufwand bedeuten, den der Anwender ggf. nicht bereit ist zu investieren. Demgegenüber stellt das algorithmische Differenzieren ein Werkzeug dar, welches vereinfacht zusammengefasst die bekannten Ableitungsregeln von Grundfunktionen (z. B. e^x $\sin(x)$) mit den Rechenregeln für zusammengesetzte Funktion (Summenregel, Produktregel, Kettenregel etc.) auf automatisierte Weise miteinander kombiniert. Auf dieser Basis sowohl möglichst effiziente als auch genaue Algorithmen zu entwickeln, ist Gegenstand eines eigenen Wissenschaftszweigs innerhalb der Mathematik und kann daher an dieser Stelle nicht eingehender behandelt werden. Interessierte Leserinnen und Leser seien daher auf die Literatur (z. B. [GW08][Prü18]) oder auch die Übersichtsseite [BH18] verwiesen.

Demgegenüber verbleibt die Nutzung des Differenzenquotienten als einzige Option zur Approximation der Ableitungen, genau dann, wenn die zu untersuchende Funktion nicht als analytische Gleichung formuliert oder als algorithmisches Programm modelliert werden kann. Dies ist z. B. der Fall, wenn das zu untersuchende Problem Ergebnis eines Experiments ist.

4.2 Statische Optimierung ohne Beschränkungen

In diesem Teilkapitel werden zunächst statische unbeschränkte Optimierungsprobleme des Typs

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} \quad J(\boldsymbol{x}) \tag{4.19}$$

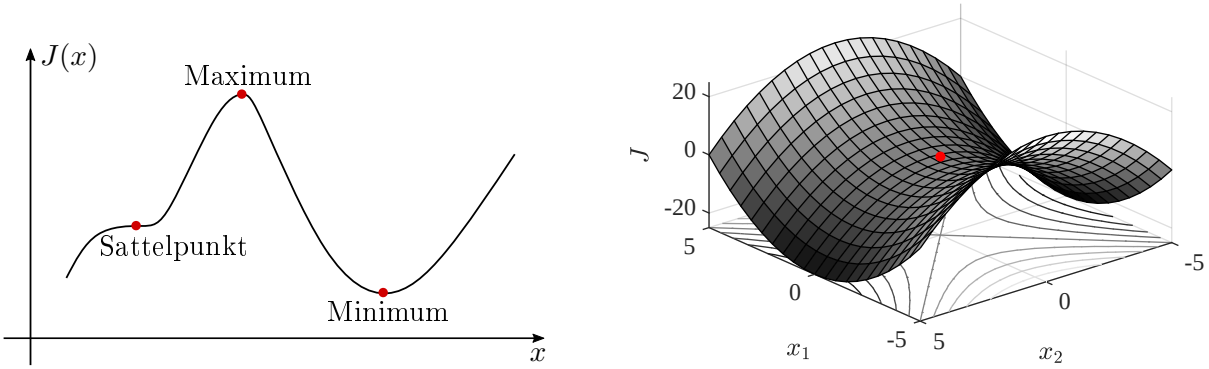
behandelt. Zur Auffindung der entsprechenden Lösung sei u. a. auf folgenden Satz verwiesen:

Satz 4.3: Notwendige Optimalitätsbedingung erster Ordnung

Sei $J : \mathbb{R}^n \to \mathbb{R}$ eine mindestens einmal stetig differenzierbare Funktion. Wenn $\boldsymbol{x}^*$ ein lokales Minimum von J auf $\mathbb{R}^n$ ist, dann gilt

$$\nabla J(\boldsymbol{x}^*) = 0. \tag{4.20}$$

Die Optimalitätsbedingung gemäß Satz 4.3 ist notwendig, aber nicht hinreichend. Die Bedingung gibt lediglich an, dass es sich bei dem betreffenden Punkt um einen *Extremalpunkt* (auch als stationärer Punkt bezeichnet) handelt, und wird von einem Minimum, Maximum oder Sattelpunkt gleichermaßen erfüllt, siehe hierzu auch Abb. 4.5.



(a) Verschiedene Beispiele in 1D (b) Sattelpunkt in 2D

Abb. 4.5: Beispiele von stationären Punkten

Zur weiteren Charakterisierung eines Extremalpunktes kann folgender Satz herangezogen werden:

Satz 4.4: Notwendige Optimalitätsbedingung zweiter Ordnung

Sei $J : \mathbb{R}^n \to \mathbb{R}$ eine mindestens zweifach stetig differenzierbare Funktion. Wenn $\boldsymbol{x}^*$ ein lokales Minimum von J auf $\mathbb{R}^n$ ist, dann gelten die Bedingungen:

(a) $\nabla J(\boldsymbol{x}^*) = 0$,

(b) $\nabla^2 J(\boldsymbol{x}^*)$ *ist positiv semi-definit.*

Auch Satz 4.4 beschreibt lediglich notwendige Optimalitätsbedingungen. Dies kann beispielhaft anhand der Funktion

$$J(x) = x^3$$

überprüft werden: Diese besitzt an $x^* = 0$ einen Extremalpunkt ($\nabla J(x^*) = 3(x^*)^2 = 0$) und, obwohl die zweite Ableitung ($\nabla^2 J(x^*) = 6x^* = 0$) positiv semi-definit ist, ist x^* kein Minimum. Die Funktion hat an der Stelle $x^* = 0$ lediglich einen Sattelpunkt. Satz 4.4 muss daher noch verschärft werden, um eine hinreichende Bedingung zu erhalten:

Satz 4.5: Hinreichende Optimalitätsbedingung zweiter Ordnung

Sei $J : \mathbb{R}^n \to \mathbb{R}$ eine mindestens zweifach stetig differenzierbare Funktion. Wenn die Bedingungen

(a) $\nabla J(\boldsymbol{x}^*) = 0$,

(b) $\nabla^2 J(\boldsymbol{x}^*)$ *ist positiv definit*

erfüllt sind, dann ist $\boldsymbol{x}^*$ ein striktes lokales Minimum von J auf $\mathbb{R}^n$.

Ist eine gegebene Funktion $J(\boldsymbol{x})$ konvex, dann ist die notwendige Optimalitätsbedingung erster Ordnung gemäß Satz 4.3 sogar hinreichend. Hierzu können Definition 4.6 sowie Satz 4.2 heran-

gezogen werden: Aufgrund der Konvexität von $J(\boldsymbol{x})$ mit dem Minimum $\boldsymbol{x}^*$ ist die Ungleichung

$$J(\boldsymbol{y}) \geq J(\boldsymbol{x}^*) + (\boldsymbol{y} - \boldsymbol{x}^*)^T \underbrace{(\nabla J)(\boldsymbol{x}^*)}_{=0} = J(\boldsymbol{x}^*) \tag{4.21}$$

für jedes beliebige $\boldsymbol{y} \in \mathbb{R}^n$ stets erfüllt. Während die vorherigen Sätze i. A. zunächst nur Aussagen zu lokalen Minima beinhalteten, können diese für konvexe Funktionen zudem auf globale Minima erweitert werden:

Satz 4.6: Globale Minima einer konvexen Funktion

Sei $J : \mathbb{R}^n \to \mathbb{R}$ eine konvexe Funktion, welche mindestens einmal stetig differenzierbar ist. Dann ist auch die Menge aller Minima $\mathcal{G} = \arg\min_{\boldsymbol{x}\in\mathbb{R}^n} J(\boldsymbol{x})$ konvex und die Bedingung

$$\nabla J(\boldsymbol{x}^*) = 0 \tag{4.22}$$

ist sowohl ausreichend als auch hinreichend, um zu zeigen, dass $\boldsymbol{x}^* \in \mathcal{G}$ sowohl lokales als auch globales Minimum ist. Ist $J(\boldsymbol{x})$ zudem strikt konvex, dann ist $\boldsymbol{x}^*$ ein striktes globales Minimum.

Da die Bestimmung eines (lokal) optimalen Punktes $\boldsymbol{x}^*$ durch analytische Lösung der Bedingung $\nabla J(\boldsymbol{x}^*) = 0$, welche n nichtlineare Gleichungen in $\boldsymbol{x}$ hervorbringt, nur in seltenen Fällen möglich ist, sind i. A. numerische Verfahren zur Suche von $\boldsymbol{x}^*$ heranzuziehen. Hierzu werden im folgenden Teilkapitel ausgewählte Algorithmen und deren Eigenschaften vorgestellt.

4.2.1 Numerische Optimierungsverfahren: Übersicht

Im Nachfolgenden werden Algorithmen diskutiert, welche versuchen mittels iterativer Rechenschritte eine Folge $\{\boldsymbol{x}[k]\}$ zu generieren, entlang der die zu optimierende Funktion $J(\boldsymbol{x})$ abnimmt, d. h.

$$J(\boldsymbol{x}[k+1]) \leq J(\boldsymbol{x}[k]), \quad k = 0, 1, 2, \ldots, \tag{4.23}$$

und die zumindest für $k \to \infty$ gegen $\boldsymbol{x}^*$ konvergieren sollen, also

$$\lim_{k\to\infty} \boldsymbol{x}[k] = \boldsymbol{x}^*. \tag{4.24}$$

Zur Untersuchung des Konvergenzverhaltens der verschiedenen Algorithmen wird die sog. *Fehlerfunktion* herangezogen:

Definition 4.8: Fehlerfunktion

Sei $e : \mathbb{R}^n \to \mathbb{R}$ eine Funktion in Abhängigkeit von $\boldsymbol{x}$. Erfüllt diese die Eigenschaften

$$e(\boldsymbol{x}^*) = 0 \quad \textit{und} \quad e(\boldsymbol{x}) \geq 0 \ \forall\, \boldsymbol{x} \in \mathbb{R}^n, \tag{4.25}$$

dann wird $e(\,\cdot\,)$ als Fehlerfunktion bezeichnet.

Als Fehlerfunktion kann z. B. der Abstand

$$e(\boldsymbol{x}) = ||\boldsymbol{x} - \boldsymbol{x}^*|| \geq 0 \tag{4.26}$$

im Sinne einer Norm (z. B. euklidische Norm) oder die Kostendifferenz

$$e(\boldsymbol{x}) = J(\boldsymbol{x}) - J(\boldsymbol{x}^*) \geq 0 \tag{4.27}$$

herangezogen werden. Das Konvergenzverhalten eines Algorithmus kann nun anhand der zu $\{\boldsymbol{x}[k]\}$ gehörenden Folge $\{e[k]\}$ mit $e[k] = e(\boldsymbol{x}[k])$ analysiert werden. Zunächst sollen dazu die Begriffe *Konvergenzordnung* und *Konvergenzrate* einer Folge von Skalaren definiert werden:

Definition 4.9: Konvergenzordnung und Konvergenzrate

Sei $\{e[k]\}$ eine Folge von Skalaren, die gegen den Grenzwert 0 konvergiert. Die Konvergenzordnung der Folge $\{e[k]\}$ ist das Supremum der nichtnegativen Zahlen p, für die gilt

$$0 \leq \lim_{k\to\infty} \frac{|\{e[k+1]\}|}{|\{e[k]\}|^p} = \mu < \infty \tag{4.28}$$

Als zugehörige Konvergenzrate bezeichnet man die Zahl $\mu \in \mathbb{R}$. Es werden folgende Fälle unterschieden:

- *Für $p = 1$ und $\mu \in (0, 1)$ liegt lineare Konvergenz vor.*
- *Für $p > 1$ und $\mu > 0$ oder $p = 1$ und $\mu = 0$ liegt superlineare Konvergenz vor.*
- *Für $p = 2$ und $\mu > 0$ spricht man von quadratischer Konvergenz.*
- *Für $p = 3$ und $\mu > 0$ spricht man von kubischer Konvergenz.*

Im Wesentlichen beschreiben die Konvergenzordnung und die Konvergenzrate das Verhalten einer Folge für $k \to \infty$. Größere Werte der Konvergenzordnung p bedeuten, dass die Folge schneller konvergiert, da die Folgenelemente $\{e[k]\}$ (zumindest für sehr große Werte von k) mit der p-ten Potenz abnehmen. Dies bedeutet, dass in jedem Iterationsschritt die Anzahl der genauen Dezimalstellen (oder die Anzahl der Stellen in einem beliebigen Stellenwertsystem) in etwa ver-p-facht wird, also beispielsweise bei quadratischer Konvergenz verdoppelt. Analoges gilt für kleinere Werte der Konvergenzrate μ.

Die Algorithmen zur Lösung des stationären unbeschränkten Optimierungsproblems (4.19) lassen sich grob in folgende Kategorien einteilen:

- **Ableitungsfreie Ansätze**, z. B.
  - Downhill-Simplex-Verfahren (auch als Nelder-Mead-Verfahren bekannt)
  - (Meta-)Heuristische (Zufalls-)Suche (siehe auch Kap. 4.4)
- **Verfahren mit 1. Ableitung**, z. B.
  - Gradientenverfahren
  - Quasi-Newton-Verfahren
- **Verfahren mit 1. und 2. Ableitung**, z. B.
  - Newton-Verfahren (auch als Newton-Raphson-Verfahren bekannt)

Da in Kap. 4.4 noch ableitungsfreie Methoden auf meta-heuristischer Basis zur Lösung von nicht-konvexen, nichtlinearen Optimierungsverfahren diskutiert werden, welche auch grundsätzlich auf (4.19) angewandt werden können, werden diese hier nicht weiter betrachtet. Das ebenfalls bekannte Nelder-Mead-Verfahren kann der weiteren Literatur (z. B. [PLB12]) entnommen werden. Stattdessen wird der nachfolgende Fokus auf die sog. *Liniensuchverfahren* gelegt, welche die 1. und ggf. 2. Ableitung der Kostenfunktion einbeziehen. Das grundsätzliche Vorgehen hierbei ist in Algorithmus 4.1 zusammengefasst:

Algorithmus 4.1 Liniensuchverfahren

Initialisierung:

1: $\boldsymbol{x}_0 = \boldsymbol{x}[k=0]$ ▷ Startlösung

2: $k = 0$ ▷ Startindex

Iterieren:

3: **while** $\boldsymbol{x}[k]$ ist nicht optimal **do**

4: Wähle geeignete Suchrichtung $\boldsymbol{s}[k]$

5: Wähle optimale Schrittweite $\alpha[k]$

6: $\boldsymbol{x}[k+1] \leftarrow \boldsymbol{x}[k] + \alpha[k]\boldsymbol{s}[k]$

7: $k \leftarrow k+1$

8: **end while**

Zum Iterationsschritt k ermittelt man vorerst eine geeignete *Suchrichtung* bzw. Abstiegsrichtung $\boldsymbol{s}[k] \in \mathbb{R}^n$. Sie soll derart gewählt werden, dass, wenn man sich hinreichend vom Punkt $\boldsymbol{x}[k]$ aus in diese Richtung bewegt, also

$$\boldsymbol{x}[k+1] = \boldsymbol{x}[k] + \alpha[k]\boldsymbol{s}[k] \tag{4.29}$$

mit einer geeigneten *Schrittweite* $\alpha[k] \in \{\mathbb{R}|\alpha > 0\}$, die Abstiegsbedingung, d. h.

$$J(\boldsymbol{x}[k+1]) = J(\boldsymbol{x}[k] + \alpha[k]\boldsymbol{s}[k]) < J(\boldsymbol{x}[k]), \tag{4.30}$$

erfüllt ist. Die Wahl der optimalen Schrittweite $\alpha[k]$ reduziert sich dann auf das skalare Problem

$$\min_{\alpha[k]>0} \quad g(\alpha[k]) = J(\boldsymbol{x}[k] + \alpha[k]\boldsymbol{s}[k])\Big|_{\boldsymbol{s}[k]} \tag{4.31}$$

bei gegebener Suchrichtung $\boldsymbol{s}[k]$. Die Iteration wird solange wiederholt, bis ein Abbruchkriterium erfüllt ist, z. B. bis eine gewählte Fehlerfunktion kleiner als ein vorgegebener Schwellwert ist.

4.2.2 Wahl der Schrittweite

Im Folgenden wird angenommen, dass die Suchrichtung $\boldsymbol{s}[k]$ bereits ermittelt wurde und ausschließlich das Problem (4.31) zu lösen sei. Hierfür werden nachfolgend verschiedene Ansätze gegenübergestellt.

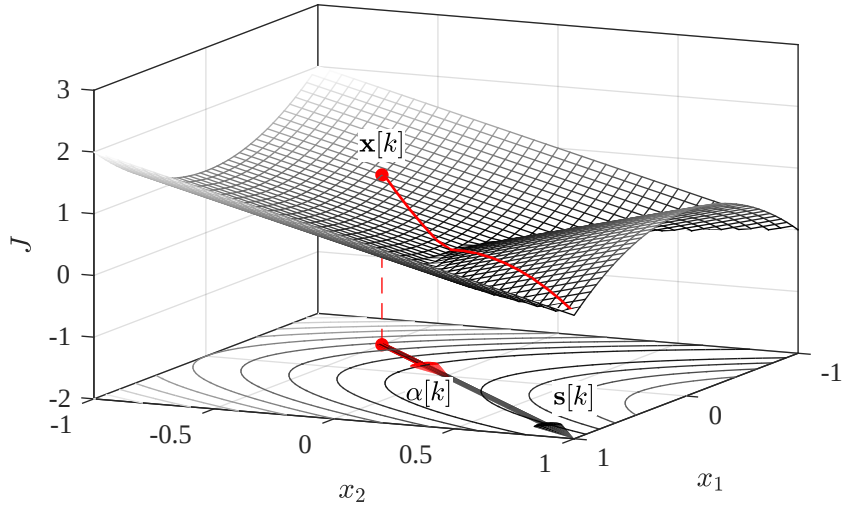


Abb. 4.6: Illustration der getrennten Wahl aus Suchrichtung $\boldsymbol{s}[k]$ und Schrittweite $\alpha[k]$ am Punkt $\boldsymbol{x}[k]$ gemäß Algorithmus 4.1

Intervallschachtelung (mittels des Goldenen Schnittes)

Das Verfahren auf Basis der Intervallschachtelung generiert für das skalare Optimierungsproblem (4.31) eine konvergierende Folge von Intervallen, um das Minimum von $g(\alpha[k])$ einzugrenzen. Hierzu muss zunächst ein Startintervall $[l[j=0], r[j=0]]$ gefunden werden, in dem die Funktion $g(\alpha[k])$ ein Minimum aufweist, siehe auch Abb. 4.6. Die triviale Lösung hierzu besteht darin, einen sehr kleinen Wert für $l[j=0] \approx 0$ anzusetzen und davon ausgehend mit vergleichsweise großen Inkrementen $r[j=0]$ zu erhöhen, bis die Funktion $g(\alpha[k])$ steigende Werte aufweist. Im Folgenden wird hierfür angenommen, dass $g(\alpha[k])$ stetig und *unimodal* auf $[l[j=0], r[j=0]]$ ist, d. h. die Funktion $g(\alpha[k])$ hat auf $[l[j=0], r[j=0]]$ exakt ein Minimum.

Zum Iterationsschritt j liege das Intervall $[l[j], r[j]]$ vor, das den gesuchten Wert $\alpha^*[k]$ beinhaltet. Nun werden zwei neue (Zwischen-)Punkte $[\hat{l}[j], \hat{r}[j]]$ durch

$$l[j] < \hat{l}[j] < \hat{r}[j] < r[j]\,, \tag{4.32a}$$

$$\hat{l}[j] = l[j] + (1-\beta)(r[j] - l[j])\,, \tag{4.32b}$$

$$\hat{r}[j] = l[j] + \beta(r[j] - l[j])\,, \tag{4.32c}$$

$$\beta \in \left\{ \mathbb{R} \middle| \frac{1}{2} < \beta < 1 \right\} \tag{4.32d}$$

ermittelt. Danach wird die Funktion $g(\,\cdot\,)$ an den Stellen $g(\hat{l}[j])$ und $g(\hat{r}[j])$ ausgewertet und der folgende Iterationsschritt durchgeführt:

$$\text{falls } g(\hat{l}[j]) \leq g(\hat{r}[j]): \quad [l[j] = l[j+1], r[j+1] = \hat{r}[j]], \tag{4.33a}$$

$$\text{falls } g(\hat{l}[j]) > g(\hat{r}[j]): \quad [l[j+1] = \hat{l}[j], r[j+1] = r[j]]. \tag{4.33b}$$

Für die weitere Betrachtung werde angenommen, dass $g(\hat{l}[j]) \leq g(\hat{r}[j])$ gelte, wie es auch in Abb. 4.7 illustriert ist. Die nachfolgenden Schritte lassen sich unmittelbar auf den anderen Fall übertragen und führen zum gleichen Ergebnis, sodass folgende einseitige Betrachtung ausreichend ist. Zunächst werde eine weitere Auswertung gemäß (4.32) zur Berechnung der Zwischen-

punkte durchgeführt:

$$\hat{l}[j+1] = \beta l[j+1] + (1-\beta) r[j+1] = (1-\beta+\beta^2) l[j] + \beta(1-\beta) r[j], \tag{4.34a}$$

$$\hat{r}[j+1] = (1-\beta) l[j+1] + \beta r[j+1] = (1-\beta^2) l[j] + \beta^2 r[j]. \tag{4.34b}$$

Durch Koeffizientenvergleich von (4.32b) und (4.34b) folgt

$$\hat{r}[j+1] = \hat{l}[j] \quad \Leftrightarrow \quad \beta^2 = 1-\beta, \tag{4.35}$$

demnach also genau dann, wenn

$$\beta = \frac{\sqrt{5}-1}{2} \approx 0,618 \tag{4.36}$$

beträgt. Der Quotient $1/\beta \approx 1,618$ ist bekannt als die Verhältniszahl des *Goldenen Schnittes*. Diese Wahl hat den Vorteil, dass für jede Iteration nur ein neuer Zwischenpunkt berechnet werden muss, was die Anzahl der notwendigen Funktionsauswertungen $g(\alpha[k])$ reduziert.

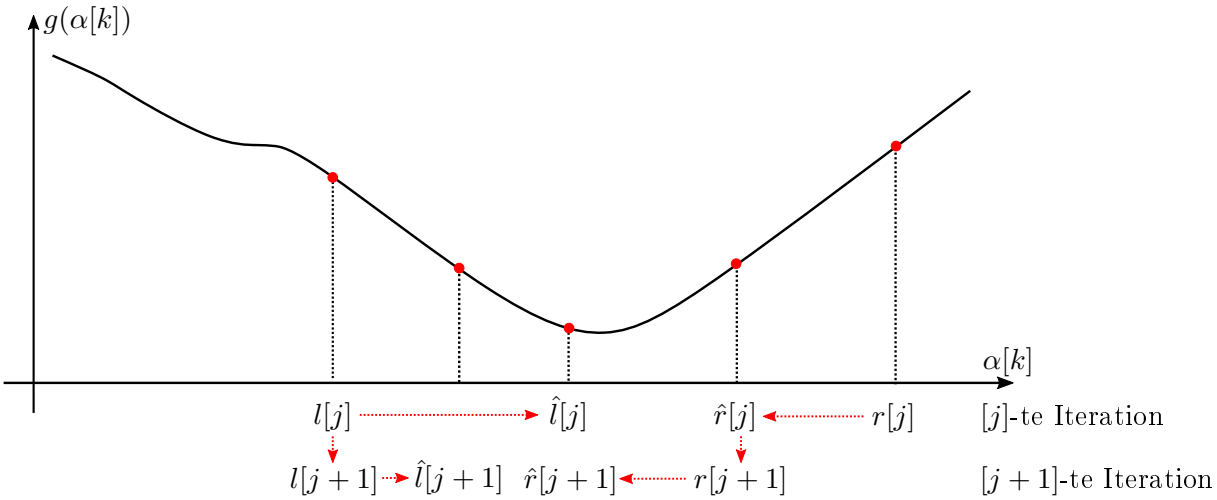


Abb. 4.7: Qualitative Darstellung der Intervallschachtelung mittels des Goldenen Schnittes

Abschließend muss der Algorithmus terminiert und der optimale Wert $\alpha[k]^*$ bestimmt werden. Hierzu bietet sich z. B. die Mittelung der letzten Intervallgrenzen an, sofern die Veränderung der Intervallgrenzen bzw. der Kostenfunktion einen Schwellwert unterschreitet

$$\alpha^*[k] = \frac{l[j]+r[j]}{2} \quad \text{falls} \begin{cases} \left\| \begin{bmatrix} l[j] & r[j] \end{bmatrix} - \begin{bmatrix} l[j-1] & r[j-1] \end{bmatrix} \right\|^2 < \varepsilon \\ \frac{g(l[j])-g(l[j-1])}{g(l[j-1])} < \tilde{\varepsilon} \quad \wedge \quad \frac{g(r[j])-g(r[j-1])}{g(r[j-1])} < \tilde{\varepsilon} \end{cases} \tag{4.37}$$

oder eine bestimmte Anzahl an Iterationen erreicht wurde.

Das Intervallschachtelungsverfahren gilt i. A. als einfaches und robustes Verfahren, das allerdings im Vergleich zu anderen Verfahren meist mehr Iterationen benötigt. Trifft zudem die Annahme einer unimodalen Kostenfunktion nicht zu, d. h., für ein gegebenes $\boldsymbol{s}[k]$ liegen mehrere lokale bzw. globale Optima für $g(\alpha[k])$ vor (also kein konvexes, sondern ein echt nichtlineares Optimierungsproblem), erhöht sich der Schwierigkeitsgrad, um $\alpha^*[k]$ in zielführender Rechenzeit zu ermitteln. Hierzu sei auch auf Kap. 4.4 verwiesen.

Eingrenzung durch Abstiegs- und Krümmungsbedingungen

Bei der Intervallschachtelung wurde u. a. das Problem eines geeigneten Startintervalls aufgezeigt. Ein zu groß gewähltes Intervall führt zu einem gesteigerten Berechnungsaufwand mit vielen Iterationen bzw. Funktionsauswertungen, während ein zu kleines Intervall die Gefahr birgt, dass das gesuchte $\alpha^*[k]$ nicht mehr Teil von $[l[j=0], r[j=0]]$ ist. Die nachfolgend behandelten Abstiegs- und Krümmungsbedingungen sollen helfen, dass Startintervall zielführend zu wählen.

Eine erste Abschätzung kann anhand der sog. *Abstiegsbedingung* (auch als *Armijo-Bedingung* bekannt) vorgenommen werden:

$$\underbrace{J(\boldsymbol{x}[k]+\alpha[k]\boldsymbol{s}[k])}_{g(\alpha[k])} \leq \underbrace{J(\boldsymbol{x}[k])}_{g(\alpha=0)} + c_1\alpha[k]\underbrace{(\nabla J)(\boldsymbol{x}[k])\boldsymbol{s}[k]}_{\partial g/\partial\alpha|_{\alpha=0}}. \tag{4.38}$$

Hier ist $c_1 \in \{\mathbb{R}|0 < c_1 < 1\}$ eine zu wählenden Konstante. Anschaulich interpretiert bedeutet die Ungleichung, dass der Abstieg in J proportional zur Schrittweite $\alpha[k]$ und der Richtungsableitung $\partial g/\partial\alpha|_{\alpha=0}$ sein sollte.

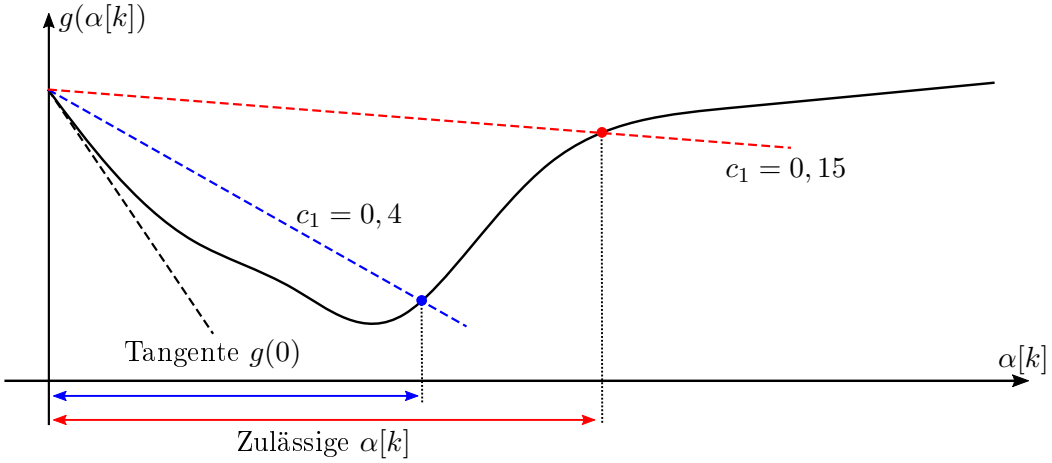


Abb. 4.8: Illustration zur Abstiegsbedingung (Armijo-Bedingung)

Die Abb. 4.8 verdeutlicht diesen Zusammenhang. Die rechte Seite der Ungleichung (4.38) stellt eine Gerade mit der negativen Steigung $c_1\partial g/\partial\alpha|_{\alpha=0}$ dar, aber aufgrund der Skalierung mit $c_1 \in (0,1)$ liegt die Gerade für hinreichend kleine Werte von $\alpha[k]$ über dem Funktionswert $g(\alpha[k])$. Die Abstiegsbedingung verlangt folglich, dass Werte von $\alpha[k]$ nur zulässig sind, wenn ihr Funktionswert $g(\alpha[k])$ unterhalb dieser Geraden liegt. In der Praxis wird c_1 häufig sehr klein gewählt (z. B. $10^{-4}\ldots10^{-3}$), damit die Abstiegsbedingung nicht zu restriktiv ist.

Die Abstiegsbedingung begrenzt das Intervall allerdings nur nach oben hin, da die Ungleichung für hinreichend kleine $\alpha[k]$ stets erfüllt ist. Es erscheint also zielführend auch eine Begrenzung des Intervalls nach unten hin vorzunehmen, um einen nennenswerten Abstieg in der Kostenfunktion zu gewährleisten und somit die Anzahl der notwendigen Iterationen zu reduzieren. Hierzu kann die sog. *Krümmungsbedingung* herangezogen werden:

$$|\underbrace{\nabla J(\boldsymbol{x}[k]+\alpha[k]\boldsymbol{s}[k])\boldsymbol{s}[k]}_{\partial g/\partial\alpha|_{\alpha[k]}}| \leq c_2|\underbrace{\nabla J(\boldsymbol{x}[k])\boldsymbol{s}[k]}_{\partial g/\partial\alpha|_{\alpha=0}}|\,. \tag{4.39}$$

Hier ist $c_2 \in \{\mathbb{R} | c_1 < c_2 < 1\}$ eine weitere Konstante, welche allerdings in Abhängig von c_1 gewählt wird. Die linke Seite von (4.39) stellt den Gradienten $\partial g(\alpha[k])/\partial\alpha$ dar, d. h. das Kriterium bedeutet demnach, dass der Betrag der Steigung $\partial g(\alpha[k])/\partial\alpha$ an der Stelle $\alpha[k]$ kleiner oder gleich dem c_2-fachen Betrag der Anfangssteigung $\partial g(0)/\partial\alpha$ sein muss. Im Falle einer stark negativen Steigung $\partial g(\alpha[k])/\partial\alpha$ ist davon auszugehen, dass g (und damit J) weiter reduziert werden kann, wenn ein Abstieg entlang der $\boldsymbol{s}[k]$-Achse erfolgt. Andererseits wird durch die Betragsbedingung vermieden, dass die Steigung $\partial g(\alpha[k])/\partial\alpha$ zu groß wird und die derart gefundenen Intervallgrenzen zu weit vom (lokalen) Minimum entfernt sind.

Die Abstiegs- und Krümmungsbedingungen sind auch als *Wolfe-Bedingungen* bekannt und können in den meisten Liniensuchverfahren eingesetzt werden. Die prinzipielle Vorgehensweise ist in Abb. 4.9 dargestellt. Über die Wolfe-Bedingungen kann zum einen ein geeignetes Startintervall für eine daran angeschlossene Intervallschachtelung gefunden werden. Zum anderen kann eine Folge $\{c_1[j], c_2[j]\}$ iterativ derart angepasst werden, um hierüber direkt das Optimum $\alpha^*[k]$ aufzufinden.

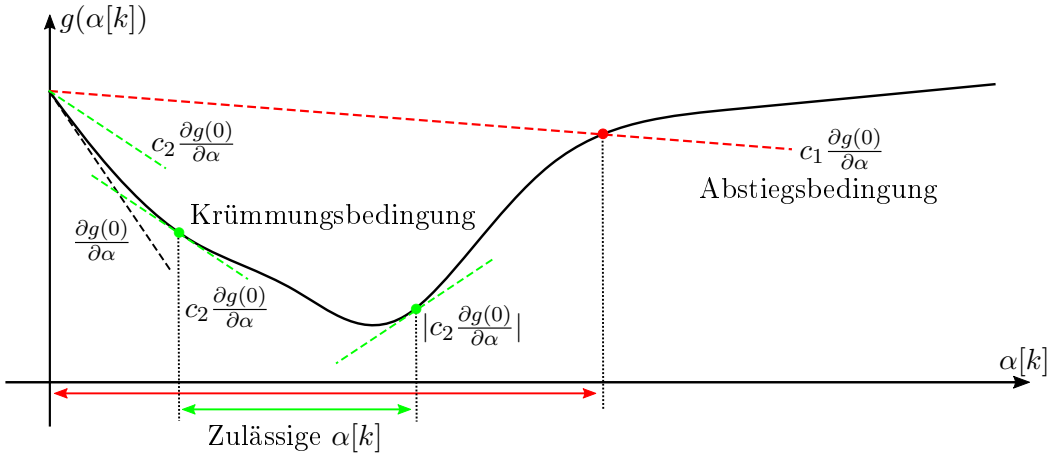


Abb. 4.9: Illustration zu den Wolfe-Bedingungen

Quadratische Interpolation

Bei der quadratischen Interpolation wird die Annahme zu Grunde gelegt, dass $g(\alpha[k])$ durch eine Hyperbel, also ein Polynom 2. Ordnung, hinreichend genau beschrieben werden kann. Die optimale Schrittweite soll daher auf einer polynomialen Interpolationsfunktion $h(\alpha[k])$ analytisch bestimmt werden. Hierzu muss zunächst die Interpolationsfunktion ermittelt werden, welche durch drei Wertepaare

$$(\tilde{\alpha}[i], g(\tilde{\alpha}[i]))$$

mit paarweise verschiedenen Stützstellen $\tilde{\alpha}[i]$ eindeutig definiert ist. Zur Lösung des Interpolationsproblems sei auf folgenden Satz verwiesen:

Satz 4.7: Lagrangesche Interpolationsformel

Gesucht sei das Polynom $h(x)$ vom Grad n, welches exakt durch die $n+1$ Wertepaare $(x[i], g(x[i]))$ mit paarweise verschiedenen Stützstellen $x[i]$ zur Funktion $g(x)$ verläuft. Diese Aufgabe ist unter Verwendung der Lagrangeschen Interpolation eindeutig lösbar und führt zu

$$h(x) = \sum_{i=0}^{n} g(x[i]) L_i(x) \tag{4.40}$$

mit dem Lagrange-Polynom

$$L_i(x) = \prod_{\substack{j=0 \\ j \neq i}}^{n} \frac{x - x[j]}{x[i] - x[j]} = \frac{x - x[0]}{x[i] - x[0]} \cdots \frac{x - x[i-1]}{x[i] - x[i-1]} \cdot \frac{x - x[i+1]}{x[i] - x[i+1]} \cdots \frac{x - x[n]}{x[i] - x[n]}.$$

Die Wahl der quadratischen Interpolation hat den Vorteil, dass auf Basis der Wertepaare $(\tilde{\alpha}[i], g(\tilde{\alpha}[i]))$ sofort eine geschlossene Lösung für die optimale Schrittweite $\alpha^*[k]$ angegeben werden kann, da für die Interpolationsfunktion ein eindeutiges Minimum besteht. Durch Extremalpunktbestimmung resultiert dann die Lösung:

$$\alpha^*[k] = \frac{1}{2} \frac{g(\tilde{\alpha}[1])(\tilde{\alpha}^2[2] - \tilde{\alpha}^2[3]) + g(\tilde{\alpha}[2])(\tilde{\alpha}^2[3] - \tilde{\alpha}^2[1]) + g(\tilde{\alpha}[3])(\tilde{\alpha}^2[1] - \tilde{\alpha}^2[2])}{g(\tilde{\alpha}[1])(\tilde{\alpha}[2] - \tilde{\alpha}[3]) + g(\tilde{\alpha}[2])(\tilde{\alpha}[3] - \tilde{\alpha}[1]) + g(\tilde{\alpha}[3])(\tilde{\alpha}[1] - \tilde{\alpha}[2])}. \tag{4.41}$$

Das derart gefundene $\alpha^*[k]$ stellt zwar für das interpolierte Problem das Minimum dar, allerdings kann $g(x)$ maßgeblich von $h(x)$ abweichen (insbesondere bei nicht konvexen Optimierungsproblemen), sodass $\alpha^*[k]$ auf $g(x)$ möglicherweise nur eine suboptimale Lösung darstellt. Diese Problematik wird in Abb. 4.10 beispielhaft illustriert.

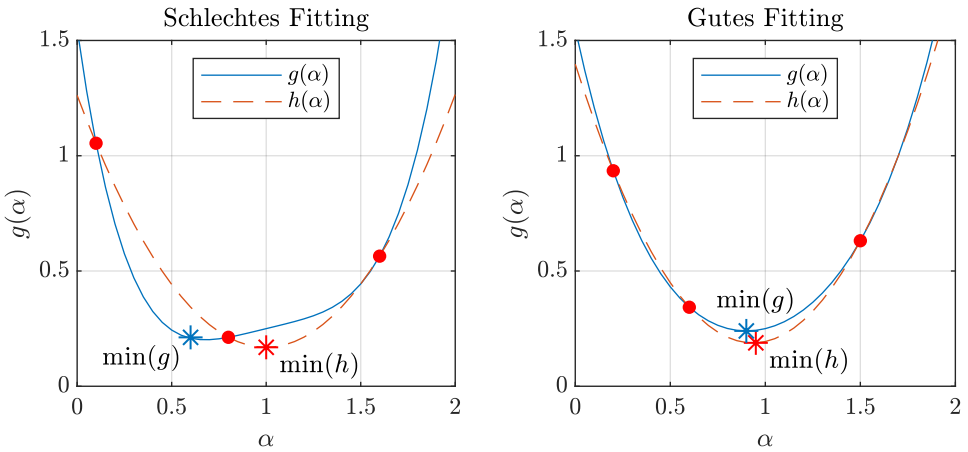


Abb. 4.10: Beispielhafte Darstellung zweier Fitting-Ergebnisse mittels quadratischer Interpolation hinsichtlich $\alpha^*[k]$ für unterschiedliche Ausgangsfunktionen $g(\alpha[k])$

Obwohl der Grad der Abweichung zwischen der optimalen Schrittweite auf dem ursprünglichen und dem interpolierten Problem im Wesentlichen von $g(\alpha)$ sowie den gewählten Stützstellen abhängt, können doch einige grundsätzliche Forderung an die Lösung $\alpha^*[k]$ gestellt werden:

(a) $\alpha^*[k] > 0$,

(b) $g(\alpha^*[k]) \leq g(\tilde{\alpha}[1]),\ g(\alpha^*[k]) \leq g(\tilde{\alpha}[2]),\ g(\alpha^*[k]) \leq g(\tilde{\alpha}[3])$,

(c) $g(\alpha^*[k]) \leq g(0)$.

Liegt ein quadratisches Optimierungsproblem vor, so kann mit der quadratischen Interpolation die tatsächlich optimale Schrittweite unmittelbar ermittelt werden, da die Interpolation $h(x)$ die Ausgangsfunktion $g(x)$ exakt nachbildet.

4.2.3 Wahl der Suchrichtung

Nachfolgend werden diverse Verfahren zur Wahl der Suchrichtung $\boldsymbol{s}[k]$ vorgestellt.

Methode des steilsten Abstiegs

Bei der Methode des steilsten Abstiegs (*steepest descent method*) entspricht die Suchrichtung $\boldsymbol{s}[k]$ dem negativen Gradienten an der Stelle $x[k]$, also

$$\boldsymbol{s}[k] = -(\nabla J)(\boldsymbol{x}[k]). \tag{4.42}$$

Für diese Wahl muss demnach lediglich die erste Ableitung bekannt sein, was zu einem vergleichsweise geringen Rechenaufwand pro Iterationsschritt führt. Allerdings stellt sich die Frage, wie viele Iterationen mit der Methode des steilsten Abstiegs benötigt werden. Hierzu sei auf folgenden Satz verwiesen:

Satz 4.8: Konvergenzrate der Methode des steilsten Abstiegs

Gegeben sei eine mindestens zweimal differenzierbare Funktion $\boldsymbol{J} : \mathbb{R}^n \to \mathbb{R}$ *mit dem lokalen Minimum* $\boldsymbol{x}^*$. *Ferner sei die Hesse-Matrix* $(\nabla^2 J)(\boldsymbol{x}^*)$ *positiv definit und* $\{\min_i \lambda_i, \max_i \lambda_i\}$ *seien dessen kleinster und größter Eigenwert. Dann ergibt sich die spektrale Konditionszahl zu* $\kappa = (\max_i \lambda_i / \min_i \lambda_i)$. *Wenn die Folge* $\{\boldsymbol{x}[k]\}$ *generiert durch die Methode des steilsten Abstiegs*

$$\boldsymbol{x}[k+1] = \boldsymbol{x}[k] - \alpha[k](\nabla J)(\boldsymbol{x}[k]) \tag{4.43}$$

gegen $\boldsymbol{x}^*$ *konvergiert, dann konvergiert die Folge* $\{J(\boldsymbol{x}[k])\}$ *linear gegen* $J(\boldsymbol{x}^*)$ *mit einer Konvergenzrate*[1] *von maximal*

$$\mu = \left(\frac{\kappa - 1}{\kappa + 1}\right)^2. \tag{4.44}$$

Es sei zudem noch auf den Spezialfall einer quadratischen Kostenfunktion der Form

$$\min_{\boldsymbol{x} \in \mathbb{R}^n} \quad J(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^T \boldsymbol{W} \boldsymbol{x} + \boldsymbol{w}^T \boldsymbol{x} + c \tag{4.45}$$

mit positiv definiter Gewichtungsmatrix $\boldsymbol{W} \in \mathbb{R}^{n \times n}$, einem Vektor $\boldsymbol{w} \in \mathbb{R}^n$ und der Konstanten $\boldsymbol{c} \in \mathbb{R}$ hingewiesen. In diesem Fall lautet die Hesse-Matrix gerade

$$(\nabla^2 J)(\boldsymbol{x}) = \boldsymbol{W}. \tag{4.46}$$

[1] Es sei an Definition 4.9 erinnert: Ein kleiner Wert für die Konvergenzrate μ bedeutet ein schnelles Auffinden der Lösung mit wenigen Iterationen. Für $\kappa = 1$ kann der stationäre Punkt sogar in einem Iterationsschritt gefunden werden.

Somit kann über die Gewichtung $\boldsymbol{W}$ die Kondition der Hesse-Matrix sowie die Konvergenzrate des Algorithmus direkt beeinflusst werden. Der prinzipielle Einfluss der Konditionierung des Optimierungsproblems auf den Iterationsverlauf für die Methode des steilsten Abstiegs wird in Abb. 4.11 verdeutlicht. Hier wird zudem klar, dass die Suchrichtung stets orthogonal zur Tangente der Höhenlinie der Kostenfunktion am jeweiligen Iterationspunkt verläuft.

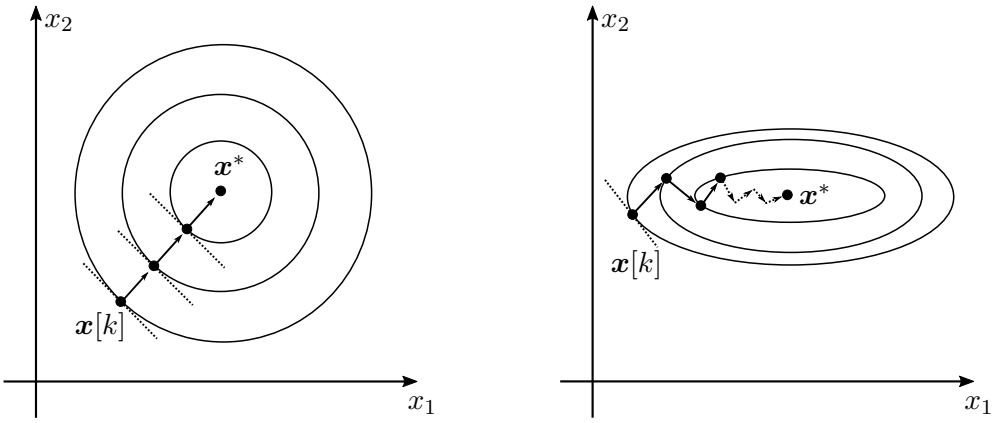


Abb. 4.11: Beispielhafte Darstellung für die Methode des steilsten Abstiegs bei ideal konditioniertem (links) und schlecht konditioniertem (rechts) Optimierungsproblem

Die Vor- und Nachteile der Methode des steilsten Abstiegs lassen sich wie folgt zusammenfassen:

Vorteile	**Nachteile**
• Einfaches Verfahren mit intuitiver Suchrichtung • Hesse-Matrix nicht erforderlich (geringer Berechnungsaufwand)	• Langsame Konvergenz bei schlecht konditionierten Problemen • Lediglich lineare Konvergenzordnung möglich

Tab. 4.2: Vor- und Nachteile der Methode des steilsten Abstiegs

Stochastisches Gradientenverfahren

Das stochastische Gradientenverfahren (*stochastic gradient descent* - SGD) ist eine Variante des Gradientenabstiegs und findet häufig Anwendung, wenn der Gradient mittels finiter Differenz auf einer großen Datenmenge berechnet werden muss. Liegen $i = 1, \ldots, N$ Datenpunkte vor, beispielsweise die Residuen zwischen einem zu parametrierenden Modell und empirischen Messungen, muss zur Berechnung eines Kostenfunktionswertes $J(\boldsymbol{x}[k])$ das Modell N-fach ausgeführt werden, beispielsweise[1]:

$$J(\boldsymbol{x}[k]) = \sum_{i=1}^{N} J_i(\boldsymbol{x}[k]). \tag{4.47}$$

[1] In diesem Beispiel wird der Einfachheit angenommen, dass sich die Kostenfunktion additiv über die N Datenpunkte zusammensetzten lässt. Diese Eigenschaft gilt zwar für viele Kostenfunktionen, aber nicht zwangsläufig für alle.

Für dieses Beispiel ergibt sich die reguläre Gradienabstiegsrichtung während der k-ten Iteration zu

$$\boldsymbol{s}[k] = -\sum_{i=1}^{N} \nabla J_i(\boldsymbol{x}[k]). \tag{4.48}$$

Muss dieser Gradient mittels finiter Differenzen ermittelt werden, potenzieren sich die Anzahl der Modellauswertungen entsprechend mit der Anzahl der Optimierungsparameter, siehe Kap. 4.1.2. Dies kann für Probleme mit einer großen Anzahl an Datenpunkten einen enormen numerischen Aufwand bedeuten. Demgegenüber werden im SGD-Ansatz nicht alle $i = 1, \ldots, N$ Datenpunkte zur Berechnung des Gradienten genutzt, sondern (im klassischen Fall) lediglich ein zufälliger Datenpunkt aus dieser Menge:

$$\boldsymbol{s}[k] = -\nabla J_i(\boldsymbol{x}[k]). \tag{4.49}$$

Der Gradient bzw. die Suchrichtung wird somit an der Stelle $\boldsymbol{x}[k]$ lediglich approximiert. Über k Iterationsschritte des SGD-Verfahrens werden die stochastischen Gradienten dann wiederum indirekt zusammengeführt und somit der tatsächliche Gradient rekonstruiert. Hierdurch sinkt i. A. zwar die Konvergenzrate des SGD-Ansatzes gegenüber dem regulären Gradientenabstieg, allerdings sind pro Optimierungsschritt deutlich weniger Modellauswertungen notwendig, sodass für einige Probleme der kumulierte numerische Aufwand verringert wird.

Eine Variante des SGD-Verfahrens ist der sog. *Mini-Batch-SGD* Ansatz, bei dem nicht nur ein Datenpunkt, sondern eine zufällige Submenge von j-Datenpunkten herangezogen wird:

$$\boldsymbol{s}[k] = -\sum_{j} \nabla J_j(\boldsymbol{x}[k]). \tag{4.50}$$

Hierdurch soll der Einfluss einzelner, nicht repräsentativer (verrauschter) Datenpunkte gemindert werden, um die Robustheit des Algorithmus zu erhöhen – also ein Mittelweg aus SGD und regulärem Gradientenabstieg. Auf Basis des an dieser Stelle nur kurz vorgestellten SGD-Ansatzes sind in der Literatur eine Reihe weiterer Verfahren für die Verarbeitung großes Datenmenge bekannt, welche häufig im Bereich des maschinellen Lernens angewandt werden. Bekannter Vertreter sind u. a. AdaGrad (*adaptive gradient algorithm*) und Adam (*adaptive moment estimation*). Für weitere Informationen zu speziellen Optimierern für das maschinelle Lernen sei auf die weiterführende Literatur verwiesen (z. B. [Agg18]).

Newton-Verfahren

Die Idee der Newton-Methode besteht darin, die Kostenfunktion $J(\boldsymbol{x})$ lokal durch eine quadratische Funktion zu approximieren und diese zu minimieren. Hierzu wird um den aktuellen Iterationspunkt $\boldsymbol{x}[k]$ eine Taylorreihe in Richtung $\boldsymbol{s}[k]$ entwickelt, welche nach dem quadratischen Term abgebrochen wird:

$$J(\boldsymbol{x}[k] + \boldsymbol{s}[k]) \approx J(\boldsymbol{x}[k]) + (\nabla J)(\boldsymbol{x}[k])\boldsymbol{s}[k] + \frac{1}{2}\boldsymbol{s}^T[k](\nabla^2 J)(\boldsymbol{x}[k])\boldsymbol{s}[k]. \tag{4.51}$$

Die sogenannte *Newton-Richtung* $\boldsymbol{s}[k]$ ergibt sich unmittelbar durch Minimierung von (4.51). Ableiten nach $\boldsymbol{s}[k]$ und Null setzen führt zu (siehe hierzu auch Kap. A.2):

$$\frac{\partial(J(\boldsymbol{x}[k] + \boldsymbol{s}[k]))}{\partial \boldsymbol{s}[k]} \approx (\nabla J)(\boldsymbol{x}[k]) + \boldsymbol{s}[k]^T(\nabla^2 J)(\boldsymbol{x}[k]) = 0. \tag{4.52}$$

Auflösen nach $\boldsymbol{s}[k]$ ergibt dann:

$$\boldsymbol{s}[k] = -[(\nabla^2 J)(\boldsymbol{x}[k])]^{-1}(\nabla J)(\boldsymbol{x}[k]). \tag{4.53}$$

Falls die Hesse-Matrix $(\nabla^2 J)(\boldsymbol{x}[k])$ am Minimum positiv definit ist, existiert in einer Umgebung um das Minimum die Inverse $[(\nabla^2 J)(\boldsymbol{x}[k])]^{-1}$ und die Methode ist wohldefiniert. Zudem sei darauf hingewiesen, dass die Inverse der Hesse-Matrix in (4.53) nicht explizit berechnet werden muss - gemäß Kap. 3.1.4 stehen zum Lösen des zugrundeliegenden Gleichungssystems weitere numerische Werkzeuge zur Verfügung.

Die hieraus resultierende Iterationsvorschrift lautet dann:

$$\boldsymbol{x}[k+1] = \boldsymbol{x}[k] - \alpha[k][(\nabla^2 J)(\boldsymbol{x}[k])]^{-1}(\nabla J)(\boldsymbol{x}[k]). \tag{4.54}$$

Es ist zu erwarten, dass in der Nähe des Minimums die optimale Schrittweite $\alpha[k] \approx 1$ entspricht, da die Kostenfunktion in hinreichender Nähe des lokalen Minimums konvexe Eigenschaften aufweist und somit die quadratische Approximation des Newton-Verfahrens zu nur einem geringen Approximationsfehler führt. Gegenüber der Methode des steilsten Abstiegs ist dies ein Vorteil, da für die anschließende Schrittweitenoptimierung ein geeigneter Startwert für die Suche von $\alpha^*[k]$ bereits intuitiv zur Verfügung steht. Die Wahl $\alpha[k] = 1$ ergibt dann einen Abstieg in die sog. *natürliche Newton-Richtung*, während man für $\alpha[k] < 1$ von der *gedämpften Newton-Methode* spricht. Eine Illustration des Verfahrens, auch im Vergleich zur Methode des steilsten Abstiegs für ein schlecht konditioniertes Beispiel, ist in Abb. 4.12 dargestellt.

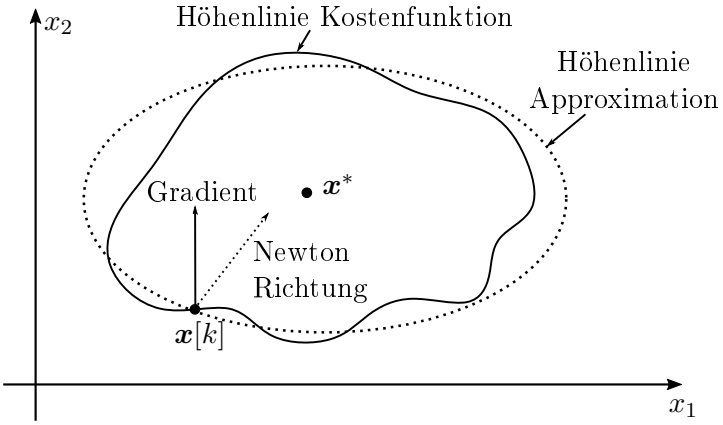


Abb. 4.12: Beispielhafte Darstellung für das Newton-Verfahren im Vergleich zur Methode des steilsten Abstiegs

Hinsichtlich der Konvergenzeigenschaften des Verfahrens sei auf folgenden Satz verwiesen:

Satz 4.9: Konvergenz der Newton-Methode

Gegeben sei eine mindestens zweimal differenzierbare Funktion $\boldsymbol{J} : \mathbb{R}^n \to \mathbb{R}$ *mit dem lokalen Minimum* $\boldsymbol{x}^*$*. Wenn die Hesse-Matrix* $(\nabla^2 J)(\boldsymbol{x}^*)$ *positiv definit ist und der Anfangswert* $\boldsymbol{x}[k=0]$ *in einer hinreichend nahen Umgebung des Minimum liegt, dann konvergiert die Newton-Iteration*

$$\boldsymbol{x}[k+1] = \boldsymbol{x}[k] - \alpha[k][(\nabla^2 J)(\boldsymbol{x}[k])]^{-1}(\nabla J)(\boldsymbol{x}[k]) \tag{4.55}$$

mit quadratischer Konvergenz gegen $\boldsymbol{x}^*$.

Problematisch kann die Newton-Methode hingegen werden, falls der aktuelle Iterationspunkt $\boldsymbol{x}[k]$ weit vom Minimum $\boldsymbol{x}^*$ entfernt ist und die Hesse-Matrix nicht mehr positiv definit ist. Dann existiert keine eindeutige Lösung von (4.52). In diesem Fall wird die modifizierte Iterationsvorschrift

$$\boldsymbol{x}[k+1] = \boldsymbol{x}[k] - \alpha[k]\left[(\nabla^2 J)(\boldsymbol{x}[k]) + \varepsilon[k]\boldsymbol{I}\right]^{-1}(\nabla J)(\boldsymbol{x}[k]) \quad \text{mit} \quad \varepsilon[k] \in \{\mathbb{R}|\varepsilon[k] > 0\} \tag{4.56}$$

mit einem geeigneten $\varepsilon[k]$ genutzt. Eine zielführende Wahl von $\varepsilon[k]$ ist in der Praxis allerdings nicht trivial. Typischerweise wird $\varepsilon[k]$ beginnend bei einem kleinen Startwert $\varepsilon[k] > 0$ sukzessive erhöht, bis die Matrix

$$(\nabla^2 J)(\boldsymbol{x}[k]) + \varepsilon[k]\boldsymbol{I}$$

positiv definit ist. Die Vor- und Nachteile der Newton-Methode sind nachfolgend zusammengefasst:

Vorteile	Nachteile
• Quadratische Konvergenz, wenn Hesse-Matrix positiv definit ist	• Außerhalb der Nähe von $\boldsymbol{x}^*$ ist die Hesse-Matrix häufig nicht positiv definit • Erhöhter Berechnungsaufwand durch explizite Berechnung der Hesse-Matrix

Tab. 4.3: Vor- und Nachteile der Newton-Methode

Quasi-Newton-Verfahren

Wesentlicher Nachteil der Newton-Methode ist die aufwändige Berechnung der Hesse-Matrix, insbesondere dann, wenn diese mittels der Differenzenquotienten approximiert werden muss. Aus diesem Grund wird bei der Quasi-Newton-Methode die inverse Hesse-Matrix iterativ bestimmt, genauer gesagt approximiert. Dazu werden Informationen über die Gradienten an den einzelnen Iterationspunkten ausgenutzt, die Informationen über die zweite Ableitung entlang der Suchrichtung liefern.

Hierzu wird der Gradient ∇J an der Stelle $\boldsymbol{x}[k]$ durch folgende Taylor-Reihe in Richtung $\boldsymbol{s}[k]$ approximiert:

$$(\nabla J)(\boldsymbol{x}[k] + \boldsymbol{s}[k]) \approx (\nabla J)(\boldsymbol{x}[k]) + (\nabla^2 J)(\boldsymbol{x}[k])\boldsymbol{s}[k]. \tag{4.57}$$

Für hinreichend kleine $\boldsymbol{s}[k] = \boldsymbol{x}[k+1] - \boldsymbol{x}[k]$ ergibt sich dann eine Approximation für den Gradienten am vorherigen Punkt $\boldsymbol{x}[k]$:

$$(\nabla J)(\boldsymbol{x}[k+1]) \approx (\nabla J)(\boldsymbol{x}[k]) + (\nabla^2 J)(\boldsymbol{x}[k])(\boldsymbol{x}[k+1] - \boldsymbol{x}[k]). \tag{4.58}$$

Auf dieser Basis kann die Hesse-Matrix durch $\boldsymbol{K}[k] \approx (\nabla^2 J)(\boldsymbol{x}[k])$ angenähert werden, indem ein $\boldsymbol{K}[k]$ gefunden wird, welches folgende Gleichung erfüllt:

$$\boldsymbol{K}([k]) \underbrace{(\boldsymbol{x}[k+1] - \boldsymbol{x}[k])}_{\boldsymbol{p}[k] \in \mathbb{R}^n} = \underbrace{(\nabla J)(\boldsymbol{x}[k+1]) - (\nabla J)(\boldsymbol{x}[k])}_{\boldsymbol{q}[k] \in \mathbb{R}^n}. \tag{4.59}$$

Diese Gleichung ist zunächst unterbestimmt, da $\boldsymbol{K} \in \mathbb{R}^{n \times n}$ insgesamt n^2 Elemente aufweist, während nur n Gleichungen zur Verfügung stehen. Durch weitere Annahmen, insbesondere der, dass die approximierte Hesse-Matrix symmetrisch sowie konstant $\boldsymbol{K}([k]) = \boldsymbol{K}$ ist, kann dennoch eine analytische Lösung gefunden werden. Hierfür werden n Stichproben (Abtastungen) hinsichtlich $\boldsymbol{p}$ und $\boldsymbol{q}$ gesammelt, also

$$\boldsymbol{P} = \begin{bmatrix} \boldsymbol{p}[0] & \boldsymbol{p}[1] & \dots & \boldsymbol{p}[n-1] \end{bmatrix}, \quad \boldsymbol{Q} = \begin{bmatrix} \boldsymbol{q}[0] & \boldsymbol{q}[1] & \dots & \boldsymbol{q}[n-1] \end{bmatrix}, \tag{4.60}$$

wobei die einzelnen Spaltenvektoren in $\boldsymbol{P} \in \mathbb{R}^{n \times n}$ und $\boldsymbol{Q} \in \mathbb{R}^{n \times n}$ linear unabhängig voneinander sein müssen. Dann folgt für $\boldsymbol{K}$:

$$\boldsymbol{K} = \boldsymbol{Q}\boldsymbol{P}^{-1}. \tag{4.61}$$

Ziel ist es nun, die Inverse $\boldsymbol{H} = \boldsymbol{K}^{-1}$ iterativ zu konstruieren. Hierfür bestehen prinzipiell mehrere Möglichkeiten. Beispielhaft soll eine symmetrische Rekonstruktion mittels

$$\boldsymbol{H}[k+1] = \boldsymbol{H}[k] + \gamma[k]\boldsymbol{z}[k]\boldsymbol{z}^T[k], \tag{4.62}$$

wobei $\gamma \in \mathbb{R}$ und $\boldsymbol{z} \in \mathbb{R}^n$ noch zu bestimmende Parameter sind, betrachtet werden. Das dyadische Produkt $\boldsymbol{z}[k]\boldsymbol{z}^T[k]$ erhält die Symmetrie und hat höchstens den Rang 1, weshalb diese Korrektur auch als *Rang 1 Korrektur* bezeichnet wird. Die Inverse von (4.61) kann hierzu für jeden Spaltenvektor in $\boldsymbol{P}$ und $\boldsymbol{Q}$ einzeln geschrieben werden:

$$\boldsymbol{H}[k+1]\boldsymbol{q}[j] = \boldsymbol{p}[j], \quad j = 0, \dots, k-1. \tag{4.63}$$

Einsetzen von (4.62) in (4.63) liefert dann für den $[k+1]$-ten Iterationsschritt:

$$\boldsymbol{p}[k] = \boldsymbol{H}[k+1]\boldsymbol{q}[k] = \boldsymbol{H}[k]\boldsymbol{q}[k] + \gamma[k]\boldsymbol{z}[k]\boldsymbol{z}^T[k]\boldsymbol{q}[k]. \tag{4.64}$$

Mit folgender Umformung unter Nutzung von (4.64)

$$\begin{aligned} (\boldsymbol{p}[k] - \boldsymbol{H}[k]\boldsymbol{q}[k])\,(\boldsymbol{p}[k] - \boldsymbol{H}[k]\boldsymbol{q}[k])^T &= \gamma^2[k]\left(\boldsymbol{z}[k]\boldsymbol{z}^T[k]\boldsymbol{q}[k]\right)\left(\boldsymbol{z}[k]\boldsymbol{z}^T[k]\boldsymbol{q}[k]\right)^T \\ &= \gamma^2[k]\boldsymbol{z}[k]\underbrace{\boldsymbol{z}^T[k]\boldsymbol{q}[k]\boldsymbol{q}^T[k]\boldsymbol{z}[k]}_{(\boldsymbol{z}^T[k]\boldsymbol{q}[k])^2}\boldsymbol{z}^T[k] \end{aligned} \tag{4.65}$$

lässt sich (4.62) wie folgt darstellen:

$$\boldsymbol{H}[k+1] = \boldsymbol{H}[k] + \frac{(\boldsymbol{p}[k] - \boldsymbol{H}[k]\boldsymbol{q}[k])\,(\boldsymbol{p}[k] - \boldsymbol{H}[k]\boldsymbol{q}[k])^T}{\gamma[k]\,(\boldsymbol{z}^T[k]\boldsymbol{q}[k])^2}. \tag{4.66}$$

Wird für (4.63) das Skalarprodukt mit $\boldsymbol{q}[k]$ gebildet,

$$\boldsymbol{q}^T[k]\boldsymbol{p}[k] = \boldsymbol{q}^T[k]\boldsymbol{H}[k]\boldsymbol{q}[k] + \gamma[k]\left(\boldsymbol{z}^T[k]\boldsymbol{q}[k]\right)^2, \tag{4.67}$$

dann lassen sich in (4.66) die unbekannten Parameter $\boldsymbol{z}$ und γ ersetzen und es kann eine geschlossene Iterationsvorschrift gefunden werden:

$$\boldsymbol{H}[k+1] = \boldsymbol{H}[k] + \frac{\left(\boldsymbol{p}[k] - \boldsymbol{H}[k]\boldsymbol{q}[k]\right)\left(\boldsymbol{p}[k] - \boldsymbol{H}[k]\boldsymbol{q}[k]\right)^T}{\boldsymbol{q}^T[k]\left(\boldsymbol{p}[k] - \boldsymbol{H}[k]\boldsymbol{q}[k]\right)}. \tag{4.68}$$

Somit lässt sich dann folgender Satz formulieren:

Satz 4.10: Quasi-Newton-Methode mit Rang 1 Korrektur

Angenommen $\boldsymbol{K} \in \mathbb{R}^{n\times n}$ ist eine konstante symmetrische Matrix und die Vektoren $\boldsymbol{p}[0], \boldsymbol{p}[1], \ldots$ sind linear unabhängig. Mit $\boldsymbol{q}[j] = \boldsymbol{K}\boldsymbol{p}[j]$ für $j = 0, 1, \ldots, k$ gilt für jede symmetrische Startmatrix $\boldsymbol{H}[0]$ und die Iterationsvorschrift

$$\boldsymbol{H}[j+1] = \boldsymbol{H}[j] + \frac{\left(\boldsymbol{p}[j] - \boldsymbol{H}[j]\boldsymbol{q}[j]\right)\left(\boldsymbol{p}[j] - \boldsymbol{H}[j]\boldsymbol{q}[j]\right)^T}{\boldsymbol{q}^T[j]\left(\boldsymbol{p}[j] - \boldsymbol{H}[j]\boldsymbol{q}[j]\right)} \tag{4.69}$$

die Beziehung

$$\boldsymbol{p}[j] = \boldsymbol{H}[k+1]\boldsymbol{q}[j], \quad j = 0, 1, \ldots, k. \tag{4.70}$$

Die resultierende Iterationsvorschrift der Quasi-Newton-Methode ist analog zur vorherigen Newton-Methode, wobei die Approximation der inversen Hesse-Matrix genutzt wird:

$$\boldsymbol{x}[k+1] = \boldsymbol{x}[k] - \alpha[k]\boldsymbol{H}[k](\nabla J)(\boldsymbol{x}[k]). \tag{4.71}$$

Ein wesentlicher Nachteil der Rang 1 Korrektur ist, dass die positive Definitheit von $\boldsymbol{H}[k+1]$ nur dann garantiert werden kann, wenn

$$\boldsymbol{q}^T[k]\left(\boldsymbol{p}[k] - \boldsymbol{H}[k]\boldsymbol{q}[k]\right) > 0 \tag{4.72}$$

gilt. Daher sind in der Literatur eine Reihe weiterer Iterationsvorschriften bekannt, welche statt (4.62) einen anderen Korrekturansatz wählen. Ohne weitere Herleitung sei hier der bekannteste Ansatz nach *Broyden-Fletcher-Goldfarb-Shanno* (BFGS) genannt:

$$\boldsymbol{H}[k+1] = \left(\boldsymbol{I} - \frac{\boldsymbol{p}[k]\boldsymbol{q}^T[k]}{\boldsymbol{q}^T[k]\boldsymbol{p}[k]}\right)\boldsymbol{H}[k]\left(\boldsymbol{I} - \frac{\boldsymbol{q}[k]\boldsymbol{p}^T[k]}{\boldsymbol{q}^T[k]\boldsymbol{p}[k]}\right) + \frac{\boldsymbol{p}[k]\boldsymbol{p}^T[k]}{\boldsymbol{q}^T[k]\boldsymbol{p}[k]}. \tag{4.73}$$

Hier wird im Korrekturschritt (4.62) eine Matrix mit Rang 2 gewählt, sodass die BFGS-Iteration der Klasse der sog. *Rang 2 Korrekturformeln* zugeordnet wird. Gegenüber dem vorherigen Ansatz wird in (4.73) deutlich, dass die BFGS-Formel genutzt werden kann, sofern

$$\boldsymbol{q}^T[k]\boldsymbol{p}[k] > 0 \tag{4.74}$$

gilt. Es kann zudem gezeigt werden, dass diese Bedingung durch eine zielführende Wahl der Schrittweite α sichergestellt werden kann, konkret dann wenn die Wolfe-Bedingungen (siehe Abb. 4.9) eingehalten werden.

Hinsichtlich der Konvergenzeigenschaften der Quasi-Newton-Methode sei folgender Satz ange-

führt:

Satz 4.11: Konvergenz der Quasi-Newton-Methode

Gegeben sei eine mindestens zweimal differenzierbare Funktion $\boldsymbol{J} : \mathbb{R}^n \to \mathbb{R}$ mit dem lokalen Minimum $\boldsymbol{x}^*$. Ist $\boldsymbol{J}$ eine allgemeine, konvexe Kostenfunktion, so konvergiert die Quasi-Newton-Methode mit superlinearer Konvergenzordnung. Liegt hingegen eine quadratische Kostenfunktion der Form

$$J(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^T\boldsymbol{W}\boldsymbol{x} + \boldsymbol{w}^T\boldsymbol{x} + c \tag{4.75}$$

mit positiv definiter Gewichtungsmatrix $\boldsymbol{W} \in \mathbb{R}^{n\times n}$, einem Vektor $\boldsymbol{w} \in \mathbb{R}^n$ und der Konstanten $\boldsymbol{c} \in \mathbb{R}$ vor, dann konvergiert die Quasi-Newton-Methode nach genau n Iterationsschritten.

Somit lassen sich abschließend die Vor- und Nachteile der Quasi-Newton-Methode wie folgt zusammenfassen:

Vorteile	Nachteile
• Berechnungsaufwand gegenüber Newton-Verfahren reduziert, da nur Gradient notwendig • Mindestens superlineare Konvergenz (also besser als Methode des steilsten Abstiegs) • Bei quadratischer Kostenfunktion: Konvergenz nach spätesten n Schritten	• Matrix $\boldsymbol{H}[k]$ muss gespeichert werden • Durch Rekursion mehr Berechnungsaufwand verglichen mit Methode des steilsten Abstiegs

Tab. 4.4: Vor- und Nachteile der Quasi-Newton-Methode

Gauss-Newton-Verfahren und Levenberg-Marquardt-Algorithmus

Bei der Gauss-Newton-Methode wird die Berechnung der Hesse-Matrix durch eine weniger rechenintensive, approximierte Berechnung ersetzt. Allerdings ist die Methode nur dann anwendbar, wenn die Kostenfunktion die Gestalt

$$J(\boldsymbol{x}) = \frac{1}{2}\left\|\boldsymbol{f}(x)\right\|_2 = \frac{1}{2}\sum_{i=1}^{m} f_i^2(\boldsymbol{x}) \tag{4.76}$$

mit einer beliebigen Funktion $\boldsymbol{f}(\boldsymbol{x}) : \mathbb{R}^n \to \mathbb{R}^m$ aufweist[1]. Hier seien die Funktionselemente f_i mindestens zwei mal differenzierbar. Die exakte Hesse-Matrix der Kostenfunktion ergibt sich dann zu

$$(\nabla^2 J)(\boldsymbol{x}) = (\nabla \boldsymbol{f})(\boldsymbol{x})(\nabla \boldsymbol{f})^T(\boldsymbol{x}) + \sum_{i=1}^{m} f_i(\boldsymbol{x})(\nabla^2 f_i)(\boldsymbol{x}), \tag{4.77}$$

[1] Während die Struktur der Kostenfunktion explizit quadratisch sein muss, kann $\boldsymbol{f}(\boldsymbol{x})$ von beliebiger Gestalt sein, d. h. auch nichtlinear. Wird $\boldsymbol{f}(\boldsymbol{x})$ als Residuum aufgefasst, entspricht das Gauss-Newton-Verfahren der Methode der kleinsten Quadrate, bei dem ein nichtlineares Gleichungssystem in eine Folge linearer Ausgleichsprobleme überführt wird.

wobei die Spalten der Jacobi-Matrix $(\nabla \boldsymbol{f})(\boldsymbol{x}) \in \mathbb{R}^{n \times m}$ die Gradienten $(\nabla f_i)(\boldsymbol{x})$ enthalten. Bei der Gauss-Newton-Methode wird der zweite Summand in (4.77) vernachlässigt, so dass sich die Näherung

$$(\nabla^2 J)(\boldsymbol{x}) \approx (\nabla \boldsymbol{f})(\boldsymbol{x})(\nabla \boldsymbol{f})^T(\boldsymbol{x}) \tag{4.78}$$

ergibt. Der damit verbundene Näherungsfehler ist also klein, wenn $f_i(\boldsymbol{x})$ oder $(\nabla^2 f_i)(\boldsymbol{x})$ betraglich kleine Werte annehmen, was typischerweise in hinreichender Nähe zum Optimum der Fall ist. Im Gegensatz zum Newton-Verfahren, welches versucht, sich durch die Minimierung einer Taylor-Approximation der eigentlichen Funktion in jedem Iterationsschritt einem Minimum zu nähern, setzt das Gauss-Newton-Verfahren eine Näherung 1. Ordnung für die Hesse-Matrix an, sodass die Minimierung durch eine analytisch, geschlossene Lösung bestimmt werden kann. Hierzu wird die Näherung aus (4.78) in die Bestimmung der Suchrichtung in (4.53) eingesetzt:

$$\begin{aligned} \boldsymbol{s}[k] &= -[(\nabla^2 J)(\boldsymbol{x}[k])]^{-1}(\nabla J)(\boldsymbol{x}[k]) \\ &= -[(\nabla \boldsymbol{f})(\boldsymbol{x}[k])(\nabla \boldsymbol{f})^T(\boldsymbol{x}[k])]^{-1}(\nabla J)(\boldsymbol{x}[k]) \\ &= -[(\nabla \boldsymbol{f})(\boldsymbol{x}[k])(\nabla \boldsymbol{f})^T(\boldsymbol{x}[k])]^{-1}(\nabla \boldsymbol{f})(\boldsymbol{x}[k])\boldsymbol{f}(\boldsymbol{x}[k]). \end{aligned} \tag{4.79}$$

Diese Wahl der Suchrichtung wir daher auch *Gauss-Newton-Richtung* genannt. Die Iterationsvorschrift der Gauss-Newton-Methode ergibt sich dann unter Ergänzung einer geeigneten Schrittweite $\alpha[k]$ zu:

$$\boldsymbol{x}[k+1] = \boldsymbol{x}[k] - \alpha[k][(\nabla \boldsymbol{f})(\boldsymbol{x}[k])(\nabla \boldsymbol{f})^T(\boldsymbol{x}[k])]^{-1}(\nabla \boldsymbol{f})(\boldsymbol{x}[k])\boldsymbol{f}(\boldsymbol{x}[k]). \tag{4.80}$$

Für $\alpha[k] \leq 1$ ergibt sich die *gedämpfte Gauss-Newton-Methode*, wobei $\alpha[k]$ dann der Dämpfungsparameter ist. Es sei zudem angemerkt, dass statt der expliziten Matrix-Inversion in (4.79) auch ein alternatives Lösungsverfahren für das zugrundeliegende Gleichungssystem aus Kap. 3.1.4 genutzt werden kann. Kann keine Lösung gefunden werden, weil die Hesse-Matrix-Näherung in (4.77) nicht mehr positiv definit ist, ist analog zum konventionellen Newton-Verfahren eine modifizierte Iteration der Form

$$\boldsymbol{x}[k+1] = \boldsymbol{x}[k] - \alpha[k][(\nabla \boldsymbol{f})(\boldsymbol{x}[k])(\nabla \boldsymbol{f})^T(\boldsymbol{x}[k]) + \varepsilon[k]\boldsymbol{I}]^{-1}(\nabla \boldsymbol{f})(\boldsymbol{x}[k])\boldsymbol{f}(\boldsymbol{x}[k]) \tag{4.81}$$

mit einem geeigneten

$$\varepsilon[k] \in \{\mathbb{R} | \varepsilon[k] > 0\} \tag{4.82}$$

möglich. Diese Modifikation ist als *Levenberg-Marquardt-Methode* bekannt. Für $\boldsymbol{\varepsilon}[k] \to 0$ geht diese in die reguläre Gauss-Newton-Methode über und für $\boldsymbol{\varepsilon}[k] >> 1$ ergibt sich die Methode des steilsten Abstiegs. Bezüglich der Konvergenzordnung sei auf folgenden Satz verwiesen:

Satz 4.12: Konvergenz der Gauss-Newton-Methode

Gegeben sei eine mindestens zweimal differenzierbare Funktion $\boldsymbol{J} : \mathbb{R}^n \to \mathbb{R}$ mit dem lokalen Minimum $\boldsymbol{x}^*$. Wenn die Hesse-Matrix $(\nabla^2 J)(\boldsymbol{x}^*)$ positiv definit ist und der Anfangswert $\boldsymbol{x}[k=0]$ in einer hinreichend nahen Umgebung des Minimums liegt, dann konvergiert die Newton-Iteration

$$\boldsymbol{x}[k+1] = \boldsymbol{x}[k] - \alpha[k][(\nabla \boldsymbol{f})(\boldsymbol{x}[k])(\nabla \boldsymbol{f})^T(\boldsymbol{x}[k])]^{-1}(\nabla \boldsymbol{f})(\boldsymbol{x}[k])\boldsymbol{f}(\boldsymbol{x}[k]) \tag{4.83}$$

mit quadratischer Konvergenz gegen $\boldsymbol{x}^*$.

Hinsichtlich obigen Satzes sei betont, dass die Konvergenz tatsächlich nur in hinreichender Nähe

zu $\boldsymbol{x}^*$ sichergestellt ist. Falls der Approximationsfehler zu signifikant ist, kann es sein, dass die Gauss-Newton-Methode nicht konvergiert. Die Vor- und Nachteile der Gauss-Newton-Methode sind wie folgt zusammenzufassen:

Vorteile	Nachteile
• Nur Gradienten-Informationen notwendig • In der Nähe von $\boldsymbol{x}^*$ besteht quadratische Konvergenz	• Konvergenz stark vom konkreten Problem abhängig und nicht garantiert • Berechnungsaufwand für die Jacobi-Matrix $\boldsymbol{f}(\boldsymbol{x})$

Tab. 4.5: Vor- und Nachteile der Gauss-Newton-Methode

4.2.4 Zusammenfassung

Die vorhergehend betrachteten Verfahren zur Lösung des unbeschränkten, statischen Optimierungsproblems (4.19) sind als Einstieg in die numerische Optimierung zu verstehen. Der Fokus der vorgestellten Verfahren lag auf der Klasse der sog. Liniensuchverfahren, welche das Optimierungsproblem in die zielführende Wahl der Suchrichtung und der Schrittweite aufspalteten. Eine kleine Gegenüberstellung diese Verfahren ist in nachfolgender Tabelle aufgezeigt:

Methode	Idee	Konvergenz
Steilster Abstieg	Negativer Gradient	Linear oder sofort für ideal konditionierte quadratische Probleme
Newton	Geschlossene Lösung auf quadratischer Approximation der Kostenfunktion	Quadratisch (in der Nähe von $\boldsymbol{x}^*$) oder sofort für quadr. Probleme
Quasi-Newton	Wie Newton plus rekursive Approximation der Hesse-Matrix	Superlinear oder in n-Schritten für quadratische Probleme
Gauss-Newton	Geschlossene Lösung durch lineare Näherung der Hesse-Matrix	Quadratisch (in der Nähe von $\boldsymbol{x}^*$) oder sofort für quadr. Probleme

Tab. 4.6: Gegenüberstellung der numerischen Verfahren zur Auffindung der Suchrichtung innerhalb der Klasse der Liniensuchverfahren

Gegenüber den in dieser Veranstaltung behandelten Liniensuchverfahren sind in der Literatur noch weitere Ansätze zu finden z. B.

- Backtracking Schrittweiten-Anpassung,

oder folgende Verfahren zur Auswahl einer zielführenden Suchrichtung:

- Konjugierte Gradientenmethode,
- weitere Varianten der (Quasi-)Newton-Implementierung.

Zudem sei die *Methode der Vertrauensbereiche (trust region method)* erwähnt. Diese nutzt zwar die 1. Ableitung der Kostenfunktion, wird klassischerweise aber nicht den Liniensuchverfahren zugeordnet, da Suchrichtung und Schrittweite gemeinsam bestimmt werden. Es kann dennoch gezeigt werden, dass diese Methode für viele Optimierungsszenarien ähnliche Eigenschaften wie die Levenberg-Marquardt-Methode aufweist. Zu Beginn des Teilkapitels wurde mit dem *Nelder-Mead-Verfahren* zudem ein Ansatz aus der Klasse der ableitungsfreien Verfahren erwähnt.

Für ein gegebenes Optimierungsproblem muss die Wahl eines zielführendes Lösungsverfahrens i. A. anwendungsabhängig geprüft werden. Ist beispielsweise die Berechnung der Hesse-Matrix mit einem hohem Berechnungsaufwand verbunden, da beispielsweise der Differenzenquotient zu dessen Näherung genutzt werden muss, sollten Verfahren vermieden werden, welche diese Information brauchen. Ein ideales Lösungsverfahren, welches für jede Anwendung optimale Eigenschaften hinsichtlich Genauigkeit, Robustheit und Geschwindigkeit aufweist, existiert nicht.

4.3 Statische Optimierung mit Beschränkungen

Das allgemeine statische Optimierungsproblem unter Nebenbedingungen wurde bereits in Definition 4.1 eingeführt. Aufgrund der attraktiven Eigenschaften konvexer Optimierungsprobleme, sei zunächst unter Verwendung der Eigenschaften von konvexen Mengen und Funktionen aus Kap. 4.1.1 noch einmal klar definiert, wann ein statisches Optimierungsproblem mit Beschränkungen (strikt) konvex ist:

Definition 4.10: Konvexe statische Optimierung mit Beschränkungen

Das statische, beschränkte Optimierungsproblem

$$\min_{\boldsymbol{x}\in\mathcal{X}} \quad J(\boldsymbol{x})\,, \tag{4.84}$$

$$\text{u. d. Nb.} \quad \boldsymbol{g}(\boldsymbol{x}) = \boldsymbol{0}\,, \tag{4.85}$$

$$\boldsymbol{h}(\boldsymbol{x}) \leq \boldsymbol{0}. \tag{4.86}$$

entsprechend der vorherigen Definition 4.1 ist (strikt) konvex, falls

(a) die Kostenfunktion $J(\boldsymbol{x})$ (strikt) konvex auf dem zulässigen Gebiet $\mathcal{X}$ ist,

(b) die Funktionen $g_i(\boldsymbol{x})(i = 1, \ldots, p)$ linear sind,

(c) die Funktionen $h_i(\boldsymbol{x})(i = 1, \ldots, q)$ konvex sind.

Hinsichtlich des Optimierungsergebnisses ergeben sich in diesem Fall folgende attraktive Eigenschaften:

(a) Jedes lokale Minimum ist ein globales Minimum.

(b) Ein strikt konvexes Optimierungsproblem kann nicht mehr als eine Lösung besitzen, d. h., falls ein lokales Minimum existiert, ist dieses das strikt globale Minimum.

Handelt es sich hingegen um ein *echt nichtlineares Optimierungsproblem*, können die nachfolgend vorgestellten Optimierungsmethoden in ungewünschte lokale Nebenminima konvergieren. In diesem Fall sei daher auf Kap. 4.4 verwiesen.

Die Berücksichtigung von allgemeinen, nichtlinearen Ungleichungsbeschränkungen mittels $\boldsymbol{h}(\boldsymbol{x})$ ist zumeist schwieriger als die Betrachtung von Gleichungsbeschränkungen. Daher kann es sinnvoll sein, Definition 4.1 umzuformulieren und Ungleichungsbeschränkungen mittels sog. Überschussvariablen (*slack variables*) $\boldsymbol{x}_s \in \mathbb{R}^d$ in die äquivalente Form

$$\begin{aligned} \min_{\boldsymbol{x},\boldsymbol{x}_s} \quad & J(\boldsymbol{x})\,, && (4.87)\\ \text{u. d. Nb.} \quad & \boldsymbol{g}(\boldsymbol{x}) = \boldsymbol{0}\,, && (4.88)\\ & \boldsymbol{h}(\boldsymbol{x}) + \boldsymbol{x}_s = \boldsymbol{0}\,, && (4.89)\\ & \boldsymbol{x}_s \geq 0 && (4.90) \end{aligned}$$

zu überführen. Die Überschussvariablen $\boldsymbol{x}_s$ stellen zusätzliche Optimierungsvariablen dar, d. h., die Dimension des Optimierungsproblems erhöht sich um d.

4.3.1 Optimalitätsbedingungen

Nachfolgend werden notwendige und hinreiche Optimalbedingungen für das Problem in Definition 4.1 entwickelt. Zur Veranschaulichung werden hierzu zunächst zwei einleitende Beispiele diskutiert.

Beispiel: Eine Gleichungsbeschränkung

Folgendes beispielhaftes Optimierungsproblem sei gegeben:

$$\begin{aligned} \min_{\boldsymbol{x}} \quad & J(\boldsymbol{x}) = (x_1 - 2)^2 + (x_2 - 1)^2\,, && (4.91a)\\ \text{u. d. Nb.} \quad & g(\boldsymbol{x}) = x_2 - 2x_1 = 0. && (4.91b) \end{aligned}$$

In Abb. 4.13 sind die entsprechenden Höhenlinien von $J(\boldsymbol{x})$ sowie die Gerade $g(\boldsymbol{x}) = 0$ dargestellt. Da die Punkte $\boldsymbol{x}$ auf der Geraden liegen müssen, existieren bis zu zwei Schnittpunkte zwischen den Höhenlinien von $J(\boldsymbol{x}) > J(\boldsymbol{x}^*)$ und der Geraden. Es ist unmittelbar ersichtlich, dass das Optimum gefunden werden kann, wenn die Höhenlinien $J(\boldsymbol{x})$ derart verengt werden, dass der Tangentialpunkt zwischen diesen und der Geraden $g(\boldsymbol{x}) = 0$ erreicht wird:

$$\boldsymbol{x}^* = \begin{bmatrix} 0,9 & 1,6 \end{bmatrix}^T, \quad J(\boldsymbol{x}^*) = 1,8. \tag{4.92}$$

An diesem Punkt verlaufen die Gradienten von $g(\boldsymbol{x})$ und $J(\boldsymbol{x})$

$$\nabla J(\boldsymbol{x}) = \begin{bmatrix} 2(x_1 - 1) \\ 2(x_2 - 1) \end{bmatrix}, \quad \nabla g(\boldsymbol{x}) = \begin{bmatrix} 2 \\ 1 \end{bmatrix} \tag{4.93}$$

kollinear. Einsetzen des optimalen Punkts aus (4.92) ergibt dementsprechend

$$\nabla J(\boldsymbol{x}^*) = \begin{bmatrix} -2,4 \\ 1,2 \end{bmatrix}, \quad \nabla g(\boldsymbol{x}^*) = \begin{bmatrix} -2 \\ 1 \end{bmatrix}, \tag{4.94}$$

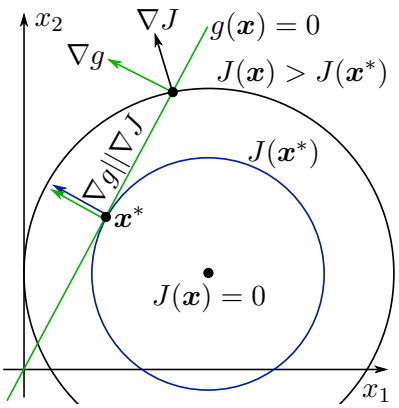


Abb. 4.13: Illustration des beispielhaften Optimierungsproblems mit einer Gleichungsbeschränkung

also $\nabla g(\boldsymbol{x}^*) || \nabla f(\boldsymbol{x}^*)$. Dieses Ergebnis des Beispiels kann mit Hilfe der *Lagrange-Funktion*

$$L(\boldsymbol{x}, \lambda) = J(\boldsymbol{x}) + \lambda g(\boldsymbol{x}) \tag{4.95}$$

verallgemeinert werden[1]. Diese addiert die Gleichungsbeschränkung mittels eines *Lagrange-Multiplikators* λ zur Kostenfunktion. Die Kollinearität von $\nabla g(\boldsymbol{x}^*)$ und $\nabla f(\boldsymbol{x}^*)$ ergibt sich dann aus der *Stationaritätsbedingung*[2]

$$\nabla_{\boldsymbol{x}} L(\boldsymbol{x}^*, \lambda^*) = \nabla J(\boldsymbol{x}^*) + \lambda^* \nabla g(\boldsymbol{x}^*) = \mathbf{0}, \tag{4.96a}$$

$$g(\boldsymbol{x}^*) = 0. \tag{4.96b}$$

Somit ergibt sich ein Gleichungssystem der Ordnung $n+1$ mit $n+1$ Unbekannten für $\boldsymbol{x}^* \in \mathbb{R}^n$ und $\lambda^* \in \mathbb{R}$. Die Stationaritätsbedingung liefert für den gegebenen Fall eine erste notwendige aber keine hinreichende Bedingung für Optimalität.

Um die Beziehung zwischen dem Lagrange-Multiplikator und der Lösung $\boldsymbol{x}^*$ zu untersuchen, wird eine Modifikation der Beschränkung in der Form

$$g(\boldsymbol{x}) = \varepsilon \quad \Rightarrow \quad \tilde{g}(\boldsymbol{x}, \varepsilon) = g(\boldsymbol{x}(\varepsilon)) - \varepsilon = 0, \tag{4.97}$$

mit

$$\varepsilon = \{\varepsilon \in \mathbb{R} | \varepsilon << 1\} \tag{4.98}$$

betrachtet. Die Variablen $\boldsymbol{x}(\varepsilon)$ hängen in diesem Fall daher auch von ε ab, wobei das modifizierte Problem für $\varepsilon \to 0$ gegen das ursprüngliche Problem konvergiert. Da $\tilde{g} = 0$ für alle ε gelten muss, gilt zudem:

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}\varepsilon} \tilde{g}(\boldsymbol{x}, \varepsilon) = 0 &= \frac{\mathrm{d}}{\mathrm{d}\varepsilon} g(\boldsymbol{x}(\varepsilon)) + \frac{\mathrm{d}\varepsilon}{\mathrm{d}\varepsilon} \\ &= \nabla_{\boldsymbol{x}} g(\boldsymbol{x}(\varepsilon)) \frac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}\varepsilon} - 1. \end{aligned} \tag{4.99}$$

[1] Es gilt zu beachten, dass hier weiterhin zunächst lediglich eine Gleichungsnebenbedingung angenommen wird.

[2] Die Nomenklatur $\nabla_{\boldsymbol{x}} L(\boldsymbol{x}, \lambda)$ bezeichnet den Gradienten von $L(\boldsymbol{x}, \lambda)$ nach $\boldsymbol{x}$, d. h., die Ableitung nach λ wird ausgelassen.

Es folgt somit unmittelbar:

$$\nabla_{\boldsymbol{x}} g(\boldsymbol{x}(\varepsilon)) \frac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}\varepsilon} = 1. \tag{4.100}$$

Nun soll untersucht werden, wie sich der Kostenwert $J(\boldsymbol{x}(\varepsilon))$ an der Stelle $\boldsymbol{x}^*$ in Abhängigkeit von ε verändert. Unter Verwendung von (4.96a) ergibt sich:

$$\frac{\mathrm{d}}{\mathrm{d}\varepsilon} J(\boldsymbol{x}^*(\varepsilon))\Big|_{\varepsilon=0} = \nabla J(\boldsymbol{x}^*) \frac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}\varepsilon}\Big|_{\varepsilon=0} = -\lambda^* \underbrace{\nabla g(\boldsymbol{x}^*) \frac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}\varepsilon}\Big|_{\varepsilon=0}}_{=1,\ (4.100)}. \tag{4.101}$$

Dies kann somit in folgendem Satz zusammengefasst werden:

Satz 4.13: Lagrange-Multiplikator

Gegeben sei ein Optimierungsproblem mit einer Gleichungsnebenbedigung analog zum Beispiel (4.91). Dann gibt der Lagrange-Multiplikator λ^* das Maß der Veränderung der Kostenfunktion $J(\boldsymbol{x}(\varepsilon))$ am Optimum $\boldsymbol{x}^*$ an, sofern die Gleichungsbeschränkung um ein hinreichend kleines $\varepsilon \in \mathbb{R}$, also

$$g(\boldsymbol{x}) = \varepsilon, \tag{4.102}$$

modifiziert wird:

$$\frac{\mathrm{d}}{\mathrm{d}\varepsilon} J(\boldsymbol{x}^*(\varepsilon))\Big|_{\varepsilon=0} = -\lambda^* \tag{4.103}$$

Der Lagrange-Multiplikator kann daher unmittelbar als Sensitivitätsmaß interpretiert werden.

Ein hoher Wert von λ^* gibt daher an, dass eine Lockerung der Nebenbedingung zu einer vergleichsweise starken Reduktion der Kostenfunktion führen würde.

Beispiel: Eine Ungleichungsbeschränkung

Nun soll folgendes Problem betrachtet werden:

$$\min_{\boldsymbol{x}} \quad J(\boldsymbol{x}) = (x_1 - 2)^2 + (x_2 - 1)^2, \tag{4.104a}$$

$$\text{u. d. Nb.} \quad h(\boldsymbol{x}) = x_1 + x_2 - 2 \leq 0. \tag{4.104b}$$

Aus der graphischen Darstellung des Problems in Abb. 4.14 wird klar, dass der optimale Punkt auf dem Rand des zulässigen Gebiets

$$\mathcal{X} = \left\{ \boldsymbol{x} \in \mathbb{R}^2 | h(\boldsymbol{x}) \leq 0 \right\} \tag{4.105}$$

liegt. Die Beschränkung (4.104b) ist somit aktiv, d. h., $h(\boldsymbol{x}^*) = 0$ und die Gradienten von $J(\boldsymbol{x}^*)$ und $g(\boldsymbol{x}^*)$

$$\nabla J(\boldsymbol{x}) = \begin{bmatrix} 2(x_1 - 1) \\ 2(x_2 - 1) \end{bmatrix}, \quad \nabla h(\boldsymbol{x}) = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \tag{4.106}$$

sind kollinear im optimalen Punkt

$$\boldsymbol{x}^* = \begin{bmatrix} 1,5 & 0,5 \end{bmatrix}^T, \quad J(\boldsymbol{x}^*) = 0,5 \tag{4.107}$$

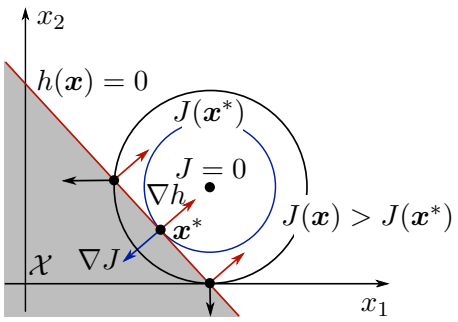


Abb. 4.14: Illustration des beispielhaften Optimierungsproblems mit einer Ungleichungsbeschränkung

mit den Werten

$$\nabla J(\boldsymbol{x}^*) = \begin{bmatrix} -1 \\ -1 \end{bmatrix}, \quad \nabla h(\boldsymbol{x}^*) = \begin{bmatrix} 1 \\ 1 \end{bmatrix}. \tag{4.108}$$

Allgemein kann eine Ungleichungsbeschränkung $h(\boldsymbol{x}) \leq 0$ analog zum vorherigen Beispiel durch die Lagrange-Funktion der Form

$$L(\boldsymbol{x}, \mu) = J(\boldsymbol{x}) + \mu h(\boldsymbol{x}) \tag{4.109}$$

abgebildet werden. Hier ist μ der Lagrange-Multiplikator im Kontext der Ungleichungsbeschränkung. Ferner ist zu differenzieren, ob in $\boldsymbol{x}^*$ diese Beschränkung aktiv ist oder nicht:

$$\text{Falls} \quad h(\boldsymbol{x}^*) < 0: \quad \nabla_{\boldsymbol{x}} L(\boldsymbol{x}^*, \mu^*) = \nabla J(\boldsymbol{x}^*) = \boldsymbol{0}, \qquad \mu^* = 0, \tag{4.110a}$$

$$\text{Falls} \quad h(\boldsymbol{x}^*) = 0: \quad \nabla_{\boldsymbol{x}} L(\boldsymbol{x}^*, \mu^*) = \nabla J(\boldsymbol{x}^*) + \mu^* \nabla h(\boldsymbol{x}^*) = \boldsymbol{0}. \tag{4.110b}$$

Das Vorzeichen von μ spielt in diesem Zusammenhang eine wichtige Rolle. Um dies zu verdeutlichen, wird $h(\boldsymbol{x})$ in eine Taylorreihe 1. Ordnung entwickelt:

$$h(\boldsymbol{x} + \boldsymbol{s}) \approx h(\boldsymbol{x}) + \nabla h(\boldsymbol{x}) \boldsymbol{s}\,. \tag{4.111}$$

Im Fall einer aktiven Beschränkung, $h(\boldsymbol{x}) = 0$, muss eine *zulässige Richtung* $\boldsymbol{s} \in \mathbb{R}^n$ die Ungleichung

$$h(\boldsymbol{x} + \boldsymbol{s}) \leq 0 \tag{4.112}$$

erfüllen, also

$$\nabla h(\boldsymbol{x}) \boldsymbol{s} \leq 0. \tag{4.113}$$

Analog wird $\boldsymbol{J}$ linear approximiert:

$$J(\boldsymbol{x} + \boldsymbol{s}) \approx J(\boldsymbol{x}) + \nabla J(\boldsymbol{x}) \boldsymbol{s}. \tag{4.114}$$

Wenn ein Punkt $\boldsymbol{x}$ kein Minimum $\boldsymbol{x}^*$ ist, muss eine *Abstiegsrichtung* $\boldsymbol{s} \in \mathbb{R}^n$ existieren mit der

$$J(\boldsymbol{x} + \boldsymbol{s}) < J(\boldsymbol{x}) \tag{4.115}$$

gilt, also

$$\nabla J(\boldsymbol{x}) \boldsymbol{s} < 0. \tag{4.116}$$

Würde dementsprechend eine Richtung $\boldsymbol{s}$ für einen Punkt $\boldsymbol{x}$ existieren, welche die Bedingungen (4.113) und (4.116) erfüllt, hieße dies, dass die Kostenfunktion innerhalb des zulässigen Gebiets

$\mathcal{X}$ reduziert werden kann. In diesem Fall kann $\boldsymbol{x}$ kein Minimum sein. Dies bedeutet, dass $\boldsymbol{x}^*$ ein optimaler Punkt ist, falls ∇J und ∇h in die umgekehrte Richtung zeigen:

$$-\nabla J(\boldsymbol{x}^*) = \mu^* \nabla h(\boldsymbol{x}^*)\,. \tag{4.117}$$

Auch mit Blick auf Abb. 4.14 kann dieser Zusammenhang nochmal verdeutlicht werden: Falls $\mu < 0$ wäre, würden ∇J und ∇h am vermeintlichen optimalen Punkt in die gleiche Richtung weisen, und eine ganze Halbebene würde entstehen, die zu einer Abnahme von J führen würde, während die Beschränkung $h(\boldsymbol{x}) \leq 0$ eingehalten wäre. Daher muss μ in $\boldsymbol{x}^*$ positiv sein. Die notwendige Bedingung für ein Minimum bei einem Optimierungsproblem mit einer Ungleichungsnebenbedingung folgt dann zu:

$$\exists \mu^* \geq 0: \quad \nabla_{\boldsymbol{x}} L(\boldsymbol{x}^*, \boldsymbol{\mu}^*) = \mathbf{0}, \quad \mu^* h(\boldsymbol{x}^*) = 0\,. \tag{4.118}$$

Der Term $\mu^* h(\boldsymbol{x}^*) = 0$ bildet die Bedingung $\mu^* = 0$ im Fall einer inaktiven Beschränkung $h(\boldsymbol{x}^*) < 0$ ab und wird als *Komplementaritätsbedingung* bezeichnet.

Des Weiteren kann analog zum Satz 4.13 gezeigt werden, dass der Lagrange-Multiplikator μ^* ein Sensitivitätsmaß für die Veränderung der Kostenfunktion bei aktiver Ungleichungsbeschränkung ist:

$$h(\boldsymbol{x}) \leq \varepsilon \quad \Rightarrow \quad \frac{\mathrm{d}}{\mathrm{d}\varepsilon} J(\boldsymbol{x}^*(\varepsilon))\Big|_{\varepsilon=0} = -\mu^*\,. \tag{4.119}$$

Auch hierbei deutet ein hoher Wert von μ^* darauf hin, dass eine Relaxation der aktiven Ungleichungsbedingung zu einer signifikanten Reduktion des Kostenwerts führen kann.

Beispiel: Zwei Ungleichungsbeschränkungen

Nun wird das vorherige Problem mit einer Ungleichgsbeschränkung auf den Fall mit zwei Ungleichungsbeschränkungen verallgemeinert:

$$\min_{\boldsymbol{x}} \quad J(\boldsymbol{x})\,, \tag{4.120a}$$

$$\text{u. d. Nb.} \quad h_1(\boldsymbol{x}) \leq 0\,, \tag{4.120b}$$

$$h_2(\boldsymbol{x}) \leq 0. \tag{4.120c}$$

Analog zum vorhergehenden Beispiel muss an Punkten $\boldsymbol{x}$ mit aktiven Beschränkungen, also $h_1(\boldsymbol{x}) = 0$ und $h_2(\boldsymbol{x}) = 0$, eine zulässige Richtung $\boldsymbol{s}$ existieren, welche die Bedingung

$$\nabla h_1(\boldsymbol{x})\boldsymbol{s} \leq 0, \quad \nabla h_2(\boldsymbol{x})\boldsymbol{s} \leq 0 \tag{4.121}$$

erfüllt. Diese Bedingung wird graphisch für $\boldsymbol{x} \in \mathbb{R}^2$ in Abb. 4.15 verdeutlicht: Die zulässigen Richtungen $\boldsymbol{s}$ ergeben sich durch die Schnittmenge der Flächen, welche durch die Tangenten an den Kurven $h_1(\boldsymbol{x}) = 0$ und $h_2(\boldsymbol{x}) = 0$ aufgespannt werden. Ebenfalls analog zum vorherigen Beispiel ist ein Punkt $\boldsymbol{x}$ kein Minimum, falls ein $\boldsymbol{s}$ mit

$$\nabla J(\boldsymbol{x})\boldsymbol{s} < 0. \tag{4.122}$$

existiert. Erneut kann $\boldsymbol{x}^*$ kein Minimum sein, falls es beide Bedingungen (4.121) und (4.122) zeitgleich erfüllt. Dies ist genau dann der Fall, wenn $-\nabla J(\boldsymbol{x}^*)$ in die Fläche zeigt, welche durch $\nabla h_1(\boldsymbol{x}^*)$ und $\nabla h_2(\boldsymbol{x}^*)$ aufgespannt wird.

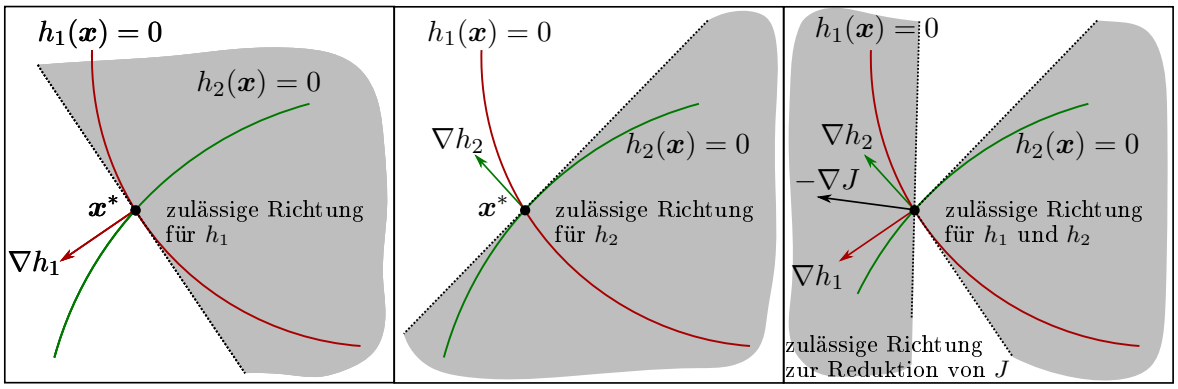


Abb. 4.15: Illustration des beispielhaften Optimierungsproblems mit zwei Ungleichungsbeschränkung

Somit muss der negative Gradient $-\nabla J(\boldsymbol{x}^*)$ mit zwei aktiven Ungleichungsbeschränkungen eine *positive Linearkombination* der Gradienten der Beschränkungsfunktionen sein:

$$\exists\,\{\mu_1^*, \mu_2^*\} \geq 0: \quad -\nabla J(\boldsymbol{x}^*) = \mu_1^* \nabla h_1(\boldsymbol{x}^*) + \mu_2^* \nabla h_2(\boldsymbol{x}^*). \tag{4.123}$$

Auch diese notwendige Bedingung kann erneut über eine Lagrange-Funktion der Form

$$L(\boldsymbol{x}, \boldsymbol{\mu}) = J(\boldsymbol{x}) + \mu_1 h_1(\boldsymbol{x}) + \mu_2 h_2(\boldsymbol{x}), \quad \boldsymbol{\mu} = \begin{bmatrix} \mu_1 & \mu_2 \end{bmatrix}^T \tag{4.124}$$

abgebildet werden. Die notwendige Optimalitätsbedingung liegt vor, falls

$$\exists\,\{\mu_1^*, \mu_2^*\} \geq 0: \quad \nabla_{\boldsymbol{x}} L(\boldsymbol{x}^*, \boldsymbol{\mu}^*) = \mathbf{0}, \quad \mu_1^* h_1(\boldsymbol{x}^*) = 0, \quad \mu_2^* h_2(\boldsymbol{x}^*) = 0 \tag{4.125}$$

gilt. Die beiden Komplementaritätsbedingungen sind erfüllt, falls entweder die entsprechende Beschränkung aktiv (und μ_i positiv) ist oder der Lagrange-Multiplikator $\mu_i = 0$ ist, was einer inaktiven Beschränkung entspricht.

Beschränkungsqualifikation

Die vorherigen Betrachtungen sollen nun auf den allgemeinen Fall

$$\min_{\boldsymbol{x} \in \mathcal{X}} \quad J(\boldsymbol{x}), \tag{4.126a}$$

$$\text{u. d. Nb.} \quad g_i(\boldsymbol{x}) = \mathbf{0}, \quad i = 1, \ldots, p, \tag{4.126b}$$

$$h_i(\boldsymbol{x}) \leq \mathbf{0}, \quad i = 1, \ldots, q. \tag{4.126c}$$

mit p Gleichungs- und q Ungleichungsbeschränkungen übertragen werden. Da ein Punkt $\boldsymbol{x} \in \mathbb{R}^n$ durch n linear unabhängige Gleichungen eindeutig beschrieben wird, bedeutet dies, dass die Summe der linear unabhängigen Gleichungs- und aktiven Ungleichungsbeschränkungen $p + q$ in $\boldsymbol{x}$ maximal n ergeben darf. Andernfalls resultiert ein überbestimmtes Gleichungssystem. Zur Abbildung dieser Problematik wird folgende Definition eingeführt:

Definition 4.11: Menge aktiver Ungleichungsbeschränkungen

Die Menge der aktiven Ungleichungsbeschränkungen des Optimierungsproblems (4.126) *an einem Punkt* $\boldsymbol{x}$ *ist definiert durch*

$$\mathcal{A} = \{i \in \{1, \ldots, q\} | h_i(\boldsymbol{x}) = 0\}\,. \tag{4.127}$$

Hierauf aufbauend kann die *linear-unabhängige Beschränkungsqualifikation (linear independence constraint qualification - LICQ)* gefordert werden:

Satz 4.14: Linear-unabhängige Beschränkungsqualifikation

Damit das Optimierungsproblem (4.126) *eindeutig lösbar ist, müssen die Gradienten der Gleichungs- und aktiven Ungleichungsbeschränkungen linear unabhängig seien, also*

$$\text{Rang} \begin{bmatrix} (\nabla g(\boldsymbol{x})^T \\ (\nabla h_{\mathcal{A}}(\boldsymbol{x}))^T \end{bmatrix} = p + \dim\{\mathcal{A}\}\,. \tag{4.128}$$

Ferner dürfen an einem Punkt $\boldsymbol{x}$ *maximal*

$$q \leq n - p$$

Ungleichungsbeschränkungen aktiv sein, wobei $p \leq n$ *gilt.*

Die LICQ-Forderung ist die gebräuchlichste Bedingung dieser Art in der Optimierung. Allerdings sei angemerkt, dass der Satz 4.14 nicht notwendig, sondern nur hinreichend ist und in manchen Fällen zu konservativ sein kann. Alternative, relaxierte Beschränkungsqualifikationen können der Literatur entnommen werden – beispielsweise die Slater-Bedingung für konvexe Optimierungsprobleme.

Optimalitätsbedingungen 1. Ordnung

Zunächst sei die Lagrange-Funktion $L(\boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{\mu})$ für das Problem (4.126) definiert:

Definition 4.12: Lagrange-Funktion

Die Lagrange-Funktion für das Problem (4.126) *ist*

$$L(\boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = J(\boldsymbol{x}) + \sum_{i=1}^{p} \lambda_i g_i(\boldsymbol{x}) + \sum_{i=1}^{q} \mu_i h_i(\boldsymbol{x}) \tag{4.129}$$

mit den Lagrange-Multiplikatoren

$$\boldsymbol{\lambda} = \begin{bmatrix} \lambda_1 & \ldots & \lambda_p \end{bmatrix}^T, \quad \boldsymbol{\mu} = \begin{bmatrix} \mu_1 & \ldots & \mu_q \end{bmatrix}^T. \tag{4.130}$$

Auf dessen Basis können die notwendigen Optimalitätsbedingungen 1. Ordnung für (4.126) angegeben werden:

Satz 4.15: Notwendige Optimalitätsbedingungen 1. Ordnung

Sei $\boldsymbol{x}^*$ ein lokales Minimum des Problems (4.126), welches die LICQ-Forderung gemäß Satz 4.14 erfüllt. Falls J, g_i und h_i stetig differenzierbar sind, existieren Lagrange-Multiplikatoren

$$\boldsymbol{\lambda}^* = \begin{bmatrix} \lambda_1^* & \dots & \lambda_p^* \end{bmatrix}^T \quad \text{und} \quad \boldsymbol{\mu}^* = \begin{bmatrix} \mu_1^* & \dots & \mu_q^* \end{bmatrix}^T , \tag{4.131}$$

welche die folgenden Bedingungen erfüllen:

$$\nabla_{\boldsymbol{x}} L(\boldsymbol{x}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*) = \mathbf{0} , \tag{4.132a}$$
$$g_i(\boldsymbol{x}^*) = 0 , \quad i = 1, \dots, p , \tag{4.132b}$$
$$h_i(\boldsymbol{x}^*) \leq 0 , \quad i = 1, \dots, q , \tag{4.132c}$$
$$\mu_i^* \geq 0 , \quad i = 1, \dots, q , \tag{4.132d}$$
$$\mu_i^* h_i(\boldsymbol{x}^*) = 0 , \quad i = 1, \dots, q . \tag{4.132e}$$

Die Bedingungen (4.132) sind als *Karush-Kuhn-Tucker* (KKT) Bedingungen bekannt. Die Bedingungen aus Satz 4.15 sind alerdings nur dann gültig, wenn die LICQ-Forderung erfüllt ist. Ist dies nicht der Fall, kann es sein, dass die KKT-Bedingungen nicht anwendbar sind. Dies soll an folgendem Beispiel demonstriert werden:

$$\min_{\boldsymbol{x} \in \mathcal{X}} \quad J(\boldsymbol{x}) = -x_1 , \tag{4.133a}$$
$$\text{u. d. Nb.} \quad h_1(\boldsymbol{x}) = x_1^3 - x_2 \leq 0 , \tag{4.133b}$$
$$h_2(\boldsymbol{x}) = x_1^3 + x_2 \leq 0 . \tag{4.133c}$$

In Abb. 4.16 ist die zulässige Menge $\mathcal{X}$ sowie der optimale Punkt $\boldsymbol{x}^* = [0 \quad 0]^T$, an dem beide

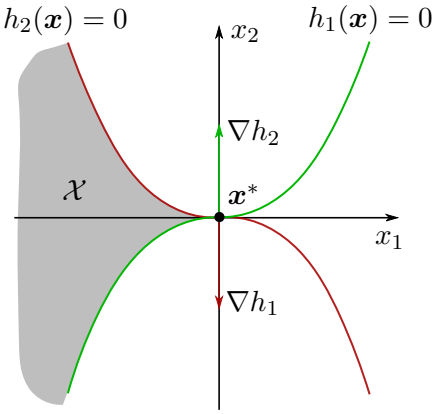


Abb. 4.16: Illustration des beispielhaften Optimierungsproblems (4.133)

Ungleichungsbeschränkungen aktiv sind, dargestellt. Die Gradienten an $\boldsymbol{x}^*$ ergeben sich zu

$$\nabla J(\boldsymbol{x}^*) = \begin{bmatrix} -1 & 0 \end{bmatrix} , \quad \nabla h_1(\boldsymbol{x}^*) = \begin{bmatrix} 0 & -1 \end{bmatrix} , \quad \nabla h_2(\boldsymbol{x}^*) = \begin{bmatrix} 0 & 1 \end{bmatrix} , \tag{4.134}$$

woraus die KKT-Teilbedingung (4.132a)

$$\nabla_{\boldsymbol{x}} L(\boldsymbol{x}^*, \boldsymbol{\mu}^*) = \begin{bmatrix} -1 & 0 \end{bmatrix} + \mu_1^* \begin{bmatrix} 0 & -1 \end{bmatrix} + \mu_2^* \begin{bmatrix} 0 & 1 \end{bmatrix} = \mathbf{0} \tag{4.135}$$

resultiert. Allerdings besitzt diese Gleichung keine Lösung, da die LICQ-Bedingung in $\boldsymbol{x}^*$ verletzt ist. Somit können die KKT-Bedingungen in Spezialfällen ggf. nicht herangezogen oder nur in Kombination mit relaxierten Beschränkungsqualifikationen genutzt werden.

Optimalitätsbedingungen 2. Ordnung

Die notwendigen Optimalitätsbedingungen 2. Ordnung lassen sich wie folgt zusammenfassen:

Satz 4.16: Notwendige Optimalitätsbedingungen 2. Ordnung

Sei $\boldsymbol{x}^*$ ein lokales Minimum des Problems (4.126), welches die LICQ-Forderung gemäß Satz 4.14 sowie die KKT-Bedingungen (4.132) mit den zugehörigen Lagrange-Multiplikatoren $\boldsymbol{\lambda}^*$ und $\boldsymbol{\mu}^*$ erfüllt. Falls J, g_i und h_i zweimal stetig differenzierbar sind, dann gilt zusätzlich

$$\boldsymbol{s}^T \nabla^2_{\boldsymbol{xx}} L(\boldsymbol{x}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*) \boldsymbol{s} \geq 0 \quad \forall \boldsymbol{s} \in \mathcal{C}(\boldsymbol{x}^*, \boldsymbol{\mu}^*) \tag{4.136}$$

mit

$$\mathcal{C}(\boldsymbol{x}^*, \boldsymbol{\mu}^*) = \boldsymbol{s} \in \mathbb{R}^n : \begin{cases} \nabla g_i(\boldsymbol{x}^*)\boldsymbol{s} & = 0, \quad i = 1, \ldots, p\,, \\ \nabla h_i(\boldsymbol{x}^*)\boldsymbol{s} & = 0, \quad i \in \mathcal{A}(\boldsymbol{x}^*), \mu_i^* > 0\,, \\ \nabla h_i(\boldsymbol{x}^*)\boldsymbol{s} & \leq 0, \quad i \in \mathcal{A}(\boldsymbol{x}^*), \mu_i^* = 0\,. \end{cases} \tag{4.137}$$

Die notwendige Bedingung (4.136) hat große Ähnlichkeit mit dem unbeschränkten Fall entsprechend Satz 4.4, bei dem die Hesse-Matrix $\nabla^2 J(\boldsymbol{x}^*)$ positiv semi-definit sein musste. Im vorliegenden Fall mit Beschränkungen wird diese Bedingung auf die Menge $\mathcal{C}(\boldsymbol{x}^*, \boldsymbol{\mu}^*)$ eingegrenzt. Die Erweiterung als hinreichende Bedingung lautet wie folgt:

Satz 4.17: Hinreichende Optimalitätsbedingungen 2. Ordnung

Sei $\boldsymbol{x}^*$ ein lokales Minimum des Problems (4.126), welches die LICQ-Forderung gemäß Satz 4.14 sowie die KKT-Bedingungen (4.132) mit den zugehörigen Lagrange-Multiplikatoren $\boldsymbol{\lambda}^*$ und $\boldsymbol{\mu}^*$ erfüllt. Falls J, g_i und h_i zweimal stetig differenzierbar sind und zudem

$$\boldsymbol{s}^T \nabla^2_{\boldsymbol{xx}} L(\boldsymbol{x}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*) \boldsymbol{s} > 0 \quad \forall \boldsymbol{s} \in \mathcal{C}(\boldsymbol{x}^*, \boldsymbol{\mu}^*), \quad \boldsymbol{s} \neq \boldsymbol{0} \tag{4.138}$$

mit $\mathcal{C}(\boldsymbol{x}^*, \boldsymbol{\mu}^*)$ entsprechend Definition (4.137) gilt, dann ist $\boldsymbol{x}^*$ ein striktes lokales Minimum.

Für konvexe Optimierungsprobleme kann darüber hinaus auf die Auswertung der Optimalitätsbedingungen 2. Ordnung verzichtet werden, da hier bereits die KKT-Bedingungen 1. Ordnung entsprechend Satz 4.15 notwendig sowie hinreichend sind. Dies lässt sich wie folgt zusammenfassen:

Satz 4.18: Hinreichende Optimalitätsbedingungen für konvexe Probleme

Gegeben sei in konvexes Optimierungsproblem entsprechend Definition 4.10. Falls ein Punkt $\boldsymbol{x}^*$ mit zugehörigen Lagrange-Multiplikatoren $\boldsymbol{\lambda}^*$ und $\boldsymbol{\mu}^*$ die KKT-Bedingnungen entsprechend Satz 4.15 erfüllt, dann ist $\boldsymbol{x}^*$ ein globales Minimum des Problems aus Definition 4.10.

Nachfolgend werden einige ausgewählte numerische Optimierungsverfahren zur Lösung des beschränkten Problems gemäß Definition 4.1 diskutiert, da eine geschlossene analytische Lösung basierend auf den obigen Optimalitätsbedingungen[1] nur in den seltensten Fällen (mit zielführendem Aufwand) möglich scheint.

4.3.2 Methode der aktiven Beschränkungen

Für die Methode der aktiven Beschränkungen (*active set method*) wird ein quadratisches Optimierungsproblem der Form

$$\min_{\boldsymbol{x}\in\mathcal{X}} \quad J(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^T\boldsymbol{W}\boldsymbol{x} + \boldsymbol{w}^T\boldsymbol{x} + c\,, \tag{4.139a}$$

$$\text{u. d. Nb.} \quad \boldsymbol{A}\boldsymbol{x} = \boldsymbol{a}\,, \tag{4.139b}$$

$$\boldsymbol{B}\boldsymbol{x} \leq \boldsymbol{b} \tag{4.139c}$$

mit $\boldsymbol{A} \in \mathbb{R}^{p\times n}$ und $\boldsymbol{a} \in \mathbb{R}^p$ zur Modellierung von linearen Gleichungsbeschränkungen sowie $\boldsymbol{B} \in \mathbb{R}^{q\times n}$ und $\boldsymbol{b} \in \mathbb{R}^q$ zur Abbildung von linearen Ungleichungsbeschränkungen angenommen. Ferner seien $\boldsymbol{W} \in \mathbb{R}^{n\times n}$ eine positiv definite Gewichtungsmatrix , $\boldsymbol{w} \in \mathbb{R}^n$ ein Vektor und $\boldsymbol{c} \in \mathbb{R}$ eine weitere Konstante. Gegenüber der vorherigen, allgemeinen Notation des Optimierungsproblems mit Beschränkungen stellt (4.139) eine Einschränkung dar – allerdings wird mit der sequentiellen quadratischen Programmierung aus Kap. 4.3.3 noch ein Verfahren vorgestellt, welches allgemeine, nichtlineare Optimierungsprobleme auf die Form (4.139) zurückführt bzw. approximiert.

Die Grundidee der Methode der aktiven Beschränkungen besteht darin, in jeder Iteration k eine Anzahl von aktiven Ungleichungsbeschränkungen

$$\boldsymbol{B}_{\mathcal{A}}\boldsymbol{x}[k] = \boldsymbol{b}_{\mathcal{A}} \tag{4.140}$$

zu bestimmen, die auch als Arbeitsmenge $\mathcal{A}$ bezeichnet wird (siehe Definition 4.11)[2]. Hierfür reduziert sich das ursprüngliche Problem (4.139) auf eine Optimierungsaufgabe, welche ausschließlich Gleichungsnebenbedingungen umfasst:

$$\min_{\boldsymbol{x}\in\mathcal{X}} \quad J(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^T\boldsymbol{W}\boldsymbol{x} + \boldsymbol{w}^T\boldsymbol{x} + c\,, \tag{4.141a}$$

$$\text{u. d. Nb.} \quad \begin{bmatrix}\boldsymbol{A}\\ \boldsymbol{B}_{\mathcal{A}}\end{bmatrix}\boldsymbol{x} = \begin{bmatrix}\boldsymbol{a}\\ \boldsymbol{b}_{\mathcal{A}}\end{bmatrix}. \tag{4.141b}$$

Hierfür kann entsprechend Definition 4.12 die Lagrange-Funktion gebildet werden:

$$L(\boldsymbol{x},\boldsymbol{\lambda},\boldsymbol{\mu}) = \frac{1}{2}\boldsymbol{x}^T\boldsymbol{W}\boldsymbol{x} + \boldsymbol{w}^T\boldsymbol{x} + c + \boldsymbol{\lambda}^T(\boldsymbol{A}\boldsymbol{x}-\boldsymbol{a}) + \boldsymbol{\mu}^T(\boldsymbol{B}_{\mathcal{A}}\boldsymbol{x}-\boldsymbol{b}_{\mathcal{A}}). \tag{4.142}$$

Daraus folgen die notwendigen Bedingungen 1. Ordnung (siehe Satz 4.15) und Verwendung von

[1] Welche eine beliebige Anzahl an nichtlinearen Gleichungen und Ungleichungen in $\boldsymbol{x}$, $\boldsymbol{\lambda}$ und $\boldsymbol{\mu}^*$ aufweisen können.

[2] Zur übersichtlicheren Darstellbarkeit wird die algorithmische Umsetzung unter Verwendung des Iterationsindexes k erst am Ende diesen Unterkapitels eingeführt.

Anhang A.2:

$$\nabla_{\boldsymbol{x}} L = \boldsymbol{W}\boldsymbol{x} + \boldsymbol{w} + \boldsymbol{A}^T\boldsymbol{\lambda} + \boldsymbol{B}_{\mathcal{A}}^T\boldsymbol{\mu} = \boldsymbol{0}\,, \tag{4.143a}$$

$$\nabla_{\boldsymbol{\lambda}} L = \boldsymbol{g}(\boldsymbol{x}) = \boldsymbol{A}\boldsymbol{x} - \boldsymbol{a} = \boldsymbol{0}\,, \tag{4.143b}$$

$$\nabla_{\boldsymbol{\mu}} L = \boldsymbol{h}_{\mathcal{A}}(\boldsymbol{x}) = \boldsymbol{B}_{\mathcal{A}}\boldsymbol{x} - \boldsymbol{b}_{\mathcal{A}} = \boldsymbol{0}\,. \tag{4.143c}$$

Hierbei entspricht $\boldsymbol{h}_{\mathcal{A}}$ den aktiven Ungleichungsbeschränkungen (UB) im jeweiligen Iterationsschritt k. Da diese Bedingungen für das Problem (4.139) linear sind, kann eine lineares Gleichungssystem aufgestellt werden:

$$\begin{bmatrix} \boldsymbol{W} & \boldsymbol{A}^T & \boldsymbol{B}_{\mathcal{A}}^T \\ \boldsymbol{A} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{B}_{\mathcal{A}} & \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{\lambda} \\ \boldsymbol{\mu} \end{bmatrix} = \begin{bmatrix} -\boldsymbol{w} \\ \boldsymbol{a} \\ \boldsymbol{b}_{\mathcal{A}} \end{bmatrix}. \tag{4.144}$$

Zur algorithmischen Umsetzung werden

$$\boldsymbol{x} = \boldsymbol{x}[k+1] = \boldsymbol{x}[k] + \boldsymbol{s}[k], \quad \boldsymbol{\lambda} = \boldsymbol{\lambda}[k+1], \quad \boldsymbol{\mu} = \boldsymbol{\mu}[k+1] \tag{4.145}$$

eingeführt, wobei angenommen wird, dass $\boldsymbol{x}[0] \in \mathcal{X}$ gilt, der Startpunkt also alle Gleichungs- und Ungleichungsbeschränkungen erfülle. Einsetzen von (4.145) in (4.144) unter Verwendung von (4.143) ergibt dann:

$$\underbrace{\begin{bmatrix} \boldsymbol{W} & \boldsymbol{A}^T & \boldsymbol{B}_{\mathcal{A}}^T \\ \boldsymbol{A} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{B}_{\mathcal{A}} & \boldsymbol{0} & \boldsymbol{0} \end{bmatrix}}_{\tilde{\boldsymbol{A}}} \underbrace{\begin{bmatrix} \boldsymbol{s}[k] \\ \boldsymbol{\lambda}[k+1] \\ \boldsymbol{\mu}[k+1] \end{bmatrix}}_{\tilde{\boldsymbol{x}}} = \underbrace{\begin{bmatrix} -\boldsymbol{w} - \boldsymbol{W}\boldsymbol{x}[k] \\ \boldsymbol{0} \\ \boldsymbol{0} \end{bmatrix}}_{\tilde{\boldsymbol{b}}}. \tag{4.146}$$

Obige Normalengleichung kann unter Verwendung der in Kap. 3.1.4 diskutierten numerischen Verfahren gelöst werden und der Lösungsvektor wird zur Berechnung des nächsten Iterationsschritts herangezogen. In jedem Iterationsschritt muss zudem geprüft werden, ob sich die Menge der aktiven Ungleichungsbeschränkungen (*active set*) verändert. Der resultierende Algorithmus ist in Algorithmus 4.2 zusammengefasst.

Die Iterationsschritte und Fallunterscheidungen in Algorithmus 4.2 lassen sich wie folgt interpretieren: So lange noch kein Optimum entsprechend Satz 4.18 gefunden wurde, wird die Suchrichtung $\boldsymbol{s}[k]$ geprüft. Ist diese Null und mindestens ein Lagrange-Multiplikator aus der aktiven Menge $\mathcal{A}$ $\mu_i[k+1] < 0$, dann kann das Optimierungsergebnis weiter verbessert werden, wenn eine aktive UB mit negativem Lagrange-Multiplikator aus $\mathcal{A}$ entfernt wird. Hierbei wird zweckmäßigerweise das kleinste $\mu_i[k+1]$ gewählt, da dieses die stärkste Verbesserung hinsichtlich des Kostenwerts verursacht (*Inaktivierungsschritt*). Falls $\boldsymbol{s}[k] \neq \boldsymbol{0}$, ist zu prüfen, ob die neue Position $\boldsymbol{x}[k+1] = \boldsymbol{x}[k] + \boldsymbol{s}[k]$ innerhalb von $\mathcal{X}$ liegt. Trifft dies zu, kann der Iterationsschritt durch entsprechende Anpassung $\boldsymbol{x}[k+1]$ abgeschlossen werden.

Ist hingegen $\{\boldsymbol{x}[k] + \boldsymbol{s}[k]\} \notin \mathcal{X}$, muss ähnlich zu den bereits bekannten Suchlinienverfahren für unbeschränkte Probleme eine Schrittweite $\alpha[k] < 1$ eingeführt werden:

$$\min_{\alpha[k]>0} \quad J(\boldsymbol{x}[k+1] = \boldsymbol{x}[k] + \alpha[k]\boldsymbol{s}[k])\Big|_{\boldsymbol{s}[k]}\,, \tag{4.147a}$$

$$\text{u. d. Nb.} \quad \boldsymbol{x}[k+1] \in \mathcal{X}\,. \tag{4.147b}$$

Algorithmus 4.2 Methode der aktiven Beschränkungen

Initialisierung:

$\boldsymbol{x}_0 = \boldsymbol{x}[k=0] \in \mathcal{X}$ ▷ Startlösung

$\boldsymbol{B}_{\mathcal{A}}[0], \boldsymbol{b}_{\mathcal{A}}[0]$ ▷ Bestimme Anfangsmenge aktiver UB ($\mathcal{A}[0]$)

$k = 0$ ▷ Startindex

Iterieren:

for $k = 1, 2, \ldots$ **do**

  Bestimme $\boldsymbol{s}[k]$, $\boldsymbol{\lambda}[k+1]$ und $\boldsymbol{\mu}[k+1]$ aus (4.146)

  switch $\{\boldsymbol{s}[k], \boldsymbol{\mu}[k+1]\}$ **do** ▷ Fallunterscheidung

    case $\{\boldsymbol{s}[k] = \boldsymbol{0} \quad \wedge \quad \mu_i[k+1] \geq 0,\ i \in \mathcal{A}[k]\}$

      $\boldsymbol{x}^* \leftarrow \boldsymbol{x}[k]$ ▷ Optimale Lösung gemäß KKT

    case $\{\boldsymbol{s}[k] = \boldsymbol{0} \quad \wedge \quad \mu_i[k+1] < 0,\ i \in \mathcal{A}[k]\}$

      Entferne aktive UB mit kleinstem $\mu_i[k+1]$ aus $\mathcal{A}[k+1]$

      $\boldsymbol{x}[k+1] \leftarrow \boldsymbol{x}[k]$

      $k \leftarrow k+1$

    case $\boldsymbol{s}[k] \neq \boldsymbol{0}$

      switch $\boldsymbol{x}[k+1]$ **do**

        case $\boldsymbol{x}[k+1] \in \{\mathcal{X} | \boldsymbol{x}[k+1] = \boldsymbol{x}[k] + \boldsymbol{s}[k]\}$

          $\boldsymbol{x}[k+1] \leftarrow \boldsymbol{x}[k] + \boldsymbol{s}[k]$

          $k \leftarrow k+1$

        case $\boldsymbol{x}[k+1] \notin \{\mathcal{X} | \boldsymbol{x}[k+1] = \boldsymbol{x}[k] + \boldsymbol{s}[k]\}$

          Bestimme $\alpha[k]$, sodass $\boldsymbol{x}[k+1] \in \{\mathcal{X} | \boldsymbol{x}[k+1] = \boldsymbol{x}[k] + \alpha[k]\boldsymbol{s}[k]\}$

          Ergänze neue aktive Beschränkung zu $\mathcal{A}[k+1]$

          $\boldsymbol{x}[k+1] \leftarrow \boldsymbol{x}[k] + \alpha[k]\boldsymbol{s}[k]$

          $k \leftarrow k+1$

end for

Gemäß (4.139) muss gelten:

$$\boldsymbol{B}\boldsymbol{x} \leq \boldsymbol{b} \quad \Leftrightarrow \quad \boldsymbol{B}_i\boldsymbol{x} \leq b_i \quad \forall i = 1, \ldots, q. \tag{4.148}$$

Hierbei ist $\boldsymbol{B}_i$ der i-te Zeilenvektor von $\boldsymbol{B}$ sowie b_i der i-te Eintrag in $\boldsymbol{b}$. Um eine geeignete Schrittweite $\alpha[k]$ zu bestimmen, müssen allerdings nur die bisher inaktiven Beschränkungen betrachtet werden, welche nicht Teil von $\mathcal{A}[k]$ sind[1]:

$$\boldsymbol{B}_i(\boldsymbol{x}[k] + \alpha[k]\boldsymbol{s}[k]) \leq b_i \quad \forall i \notin \mathcal{A}[k]. \tag{4.149}$$

Für die bisher nicht aktive Beschränkung $i \notin \mathcal{A}[k]$ muss daher gelten:

$$\alpha[k] \leq \frac{b_i - \boldsymbol{B}_i\boldsymbol{x}[k]}{\boldsymbol{B}_i\boldsymbol{s}[k]}. \tag{4.150}$$

Zudem sind nur diejenigen bisher inaktiven UBs von Interesse bei denen

$$\boldsymbol{B}_i\boldsymbol{s}[k] > 0 \tag{4.151}$$

vorliegt, da nur hier die Suchrichtung $\boldsymbol{s}[k]$ im gegebenen Iterationsschritt in Richtung der Ungleichungsbeschränkung zeigt. Die größtmöglich wählbare Schrittweite in ergibt sich dann zu:

$$\alpha[k] = \min\left\{1, \quad \min_{i \notin \mathcal{A}[k], \boldsymbol{B}_i\boldsymbol{s}[k] > 0} \frac{b_i - \boldsymbol{B}_i\boldsymbol{x}[k]}{\boldsymbol{B}_i\boldsymbol{s}[k]}\right\}. \tag{4.152}$$

Die hinsichtlich $\alpha[k]$ dann limitierende Ungleichungsbeschränkung $i \notin \mathcal{A}[k]$ wird dann zur Arbeitsmenge $\mathcal{A}[k+1]$ hinzugefügt (*Aktivierungsschritt*).

Die Methode der aktiven Beschränkungen kann als verschachtelte Variante des Newton-Algorithmus aus Kap. 4.2.3 verstanden werden, insbesondere die Bestimmung des Lösungsvektors gemäß (4.146) erfolgt analog zur Bestimmung der Newton-Richtung für unbeschränkte Probleme. Der wesentliche Mechanismus der Methode der aktiven Beschränkungen ist die Anpassung der aktiven Menge $\mathcal{A}$ in jedem Iterationsschritt, sodass sich die Gleichungsstruktur (4.146), welche zur Berechnung der Newton-Richtung herangezogen wird, zwischen den Iterationsschritten verändern kann. Zudem kann in jedem Iterationsschritt nur jeweils eine Ungleichungsbeschränkung (de-)aktiviert werden, sodass die Methode der aktiven Beschränkungen für hoch-dimensionale Probleme als langsam gilt. Allerdings kann gezeigt werden, dass der Ansatz für strikt konvexe quadratische Probleme in einer endlichen Anzahl von Iterationen ins Optimum konvergiert.

4.3.3 Sequentielle quadratische Programmierung

Gegenüber der Methode der aktiven Beschränkungen, welche sich explizit auf quadratische Probleme bezog, soll nun wieder der allgemeine Optimierungsfall

$$\min_{\boldsymbol{x} \in \mathcal{X}} \quad J(\boldsymbol{x}), \tag{4.153a}$$

$$\text{u. d. Nb.} \quad g_i(\boldsymbol{x}) = \boldsymbol{0}, \quad i = 1, \ldots, p, \tag{4.153b}$$

$$h_i(\boldsymbol{x}) \leq \boldsymbol{0}, \quad i = 1, \ldots, q \tag{4.153c}$$

[1] Da ein quadratisches Problem der Klasse der konvexen Probleme entspricht, bleiben alle bereits aktiven UB in $\mathcal{A}[k]$ für jedes $0 < \alpha[k] < 1$ erfüllt.

betrachtet werden. Hierbei können J, g_i und h_i beliebige (nichtlineare) Funktionen sein. Kernidee der sequentiellen quadratischen Programmierung (SQP) ist es, (4.153) am jeweiligen Iterationspunkt $\boldsymbol{x}[k]$ durch ein quadratisches Problem zu approximieren. Hierfür wird

$$\boldsymbol{x}[k+1] = \boldsymbol{x}[k] + \boldsymbol{s}[k] \tag{4.154}$$

angesetzt und die Kostenfunktion durch eine Taylor-Reihe 2. Ordnung sowie die Nebenbedigung durch Taylor-Reihen 1. Ordnung vereinfacht:

$$\min_{\boldsymbol{s}[k]} \quad \frac{1}{2}\boldsymbol{s}[k]^T(\nabla^2 J)(\boldsymbol{x}[k])\boldsymbol{s}[k] + (\nabla J)(\boldsymbol{x}[k])\boldsymbol{s}[k] + J(\boldsymbol{x}[k])\,, \tag{4.155a}$$

$$\text{u. d. Nb.} \quad \nabla g_i(\boldsymbol{x}[k])\boldsymbol{s}[k] + g_i(\boldsymbol{x}[k]) = \boldsymbol{0}, \quad \forall i = 1,\ldots,p\,, \tag{4.155b}$$

$$\nabla h_i(\boldsymbol{x}[k])\boldsymbol{s}[k] + h_i(\boldsymbol{x}[k]) \leq \boldsymbol{0}, \quad \forall i = 1,\ldots,q\,. \tag{4.155c}$$

Das derart gefundene, approximierte quadratische Problem entspricht der Form (4.139). Für einen Iterationsschritt kann daher ein Verfahren zur Lösung quadratischer Probleme herangezogen werden, wie beispielsweise die zuvor diskutierte Methode der aktiven Beschränkungen. Dies wird solange iteriert bis $\boldsymbol{s}[k] \approx \boldsymbol{0}$ vorliegt, da dann (4.155) dem ursprünglichen Problem (4.153) entspricht. Darüber hinaus kann gezeigt werden, dass für geeignete Startwerte $\{\boldsymbol{x}_0, \boldsymbol{\lambda}_0, \boldsymbol{\mu}_0\}$, welche sich in einer hinreichend kleinen Umgebung von $\{\boldsymbol{x}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*\}$ befinden, der SQP-Ansatz quadratisch in das (lokale) Optimum konvergiert (sogenanntes *lokales SQP*). Der SQP-Algorithmus lässt sich wie folgt zusammenfassen:

Algorithmus 4.3 Sequentielle quadratische Programmierung

Initialisierung:

1: $\boldsymbol{x}_0 = \boldsymbol{x}[k=0]$ ▷ Startlösung

2: $\boldsymbol{\lambda}_0 = \boldsymbol{\lambda}[k=0], \boldsymbol{\mu}_0 = \boldsymbol{\mu}[k=0]$ ▷ Startwerte Lagrange-Multiplikatoren

3: $k = 0$ ▷ Startindex

4: ε ▷ Abbruchkriterium

Iterieren:

5: **while** $\|\boldsymbol{s}[k]\| \geq \varepsilon$ **do**

6: Berechne $J(\boldsymbol{x}[k]), (\nabla J)(\boldsymbol{x}[k]), (\nabla^2 J)(\boldsymbol{x}[k]), \boldsymbol{g}(\boldsymbol{x}[k]), (\nabla g)(\boldsymbol{x}[k]), \boldsymbol{h}(\boldsymbol{x}[k]), (\nabla h)(\boldsymbol{x}[k])$

7: Berechne $\{\boldsymbol{s}[k], \boldsymbol{\lambda}[k], \boldsymbol{\mu}[k]\}$ durch Lösen von (4.155)

8: $\boldsymbol{x}[k+1] \leftarrow \boldsymbol{x}[k] + \boldsymbol{s}[k]$

9: $\{\boldsymbol{\lambda}[k+1], \boldsymbol{\mu}[k+1]\} \leftarrow \{\boldsymbol{\lambda}[k], \boldsymbol{\mu}[k]\}$

10: $k \leftarrow k+1$

11: **end while**

Für die Approximation des ursprünglichen Problems wird in jedem Iterationsschritt die Hesse-Matrix benötigt. In vielen Anwendungen ist diese nicht analytisch bekannt und müsste aufwendig über finite Differenzen approximiert werden, was ggf. auch zu einer indefiniten Hesse-Matrix führen kann (insbesondere, falls der Abstand zu $\{\boldsymbol{x}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*\}$ noch groß ist). Analog zur Quasi-Newton-Methode für unbeschränkte Optimierungsprobleme wird daher in der Praxis die Hesse-Matrix häufig iterativ approximiert. Dies ist als *modifizierte BFGS Methode* oder auch

gedämpfte BFGS Methode bekannt[1]:

$$(\nabla^2 J)(\boldsymbol{x}[k]) \approx \boldsymbol{H}[k] \tag{4.156a}$$

$$\boldsymbol{H}[k+1] = \boldsymbol{H}[k] + \frac{\boldsymbol{q}[k]\boldsymbol{q}^T[k]}{\boldsymbol{q}^T[k]\boldsymbol{d}[k]} - \frac{\boldsymbol{H}[k]\boldsymbol{d}[k]\boldsymbol{d}^T[k]\boldsymbol{H}[k]}{\boldsymbol{d}^T[k]\boldsymbol{H}[k]\boldsymbol{d}[k]} \tag{4.156b}$$

mit

$$\boldsymbol{d}[k] = \boldsymbol{x}[k+1] - \boldsymbol{x}[k] \tag{4.157a}$$

$$\boldsymbol{y}[k] = (\nabla J)(\boldsymbol{x}[k+1]) - (\nabla J)(\boldsymbol{x}[k]) \tag{4.157b}$$

$$\theta[k] = \begin{cases} 1, & \boldsymbol{d}^T[k]\boldsymbol{y}[k] \geq \frac{1}{5}\boldsymbol{d}^T[k]\boldsymbol{H}[k]\boldsymbol{d}[k] \\ \frac{\frac{4}{5}\boldsymbol{d}^T[k]\boldsymbol{H}[k]\boldsymbol{d}[k]}{\boldsymbol{d}^T[k]\boldsymbol{H}[k]\boldsymbol{d}[k] - \boldsymbol{d}^T[k]\boldsymbol{y}[k]}, & \boldsymbol{d}^T[k]\boldsymbol{y}[k] < \frac{1}{5}\boldsymbol{d}^T[k]\boldsymbol{H}[k]\boldsymbol{d}[k] \end{cases} \tag{4.157c}$$

$$\boldsymbol{q}[k] = \theta[k]\boldsymbol{y}[k] + (1-\theta[k])\boldsymbol{H}[k]\boldsymbol{d}[k]. \tag{4.157d}$$

Die obige Korrekturvorschrift garantiert, dass $\boldsymbol{H}[k+1]$ symmetrisch und positiv definit bleibt, sofern dies zuvor auf $\boldsymbol{H}[k]$ zutrifft. Daher muss während der Initialisierung von Algorithmus 4.3 sichergestellt werden, dass $\boldsymbol{H}[0]$ symmetrisch und positiv definit ist. Ähnlich wie auch schon bei der Quasi-Newton-Methode verschlechtert sich die Konvergenz des SQP-Verfahren bei Anwendung der Hesse-Approximation hin zur superlinearen Konvergenz in der Nähe von $\{\boldsymbol{x}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*\}$.

Globalisierung

Der obige lokale SQP-Ansatz hat die folgenden, wesentlichen Nachteile:

(a) Konvergiert nur, falls $\{\boldsymbol{x}_0, \boldsymbol{\lambda}_0, \boldsymbol{\mu}_0\}$ in hinreichender Nähe von $\{\boldsymbol{x}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*\}$.

(b) Die Folge $\{\boldsymbol{x}[k]\}$ erfüllt aufgrund der Linearisierung der Nebenbedingung i. d. R. diese nicht exakt, also $\boldsymbol{x}[k] \notin \mathcal{X}$.

Zur Kompensation dieser Nachteile kann eine *Globalisierung* der SQP-Methode vorgenommen werden, welche im Kern die Einführung einer Schrittweite vorsieht:

$$\boldsymbol{x}[k+1] = \boldsymbol{x}[k] + \alpha[k]\boldsymbol{s}[k]. \tag{4.158}$$

Gegenüber dem unbeschränkten Fall z. B. bei Newton-Verfahren, muss das resultierende Liniensuchverfahren allerdings modifiziert werden: Bei beschränkten Problemen muss $\alpha^*[k]$ nicht nur die Kostenfunktion $J(\boldsymbol{x})$ minimieren, sondern auch die Verletzung der Nebenbedingungen $\boldsymbol{g}(\boldsymbol{x})$ und $\boldsymbol{h}(\boldsymbol{x})$. Um diese ggf. auch divergierenden Ziele abzubilden, wird eine *Bewertungsfunktion* oder auch *Straffunktion* eingeführt:

$$P(\boldsymbol{x}, \sigma) = J(\boldsymbol{x}) + \sigma \left[\sum_{i=1}^{p} |g_i(\boldsymbol{x})| + \sum_{i=1}^{q} \max\{0, h_i(\boldsymbol{x})\} \right]. \tag{4.159}$$

[1] Es gilt zu beachten, dass gegenüber der Quasi-Newton-Methode im SQP-Kontext direkt die Hesse-Matrix approximiert wird und nicht ihre Inverse. Trotz dieses Unterschieds hat sich in der Literatur der gleiche Formelbuchstabe $\boldsymbol{H}$ für beide Fälle etabliert.

Eine Minimierung dieser Funktion führt zum gleichen Lösungspunkt $\boldsymbol{x}^*$ wie (4.153), sofern die gefundene Lösung $\{\boldsymbol{x}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*\}$ die Bedingungen aus Satz 4.17 erfüllt und zudem

$$\sigma > |\lambda_i|,\ i = 1, \ldots, p \quad \wedge \quad \sigma > |\mu_i|,\ i = 1, \ldots, q \tag{4.160}$$

gilt. Ist dies der Fall, wird (4.160) auch *exakte Straffunktion* genannt. Die Gewichtung σ wird daher i. A. in jedem Iterationsschritt angepasst, um obige Eigenschaften zu erfüllen. Das resultierende Liniensuchproblem lautet dann

$$\min_{\alpha[k] \in \mathbb{R}} P(\boldsymbol{x}[k] + \alpha[k]\boldsymbol{s}[k], \sigma)\Big|_{\boldsymbol{x}[k], \boldsymbol{s}[k]}, \tag{4.161}$$

welches anschließend mit einem der in Kap. 4.2.2 diskutierten Verfahren gelöst werden kann.

4.3.4 Methode der Straf- und Barrierefunktionen

Mit Hilfe von Straf- und Barrierefunktionen lassen sich beschränkte in unbeschränkte Optimierungsprobleme überführen, welche dann mit den in Kap. 4.2 beschriebenen Methoden gelöst werden können.

Straffunktionen

Ähnlich wie bei der Globalisierung des SQP-Verfahrens, können Strafterme ganz allgemein genutzt werden, um das beschränkte Probleme (4.153) in ein unbeschränktes Problem zu überführen[1]:

$$\min_{\boldsymbol{x} \in \mathbb{R}} \quad P(\boldsymbol{x}, \tau) = J(\boldsymbol{x}) + \frac{1}{\tau}\left[\sum_{i=1}^{p} \phi_g(g_i(\boldsymbol{x})) + \sum_{i=1}^{q} \phi_h(h_i(\boldsymbol{x}))\right]. \tag{4.162}$$

Die Funktionen ϕ_g und ϕ_h stellen die sog. *(äußeren) Straffunktionen* dar, die Beschränkungen von außen bestrafen und somit ein Verletzten der Nebenbedingungen grundsätzlich erlauben. Typische Straffunktionen sind Potenzfunktionen der Form

$$\phi_g(g_i(\boldsymbol{x})) = |g_i(\boldsymbol{x})|^r, \tag{4.163a}$$

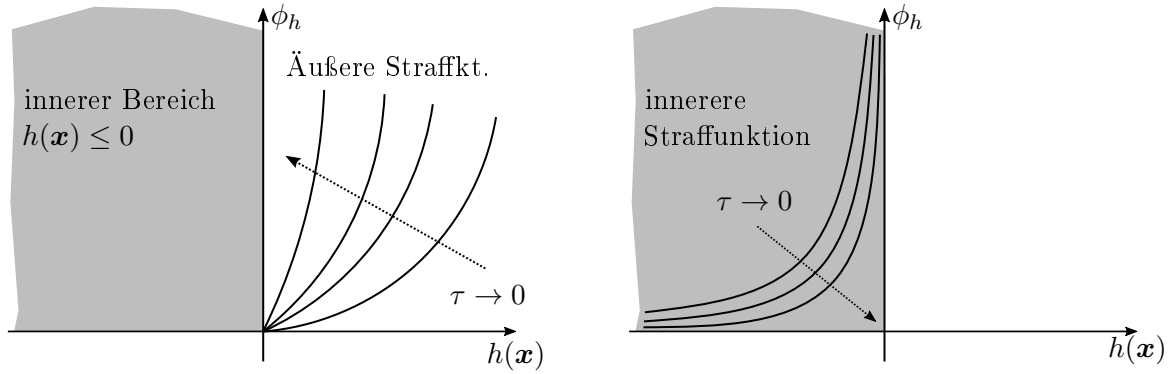
$$\phi_h(h_i(\boldsymbol{x})) = (\max\{0, h_i(\boldsymbol{x})\})^r \tag{4.163b}$$

mit $r \in \{\mathbb{R} | r \geq 1\}$. In Abb. 4.17a wird die Straffunktion unter Variation von τ veranschaulicht. Die Lösung $\boldsymbol{x}^*(\tau)$ des neuen unbeschränkten Problems (4.162) wird i. d. R. von der optimalen Lösung $\boldsymbol{x}^*$ des Originalproblems (4.153) abweichen, da der Strafterm die originale Kostenfunktion verfälscht. Wird hingegen eine abnehmende Folge $\{\tau[k]\}$ des Strafparameters eingeführt, nähert sich die unbeschränkte Kostenfunktion aufgrund der zunehmenden Bestrafung der Beschränkungsverletzung der beschränkten Originalfunktion an:

$$\lim_{k \to \infty} \boldsymbol{x}^*(\tau) = \boldsymbol{x}^*. \tag{4.164}$$

[1] Also unabhängig von einer weiteren Approximation der Kostenfunktion und/oder der Nebenbedingung im Sinne des SQP-Verfahrens.

Ein bekannter Nachteil des Verfahrens ist allerdings, dass das Problem (4.162) für $\tau \to 0$ zunehmend schlecht konditioniert wird, was eine genaue Approximation der optimalen Lösung in der Praxis schwierig macht.



(a) (Äußere) Straffunktion (b) Barrierefunktion / Innere Straffunktion

Abb. 4.17: Veranschaulichung zur Methode der Straf- und Barrierefunktionen

Barrierefunktionen

Barrierefunktionen, auch *innere Straffunktionen* genannt, bestrafen die Annäherung an die Beschränkungsgrenze und gehen für $h_i(\boldsymbol{x}) \to 0$ gegen Unendlich, siehe hierzu auch Abb. 4.17b. Typische Beispiele sind:

$$\phi_h(h_i(\boldsymbol{x})) = \frac{-1}{h_i(\boldsymbol{x})}, \qquad \phi_h(h_i(\boldsymbol{x})) = -\ln(-h_i(\boldsymbol{x})). \tag{4.165}$$

Da diese nur im Inneren des zulässigen Bereichs definiert sind, können nur Ungleichungsbeschränkungen mittels Barrierefunktionen abgebildet werden:

$$\min_{\boldsymbol{x}\in\mathbb{R}} \quad P(\boldsymbol{x},\tau) = J(\boldsymbol{x}) + \tau \sum_{i=1}^{q} \phi_h(h_i(\boldsymbol{x})). \tag{4.166}$$

Durch sukzessive Reduktion des Strafparameters τ reduziert sich die Bestrafung des zulässigen Bereiches für $h_i(\boldsymbol{x}) < 0$, während $\phi_h \to \infty$ bei Annäherung an die Beschränkung $h_i(\boldsymbol{x}) \to 0$ erhalten bleibt. Analog zu den äußeren Straffunktionen kann es zudem zielführend sein, eine abnehmende Folge $\{\tau[k]\}$ einzuführen, damit $\lim_{k\to\infty} \boldsymbol{x}^*(\tau) = \boldsymbol{x}^*$ erzielt wird. Des Weiteren gilt es bei der Verwendung der Barrierefunktionen zu bedenken, dass die Initialisierung $\boldsymbol{x}_0$ zwingend $\boldsymbol{h}(\boldsymbol{x}_0) \leq 0$ einhalten muss, damit (4.166) überhaupt definiert ist.

4.3.5 Zusammenfassung

Die vorangegangenen Methoden zur statischen Optimierung unter Nebenbedingungen stellen nur einen kleinen Ausschnitt der verfügbaren Verfahren dar. Je nach Optimierungsklasse lassen sich weitere Verfahren aus der Literatur (z. B. [NW06][PLB12]) entnehmen und anwenden, unter anderem:

Lineare Optimierungsprobleme

- Simplex-Algorithmus
- Innere-Punkt-Verfahren (*interior point methods*)

Quadratische Optimierungsprobleme

- Verfahren der konjugierten Gradienten
- Schurkomplement-Methode

Nichtlineare Optimierungsprobleme

- Innere-Punkt-Verfahren für nichtlineare Probleme
- Primal-Dual-Active-Set-Algorithmus

Ferner existieren eine Reihe von kostenfreien sowie -pflichtigen Software-Lösungen, welche u. a. obige Verfahren einsetzen. Häufig sind entsprechende Algorithmen in vergleichsweise maschinennahen Programmiersprachen, wie C/C++ oder Fortran, umgesetzt, um eine zügige Abarbeitung der notwendigen Rechenoperationen zu gewährleisten. Umfassende Auflistungen können u. a. [Uni18][Wik18b] entnommen werden. Wesentliche Softwarepakete sind:

Toolboxen für lineare Optimierungsprobleme

- linprog: Matlab Optimization Toolbox (kostenpflichtig)
  `https://de.mathworks.com/help/optim/ug/linprog.html`
- GLPK: GNU Linear Programming Kit (frei zugänglich)
  `https://www.gnu.org/software/glpk/`

Toolboxen für quadratische Optimierungsprobleme

- quadprog: Matlab Optimization Toolbox (kostenpflichtig)
  `https://de.mathworks.com/help/optim/ug/quadprog.html`
- qpOASES (frei zugänglich)
  `https://projects.coin-or.org/qpOASES`

Toolboxen für nichtlineare Optimierungsprobleme

- fmincon: Matlab Optimization Toolbox (kostenpflichtig)
  `https://de.mathworks.com/help/optim/ug/fmincon.html`
- Ipopt (frei zugänglich)
  `https://github.com/coin-or/Ipopt`

Toolboxen für gemischt-ganzzahlige Optimierung

- intlinprog: Matlab Optimization Toolbox (kostenpflichtig)
  `https://de.mathworks.com/help/optim/ug/intlinprog.html`
- Cbc (frei zugänglich)
  `https://github.com/coin-or/Cbc`

4.4 Globale Optimierung bei nichtlinearen Problemen

Die bisher betrachteten Lösungsverfahren operieren auf lokaler Ebene, d. h. abhängig von der jeweiligen Initialisierung konvergieren diese (unter entsprechenden Voraussetzungen) in eines der umliegenden (lokalen) Minima. In vielen realen, technischen Anwendungen liegen nichtlineare Probleme vor, welche sich durch eine Vielzahl von lokalen Minima auszeichnen können. Im Sinne der jeweiligen Optimierungsaufgabe sind daher Maßnahmen zu treffen, um das Konvergieren in ungünstige lokale Optima zu verhindern bzw. idealerweise in das globale Optimum zu erreichen. Hierzu sei auf folgenden Satz verwiesen:

Satz 4.19: Globale Optimierung nichtlinearer Probleme

Im Gegensatz zur lokalen Optimierung ist die globale Optimierung ein quasi ungelöstes Problem der Mathematik: Es gibt praktisch keinerlei Methoden, bei deren Anwendung man in den meisten Fällen als Ergebnis einen Punkt erhält, der mit Sicherheit oder auch nur großer Wahrscheinlichkeit das globale Optimum darstellt.

Dieses Problem wird in Abb. 4.18 anhand der Rastrigin-Funktion verdeutlicht – durch die Vielzahl lokaler Minima ist das Optimierungsergebnis auf Basis der bisher diskutierten Verfahren maßgeblich durch die Initialisierung $\boldsymbol{x}_0$ abhängig. Ist $\boldsymbol{x}_0$ nicht in hinreichender Nähe des globalen Optimums $\boldsymbol{x}^*$, findet ein Abstieg in das lokale Minimum in direkter Umgebung des initialen Punktes statt.

Daher gilt es globale Optimierungsstrategien heranzuziehen, um zumindest zielführende Lösungspunkte zu identifizieren. Wesentliches Charakteristikum der Methoden zur globalen Optimierung ist daher, dass sie wiederholt nach einem bestimmten System lokale Minima aufsuchen bis eine gegebene Abbruchbedingung erfüllt ist. Diese Bedingung kann z. B. ein a-priori an das Problem gestelltes Mindestmaß hinsichtlich des Kostenfunktionswerts oder auch eine maximal erlaubte Iterations-/Zeitbeschränkung sein. Hierbei wird demnach die Annahme getroffen, dass in der Menge der lokalen Extrempunkte ein Lösungskandidat des globalen Optimierungsproblems enthalten ist, welcher entweder tatsächlich das globale Optimum enthält oder diesem zumindest hinsichtlich der Lösungseigenschaften hinreichend ähnlich ist.

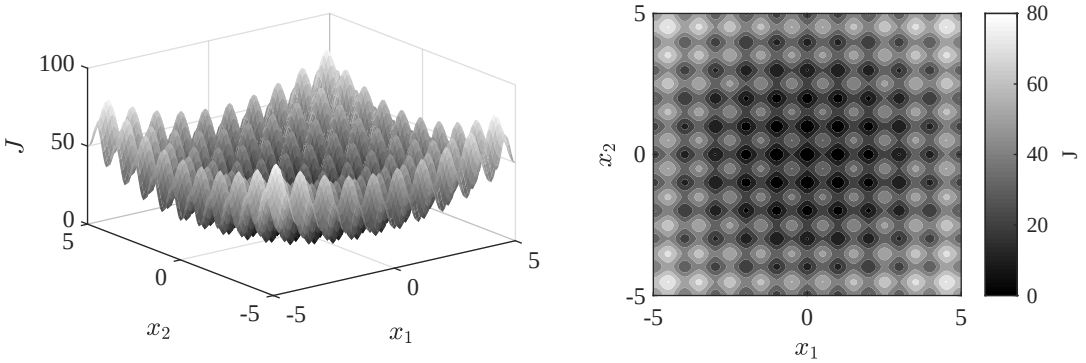


Abb. 4.18: Rastrigin-Funktion $J(\boldsymbol{x}) = An + \sum_{i=1}^{n} \left(x_i^2 - A\cos(2\pi x_i)\right)$ mit $A \in \{\mathbb{R}|A > 0\}$ als beispielhaftes Optimierungsproblem mit zahlreichen lokalen Minima

Im Folgenden werden Lösungskategorien für globale Optimierungsprobleme aufgezeigt und ausgewählte Algorithmen näher vorgestellt. Diese Auswahl ist keineswegs vollständig und wirft

lediglich ein kurzes Schlaglicht auf die Thematik – für weitere Informationen zur globalen Optimierung sei auf die entsprechende Fachliteratur (z. B. [HP95][PR02]) verwiesen. Die nachfolgend verwendete Taxonomie zur Kategorisierung der Verfahren ist zudem als nicht abschließend zu bewerten und lediglich beispielhaft. Auch hier finden sich in der Literatur umfassendere Taxonomien, dessen Studium empfehlenswert ist (z. B. [SEBB18]).

4.4.1 Deterministische Ansätze

Gittersuche (*grid search*)

Die Idee der Gittersuche (*grid search*) ist sehr einfach: Der Suchbereich $\mathcal{X}$ wird in immer kleinere Teile durch Halbierung in allen Dimensionen strukturiert. Nach der m-ten Unterteilung ist die Kantenlänge der einzelnen Boxen, der Gitterabstand, $1/(2^m)$ Mal des Originalabstandes – siehe Abb. 4.19. Das Verfahren ist offensichtlich konvergent, wenn das Gitter hinreichend fein ist und somit mindestens ein Gitterpunkt in hinreichender Nähe zum globalen Optimum ist. Die Gittersuche kann eigenständig verwendet werden, d. h. die Kostenfunktion wird an den Gitterpunkten ausgewertet und der Gitterpunkt mit den geringsten Kosten wird als Optimum definiert, oder die Gitterpunkte werden als Startwerte für lokale Optimierungsverfahren herangezogen.

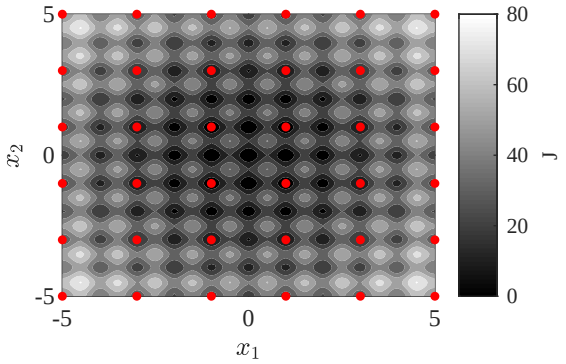


Abb. 4.19: Illustration zur Gittersuche für die Rastrigin-Funktion

Wesentlicher Nachteil der Gittersuche ist der damit verbundene Aufwand. Wird angenommen, dass jede Optimierungsvariable eines Optimierungsproblems mit n-Dimensionen m-fach gleichmäßig geteilt wird, ergeben sich $(m+1)^n$ Gitterpunkte. Für hoch-dimensionale Probleme ist das Verfahren daher praktisch nicht anzuwenden.

Verzweigung und Schranke (*branch-and-bound*)

Die Methode der Verzweigungen und Schranken (*branch-and-bound*) stellt eine Modifikation der Gittersuche dar. Hierfür wird das zulässige Gebiet $\mathcal{X}$ in mehrere Teilmengen (*branches*) aufgespalten. Mittels geeigneter Schranken (*bounds*) sollen viele suboptimale Teilbereiche von $\mathcal{X}$ frühzeitig erkannt und ausgesondert werden, sodass der zu durchsuchende Lösungsraum klein gehalten wird. Es bestehen mehrere Varianten hinsichtlich der Ausführungen von Branch- und Bound-Schritten, allen Ansätzen gemein ist allerdings, dass ein Entscheidungsbaum aus dem Verfahren resultiert – siehe hierzu Abb. 4.20.

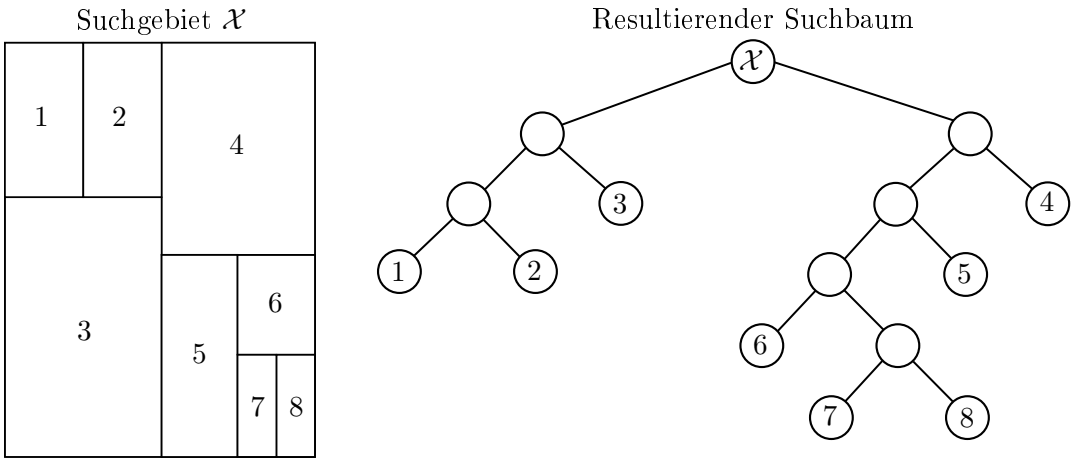


Abb. 4.20: Illustration zum Branch-and-Bound Ansatz für ein Beispiel im $\mathbb{R}^2$

Analog zur Gittersuche können die Positionen im Suchraum explizit ausgewertet oder als Startpunkte für lokale Suchalgorithmen herangezogen werden. Je nach vorliegendem Problem und Strukturierung der Methode besteht allerdings eine gewisse Gefahr, dass beim Bound-Schritt der Teil von $\mathcal{X}$ zu frühzeitig abgeschnitten wird, welcher das globale Optimum enthält.

4.4.2 Stochastische Ansätze

Zufallssuche (*random search*)

Unter dem Begriff Zufallssuche (*random search*) können zwei verschiedene Ansätze verstanden werden: Die erste Interpretation entspricht einer Variante der Gittersuche. Hier wird der Suchraum nicht gleichmäßig unterteilt, sondern es werden zufällig Punkte in diesem verteilt (siehe Abb. 4.21). Diese Punkte können dann wieder als Startwerte für lokale Suchverfahren genutzt oder lediglich explizit zur Funktionsauswertung genutzt werden. In Kombination mit lokalen Suchverfahren ist die Herangehensweise auch als *Multi-Start-Ansatz* bekannt.

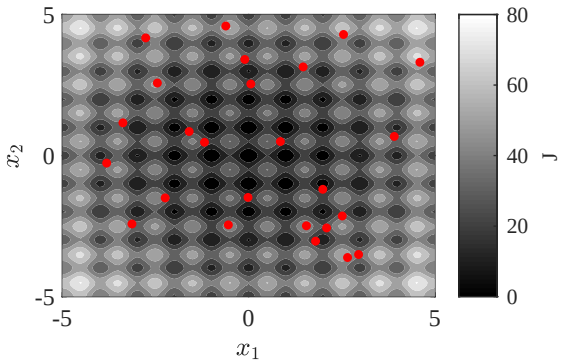


Abb. 4.21: Illustration zur Zufallssuche für die Rastrigin-Funktion

Hieran schließt sich dann auch die zweite Interpretation der Zufallssuche an, nämlich ein iterativer Ansatz. Dieser lässt sich wie folgt zusammenfassen:

Algorithmus 4.4 Zufallssuche

Initialisierung:

1: $\boldsymbol{x}_0 = \boldsymbol{x}[k=0]$ ▷ Startlösung

2: $k = 0$ ▷ Startindex

Iterieren:

3: **while** Abbruchbedingung noch nicht erfüllt (z. B. Anzahl Iterationen oder Kostenwert) **do**

4: Teste neue, zufällige Position $\tilde{\boldsymbol{x}}[k+1] \in \mathcal{X}$ in Nähe (z. B. in einer n-Sphäre) von $\boldsymbol{x}[k]$

5: Falls $\tilde{\boldsymbol{x}}[k+1] < \boldsymbol{x}[k]$: $\boldsymbol{x}[k+1] \leftarrow \tilde{\boldsymbol{x}}[k+1]$

6: $k \leftarrow k+1$

7: **end while**

Dieser Basisalgorithmus lässt sich beliebig erweitern, z. B. hinsichtlich einer optimalen bzw. adaptiven Schrittweite. Zudem stellt er die Basis für viele metaheureistische Anästze aus Kap. 4.4.3 dar.

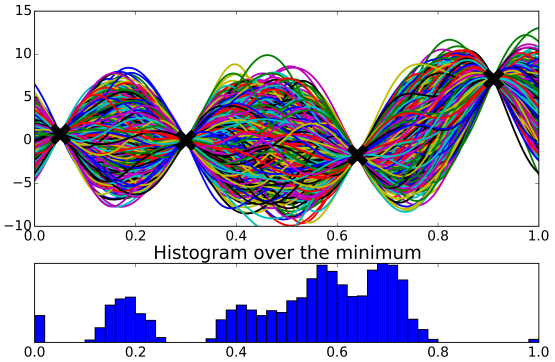
Bayessche Optimierung (*Bayesian optimization*)

Die Bayessche Optimierung (*Bayesian optimization*) kann als zielgerichtete Erweiterung der Zufallssuche verstanden werden. Ziel ist es, den Suchraum möglichst geschickt abzutasten, wobei ein Kompromiss aus Exploration, d. h. die Abtastung bisher unbekannter Regionen zur Minimierung der Unsicherheit, und Konvergenz (*exploitation*), d. h. die stetig engmaschiger werdende Begutachtung von Regionen mit vergleichsweise geringen Kostenwerten, zu wählen ist. Das Vorgehen hierbei kann wie folgt zusammengefasst werden:

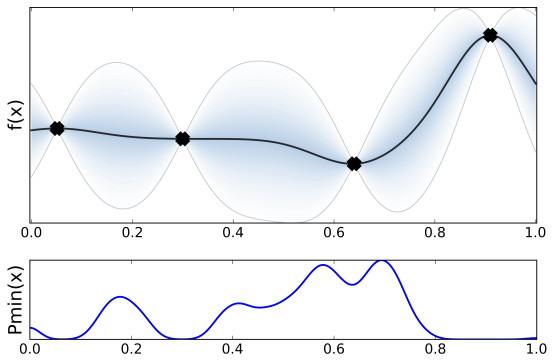
1. Bestimme ein Ersatzmodell mit Mittelwert- und Kovarianzschätzung der Kosten zur Abbildung des Optimierungsproblems (i.A. Gaußscher Prozess).
2. Fitte Mittelwert- und Kovarianzschätzung des Ersatzmodells auf Basis von initialen Stichproben im Suchraum (i.A. Zufallssuche).
3. Wiederhole:
   a) Platziere neue Stichprobe im Suchraum auf auf Basis der Kompromissbewertung von Exploration und Konvergenz mittels des Ersatzmodells
   b) Fitte Mittelwert- und Kovarianzschätzung des Ersatzmodells unter Berücksichtigung der neuen Stichprobe (Verbesserung der Modellvorhersage)
   c) Überprüfung der Abbruchbedingung (z. B. Iterationsanzahl)

Die Abbildung des Suchraums über einen Gaußschen Prozess als Ersatzmodell ist zudem in Abb. 4.22 für den eindimensionalen Fall illustriert. Der Bayessche Ansatz wird häufig gewählt, wenn die Funktionsauswertungen teuer, also zeit-, rechen- und/oder kostenintensiv sind, um mit einem kleinen Stichprobenumfang möglichst gute Ergebnisse zu erzielen. Diese bilden zwar nicht notwendigerweise das globale Optimum hinreichend genau ab, aber sie können dennoch ein vorher definiertes Optimierungsziel erreichen. Eine typische Anwendung hierfür ist die Hyperparameter-Optimierung im Kontext des maschinellen Lernens. Vorkonfigurierte Toolboxen zur Bayesschen Optimierung sind u. a.:

- Bayesian Optimization: Python Implementierung (frei zugänglich)
  `https://github.com/fmfn/BayesianOptimization`
- GPyOpt: Alternative Python Implementierung (frei zugänglich)
  `https://sheffieldml.github.io/GPyOpt/`
- BADS: Externe Matlab Implementierung (frei zugänglich)
  `https://github.com/lacerbi/bads`
- bayesopt: Interne Matlab Implementierung (kostenpflichtig)
  `https://de.mathworks.com/help/stats/bayesopt.html`



(a) Empirische Abbildungen möglicher Funktionsverläufe zwischen gegebenen Stützstellen und Histogramm der Minima-Stellen



(b) Überführung der resultierenden Minimum-Häufigkeitsverteilung in eine Wahrscheinlichkeitsdichteverteilung

Abb. 4.22: Veranschaulichung zur Bayesschen Optimierung (Quelle: [Gon17])

4.4.3 Metaheuristische Ansätze

Eine Metaheuristik ist ein Algorithmus zur näherungsweisen Lösung von Optimierungsproblemen. Im Gegensatz zu problemspezifischen Heuristiken, die nur auf ein bestimmtes Optimierungsproblem angewandt werden können, definieren Metaheuristiken eine abstrakte Folge von Schritten, die (theoretisch) auf beliebige Problemstellungen angewandt werden können. Die einzelnen Schritte müssen allerdings wieder problemspezifisch implementiert werden. In der Regel ist nicht garantiert, dass eine Metaheuristik eine optimale Lösung findet. Prinzipiell können all diese Verfahren gute Lösungen berechnen, aber auch beliebig schlecht im Vergleich zu einer Alternative sein. Generell hängen der Erfolg und die Laufzeit metaheuristischer Verfahren entscheidend von der Definition und Implementierung der einzelnen Schritte ab. Hierzu sei auch auf folgenden Satz verwiesen:

Satz 4.20: No-Free-Lunch-Theorem (NFL) nach Wolpert und Mcready

Vereinfacht ausgedrückt besagt das NFL-Teorem, dass kein universell gutes Verfahren zur Lösung von Optimierungsproblemen über die Menge aller Probleme existiert. Ist eine bestimmte Strategie in einem Teilbereich besser als eine andere, so muss sie in einem anderen Teilbereich schlechter sein (nichts ist umsonst / no-free-lunch).

Im Kontext der Metaheuristiken für globale Optimierungsaufgaben finden sich häufig *naturanaloge* Ansätze, also Verfahren, deren grundsätzliche Funktionsweise von natürlichen Vorbildern (beispielsweise Biologie) inspiriert ist. Typische Vertreter hierbei sind:

- Evolutionäre Algorithmen (z. B. Genetische Algorithmen),
- Schwarmintelligente Algorithmen (z. B. Partikelschwarmoptimierung),
- Simulierte Abkühlung (und Varianten wie z. B. Sintflutalgorithmus).

Weitere, nicht-naturanaloge Methoden sind z. B.

- Bergsteigeralgorithmus,
- Stochastisches Tunneln.

Nachfolgend soll die Partikelschwarmoptimierung als ein Beispiel der metaheuristischen Ansätze näher erläutert werden. Für Informationen zu den weiteren Verfahren sei auf die Literatur (z. B. [BLR04][Tal09]) verwiesen.

Partikelschwarmoptimierung (*Particle swarm optimization*)

Die Grundidee der Partikelschwarmoptimierung (PSO) basiert auf [KE95]: Es wird eine Anzahl von Entitäten, die Partikel, zufällig im Suchraum einer gegebenen Optimierungsfunktion platziert. Diese Funktion wird für jede Partikelposition ausgewertet, anschließend wird die Bewegung jedes Partikels basierend auf den historisch-individuellen Funktionsauswertungen und den Informationen eines oder mehrerer Nachbarpartikel berechnet. Diese Bewegungsanpassungen haben i. A. eine zufällige Komponente, um die Exploration des Suchraums voranzutreiben und das Konvergieren in lokale Minima zu verhindern. Die nächste Iteration startet nach der Aktualisierung aller Partikelpositionen mit der erneuten Funktionsauswertung. Hierdurch soll, in Analogie zu Vögeln oder anderen Tiergruppen, ein Schwarmverhalten nachgeahmt werden, welches mit hoher Wahrscheinlichkeit eine Partikelbewegung zum globalen Optimum vorantreibt.

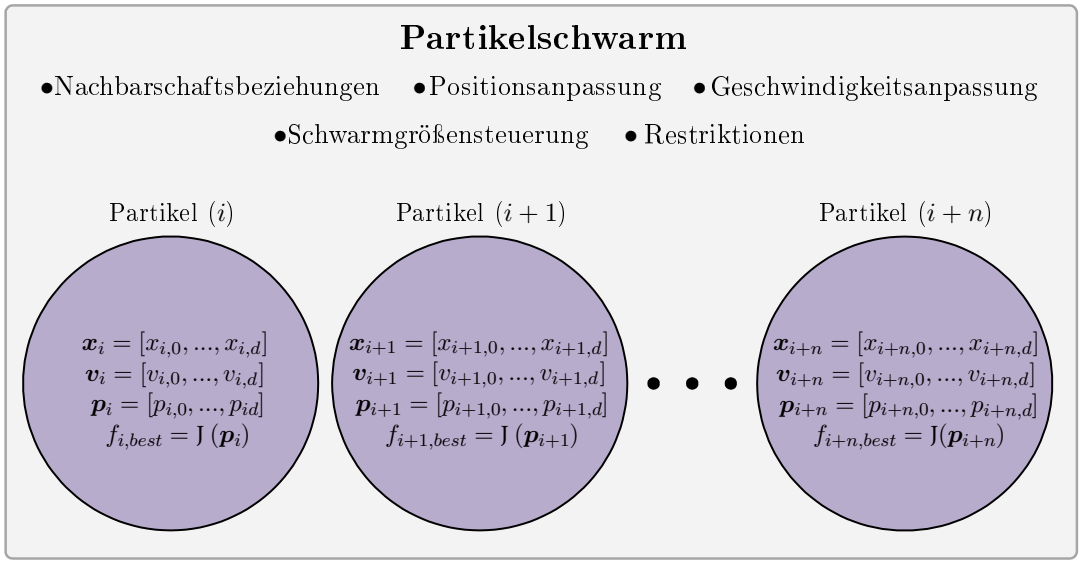
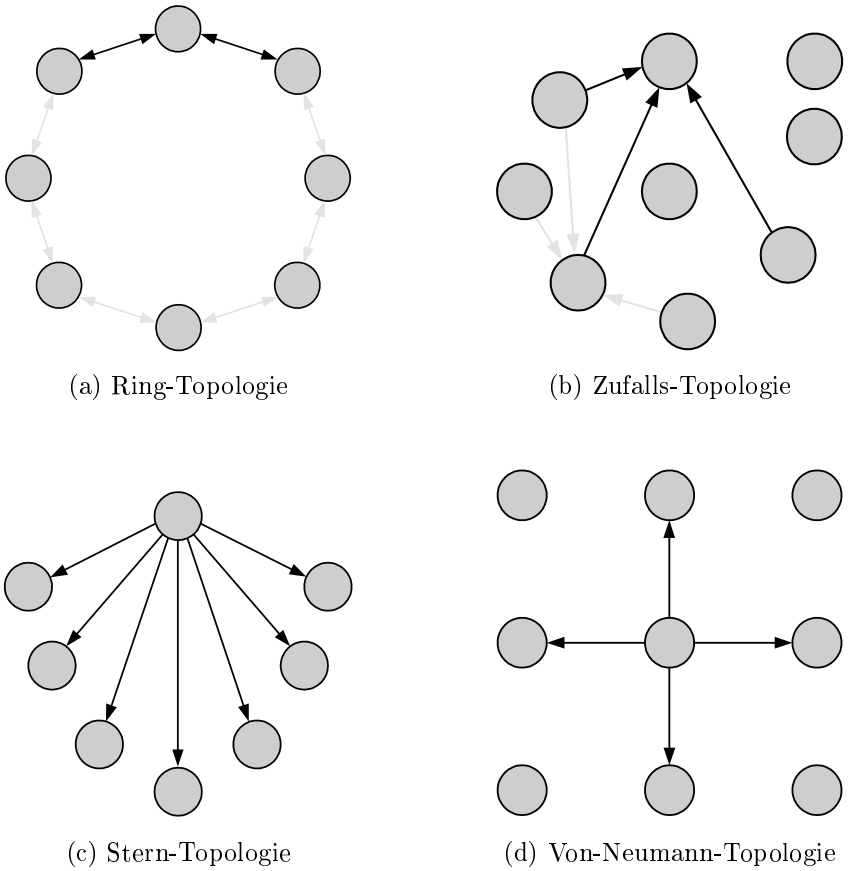


Abb. 4.23: Grundsätzlicher Aufbau eines Partikels und des Schwarms

Jedes i-te Partikel besteht aus drei d-dimensionalen Vektoren, wobei d der Anzahl der Optimierungsvariablen entspricht. Diese Vektoren sind die aktuelle Position $\boldsymbol{x}_i$, die aktuelle Geschwindigkeit $\boldsymbol{v}_i$ und die individuell beste Position der bisherigen Iterationen $\boldsymbol{p}_i$. Die Geschwindigkeit

kann dabei sehr anschaulich als gewichteter Richtungsvektor im Suchraum interpretiert werden, welcher die Positionsanpassung in der kommenden Iteration bestimmt. Des Weiteren wird der zugehörige Funktionswert $f_{i,best}$ an der historisch-individuell besten Position $\boldsymbol{p}_i$ ebenfalls gespeichert.



(a) Ring-Topologie (b) Zufalls-Topologie

(c) Stern-Topologie (d) Von-Neumann-Topologie

Abb. 4.24: Auswahl bekannter Nachbarschafts-Topologien der PSO

Das globale Optimum soll dann durch Interaktion der Partikel innerhalb definierter Nachbarschaften gefunden werden. Die einfachste Form ist die globale Nachbarschaft, d. h., alle Partikel tauschen sich untereinander aus. Weitere Nachbarschaftsbeziehungen sind in Abb. 4.24 dargestellt. Einzelne und unabhängige Partikel haben in komplexen und mehrdimensionalen Suchräumen kaum Chancen, das globale Optimum aufzufinden – erst durch den Zusammenschluss vieler Partikel in einem Schwarm wird ein intelligentes Suchverfahren erzeugt. Während die Partikel durch die oben beschriebenen Parameter in allen PSO-Varianten klar definiert sind, existieren vielfältige Variationen hinsichtlich der Schwarmgestaltung. Dieses umfasst insbesondere die Art der Nachbarschaften, der Umgang mit Restriktionen sowie die Anpassungsregeln der Position, der Geschwindigkeit und der Schwarmgröße zur Laufzeit. Unabhängig von diesen Gestaltungsmöglichkeiten sind die wichtigsten Schritte des Basis-PSO-Algorithmus wie folgt gekennzeichnet:

1. Initialisierung:
Alle Partikel werden zufällig im Suchraum platziert und es findet eine erstmalige Funktionsauswertung statt. Auch der Geschwindigkeitsvektor wird zufällig besetzt.

2. Aktualisierung der Geschwindigkeit:

Alle Geschwindigkeitsvektoren werden neu berechnet. Hierfür wird zum einen der vorherige Geschwindigkeitsvektor $\boldsymbol{v}_i[k]$ verwendet und zum anderen wird die aktuelle Partikelposition $\boldsymbol{x}_i[k]$ gegenüber der historisch-individuell besten Position $\boldsymbol{p}_i$ sowie der besten Position innerhalb einer definierten Nachbarschaft $\boldsymbol{p}_g$ verglichen. Die klassische Adaptionsvorschrift lautet dann:

$$\begin{aligned}
&\boldsymbol{v}_i[k+1] = \boldsymbol{v}_i[k] + \varphi_1(\boldsymbol{p}_i - \boldsymbol{x}_i[k]) + \varphi_2(\boldsymbol{p}_g - \boldsymbol{x}_i[k]) \quad \text{mit} \\
&\varphi_1 = c_1 \cdot r(0,1), \quad \varphi_2 = c_2 \cdot r(0,1) \quad \text{(Lerngesetz)}\,, \\
&c_1 + c_2 \approx 4 \quad \text{(Lernfaktoren)}\,, \\
&r(a,b) = \text{Gleichverteilte Zufallszahl im Intervall } [a,b]\,.
\end{aligned} \tag{4.167}$$

Die exakte Wahl der Lernfaktoren c_1 und c_2 ist stark anwendungsabhängig. Hierbei ist zu berücksichtigen, ob der Algorithmus stärker zum Optimum der Nachbarschaft oder zu den lokalen Partikeloptima konvergieren soll. Ferner wird die Konvergenzgeschwindigkeit maßgeblich beeinflusst, wobei in der Literatur häufig die Empfehlung $c_1 + c_2 \approx 4$ als geeigneter Kompromiss zwischen Exploration und lokaler Optimierung zu finden ist. Des Weiteren kann eine Beschränkung der Geschwindigkeit erfolgen, um zu verhindern, dass die Partikel zu schnell den definierten Suchraum verlassen:

$$v_{j,min} \leq v_{i,j} \leq v_{j,max} \quad \forall\, \{i = 1, ..., n;\ j = 1, ..., D\}\,. \tag{4.168}$$

Herbei bezeichnet $v_{i,j}$ die Geschwindigkeit der j-ten Ausprägung (Optimierungsvariable) des i-ten Partikels. Eine typische Festlegung der Schrankenwerte ist

$$-\,v_{j,min} = v_{j,max} = \frac{|x_{j,max} - x_{j,min}|}{2} \tag{4.169}$$

wobei $x_{j,max}$ und $x_{j,min}$ die obere und untere Begrenzung der j-ten Ausprägung der Partikel darstellen.

3. Aktualisierung der Position:
Mittels des neuen Geschwindigkeitsvektors werden alle Partikelpositionen aktualisiert:

$$\boldsymbol{x}_i[k+1] = \boldsymbol{x}_i[k] + \boldsymbol{v}_i[k+1]. \tag{4.170}$$

Anschließend ist zu prüfen, ob ein Partikel den Suchraum verlassen hat:

$$x_{j,min} \leq x_{i,j} \leq x_{j,max} \quad \forall\, \{i = 1, ..., n;\ j = 1, ..., D\}\,. \tag{4.171}$$

Ist dies der Fall, gibt es drei Möglichkeiten das betrachtete Partikel in den Suchraum zurückzuführen [Cle06]:

1. zufällige Platzierung innerhalb des Suchraums, ggf. auch Anpassung der Geschwindigkeit,
2. Begrenzung auf die Grenze der relevanten Ausprägung ($x_{i,j} = x_{j,min}$ bzw. $x_{i,j} = x_{j,max}$) und Null setzen des zugehörigen Geschwindigkeitswerts $v_{i,j}$,
3. indirekte Begrenzung durch Bestrafungsterm innerhalb der Kostenfunktion $J(\boldsymbol{x}_i)$.

4. Funktionsauswertung:
Alle Partikel werden an ihren neuen Positionen ausgewertet: $J(\boldsymbol{x}_i[k+1])$. Dann ist zu prüfen,

ob die Partikel ihre historisch-individuell beste Position verbessern konnten:

$$\boldsymbol{p}_i = \boldsymbol{x}_i[k+1], \quad \forall\, \boldsymbol{x}_i[k+1] = \{\boldsymbol{x}_i[k+1] \,|\, J(\boldsymbol{x}_i[k+1]) < f_{i,best}\}\,. \tag{4.172}$$

Ferner ist zu prüfen, ob die bisher beste Position einer gegebenen Nachbarschaft P_h verbessert wurde:

$$\boldsymbol{p}_{g,h} = \boldsymbol{p}_i, \quad \forall\, \boldsymbol{p}_i = \{\boldsymbol{p}_i \,|\, J(\boldsymbol{p}_i) < f_{h,best} \wedge \boldsymbol{p}_i \in P_h\}\,. \tag{4.173}$$

Hierbei bezeichnet der Begriff der Nachbarschaft kein definiertes Teilgebiet im Suchraum, sondern einen Zusammenschluss mehrerer Partikel, welche in einem spezifischen Verhältnis zueinander stehen und sich durchaus in sehr unterschiedlichen Bereichen des Suchraums befinden können (s. Abb. 4.24).

5. Abbruch oder nächste Iteration:
Es wird geprüft, ob ein definiertes Abbruchkriterium, z. B. die maximale Anzahl erlaubter Iterationen, erreicht wurde. Falls ja, wird die Optimierung beendet und eine Auswertung ausgegeben. Andernfalls startet die nächste Iteration mit der erneuten Ausführung ab Schritt 2.

Eine der grundlegenden Fragestellungen der PSO stellt die geeignete Anpassung des Schwarmverhaltens während der Optimierung dar, um sowohl eine ausreichende Exploration des gesamten Suchraums sicherzustellen als auch die Konvergenz des Schwarms zu garantieren. In der klassischen PSO-Implementierung werden Lernfaktoren (c_1, c_2) verwendet, welche bei der Aktualisierung der Geschwindigkeit eine Gewichtung zwischen dem historisch-individuellen Optimum und dem historischen Optimum der Nachbarschaft erlauben. Da diese Lernfaktoren allerdings konstant sind, besteht die Gefahr, dass sich die Partikel mit einer stetig hohen Geschwindigkeit durch den Suchraum bewegen und eine feinmaschige Optimierung in der Nähe der bisher vielversprechendsten Positionen ausbleibt. Für diesen Übergang von der explorativen Suche zur lokalen Optimierung sollen nachfolgend die bekanntesten Methoden vorgestellt werden:

Einfache Trägheitsgewichte:
Diese Methode wurde bereits in [KE95] vorgestellt und als *inertia weight* bezeichnet. Hierbei wird ein Gewichtungsfaktor α eingeführt, der zu einer Reduktion der Partikelgeschwindigkeiten mit zunehmender Iterationsanzahl führt:

$$\boldsymbol{v}_i[k+1] = \alpha \boldsymbol{v}_i[k] + \varphi_1(\boldsymbol{p}_i - \boldsymbol{x}_i[k]) + \varphi_2(\boldsymbol{p}_g - \boldsymbol{x}_i[k]). \tag{4.174}$$

Folglich wird die Bewegung in Richtung von $\boldsymbol{p}_i$ und $\boldsymbol{p}_g$ stärker gewichtet und die Konvergenz in ein (lokales) Optimum fokussiert. Als weitere Variante ist eine lineare Reduzierung von α in Abhängigkeit der Iterationsanzahl bekannt, z. B. im Intervall $\alpha = [0,9 \rightarrow 0,4]$. Allerdings ist es schwierig vor der eigentlichen Optimierung abzuschätzen, wie lange die explorative Phase dauern sollte, da eine konkrete Iterationsanzahl vorgegeben werden muss.

Dynamische Trägheitsgewichte:
In vielen Anwendungen kann die Kostenfunktion nur schwer beurteilt werden, sodass wenig a-priori Informationen über den anstehenden Optimierungsprozess verfügbar sind. In diesen Fällen ist es erstrebenswert nicht eine feste Anzahl von Iterationen als Abbruchkriterium vorzugeben, sondern die Optimierung über statistische Gütemaße, z. B. die relative Verbesserung der Kostenfunktion innerhalb der letzten Iterationen, zu terminieren. Eine Variante der *inertia*

weight-Methode führt daher zu einer adaptiven Anpassung des Trägheitskoeffizienten:

$$\alpha_i = 1,1 - \left| \frac{f(\boldsymbol{p}_g)}{f(\boldsymbol{p}_i)} \right|. \tag{4.175}$$

Im Unterschied zur einfachen *inertia weight*-Methode wird ein individuelles α_i für jedes Partikel in jeder Iteration neu berechnet. Der Quotient des individuell besten Funktionswertes zum global besten Funktionswert soll hierbei als eine Abschätzung dienen, wie homogen der Schwarm bereits ist und ob die lokale Suche stärker zu fokussieren sei.

Verengungsansatz
Beim sog. Verengungsansatz (*constriction factor*) werden im Unterschied zu den Trägheitsgewichten nicht nur die vorherigen Geschwindigkeiten, sondern auch die Beziehung zum bisherigen historisch-individuellen Optimum und dem der Nachbarschaft gewichtet:

$$\boldsymbol{v}_i[k+1] = \boldsymbol{v}_i[k] - \chi \left[\boldsymbol{v}_i[k] + \varphi_1(\boldsymbol{p}_i - \boldsymbol{x}_i[k]) + \varphi_2(\boldsymbol{p}_g - \boldsymbol{x}_i[k]) \right] \tag{4.176}$$

mit χ als der sog. Verengungsfaktor:

$$\chi = \frac{1}{\varphi - 1 + \sqrt{\varphi^2 - 4\varphi}} \quad \text{mit: } \varphi = \varphi_1 + \varphi_2. \tag{4.177}$$

In [Cle06] wurde gezeigt, dass (4.176) gegenüber dem Standardansatz (4.167) zu einer Stauchung der Geschwindigkeitsvektoren im Suchraum führt. Diese Stauchung führt mit fortschreitender Anzahl von Iterationen ebenfalls zum Übergang hin zu einer lokalen Optimierung.

4.4.4 Zusammenfassung

Die vorangegangenen Methoden zur globalen Optimierung statischer Probleme stellen nur einen kleinen Ausschnitt der verfügbaren Verfahren dar und wurden zudem nicht tiefer gehend behandelt. Neben dem erneuten Verweis auf die bereits zitierte Literatur für weitere Informationen zu diesem Themenfeld, sei auch an dieser Stelle eine Auswahl an vorkonfigurierten Toolboxen zusammengefasst, welche bei der Lösung globaler Optimierungsprobleme hilfreich sein können:

Toolboxen für globale Optimierungsprobleme

- GADS: Matlab Global Optimization Toolbox (verschiedene Solver, kostenpflichtig)
  `https://de.mathworks.com/help/gads/`
- NLopt (verschiedene Solver, u. a. Matlab/Python Schnittstelle, frei zugänglich)
  `https://nlopt.readthedocs.io/`
- Pagmo / pygmo (verschiedene Solver, u. a. C++/Python Schnittstelle, frei zugänglich)
  `https://esa.github.io/pagmo2/`
- SciPy (verschiedene Solver, Python-Softwareumgebung, frei zugänglich)
  `https://docs.scipy.org/doc/scipy/reference/optimize.html`
- NEOS (verschiedene Solver, frei zugänglich inkl. Server-Kapazitäten und Web-App)
  `https://neos-guide.org/content/global-optimization`

5 Zustandsschätzung mittels Kalman-Filter

In vielen Anwendungen ist eine Schätzung des Zustands $\hat{\boldsymbol{x}}[k]$ basierend auf Messwerten $\boldsymbol{y}[j]$ sowie Eingangsgrößen $\boldsymbol{u}[j]$ von Interesse, z. B. dann, wenn ein Zustand aufgrund der damit verbundenen Kosten oder aus konstruktiven Gründen nicht messtechnisch erfasst werden kann. Hierbei können folgende Szenarien unterschieden werden:

$$\hat{\boldsymbol{x}}[k] = \boldsymbol{f}(\boldsymbol{u}[j], \boldsymbol{u}[j-1], \ldots, \boldsymbol{y}[j], \boldsymbol{y}[j-1], \ldots) \quad \text{mit} \quad \begin{cases} k > j : \text{Prädiktion}, \\ k = j : \text{Filtern}, \\ k < j : \text{Glättung}. \end{cases} \tag{5.1}$$

Das nachfolgend vorgestellte Kalman-Filter[1] basiert auf einer 1-Schritt-Prädiktion, welche anschließend auf Basis eingehender Messwerte korrigiert wird[2]. Hierfür werden die nachfolgenden Betrachtungen ausschließlich im zeitdiskreten Zustandsraum stattfinden, wobei folgende Systemdarstellung angenommen wird[3]:

Definition 5.1: Zeitdiskretes Modell mit additiven Rauscheinflüssen

Gegeben sei eine zeitdiskrete Modellbeschreibung durch

$$\begin{aligned} \boldsymbol{x}[k+1] &= \boldsymbol{f}\left(\boldsymbol{x}[k], \boldsymbol{u}[k]\right) + \boldsymbol{m}[k], \quad \boldsymbol{x}[k=0] = \boldsymbol{x}_0\,, \\ \boldsymbol{y}[k] &= \boldsymbol{g}\left(\boldsymbol{x}[k], \boldsymbol{u}[k]\right) + \boldsymbol{n}[k] \end{aligned} \tag{5.2}$$

mit einer Zustandsfunktion $\boldsymbol{f} : \mathbb{R}^n \to \mathbb{R}^n$ *sowie der Ausgangsfunktion* $\boldsymbol{g} : \mathbb{R}^n \to \mathbb{R}^p$. *Ferner sind* $\boldsymbol{m}[k] \in \mathbb{R}^n$ *und* $\boldsymbol{n}[k] \in \mathbb{R}^p$ *weißes, additives System- bzw. Messrauschen. Hierbei seien* $\boldsymbol{m}[k]$ *und* $\boldsymbol{n}[k]$ *unkorrelierte und mittelwertfreie weiße Rauschvorgänge,*

$$\begin{aligned} \mathrm{Cov}(\boldsymbol{m}[k], \boldsymbol{n}[k]) = \boldsymbol{0}\,, \quad \mathrm{E}(\boldsymbol{m}[k]) = \boldsymbol{0}\,, \quad \mathrm{E}(\boldsymbol{n}[k]) = \boldsymbol{0}\,, \\ \mathrm{Cov}(\boldsymbol{m}[k], \boldsymbol{m}[l]) = \boldsymbol{0}\,, \quad \mathrm{Cov}(\boldsymbol{n}[k], \boldsymbol{n}[l]) = \boldsymbol{0}\,, \ \text{für } k \neq l \end{aligned} \tag{5.3}$$

mit konstanter Kovarianz

$$\mathrm{E}(\boldsymbol{m}[k]\boldsymbol{m}^T[k]) = \boldsymbol{M}, \quad \mathrm{E}(\boldsymbol{n}[k]\boldsymbol{n}^T[k]) = \boldsymbol{N}. \tag{5.4}$$

Nachfolgend werden verschiedenen Ausführungsformen des Kalman-Filters vorgestellt. Hierbei wird gezeigt, dass das Filter sowohl zur Schätzung nicht messbarer Zustände, zur Reduktion des Rauscheinflusses in Messgrößen als auch zur Vorhersage von Zuständen zwischen Abtastungen genutzt werden kann. Typische industrielle Anwendungen des Filters sind:

---

[1] Nach Rudolf Emil Kalman, US-amerikanischer Elektroingenieur und Mathematiker ungarischer Herkunft. Erlangte große Bekanntheit als seine Erfindung während des Apollo-Programms ab den 1960ern eingesetzt wurde.

[2] In seiner Standardformulierung gibt es daher sowohl prädizierte als auch gefilterte Zustände.

[3] Das *Kalman-Bucy-Filter* ist das entsprechend Pendant für kontinuierliche Systeme.

- **Inertialnavigation**: Während des Flugs werden Beschleunigungen und Drehraten eines Flugzeugs von einer inertialen Messeinheit mit hohen Frequenzen gemessen, um eine Kurzzeit-Navigation zu ermöglichen. Weitere Sensoren, insbesondere satellitengestützte Positionsbestimmung (z. B. GPS), liefern Stützdaten. Diese verschiedenen Messungen müssen fusioniert werden, um eine möglichst optimale Schätzung der aktuellen Position und Orientierung zu gewährleisten.
- **Tracking**: Sicherheits- oder Komfortanwendungen, die auf umfelderkennenden Systemen basieren, sind auf verlässliche Informationen (z. B. Position, Geschwindigkeit) bezüglich der Objekte in ihrem Umfeld angewiesen. Bei autonomen Landfahrzeugen werden Kalman-Filter zur Reduzierung des Rauschens von Lidar- und Radargeräten eingesetzt.
- **Sensorersatz**: In der elektrischen Antriebstechnik werden bei Drehstrommotoren i. A. die Lage und/oder die Geschwindigkeit des Rotors benötigt, um eine funktionierende Regelung sicherzustellen. Entsprechende Sensoren (z. B. Resolver) verursachen z. T. nicht unerhebliche Kosten und benötigen Bauraum. Das Kalman-Filter wird daher gerne genutzt, um Geschwindigkeit und/oder Position des Rotors auf Basis anderer Messgrößen (z. B. Strom, Spannung) zu beobachten und folglich den Sensor entfallen zu lassen.

5.1 Das linear-zeitdiskrete Kalman-Filter

Gegenüber der allgemeinen und potentiell nichtlinearen Darstellung in (5.2), bezieht sich das *lineare* Kalman-Filter (KF) ausschließlich auf zeitdiskrete LTI-Systeme der Form:

$$\begin{aligned} \boldsymbol{x}[k+1] &= \boldsymbol{A}\boldsymbol{x}[k] + \boldsymbol{B}\boldsymbol{u}[k] + \boldsymbol{m}[k]\,, \quad \boldsymbol{x}[k=0] = \boldsymbol{x}_0\,, \\ \boldsymbol{y}[k] &= \boldsymbol{C}\boldsymbol{x}[k] + \boldsymbol{D}\boldsymbol{u}[k] + \boldsymbol{n}[k]\,. \end{aligned} \tag{5.5}$$

Die vorangegangenen Annahmen hinsichtlich der Rauschprozesse bleiben erhalten. Ziel des KFs ist die Minimierung des quadratischen Prädiktionsfehlers hinsichtlich der Zustände:

$$J[k+1] = \mathrm{E}\left((\hat{\boldsymbol{x}}[k+1] - \boldsymbol{x}[k+1])^T(\hat{\boldsymbol{x}}[k+1] - \boldsymbol{x}[k+1])\right). \tag{5.6}$$

Hierbei bezeichnet $\hat{\boldsymbol{x}}$ einen beobachteten Schätzwert. Aufgrund der stochastischen Rauscheinflüsse wird bei der Berechnung der Kostenfunktion J zudem der Erwartungswert angesetzt. Kernidee des KFs ist der sog. *Prädiktor-Korrektor-Ansatz*, d. h., die Zustände werden für den zukünftigen Zeitpunkt $k+1$ vorhergesagt, um dann auf Basis einer neuen Messung $\boldsymbol{y}[k+1]$ korrigiert zu werden. Hierbei soll folgende Notation genutzt werden:

- $\hat{\boldsymbol{x}}[k+1|k]$: Prädizierter Zustand zum Zeitpunkt $k+1$ basierend auf den Messungen bis zum Zeitpunkt k (sog. *a-priori*[1] Schätzung).
- $\hat{\boldsymbol{x}}[k+1|k+1]$: Prädizierter und korrigierter Zustand zum Zeitpunkt $k+1$ basierend auf den Messungen bis zum Zeitpunkt $k+1$ (sog. *a-posteriori*[2] Schätzung).

[1] Lateinisch *a* „von ... her“ und lateinisch *prior* „der vordere, der frühere“.

[2] Lateinisch *a* „von ... her“ und lateinisch *posterior* „der spätere, der hintere“. Es sei zudem darauf hingewiesen, dass $\hat{\boldsymbol{x}}[k+1|k+1]$ sowie $\hat{\boldsymbol{x}}[k+1|k]$ bzw. $\boldsymbol{P}[k+1|k+1]$ und $\boldsymbol{P}[k+1|k]$ trotz der gewählten Darstellung mit doppeltem Zeitindex unterschiedliche Größen sind und nicht die selbe Größen zu jeweils nur unterschiedlichen Zeitpunkten darstellen. In der Literatur ist daher ebenfalls die Notation $\hat{\boldsymbol{x}}^-[k+1] = \hat{\boldsymbol{x}}[k+1|k]$ bzw. $\boldsymbol{P}^-[k+1] = \boldsymbol{P}[k+1|k]$ anzutreffen, um diese Differenzierung noch prägnanter zum Ausdruck zu bringen. Bei der praktischen Implementierung des Filters sind daher auch i. A. getrennte Variablen für die genannten Größen

Ferner wird die Kovarianzmatrix

$$\boldsymbol{P}[k|k] = \mathrm{E}\left((\hat{\boldsymbol{x}}[k|k] - \boldsymbol{x}[k])(\hat{\boldsymbol{x}}[k|k] - \boldsymbol{x}[k])^T\right) \tag{5.7}$$

der Zustände zum Zeitpunkt k benötigt. Der *Prädiktionsschritt* für das tatsächliche und modellierte System lautet:

$$\boldsymbol{x}[k+1] = \boldsymbol{A}\boldsymbol{x}[k] + \boldsymbol{B}\boldsymbol{u}[k] + \boldsymbol{m}[k], \tag{5.8}$$

$$\hat{\boldsymbol{x}}[k+1|k] = \boldsymbol{A}\hat{\boldsymbol{x}}[k|k] + \boldsymbol{B}\boldsymbol{u}[k]. \tag{5.9}$$

Die Kovarianzmatrix für den Prädiktionsschritt ergibt sich dann zu

$$\begin{aligned}\boldsymbol{P}[k+1|k] &= \mathrm{E}\left((\hat{\boldsymbol{x}}[k+1|k] - \boldsymbol{x}[k+1])(\hat{\boldsymbol{x}}[k+1|k] - \boldsymbol{x}[k+1])^T\right) \\ &= \mathrm{E}\left((\boldsymbol{A}\hat{\boldsymbol{x}}[k|k] - \boldsymbol{A}\boldsymbol{x}[k] - \boldsymbol{m}[k])(\boldsymbol{A}\hat{\boldsymbol{x}}[k|k] - \boldsymbol{A}\boldsymbol{x}[k] - \boldsymbol{m}[k])^T\right)\end{aligned} \tag{5.10}$$

und weiteres Ausmultiplizieren unter Verwendung der Annahme, dass $\boldsymbol{x}$ und $\boldsymbol{m}$ unkorreliert sind, führt dann zu:

$$\begin{aligned}\boldsymbol{P}[k+1|k] = {} & \boldsymbol{A}\mathrm{E}\left((\hat{\boldsymbol{x}}[k|k] - \boldsymbol{x}[k])(\hat{\boldsymbol{x}}[k|k] - \boldsymbol{x}[k])^T\right)\boldsymbol{A}^T + \mathrm{E}\left(\boldsymbol{m}[k]\boldsymbol{m}^T[k]\right) \\ & + \underbrace{\boldsymbol{A}\mathrm{E}\left((\hat{\boldsymbol{x}}[k|k] - \boldsymbol{x}[k])\boldsymbol{m}^T[k]\right)}_{=\boldsymbol{0}} + \underbrace{\mathrm{E}\left(\boldsymbol{m}[k](\hat{\boldsymbol{x}}[k|k] - \boldsymbol{x}[k])^T\right)\boldsymbol{A}^T}_{=\boldsymbol{0}}.\end{aligned} \tag{5.11}$$

Es ergibt sich somit für die prädizierte Kovarianzmatrix:

$$\boldsymbol{P}[k+1|k] = \boldsymbol{A}\boldsymbol{P}[k|k]\boldsymbol{A}^T + \boldsymbol{M}. \tag{5.12}$$

Sobald die Messung $\boldsymbol{y}[k+1]$ verfügbar ist, erfolgt der *Korrekturschritt*:

$$\hat{\boldsymbol{x}}[k+1|k+1] = \hat{\boldsymbol{x}}[k+1|k] + \boldsymbol{K}[k+1](\boldsymbol{y}[k+1] - \boldsymbol{C}\hat{\boldsymbol{x}}[k+1|k] - \boldsymbol{D}\boldsymbol{u}[k+1]). \tag{5.13}$$

Die noch unbekannte Matrix $\boldsymbol{K}[k+1]$ ist die sog. *Kalman-Matrix*, welche den Fehler zwischen Messung und Prädiktion nutzt, um eine korrigierte Zustandsschätzung zu ermitteln. Das KF kann daher auch als rekursiver LS-Schätzer, analog zur Anwendung des RLS auf statische Systeme in Kap. 3.2.2, für dynamische Systeme verstanden werden. Das grundsätzliche Kalman-Filter Konzept ist in Abb. 5.1 graphisch dargestellt. Im Folgenden wird die optimale Wahl für $\boldsymbol{K}[k+1]$ bestimmt, um (5.6) zu minimieren. Hierfür wird (5.13) unter Verwendung von (5.5) zunächst umgeschrieben:

$$\hat{\boldsymbol{x}}[k+1|k+1] = \hat{\boldsymbol{x}}[k+1|k] + \boldsymbol{K}[k+1](\boldsymbol{C}\boldsymbol{x}[k+1] + \boldsymbol{n}[k+1] - \boldsymbol{C}\hat{\boldsymbol{x}}[k+1|k]). \tag{5.14}$$

Die prädizierte und korrigierte Kovarianzmatrix ist dann:

$$\boldsymbol{P}[k+1|k+1] = \mathrm{E}\left((\hat{\boldsymbol{x}}[k+1|k+1] - \boldsymbol{x}[k+1])(\hat{\boldsymbol{x}}[k+1|k+1] - \boldsymbol{x}[k+1])^T\right). \tag{5.15}$$

---

anzulegen.

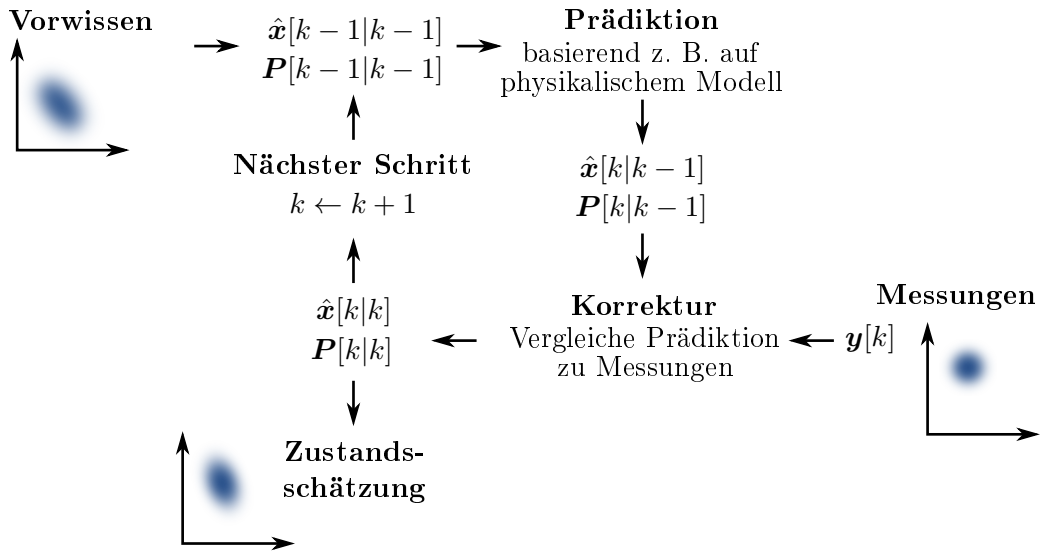


Abb. 5.1: Illustration zum Prädiktor-Korrektor-Ansatz des KFs (vgl. [Wik18c])

Unter Verwendung der Definition 2.18 kann dann die Kostenfunktion (5.6) in Abhängigkeit der Kovarianzmatrix (5.15) ausgedrückt werden[1]:

$$\begin{aligned} J[k+1] &= \mathrm{E}\left((\hat{\boldsymbol{x}}[k+1|k+1]-\boldsymbol{x}[k+1])^T(\hat{\boldsymbol{x}}[k+1|k+1]-\boldsymbol{x}[k+1])\right) \\ &= \mathrm{E}\left(\mathrm{Spur}\left((\hat{\boldsymbol{x}}[k+1|k+1]-\boldsymbol{x}[k+1])(\hat{\boldsymbol{x}}[k+1|k+1]-\boldsymbol{x}[k+1])^T\right)\right) \\ &= \mathrm{Spur}\left(\mathrm{E}\left((\hat{\boldsymbol{x}}[k+1|k+1]-\boldsymbol{x}[k+1])(\hat{\boldsymbol{x}}[k+1|k+1]-\boldsymbol{x}[k+1])^T\right)\right) \\ &= \mathrm{Spur}\left(\boldsymbol{P}[k+1|k+1]\right). \end{aligned} \tag{5.16}$$

Für die nachfolgenden Berechnungen werden zwischenzeitlich die Indizes weggelassen, um eine kompaktere Schreibweise zu erlauben. Zunächst werde die Kovarianzmatrix (5.15) unter Verwendung von (5.14) noch weiter umgeschrieben:

$$\begin{aligned} \boldsymbol{P}[k+1|k+1] &= \mathrm{E}\left((\hat{\boldsymbol{x}}-\boldsymbol{x}-\boldsymbol{KC}(\hat{\boldsymbol{x}}-\boldsymbol{x})+\boldsymbol{Kn}))(\hat{\boldsymbol{x}}-\boldsymbol{x}-\boldsymbol{KC}(\hat{\boldsymbol{x}}-\boldsymbol{x})+\boldsymbol{Kn})^T\right) \\ &= \mathrm{E}\left(((\boldsymbol{I}-\boldsymbol{KC})(\hat{\boldsymbol{x}}-\boldsymbol{x})+\boldsymbol{Kn}))((\boldsymbol{I}-\boldsymbol{KC})(\hat{\boldsymbol{x}}-\boldsymbol{x})+\boldsymbol{Kn})^T\right). \end{aligned} \tag{5.17}$$

Mittels (5.17) und erneuter Ausnutzung, dass $\boldsymbol{x}$ und $\boldsymbol{n}$ unkorreliert sind, folgt dann:

$$\boldsymbol{P}[k+1|k+1] = (\boldsymbol{I}-\boldsymbol{K}[k+1]\boldsymbol{C})\boldsymbol{P}[k+1|k](\boldsymbol{I}-\boldsymbol{K}[k+1]\boldsymbol{C})^T + \boldsymbol{K}[k+1]\boldsymbol{N}\boldsymbol{K}^T[k+1]. \tag{5.18}$$

Zum Auffinden der optimalen Kalman-Matrix, muss (5.16) nach $\boldsymbol{K}[k+1] = \boldsymbol{K}$ abgeleitet und zu Null gesetzt werden:

$$\frac{\partial}{\partial \boldsymbol{K}}\mathrm{Spur}\left(\boldsymbol{P}[k+1|k+1]\right) = \begin{bmatrix} \frac{\partial}{\partial k_{11}} & \frac{\partial}{\partial k_{12}} & \cdots & \frac{\partial}{\partial k_{1p}} \\ \frac{\partial}{\partial k_{21}} & \ddots & \cdots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial}{\partial k_{n1}} & \frac{\partial}{\partial k_{n2}} & \cdots & \frac{\partial}{\partial k_{np}} \end{bmatrix} \mathrm{Spur}\left(\boldsymbol{P}[k+1|k+1]\right) = \boldsymbol{0} \tag{5.19}$$

[1] Sei $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ eine quadratische Matrix. Dann bezeichnet die *Spur* der Matrix die Summe ihrer Diagonalelemente: $\mathrm{Spur}(\boldsymbol{A}) = \sum_{i=1}^{n} a_{ii}$.

mit

$$\boldsymbol{K} = \begin{bmatrix} k_{11} & k_{12} & \cdots & k_{1p} \\ k_{21} & \ddots & \cdots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ k_{n1} & k_{n2} & \cdots & k_{np} \end{bmatrix}. \tag{5.20}$$

Es ergibt sich demnach ein Lösungssystem mit $n \times p$ Unbekannten und entsprechenden Gleichungen. Ausmultiplizieren von (5.18) ergibt:

$$\begin{aligned} \boldsymbol{P}[k+1|k+1] = \boldsymbol{P}[k+1|k] - \boldsymbol{K}[k+1]\boldsymbol{C}\boldsymbol{P}[k+1|k] - \boldsymbol{P}[k+1|k]\boldsymbol{C}^T\boldsymbol{K}^T[k+1] \\ + \boldsymbol{K}[k+1]\boldsymbol{C}\boldsymbol{P}[k+1|k]\boldsymbol{C}^T\boldsymbol{K}^T[k+1] + \boldsymbol{K}[k+1]\boldsymbol{N}\boldsymbol{K}^T[k+1]. \end{aligned} \tag{5.21}$$

Unter Verwendung der in Anhang A.3 zusammengefassten Rechenregeln und unter Nutzung der Eigenschaft, dass reelle Kovarianzmatrizen symmetrisch sind, folgt dann für die einzelnen Terme in (5.19):

$$\begin{aligned} &\frac{\partial}{\partial \boldsymbol{K}}\text{Spur}\left(\boldsymbol{P}[k+1|k]\right) = 0, \\ &\frac{\partial}{\partial \boldsymbol{K}}\text{Spur}\left(\boldsymbol{K}[k+1]\boldsymbol{C}\boldsymbol{P}[k+1|k]\right) = \boldsymbol{P}[k+1|k]\boldsymbol{C}^T, \\ &\frac{\partial}{\partial \boldsymbol{K}}\text{Spur}\left(\boldsymbol{P}[k+1|k]\boldsymbol{C}^T\boldsymbol{K}^T[k+1]\right) = \boldsymbol{P}[k+1|k]\boldsymbol{C}^T, \\ &\frac{\partial}{\partial \boldsymbol{K}}\text{Spur}\left(\boldsymbol{K}[k+1]\boldsymbol{C}\boldsymbol{P}[k+1|k]\boldsymbol{C}^T\boldsymbol{K}^T[k+1]\right) = 2\boldsymbol{K}[k+1]\boldsymbol{C}\boldsymbol{P}[k+1|k]\boldsymbol{C}^T, \\ &\frac{\partial}{\partial \boldsymbol{K}}\text{Spur}\left(\boldsymbol{K}[k+1]\boldsymbol{N}\boldsymbol{K}^T[k+1]\right) = 2\boldsymbol{K}[k+1]\boldsymbol{N}. \end{aligned} \tag{5.22}$$

Einsetzen in (5.19) und Umstellen ergibt dann:

$$\boldsymbol{K}[k+1]\left(\boldsymbol{C}\boldsymbol{P}[k+1|k]\boldsymbol{C}^T + \boldsymbol{N}\right) = \boldsymbol{P}[k+1|k]\boldsymbol{C}^T. \tag{5.23}$$

Auflösen führt schließlich zur optimalen Kalman-Matrix:

$$\boldsymbol{K}[k+1] = \boldsymbol{P}[k+1|k]\boldsymbol{C}^T\left(\boldsymbol{C}\boldsymbol{P}[k+1|k]\boldsymbol{C}^T + \boldsymbol{N}\right)^{-1}. \tag{5.24}$$

Zusammenfassend ergibt sich somit das Kalman-Filter für LTI-Systeme gemäß (5.5) entsprechend des nachfolgendes Satzes:

Satz 5.1: Linear-zeitdiskretes Kalman-Filter für LTI-Systeme

Für das LTI-System (5.5) *kann ein optimaler, linearer und rekursiver LS-Schätzer zur Minimierung von* (5.6) *unter den Annahmen aus Definition 5.1 durch folgenden Prädiktor-Korrektor-Ansatz gewonnen werden:*

Prädiktion:

$$\hat{\boldsymbol{x}}[k+1|k] = \boldsymbol{A}\hat{\boldsymbol{x}}[k] + \boldsymbol{B}\boldsymbol{u}[k] \tag{5.25a}$$

$$\boldsymbol{P}[k+1|k] = \boldsymbol{A}\boldsymbol{P}[k|k]\boldsymbol{A}^T + \boldsymbol{M} \tag{5.25b}$$

Korrektur:

$$\boldsymbol{K}[k+1] = \boldsymbol{P}[k+1|k]\boldsymbol{C}^T\left(\boldsymbol{C}\boldsymbol{P}[k+1|k]\boldsymbol{C}^T + \boldsymbol{N}\right)^{-1} \tag{5.25c}$$

$$\hat{\boldsymbol{x}}[k+1|k+1] = \hat{\boldsymbol{x}}[k+1|k] + \boldsymbol{K}[k+1]\left(\boldsymbol{y}[k+1] - \boldsymbol{C}\hat{\boldsymbol{x}}[k+1|k] - \boldsymbol{D}\boldsymbol{u}[k+1]\right) \tag{5.25d}$$

$$\boldsymbol{P}[k+1|k+1] = \left(\boldsymbol{I} - \boldsymbol{K}[k+1]\boldsymbol{C}\right)\boldsymbol{P}[k+1|k] \tag{5.25e}$$

mit entsprechender Initialisierung

$$\boldsymbol{P}[0|0] = \boldsymbol{P}_0, \quad \hat{\boldsymbol{x}}[0|0] = \boldsymbol{x}_0. \tag{5.26}$$

Es sei betont, dass für die Herleitung keine Annahme hinsichtlich der Zufallsverteilung für $\boldsymbol{m}[k]$ und $\boldsymbol{n}[k]$ getroffen wurde. Wichtig ist, dass diese mittelwertfrei, weiß und unkorreliert sind, dann folgt, dass das KF die beste lineare Lösung zur gegebene Aufgabe ist – was nicht ausschließt, dass ein nichtlinearer Schätzer bessere Ergebnisse liefern kann. Eine gelegentlich anzutreffende Forderung, dass das Rauschen Gauß-verteilt sein muss, damit das KF der beste lineare Schätzer ist, ist nicht zutreffend. Ist das Rauschen allerdings tatsächlich Gauß-verteilt, kann gezeigt werden, dass es in der Klasse der nichtlinearen Schätzer keinen besseren Ansatz als das hier vorgestellte lineare KF gibt. Ist das Rauschen hingegen korreliert oder nicht-weiß, existieren modifizierte KF-Formulierungen, welche auch diese Fälle abdecken (siehe z. B. [Sim06]).

Das resultierende Blockdiagramm[1] zum Kalman-Filter ist in Abb. 5.2 dargestellt. Es ist ersichtlich, dass aus der Struktur sowohl der prädizierte Zustand $\hat{\boldsymbol{x}}[k+1|k]$ vor Eintreffen der Messung $\boldsymbol{y}[k+1]$ als auch der entsprechend korrigierte Zustand $\hat{\boldsymbol{x}}[k+1|k+1]$ nach Eintreffen der Messung $\boldsymbol{y}[k+1]$ hervorgehen – es handelt sich, wie eingangs beschrieben, daher um den 1-Schritt-KF. In der Literatur sind daneben Mehrschritt-KF bekannt, welche im Prädiktionsschritt bis zu N-Schritte in die Zukunft prädizieren, während der Korrekturschritt erhalten bleibt (siehe z. B. [Tar10]).

Für die Verwendung des Kalman-Filters seien zudem noch folgende Anmerkungen gegeben:

- Die Matrizen $\boldsymbol{M}$ und $\boldsymbol{N}$ zur Abbildung des System- und Messrauschens stellen die wesentlichen Freiheitsgrade beim KF dar.
- Idealerweise müssen die beiden Rauschprozesse im Vorhinein exakt bekannt sein, um eine entsprechende Parametrierung durchzuführen.

[1] Die Struktur des Kalman-Filters für LTI-Systeme entspricht exakt der eines Luenberger-Beobachters. Bei Letzterem erfolgt allerdings die Bestimmung der Korrekturmatrix durch Polvorgabe, d. h. insbesondere, dass die Korrekturmatrix für ein LTI-System während der Laufzeit konstant ist.

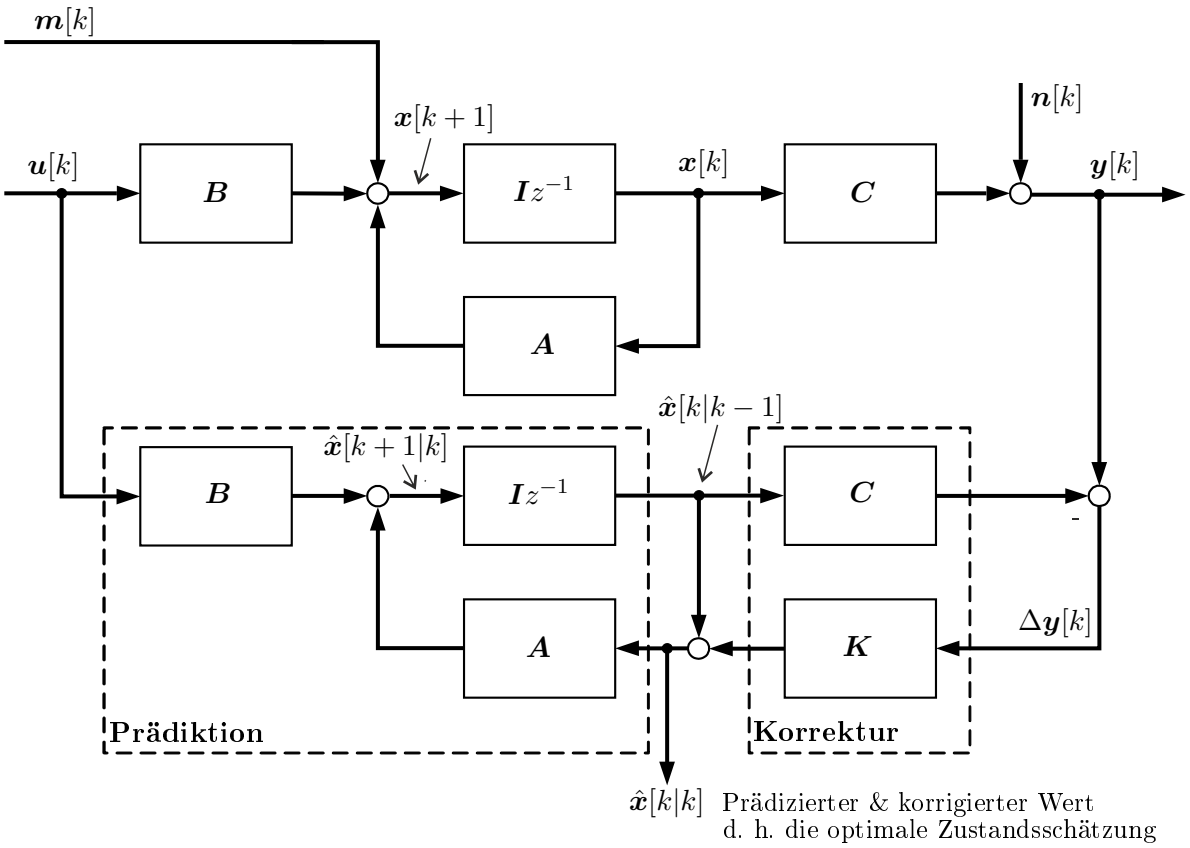


Abb. 5.2: Strukturdiagramm des KFs für zeitdiskrete LTI-Systeme für $\boldsymbol{D} = \boldsymbol{0}$ (vgl. [IM11])

- Diese Parametrierung kann beispielsweise über systemspezifisches Vorwissen oder durch Stichprobenanalyse erfolgen.
- Allerdings können in vielen Systemen die Annahmen aus Definition 5.1 nicht eingehalten werden (z. B. aufgrund nicht idealer Messung mit Gain-/Offset-Fehler).
- In diesen Fällen kann die Wahl von $\boldsymbol{M}$ und $\boldsymbol{N}$ auf die Frage, wie sehr dem Modell bzw. der Messung zu vertrauen ist, reduziert werden.

Analog repräsentiert die Wahl von $\boldsymbol{P}_0$, wie sehr der Zustandsinitialisierung $\boldsymbol{x}_0$ vertraut wird. Für kleine Werte von $\boldsymbol{P}_0$ resultiert zu Beginn des KFs eine nur geringfügige Korrektur über die Kalman-Matrix $\boldsymbol{K}$. Für $k \to \infty$ reduziert sich der Einfluss von $\boldsymbol{P}_0$ zu Null, während $\boldsymbol{M}$ und $\boldsymbol{N}$ die Berechnung von $\boldsymbol{K}$ bestimmen. Es sei daher betont, dass $\boldsymbol{K} = \boldsymbol{K}[k]$ prinzipiell keine konstante Matrix ist, sondern dass diese zur Laufzeit veränderlich ist.

In der Praxis ist das Auffinden einer geschlossenen Auslegungsform für $\boldsymbol{M}$ und $\boldsymbol{N}$ häufig schwierig, insbesondere dann, wenn die KF-Annahmen aus Definition 5.1 (besonders stark) verletzt werden. Während beim Messrauschen häufig noch Datenblattangaben oder empirische Stichproben gezogen werden können, um dass Rauschen zu modellieren, ist dieses beim Systemrauschen i. d. R. nicht ohne Weiteres möglich. Daher findet man in der Praxis häufig *Versuch-und-Irrtum-Ansätze* zur anwendungsspezifischen Auslegung des KFs. Ist es ggf. möglich Experimente zur Auslegung des KFs durchzuführen, können zudem die Optimierungsmethoden aus Kap. 4 herangezogen werden, um einen strukturierten Weg zum Auffinden zielführender Einstellungen für $\boldsymbol{M}$ und $\boldsymbol{N}$ zu beschreiten.

Eine Überprüfungsmöglichkeit des Kalman-Filters zur Laufzeit stellt zudem die Analyse des sog. *Innovationsterms*

$$\boldsymbol{r}[k+1] = \boldsymbol{y}[k+1] - \boldsymbol{C}\hat{\boldsymbol{x}}[k+1|k] \tag{5.27}$$

als Teil von (5.25d) dar. Es kann gezeigt werden, dass dieser unter den obigen KF-Annahmen ein weißes, mittelwertfreies Rauschen mit der Kovarianz

$$\mathrm{Cov}(\boldsymbol{r}[k+1], \boldsymbol{r}[k+1]) = \boldsymbol{A}\boldsymbol{P}[k|k]\boldsymbol{A}^T + \boldsymbol{M} \tag{5.28}$$

darstellt. Sollte der Innovationsterm zur Laufzeit abweichende Eigenschaften aufzeigen, ist dies ein Hinweis auf eine unzureichende Modellierung des Systems oder der Rauschprozesse. Daher kann eine Anpassung der System- und Rauschbeschreibung, welche die beschriebenen Eigenschaften für den Innovationsterm sicherstellen, ebenfalls als Metrik zur Modellbildung bzw. Modellkorrektur im Kalman-Filter-Kontext verstanden werden.

5.1.1 Anpassung für den stationären Fall

Ein Nachteil des KFs ist, dass $\boldsymbol{K}[k]$ zur Laufzeit stetig neu berechnet wird, was zu einem höheren Berechnungsaufwand führt als dies beispielsweise beim strukturäquivalente Luenberger-Beobachter der Fall ist. Mit Blick auf Satz 5.1 wird allerdings deutlich, dass das KF für ein LTI-System mit konstanten System- und Rauschmatrizen für $k \to \infty$ gegen konstante Werte konvergiert. Daher kann es vorteilhaft sein, direkt das stationäre $\boldsymbol{K}[k \to \infty]$ zu berechnen. Hierfür wird zunächst (5.25c) in (5.25e) eingesetzt

$$\boldsymbol{P}[k+1|k+1] = \boldsymbol{P}[k+1|k] - \boldsymbol{P}[k+1|k]\boldsymbol{C}^T \left(\boldsymbol{C}\boldsymbol{P}[k+1|k]\boldsymbol{C}^T + \boldsymbol{N}\right)^{-1} \boldsymbol{C}\boldsymbol{P}[k+1|k], \tag{5.29}$$

was wiederum mit (5.25b) verrechnet werden kann:

$$\boldsymbol{P}[k+1|k+1] = \boldsymbol{A}\boldsymbol{P}[k|k]\boldsymbol{A}^T - \boldsymbol{A}\boldsymbol{P}[k|k]\boldsymbol{C}^T \left(\boldsymbol{C}\boldsymbol{P}[k|k]\boldsymbol{C}^T + \boldsymbol{N}\right)^{-1} \boldsymbol{C}\boldsymbol{P}[k|k]\boldsymbol{A}^T + \boldsymbol{M}. \tag{5.30}$$

Im stationären Zustand gilt dann

$$\boldsymbol{P}[k+1|k+1] = \boldsymbol{P}[k|k] = \boldsymbol{P}, \quad \boldsymbol{K}[k] = \boldsymbol{K} \tag{5.31}$$

und somit in Kurzschreibweise:

$$\boldsymbol{P} = \boldsymbol{A}\boldsymbol{P}\boldsymbol{A}^T - \boldsymbol{A}\boldsymbol{P}\boldsymbol{C}^T \left(\boldsymbol{C}\boldsymbol{P}\boldsymbol{C}^T + \boldsymbol{N}\right)^{-1} \boldsymbol{C}\boldsymbol{P}\boldsymbol{A}^T + \boldsymbol{M}. \tag{5.32}$$

Diese Gleichung weist die Form der zeitdiskrekten Riccati-Gleichung (*discrete-time algebraic Riccati equation* - DARE) auf. Bei dieser wird die unbekannte Matrix $\boldsymbol{X}$, durch Lösen der algebraischen Gleichung

$$\tilde{\boldsymbol{A}}^T\boldsymbol{X}\tilde{\boldsymbol{A}} - \boldsymbol{X} - \tilde{\boldsymbol{A}}^T\boldsymbol{X}\tilde{\boldsymbol{B}} \left(\tilde{\boldsymbol{B}}^T\boldsymbol{X}\tilde{\boldsymbol{B}} + \boldsymbol{R}\right)^{-1} \tilde{\boldsymbol{B}}^T\boldsymbol{X}\tilde{\boldsymbol{A}} + \boldsymbol{Q} = \boldsymbol{0} \tag{5.33}$$

mit entsprechend bekannten Matrizen $\tilde{\boldsymbol{A}}, \tilde{\boldsymbol{B}}, \boldsymbol{R}$ und $\boldsymbol{Q}$, gesucht. Mit Bezug auf (5.32) können folgende Zuordnungen getroffen werden:

$$\boldsymbol{X} = \boldsymbol{P}, \quad \tilde{\boldsymbol{A}} = \boldsymbol{A}^T, \quad \tilde{\boldsymbol{B}} = \boldsymbol{C}^T, \quad \boldsymbol{R} = \boldsymbol{N}, \quad \boldsymbol{Q} = \boldsymbol{M}. \tag{5.34}$$

Die Aufgabe ist ähnlich zum Auffinden eines optimalen Regelgesetzes im Kontext von linear-quadratischen Reglern (LQ-Ansatz). Zur numerischen Lösung stehen vorgefertigte Solver zur Verfügung, z. B. `[X,L,G]=dare(A,B,Q,R)` in Matlab oder der äquivalente Befehl aus der Python Toolbox *scipy*. Nach Erlangung der Lösung für $\boldsymbol{P}$ kann die konstante Kalman-Matrix direkt berechnet werden:

$$\boldsymbol{K} = \boldsymbol{P}\boldsymbol{C}^T \left(\boldsymbol{C}\boldsymbol{P}\boldsymbol{C}^T + \boldsymbol{N}\right)^{-1}. \tag{5.35}$$

Die Kalman-Filter Vorschrift reduziert sich dann zu

Prädiktion:

$$\hat{\boldsymbol{x}}[k+1|k] = \boldsymbol{A}\hat{\boldsymbol{x}}[k] + \boldsymbol{B}\boldsymbol{u}[k] \tag{5.36a}$$

Korrektur:

$$\hat{\boldsymbol{x}}[k+1|k+1] = \hat{\boldsymbol{x}}[k+1|k] + \boldsymbol{K}\left(\boldsymbol{y}[k+1] - \boldsymbol{C}\hat{\boldsymbol{x}}[k+1|k] - \boldsymbol{D}\boldsymbol{u}[k+1]\right) \tag{5.36b}$$

mit entsprechender Initialisierung

$$\hat{\boldsymbol{x}}[0|0] = \boldsymbol{x}_0\,. \tag{5.37}$$

Verglichen mit Satz 5.1 resultiert somit ein erheblich verringerter Berechnungsaufwand in jedem Iterationsschritt.

5.1.2 Anpassung für LPV-Systeme

In vielen technischen Systemen verändern sich die Parameter zur Laufzeit, z. B. aufgrund thermischer Effekte in elektrotechnischen Anwendungen oder generell durch Abnutzung. Handelt es sich hierbei weiterhin um ein lineares System, kann dieses in folgender LPV-Form dargestellt werden:

$$\begin{aligned} \boldsymbol{x}[k+1] &= \boldsymbol{A}[k]\boldsymbol{x}[k] + \boldsymbol{B}[k]\boldsymbol{u}[k] + \boldsymbol{m}[k]\,, \quad \boldsymbol{x}[k=0] = \boldsymbol{x}_0\,, \\ \boldsymbol{y}[k] &= \boldsymbol{C}[k]\boldsymbol{x}[k] + \boldsymbol{D}[k]\boldsymbol{u}[k] + \boldsymbol{n}[k]\,. \end{aligned} \tag{5.38}$$

Analog kann eine zeitvariante Veränderung für die Rauschprozesse abgebildet werden:

$$\mathrm{E}(\boldsymbol{m}[k]\boldsymbol{m}^T[k]) = \boldsymbol{M}[k], \quad \mathrm{E}(\boldsymbol{n}[k]\boldsymbol{n}^T[k]) = \boldsymbol{N}[k]. \tag{5.39}$$

Gelten darüber hinaus die weiteren Annahmen aus Definition 5.1, kann das Kalman-Filter für LPV-Systeme analog zum vorherigen Fall der LTI-Systeme hergeleitet werden. Die KF-Vorschrift ergibt sich dann zu:

Prädiktion:

$$\hat{\boldsymbol{x}}[k+1|k] = \boldsymbol{A}[k]\hat{\boldsymbol{x}}[k] + \boldsymbol{B}[k]\boldsymbol{u}[k]$$
$$\boldsymbol{P}[k+1|k] = \boldsymbol{A}[k]\boldsymbol{P}[k|k]\boldsymbol{A}^T[k] + \boldsymbol{M}[k]$$

Korrektur:

$$\boldsymbol{K}[k+1] = \boldsymbol{P}[k+1|k]\boldsymbol{C}^T[k+1]\left(\boldsymbol{C}[k+1]\boldsymbol{P}[k+1|k]\boldsymbol{C}^T[k+1] + \boldsymbol{N}[k+1]\right)^{-1}$$
$$\hat{\boldsymbol{x}}[k+1|k+1] = \hat{\boldsymbol{x}}[k+1|k] + \boldsymbol{K}[k+1]\left(\boldsymbol{y}[k+1] - \boldsymbol{C}[k+1]\hat{\boldsymbol{x}}[k+1|k] - \boldsymbol{D}[k+1]\boldsymbol{u}[k+1]\right)$$
$$\boldsymbol{P}[k+1|k+1] = \left(\boldsymbol{I} - \boldsymbol{K}[k+1]\boldsymbol{C}[k+1]\right)\boldsymbol{P}[k+1|k]$$

mit entsprechender Initialisierung

$$\boldsymbol{P}[0|0] = \boldsymbol{P}_0, \quad \hat{\boldsymbol{x}}[0|0] = \boldsymbol{x}_0.$$

Durch den LPV-Charakter des Systems muss jeweils der gesamte Prädiktions- und Korrekturschritt berechnet werden, wobei die Kalman-Matrix $\boldsymbol{K}[k]$ in Folge variierende Rausch- und Systemmatrizen nicht in einen stationären Endwert konvergiert. Der erhöhte Berechnungsaufwand gegenüber dem stationären KF muss daher in Kauf genommen werden.

5.1.3 Sequentielle Implementierung

Für das KF gemäß (5.25) muss für die Berechnung der Kalman-Matrix $\boldsymbol{K}[k] \in \mathbb{R}^{n\times p}$ im Schritt (5.25c) eine $\{p \times p\}$ Matrix invertiert werden, dessen Dimension der Anzahl der Messgrößen entspricht. Diese Invertierung kann aus Implementierungssicht anspruchsvoll sein, insbesondere da sie in jedem Rechenschritt erneut ausgeführt werden muss. Daher steht mit dem *sequentiellen Kalman-Filter* eine alternative Implementierungsweise zur Verfügung, welche ohne Invertierung auskommt. Die Idee hierbei ist, dass die Messung $\boldsymbol{y}[k]$ nicht vollständig in einem Schritt zur Korrektur verwendet wird, sondern die einzelnen Messwerteinträge $y_i[k], ..., y_p[k]$ sequentiell nacheinander verarbeitet werden. Hierfür wird die Annahme getätigt, dass die Kovarianzmatrix des Messrauschens Diagonalgestalt aufweise, d. h., die einzelnen Messgrößen sind untereinander unkorreliert:

$$\boldsymbol{N} = \operatorname{diag}\left(\begin{bmatrix} n_{11} & n_{22} & \dots & n_{pp} \end{bmatrix}\right). \tag{5.41}$$

Im Vergleich zu (5.25) verbleibt der Prädiktionsschritt unverändert. Die Korrektur wird hingegen für jeden k-ten Rekursionsschritt wie folgt durchgeführt:

(a) Initialisiere die sequentielle a-posteriori Schätzung mit

$$\hat{\boldsymbol{x}}[k+1|k+1] = \hat{\boldsymbol{x}}_i[k+1|k], \quad \boldsymbol{P}[k+1|k+1] = \boldsymbol{P}[k+1|k]. \tag{5.42}$$

Dies entspricht zunächst der a-priori Schätzung (Prädiktion aus (5.25)), welche nachfolgend sequentiell korrigiert wird.

(b) Für $i = 1, \dots, p$ (mit p als Anzahl der messbaren Zustände), erfolgt die folgende sequentielle Korrektur:

$$\begin{aligned} \boldsymbol{K}_i[k+1] &= \frac{\boldsymbol{P}_{i-1}[k+1|k+1]\boldsymbol{c}_i^T}{\boldsymbol{c}_i\boldsymbol{P}_{i-1}[k+1|k+1]\boldsymbol{c}_i^T + n_{ii}}, \\ \hat{\boldsymbol{x}}_i[k+1|k+1] &= \hat{\boldsymbol{x}}_{i-1}[k+1|k+1] + \boldsymbol{K}_i[k+1]\left(y_i[k] - \boldsymbol{c}_i\hat{\boldsymbol{x}}_{i-1}[k+1|k+1]\right), \\ \boldsymbol{P}_i[k+1|k+1] &= \left(\boldsymbol{I} - \boldsymbol{K}_i[k+1]\boldsymbol{c}_i\right)\boldsymbol{P}_{i-1}[k+1|k+1]. \end{aligned} \tag{5.43}$$

Hierbei ist $\boldsymbol{c}_i$ die i-te Zeile der Ausgangsmatrix $\boldsymbol{C}$ und y_i der i-te Messwert. Die Notation $\boldsymbol{K}_i$, $\boldsymbol{P}_i$ bzw. $\hat{\boldsymbol{x}}_i$ bedeutet hingegen, dass es sich um die jeweilige Matrix bzw. den Vektor nach dem i-ten sequentiell durchgeführten Korrekturschritt handelt.

(c) Nachdem alle p-Korrekturen durchgeführt wurden, kann die finale a-posteriori Schätzung zugeordnet werden:

$$\hat{\boldsymbol{x}}[k+1|k+1] = \hat{\boldsymbol{x}}_p[k+1|k+1], \quad \boldsymbol{P}[k+1|k+1] = \boldsymbol{P}_p[k+1|k+1]. \tag{5.44}$$

Auf eine detaillierte Herleitung der sequentiellen Implementierung wurde an dieser Stelle verzichtet – interessierte Leser seien hier z. B. auf [Sim06] verwiesen. Die anfängliche Annahme einer diagonalen Messrauschmatrix $\boldsymbol{N}$ trifft dann zu, wenn die Messungen untereinander nicht korreliert sind. Für den Fall, dass $\boldsymbol{N}$ keine Diagonalform aufweist aber symmetrisch und positiv definit ist, kann eine Jordan-Zerlegung

$$\boldsymbol{N} = \boldsymbol{S}\tilde{\boldsymbol{N}}\boldsymbol{S}^{-1} \tag{5.45}$$

durchgeführt werden. Hier ist $\tilde{\boldsymbol{N}}$ eine Diagonalmatrix, welche die Eigenwerte von $\boldsymbol{N}$ besitzt und $\boldsymbol{S}$ ist eine orthogonale Matrix mit den Eigenvektoren von $\boldsymbol{N}$. Mit entsprechender Normalisierung der Messungen

$$\begin{aligned} \tilde{\boldsymbol{y}}[k] &= \boldsymbol{S}^{-1}\boldsymbol{y}[k] = \boldsymbol{S}^{-1}\left(\boldsymbol{C}\hat{\boldsymbol{x}}[k] + \boldsymbol{n}[k]\right) \\ &= \tilde{\boldsymbol{C}}\hat{\boldsymbol{x}}[k] + \tilde{\boldsymbol{n}}[k] \end{aligned} \tag{5.46}$$

folgt für die Kovarianz des Messrauschens

$$\begin{aligned} \mathrm{Cov}(\tilde{\boldsymbol{n}}[k], \tilde{\boldsymbol{n}}[k]) &= \mathrm{E}\left(\tilde{\boldsymbol{n}}[k]\tilde{\boldsymbol{n}}^T[k]\right) \\ &= \mathrm{E}\left(\boldsymbol{S}^{-1}\boldsymbol{n}[k]\boldsymbol{n}^T[k]\boldsymbol{S}^{-T}\right) \\ &= \mathrm{E}\left(\boldsymbol{S}^{-1}\boldsymbol{n}[k]\boldsymbol{n}^T[k]\boldsymbol{S}\right) \\ &= \boldsymbol{S}^{-1}\boldsymbol{R}\boldsymbol{S} \\ &= \tilde{\boldsymbol{R}}. \end{aligned} \tag{5.47}$$

Ist die Jordan-Zerlegung möglich, kann somit der Korrekturschritt (5.43) auf Basis der normalisierten Messungen $\tilde{\boldsymbol{y}}[k]$ bzw. Ausgangsmatrix $\tilde{\boldsymbol{C}}$ erfolgen. Allerdings ist eine derartige Zerlegung auch wieder rechenaufwendig, sodass das ursprüngliche Ziel einer schlanken Implementierung nur dann sinnvoll verfolgt werden kann, wenn die Zerlegung einmal in der Vorbereitungsphase durchgeführt wird – sprich das Messrauschen $\boldsymbol{N}$ muss konstant sein. Somit lässt sich das sequentielle KF dann sinnvoll anwenden, falls

- die Messrauschmatrix $\boldsymbol{N}[k]$ Diagonalgestalt aufweist oder
- die Messrauschmatrix $\boldsymbol{N}$ konstant, symmetrisch und positiv definit ist.

Neben dem sequentiellen KF gibt es eine Reihe weiterer Varianten der KF-Implementierung. Erwähnt sei u. a. das sog. *Informations-Filter*, welches anstatt der Verwendung von $\boldsymbol{P}$ dessen Inverse $\boldsymbol{P}^{-1}$ rekursiv anpasst. Dies bewirkt, dass im Korrekturschritt statt einer $\{p \times p\}$ Matrix-Inversion eine Reihe von $\{n \times n\}$ Matrix-Inversionen durchgeführt werden muss. Dies erscheint demnach genau dann zielführend, wenn die Anzahl der Messungen deutlich größer als die Anzahl der Systemzustände ist.

Eine weitere Variante ist das sog. *Square-Root-Filter*, welches nicht direkt $\boldsymbol{P}$, sondern eine daraus abgeleitete Matrix-Zerlegung (z. B. nach Cholesky) nutzt. Es kann gezeigt werden, dass die numerische Stabilität (Kondition der Kovarianzmatrix) des transformierten Problems deutlich verbessert wird, was insbesondere bei Festkomma-Implementierung mit geringen Rechenressourcen von Vorteil sein kann, sofern $\boldsymbol{P}$ schlecht konditioniert ist. Der Zugewinn an numerischer Stabilität wird allerdings durch einen erhöhten Berechnungsaufwand erkauft.

5.2 Das erweiterte Kalman-Filter

Das lineare KF kann ausschließlich auf lineare Systeme angewandt werden. In realen, technischen Anwendungen ist die maßgebliche Anzahl der Probleme hingegen von nichtlinearer Natur. Das *erweiterten Kalman-Filter* (EKF) stellt eine vergleichsweise einfache Möglichkeit dar, um die zuvor hergeleitete KF-Struktur für lineare Systeme auf nichtlineare Probleme zu übertragen. Hier wird zunächst von einer allgemeinen, zeitdiskreten und nichtlinearen Systembeschreibung

$$\begin{aligned} \boldsymbol{x}[k+1] &= \boldsymbol{f}\left(\boldsymbol{x}[k], \boldsymbol{u}[k]\right) + \boldsymbol{m}[k]\,, \quad \boldsymbol{x}[k=0] = \boldsymbol{x}_0\,, \\ \boldsymbol{y}[k+1] &= \boldsymbol{g}\left(\boldsymbol{x}[k+1], \boldsymbol{u}[k+1]\right) + \boldsymbol{n}[k+1]\,, \end{aligned} \tag{5.48}$$

mit additivem Messrauschen entsprechend Definition 5.1 ausgegangen. Kernidee des EKFs ist eine lineare Näherung von (5.48) um den jeweiligen Schätzpunkt. Hierzu wird eine Taylor-Reihe 1. Ordnung herangezogen und die nichtlineare System- bzw. Ausgangsfunktion durch ihre jeweiligen Jacobi-Matrizen angenähert:

$$\boldsymbol{F}[k] = \frac{\partial \boldsymbol{f}\left(\boldsymbol{x}[k], \boldsymbol{u}[k]\right)}{\partial \boldsymbol{x}}\Big|_{\boldsymbol{x}=\hat{\boldsymbol{x}}[k|k], \boldsymbol{u}=\boldsymbol{u}[k]}\,, \tag{5.49}$$

$$\boldsymbol{G}[k+1] = \frac{\partial \boldsymbol{g}\left(\boldsymbol{x}[k+1], \boldsymbol{u}[k+1]\right)}{\partial \boldsymbol{x}}\Big|_{\boldsymbol{x}=\hat{\boldsymbol{x}}[k+1|k], \boldsymbol{u}=\boldsymbol{u}[k+1]}\,. \tag{5.50}$$

Mit dieser Linearisierung kann die bereits bekannte KF-Vorschrift auf das nichtlineare System (5.48) übertragen werden. Die EKF-Vorschrift lautet daher wie folgt:

Satz 5.2: Erweitertes Kalman-Filter für zeitdiskrete, nichtlineare Systeme

Für das zeitdiskrete, nichtlineare System (5.48) *kann ein rekursiver LS-Schätzer zur Minimierung von* (5.6) *unter den Annahmen aus Definition 5.1 durch folgenden linearsierten Prädiktor-Korrektor-Ansatz gewonnen werden:*

Prädiktion:

$$\hat{\boldsymbol{x}}[k+1|k] = \boldsymbol{f}\left(\hat{\boldsymbol{x}}[k], \boldsymbol{u}[k]\right) \tag{5.51a}$$

$$\boldsymbol{F}[k] = \frac{\partial \boldsymbol{f}\left(\boldsymbol{x}[k], \boldsymbol{u}[k]\right)}{\partial \boldsymbol{x}}\Big|_{\boldsymbol{x}=\hat{\boldsymbol{x}}[k|k], \boldsymbol{u}=\boldsymbol{u}[k]} \tag{5.51b}$$

$$\boldsymbol{P}[k+1|k] = \boldsymbol{F}[k]\boldsymbol{P}[k|k]\boldsymbol{F}^T[k] + \boldsymbol{M}[k] \tag{5.51c}$$

Korrektur:

$$\boldsymbol{G}[k+1] = \frac{\partial \boldsymbol{g}\left(\boldsymbol{x}[k+1], \boldsymbol{u}[k+1]\right)}{\partial \boldsymbol{x}}\Big|_{\boldsymbol{x}=\hat{\boldsymbol{x}}[k+1|k], \boldsymbol{u}=\boldsymbol{u}[k+1]} \tag{5.51d}$$

$$\boldsymbol{K}[k+1] = \boldsymbol{P}[k+1|k]\boldsymbol{G}^T[k+1]\left(\boldsymbol{G}[k+1]\boldsymbol{P}[k+1|k]\boldsymbol{G}^T[k+1] + \boldsymbol{N}[k]\right)^{-1} \tag{5.51e}$$

$$\hat{\boldsymbol{x}}[k+1|k+1] = \hat{\boldsymbol{x}}[k+1|k] + \boldsymbol{K}[k+1]\left(\boldsymbol{y}[k+1] - \boldsymbol{g}\left(\hat{\boldsymbol{x}}[k+1|k], \boldsymbol{u}[k+1]\right)\right) \tag{5.51f}$$

$$\boldsymbol{P}[k+1|k+1] = \left(\boldsymbol{I} - \boldsymbol{K}[k+1]\boldsymbol{G}[k+1]\right)\boldsymbol{P}[k+1|k] \tag{5.51g}$$

mit entsprechender Initialisierung

$$\boldsymbol{P}[0|0] = \boldsymbol{P}_0, \quad \hat{\boldsymbol{x}}[0|0] = \boldsymbol{x}_0. \tag{5.52}$$

Gegenüber dem linearen KF ist zu betonen, dass das EKF kein optimaler Schätzer ist. Durch die Linearisierung wird die zugrundeliegende Zufallsverteilung der Rauschprozesse nicht korrekt abgebildet und es kommt zu systematischen Abweichungen zwischen Modell und System. Sollte in einem Iterationsschritt zudem ein vergleichsweise weiter Bereich innerhalb des Zustandsraums überschritten werden, ist es möglich, dass das EKF divergiert, da die Linearisierung um den letzten Schätzpunkt das Systemverhalten nicht ausreichend genau abbildet.

5.2.1 Parameterschätzung mittels Zustandsaugmentation

Das EKF sowie auch weitere nichtlineare KF-Varianten sind prinzipiell in der Lage, Parameter $\boldsymbol{\theta}$ des Systemmodells zur Laufzeit zu identifizieren bzw. nachzuführen. Hierfür wird eine augmentierte[1] Modellbeschreibung herangezogen:

$$\begin{aligned} \hat{\tilde{\boldsymbol{x}}}[k+1] = \begin{bmatrix} \hat{\boldsymbol{x}}[k+1] \\ \hat{\boldsymbol{\theta}}[k+1] \end{bmatrix} = \begin{bmatrix} \boldsymbol{f}\left(\hat{\boldsymbol{x}}[k], \hat{\boldsymbol{\theta}}[k], \boldsymbol{u}[k]\right) \\ \hat{\boldsymbol{\theta}}[k] \end{bmatrix} = \tilde{\boldsymbol{f}}\left(\hat{\boldsymbol{x}}[k], \hat{\boldsymbol{\theta}}[k], \boldsymbol{u}[k]\right), \\ \hat{\boldsymbol{y}}[k+1] = \boldsymbol{g}\left(\hat{\boldsymbol{x}}[k+1], \boldsymbol{u}[k+1]\right). \end{aligned} \tag{5.53}$$

Für die Parameter selbst wird ebenfalls ein additives Rauschen angenommen, sodass sich folgendes tatsächliches System

$$\begin{aligned} \tilde{\boldsymbol{x}}[k+1] = \begin{bmatrix} \boldsymbol{x}[k+1] \\ \boldsymbol{\theta}[k+1] \end{bmatrix} = \begin{bmatrix} \boldsymbol{f}\left(\boldsymbol{x}[k], \boldsymbol{\theta}[k], \boldsymbol{u}[k]\right) + \boldsymbol{m}[k] \\ \boldsymbol{\theta}[k] + \boldsymbol{m}_{\boldsymbol{\theta}}[k] \end{bmatrix} = \tilde{\boldsymbol{f}}\left(\boldsymbol{x}[k], \boldsymbol{\theta}[k], \boldsymbol{u}[k]\right) + \tilde{\boldsymbol{m}}, \\ \boldsymbol{y}[k+1] = \boldsymbol{g}\left(\boldsymbol{x}[k+1], \boldsymbol{u}[k+1]\right) \end{aligned} \tag{5.54}$$

mit $\boldsymbol{m}_{\boldsymbol{\theta}}[k]$ als Rauschterme der modellierten Parameter ergibt. Demnach resultiert eine augmentierte Systemrauschmatrix

$$\mathrm{E}(\tilde{\boldsymbol{m}}[k]\tilde{\boldsymbol{m}}^T[k]) = \tilde{\boldsymbol{M}}[k], \tag{5.55}$$

welche sowohl das Rauschverhalten der eigentlichen Zustände $\boldsymbol{x}$ als auch der Parameter $\boldsymbol{\theta}$ beschreibt[2]. Im Prädiktionsschritt des EKFs werden die Parameter $\boldsymbol{\theta}$ nicht verändert und somit als Konstanten behandelt. Die (augmentierte) Jacobi-Matrix für den Prädiktionsschritt ergibt sich dann zu:

$$\tilde{\boldsymbol{F}}[k] = \begin{bmatrix} \frac{\partial \boldsymbol{f}\left(\hat{\boldsymbol{x}}[k], \hat{\boldsymbol{\theta}}[k], \boldsymbol{u}[k]\right)}{\partial \hat{\boldsymbol{x}}[k]} & \frac{\partial \boldsymbol{f}\left(\hat{\boldsymbol{x}}[k], \hat{\boldsymbol{\theta}}[k], \boldsymbol{u}[k]\right)}{\partial \hat{\boldsymbol{\theta}}[k]} \\ \frac{\partial \hat{\boldsymbol{\theta}}[k]}{\partial \hat{\boldsymbol{x}}[k]} & \frac{\partial \hat{\boldsymbol{\theta}}[k]}{\partial \hat{\boldsymbol{\theta}}[k]} \end{bmatrix} = \begin{bmatrix} \frac{\partial \boldsymbol{f}\left(\hat{\boldsymbol{x}}[k], \hat{\boldsymbol{\theta}}[k], \boldsymbol{u}[k]\right)}{\partial \hat{\boldsymbol{x}}[k]} & \frac{\partial \boldsymbol{f}\left(\hat{\boldsymbol{x}}[k], \hat{\boldsymbol{\theta}}[k], \boldsymbol{u}[k]\right)}{\partial \hat{\boldsymbol{\theta}}[k]} \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix}. \tag{5.56}$$

Die weiteren Schritte des EKFs gemäß Satz 5.2 verbleiben unverändert. Durch das modellierte Parameterrauschen in (5.53) wird ein Teil der System- und Messunsicherheit in $\boldsymbol{P}[k+1|k]$ sowie $\boldsymbol{P}[k+1|k+1]$ durch die Parameter $\boldsymbol{\theta}$ abgebildet. Dies wiederum hat zur Folge, dass in (5.51e) bzw. (5.51f) ein Teil dieser Unsicherheit durch den Korrekturschritt behoben wird, sodass es zu einer dynamischen Anpassung von $\boldsymbol{\theta}$ kommt, bis dessen Varianzanteil nur noch einem mittelwertfreien Rauschen entspricht.

---

[1] Augementation: abgeleitet von dem spätlateinischen Substantiv „augmentatio“ (deutsch: die Vermehrung).

[2] Wird angenommen, dass alle Zustände und Parameter unkorreliert seien, hat $\tilde{\boldsymbol{M}}[k]$ Diagonalform. In diesem Fall können die entsprechenden Einträge $\tilde{m}_{ii}$ zum jeweiligen Parameter θ_i als Maß der Unsicherheit interpretiert werden, mit dem dieser Parameter bekannt sei.

Es sei zudem angemerkt, dass es ebenfalls für die Parameterschätzung mittels EKF keinerlei Garantien auf Konvergenz oder Optimalität gibt. Durch die Linearisierung kann das Filter und somit die Parameterschätzung sogar divergieren und instabil werden. Um dies nach Möglichkeit zu vermeiden, sollte zumindest sichergestellt sein, dass sich die Initialisierung $\boldsymbol{\theta_0} = \boldsymbol{\theta}[k=0]$ in hinreichender Nähe zu $\boldsymbol{\theta}^*$ befindet. Die Parameterschätzung mittels augmentierten EKFs hat sich daher u. a. besonders bei linearen Systemen zur Nachführung von veränderlichen Parametern etabliert, sofern gesichertes Wissens über die zu beobachtenden Parameter zum Startzeitpunkt verfügbar ist[1]. Typische Beispiele sind temperaturveränderliche Widerstandswerte oder die Veränderung von Induktivitätswerten in Folge magnetischer Sättigung in Elektromotoren.

5.2.2 Iterierendes EKF

Das iterierende EKF (*iterated EKF* - IEKF) stellt eine Variante des EKFs dar, bei dem der Korrekturschritt N-fach ausgeführt wird. Dies wird beispielhaft in Abb. 5.3 verdeutlicht: Nach erfolgter Prädiktion wird am Punkt $\boldsymbol{x}[k+1|k]$ die Korrektur mehrfach ausgeführt, bis eine gegebene Iterationsanzahl erreicht oder die Zustandsänderung zwischen den Iterationsschritten hinreichend klein ist.

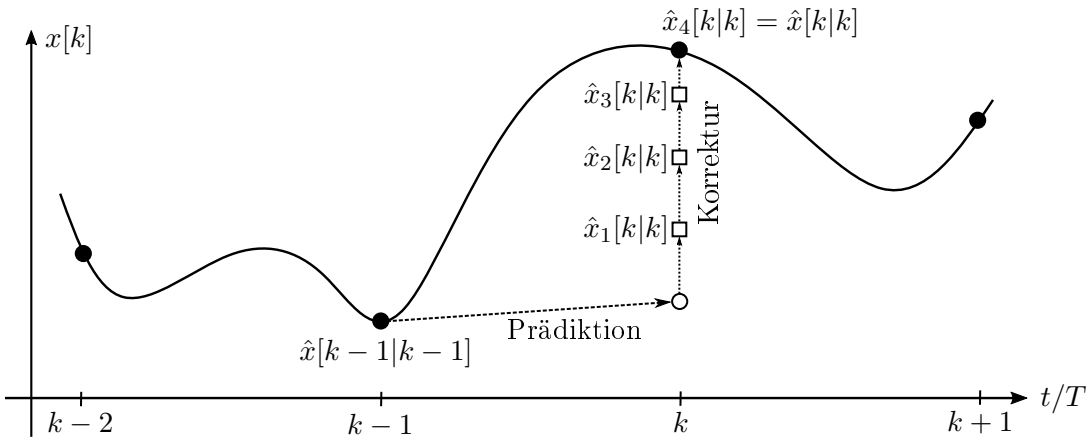


Abb. 5.3: Illustration zum iterierenden EKF (IEKF) am Beispiel eines Systems mit einem Zustand (vgl. [Hol18])

Der entsprechende Algorithmus zum IEKF ist zudem im Algorithmus 5.5 bzw. im Schaubild Abb. 5.4 in allgemeiner Form dargestellt. Das IEKF führt demnach die Linearisierung der Ausgangsgleichung am neuen Schätzpunkt wiederholt aus, um eine verbesserte Korrektur zu erhalten. Der Prädiktionsschritt verbleibt allerdings vollkommen gleich, sodass das IEKF nur für Systeme mit signifikanter Nichtlinearität in der Ausgangsgleichung von Vorteil sein kann. Sind demgegenüber die Zustände vollständig messbar oder (nahezu) linear mit den Ausgangsgrößen verknüpft, bietet das IEKF gegenüber dem EKF keinerlei Vorteile. Zudem sei in Abgrenzung zu Abb. 5.3 darauf hingewiesen, dass bei nichtlinearen und mehrdimensionalen Problemen die wiederholende Ausgangslinearisierung nicht zwangsläufig zum wahren Zustandswert führen muss. Ist der prädizierte Zustand zu Beginn der iterierenden Korrektur nicht in hinreichender Nähe zum wahren Zustand, kann eine Iteration in ein lokales Nebenminimum erfolgen.

[1] Beachte: Durch die Zustandsaugmentation wird ein ursprünglich lineares Systemmodell nichtlinear, sodass die Parameterschätzung mittels des linearen KFs prinzipbedingt nicht möglich ist.

Algorithmus 5.5 Iterierendes EKF

Initialisierung:

1: $\hat{\boldsymbol{x}}_0 = \boldsymbol{x}[k=0], \hat{\boldsymbol{P}}_0 = \boldsymbol{P}[k=0]$

2: $k = 0$

Prädiktion:

3: $\hat{\boldsymbol{x}}[k+1|k] = \boldsymbol{f}\left(\hat{\boldsymbol{x}}[k], \boldsymbol{u}[k]\right)$

4: $\boldsymbol{F}[k] = \frac{\partial \boldsymbol{f}(\boldsymbol{x}[k], \boldsymbol{u}[k])}{\partial \boldsymbol{x}}\Big|_{\boldsymbol{x}=\hat{\boldsymbol{x}}[k|k], \boldsymbol{u}=\boldsymbol{u}[k]}$

5: $\boldsymbol{P}[k+1|k] = \boldsymbol{F}[k]\boldsymbol{P}[k|k]\boldsymbol{F}^T[k] + \boldsymbol{M}[k]$

Korrektur:

6: $\boldsymbol{G}_0[k+1] = \frac{\partial \boldsymbol{g}(\boldsymbol{x}[k+1], \boldsymbol{u}[k+1])}{\partial \boldsymbol{x}}\Big|_{\boldsymbol{x}=\hat{\boldsymbol{x}}[k+1|k], \boldsymbol{u}=\boldsymbol{u}[k+1]}$

7: $\hat{\boldsymbol{x}}[k+1|k+1] = \hat{\boldsymbol{x}}[k+1|k]$

8: **while** $i \leq N$ **do**

9: $\quad \boldsymbol{G}_i[k+1] = \frac{\partial \boldsymbol{g}(\boldsymbol{x}[k+1], \boldsymbol{u}[k+1])}{\partial \boldsymbol{x}}\Big|_{\boldsymbol{x}=\hat{\boldsymbol{x}}_{i-1}[k+1|k+1], \boldsymbol{u}=\boldsymbol{u}[k+1]}$

10: $\quad \boldsymbol{K}_i[k+1] = \boldsymbol{P}[k+1|k]\boldsymbol{G}_i^T[k+1]\left(\boldsymbol{G}_i[k+1]\boldsymbol{P}[k+1|k]\boldsymbol{G}_i^T[k+1] + \boldsymbol{N}[k]\right)^{-1}$

11: $\quad \hat{\boldsymbol{x}}_i[k+1|k+1] = \hat{\boldsymbol{x}}_i[k+1|k] + \boldsymbol{K}_i[k+1]\left(\boldsymbol{y}[k+1] - \boldsymbol{g}\left(\hat{\boldsymbol{x}}_i[k+1|k], \boldsymbol{u}[k+1]\right)\right)$

12: $\quad \boldsymbol{P}_i[k+1|k+1] = \left(\boldsymbol{I} - \boldsymbol{K}_i[k+1]\boldsymbol{G}_i[k+1]\right)\boldsymbol{P}[k+1|k]$

13: **end while**

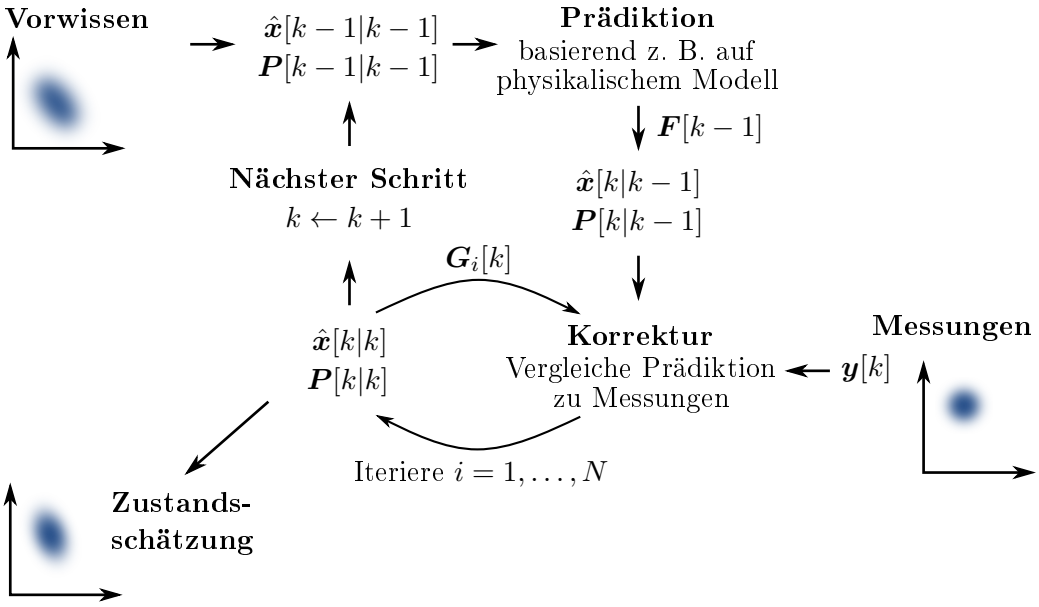


Abb. 5.4: Prinzipielles Vorgehen beim IEKF

5.3 Das Unscented Kalman-Filter

Die Kernidee des sogenannten *unscented Kalman-filters*[1] (UKF) lässt sich einfach zusammenfassen: Anstatt die nichtlineare Systemdynamik zu linearisieren und so die Rechenregeln des linearen KFs anwenden zu können (EKF-Ansatz), werden sorgsam gewählte Stichprobenpunkte (*Sigma-Punkte*) direkt durch die nichtlinearen System- und Korrekturgleichungen propagiert. Der Mittelwert und die Kovarianz der Zustände werden dann unmittelbar aus dieser Stichprobe rekonstruiert. Hierdurch wird der nichtlineare Charakter des Systems berücksichtigt, allerdings i. d. R. auf Kosten eines höheren Berechnungsaufwands verglichen mit dem des EKF. Das grundsätzliche Vorgehen wird in Abb. 5.5 illustriert.

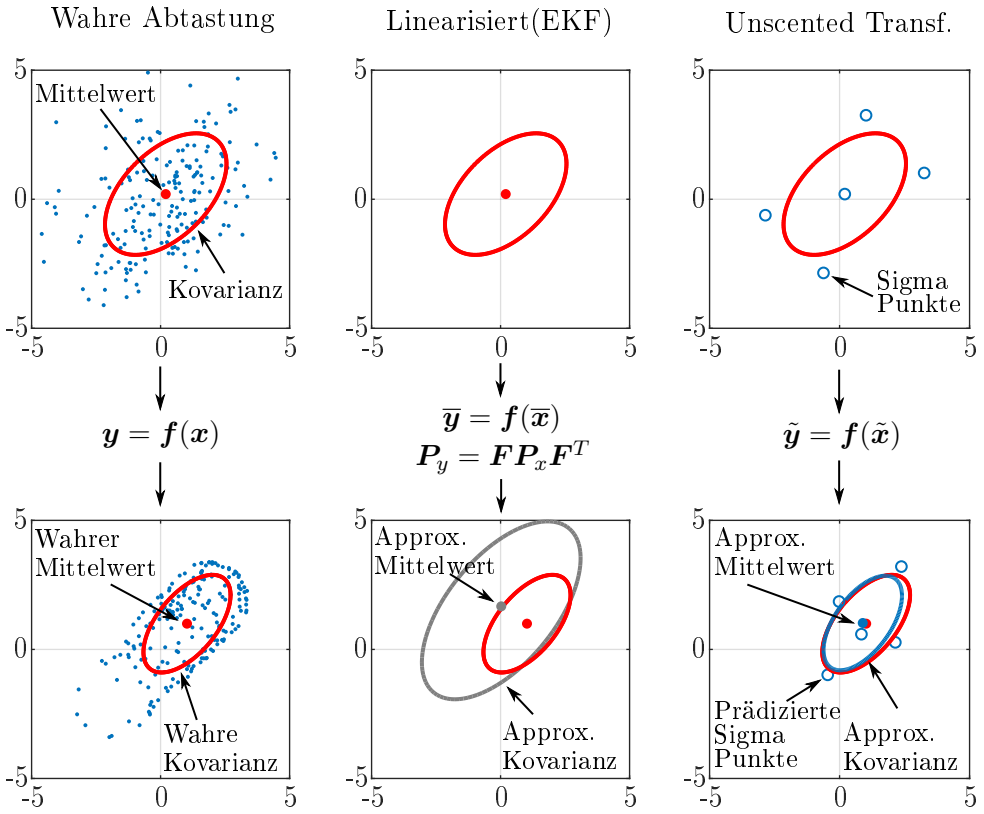


Abb. 5.5: Prinzipielle Unterschiede der verschiedenen Methoden zur Prädiktion von Mittelwert und Kovarianz in nichtlinearen Systemen

Zunächst soll die Unscented-Transformation (UT) anhand des nichtlinearen Beispiels

$$\boldsymbol{y} = \boldsymbol{f}(\boldsymbol{x}) \tag{5.57}$$

[1] Die direkte Übersetzung *unparfümiert* ist nicht selbsterklärend. Der Namensgeber und Erfinder Jeffrey Uhlmann beantwortete die Frage hinsichtlich der Namensgebung im Zuge eines Interviews zur Veröffentlichung [JU04] wie folgt: Zu Beginn hätten seine Mitarbeiter das Filter als *Uhlmann-Filter* bezeichnet, was er so nicht akzeptieren konnte. Als er eines Abends allein im Labor war, fiel sein Blick auf das zurückgelassene Deo eines Kollegen mit der Aufschrift *unscented*, was nichts mit dem eigentlichen Filter zu tun hat, aber dennoch für Uhlmann den perfekten Begriff darstellte. Dieser etablierte sich, trotz anfänglicher Ablehnung, in der Fachwelt als technischer Fachbegriff zügig. Das UKF wird allgemein auch der Klasse der Sigma-Punkt Filter zugeordnet, welche noch eine Reihe weiterer Filter-Ansätze zur Transformation der statistischen Eigenschaften einer Zufallsvariablen durch eine nichtlineare Funktion mittels einer geringen, deterministischen Anzahl an Stützstellen beinhaltet.

verdeutlicht werden. Es gelte $\boldsymbol{f} : \mathbb{R}^n \to \mathbb{R}^n$ und $\{\boldsymbol{x}, \boldsymbol{y}\} \in \mathbb{R}^n$ werden als Zufallsvariablen mit gegebener Wahrscheinlichkeitsverteilung angenommen. Der Mittelwert $\overline{\boldsymbol{x}}$ sowie die Kovarianz $\boldsymbol{P}_x$ der Zufallsvariable $\boldsymbol{x}$ seien bekannt. Ziel der UT ist es, den Mittelwert $\overline{\boldsymbol{y}}$ sowie die Kovarianz $\boldsymbol{P}_y$ der transformierten Zufallsvariable $\boldsymbol{y}$ abzuschätzen. Auf eine vollständige Herleitung der UT-Vorschrift wird an dieser Stelle verzichtet – für diese sei auf [Sim06] verwiesen. Die UT-Abfolge lautet für das Beispiel (5.57) wie folgt:

1. Berechne $2n$ Sigma-Punkt-Vektoren $\boldsymbol{x}^{(i)}$ gemäß

$$\begin{aligned} \boldsymbol{x}^{(i)} &= \overline{\boldsymbol{x}} + \tilde{\boldsymbol{x}}^{(i)}, \quad i = 1, \ldots, 2n\,, \\ \tilde{\boldsymbol{x}}^{(i)} &= \left(\sqrt{n\boldsymbol{P}_x}\right)_i^T, \quad i = 1, \ldots, n\,, \\ \tilde{\boldsymbol{x}}^{(i)} &= -\left(\sqrt{n\boldsymbol{P}_x}\right)_{i-n}^T, \quad i = (n+1), \ldots, 2n\,, \end{aligned} \tag{5.58}$$

   mit der Quadratwurzel der Matrix $(n\boldsymbol{P}_x) = \left(\sqrt{n\boldsymbol{P}_x}\right)^T \left(\sqrt{n\boldsymbol{P}_x}\right)$ und $\left(\sqrt{n\boldsymbol{P}_x}\right)_i$ als i-te Reihe von $\left(\sqrt{n\boldsymbol{P}_x}\right)$.

2. Propagiere die Sigma-Punkte durch das (nichtlineare) Systemmodell:

$$\boldsymbol{y}^{(i)} = \boldsymbol{f}(\boldsymbol{x}^{(i)}), \quad i = 1, \ldots, 2n. \tag{5.59}$$

3. Rekonstruiere den Mittelwert und die Kovarianz von $\boldsymbol{y}$ aus der Stichprobe:

$$\begin{aligned} \hat{\overline{\boldsymbol{y}}} &= \frac{1}{2n} \sum_{i=1}^{2n} \boldsymbol{y}^{(i)}\,, \\ \hat{\boldsymbol{P}}_y &= \frac{1}{2n} \sum_{i=1}^{2n} \left(\boldsymbol{y}^{(i)} - \hat{\overline{\boldsymbol{y}}}\right) \left(\boldsymbol{y}^{(i)} - \hat{\overline{\boldsymbol{y}}}\right)^T. \end{aligned} \tag{5.60}$$

Zur UT-Anwendung seien folgende Hinweise gegeben:

- Die Berechnung der Matrixwurzel in (5.58) stellt i. d. R. ein numerisch aufwendiges Vorhaben dar. Nur in Spezialfällen (z. B. wenn $\boldsymbol{P}$ eine Diagonalmatrix ist) lässt sich dieses mit vergleichsweise geringem Aufwand realisieren. Im Allgemeinen bieten sich zu dessen Lösung daher entsprechende Matrixzerlegungen an, wie beispielsweise die Cholesky-Zerlegung (siehe Kap. 3.1.4).
- Bei der Definition der Sigma-Punkte $\boldsymbol{x}^{(i)}$ in (5.58) wurde die Wahl derart getroffen, dass der empirische Mittelwert und die Kovarianz der Sigma-Punkte denen der eigentlichen Zufallsvariable $\boldsymbol{x}$ entsprechen. In Kap. 5.3.1 werden noch weitere Varianten hinsichtlich der Sigma-Punkte Verteilung der UT vorgestellt, welche sich hinsichtlich Rechenaufwand und Genauigkeit der Schätzung unterscheiden.
- Es sei angemerkt, dass alle UKF-Varianten stets eine Approximation von Mittelwert und Kovarianz, also der ersten beiden statistischen Momente, zum Ziel haben. Es wird somit angenommen, dass sich die zugrundeliegende Wahrscheinlichkeitsverteilung zielführend durch diese Parameter beschreiben lässt. Sollte hingegen eine Verteilung vorliegen, bei der dies nicht der Fall ist (z. B. multimodale Verteilung), sind KF-basierte Ansätze

nicht zielführend. Dann muss die Verteilung umfassender und somit aufwendiger abgeschätzt werden, was typischerweise durch die *Sequenzielle Monte-Carlo-Methode* (oder auch *Partikel-Filter* genannt) realisiert wird[1].

Mit der UT lässt sich der Prädiktionsschritt des KFs unmittelbar ausführen, wobei hier eine Schätzung sowohl des Mittelwerts und der Kovarianz des prädizierten Zustands als auch der prädizierten Ausgangsgrößen resultiert.

Nun muss noch der Korrekturschritt für das UKF hergeleitet werden. Gemäß des Ansatzes aus (5.16) wird hierfür die korrigierte Kovarianzmatrix $\boldsymbol{P}[k+1|k+1]$ benötigt:

$$\boldsymbol{P}[k+1|k+1] = \mathrm{Cov}\left(\hat{\boldsymbol{x}}[k+1|k+1], \hat{\boldsymbol{x}}[k+1|k+1]\right) = \mathrm{Var}\left(\hat{\boldsymbol{x}}[k+1|k+1]\right). \tag{5.61}$$

Die hierfür notwendige Zustandsschätzung $\hat{\boldsymbol{x}}[k+1|k+1]$ ergibt sich aus dem bekannten KF-Korrekturansatz (5.13):

$$\hat{\boldsymbol{x}}[k+1|k+1] = \hat{\boldsymbol{x}}[k+1|k] + \boldsymbol{K}[k+1](\boldsymbol{y}[k+1] - \hat{\boldsymbol{y}}[k+1|k]). \tag{5.62}$$

Durch die allgemeine Beziehung

$$\mathrm{Var}\left(\boldsymbol{x}+\boldsymbol{y}\right) = \mathrm{Var}\left(\boldsymbol{x}\right) + \mathrm{Var}\left(\boldsymbol{y}\right) + 2\mathrm{Cov}\left(\boldsymbol{x},\boldsymbol{y}\right) \tag{5.63}$$

folgt dann:

$$\begin{aligned}\boldsymbol{P}[k+1|k+1] = \mathrm{Var}\left(\hat{\boldsymbol{x}}[k+1|k]\right) + \mathrm{Var}\left(\boldsymbol{K}[k+1](\boldsymbol{y}[k+1] - \hat{\boldsymbol{y}}[k+1|k])\right) \\ + 2\mathrm{Cov}\left(\hat{\boldsymbol{x}}[k+1|k], \boldsymbol{K}[k+1](\boldsymbol{y}[k+1] - \hat{\boldsymbol{y}}[k+1|k])\right).\end{aligned} \tag{5.64}$$

In (5.64) entspricht der erste Term gerade der Kovarianzmatrix des prädizierten Zustands:

$$\mathrm{Var}\left(\hat{\boldsymbol{x}}[k+1|k]\right) = \boldsymbol{P}_{\hat{x}}[k+1|k]. \tag{5.65}$$

Für den Innovationsterm folgt

$$\begin{aligned}\mathrm{Var}\left(\boldsymbol{y}[k+1] - \hat{\boldsymbol{y}}[k+1|k]\right) &= \mathrm{Var}\left(\boldsymbol{y}[k+1|k]\right) + \mathrm{Var}\left(\hat{\boldsymbol{y}}[k+1]\right) - 2\mathrm{Cov}\left(\boldsymbol{y}[k+1], \hat{\boldsymbol{y}}[k+1|k]\right) \\ &= \tilde{\boldsymbol{P}}_{\hat{y}}[k+1|k] + \boldsymbol{N}[k] \\ &= \boldsymbol{P}_{\hat{y}}[k+1|k]\end{aligned}$$

mit der Kovarianz

$$\tilde{\boldsymbol{P}}_{\hat{y}}[k+1|k] = \frac{1}{2n}\sum_{i=1}^{2n}\left(\hat{\boldsymbol{y}}^{(i)}[k+1|k] - \hat{\boldsymbol{y}}[k+1|k]\right)\left(\hat{\boldsymbol{y}}^{(i)}[k+1|k] - \hat{\boldsymbol{y}}[k+1|k]\right)^T \tag{5.66}$$

der Sigma-Punkte der prädizierten Ausgänge. Das obige Ergebnis folgt unmittelbar aus der Annahme, dass ein additives Messrauschen auf dem Systemausgang liegt, d. h., die Gesamtvarianz resultiert aus dem Rauschanteil plus der inhärenten Varianz der Sigma-Punkte Verteilung.

[1] Das prinzipielle Vorgehen ist allerdings ähnlich wie beim UKF, d. h., es wird eine Anzahl an Stichprobenpunkten durch das nichtlineare System propagiert und danach analysiert, wobei der Umfang der Stichprobe beim Partikel-Filter i. A. deutlich umfangreicher ist wie beim UKF.

Unter Berücksichtigung der Multiplikation mit $\boldsymbol{K}$ folgt dann[1]:

$$\mathrm{Var}\left(\boldsymbol{K}[k+1](\boldsymbol{y}[k+1]-\hat{\boldsymbol{y}}[k+1|k])\right)=\boldsymbol{K}[k+1]\left(\boldsymbol{P}_{\hat{y}}[k+1|k]\right)\boldsymbol{K}^T[k+1]. \tag{5.67}$$

Unter Ausnutzung der Bilinearität[2] der Kovarianz sowie der Annahme der Unkorreliertheit zwischen Messungen und Zuständen folgt für den letzten Term in (5.64)

$$\begin{aligned}\mathrm{Cov}\left(\hat{\boldsymbol{x}},\boldsymbol{K}(\boldsymbol{y}-\hat{\boldsymbol{y}})\right)&=\mathrm{Cov}\left(\hat{\boldsymbol{x}},\boldsymbol{K}\boldsymbol{y}\right)-\mathrm{Cov}\left(\hat{\boldsymbol{x}},\boldsymbol{K}\hat{\boldsymbol{y}}\right)\\&=\underbrace{\mathrm{Cov}\left(\hat{\boldsymbol{x}},\boldsymbol{y}\right)}_{=\boldsymbol{0}}\boldsymbol{K}^T-\underbrace{\mathrm{Cov}\left(\hat{\boldsymbol{x}},\hat{\boldsymbol{y}}\right)}_{\boldsymbol{P}_{\hat{x}\hat{y}}}\boldsymbol{K}^T\end{aligned} \tag{5.68}$$

mit der Kovarianz

$$\boldsymbol{P}_{\hat{x}\hat{y}}=\boldsymbol{P}_{\hat{x}\hat{y}}[k+1|k]=\frac{1}{2n}\sum_{i=1}^{2n}\left(\hat{\boldsymbol{x}}^{(i)}[k+1|k]-\hat{\boldsymbol{x}}[k+1|k]\right)\left(\hat{\boldsymbol{y}}^{(i)}[k+1|k]-\hat{\boldsymbol{y}}[k+1|k]\right)^T \tag{5.69}$$

zwischen den prädizierten Sigma-Punkten der Zustände $\hat{\boldsymbol{x}}^{(i)}$ sowie den prädizierten Sigma-Punkten der Ausgänge $\hat{\boldsymbol{y}}^{(i)}$. Einsetzen der obigen Terme in (5.64) ergibt dann:

$$\boldsymbol{P}[k+1|k+1]=\boldsymbol{P}_{\hat{x}}[k+1|k]+\boldsymbol{K}[k+1]\left(\boldsymbol{P}_{\hat{y}}[k+1|k]\right)\boldsymbol{K}^T[k+1]-2\boldsymbol{P}_{\hat{x}\hat{y}}[k+1|k]\boldsymbol{K}^T[k+1]. \tag{5.70}$$

Entsprechend (5.16) muss zur Minimierung der quadratischen Kostenfunktion $J[k+1]$ die Spur von $\boldsymbol{P}[k+1|k+1]$ minimiert werden. Es folgt demnach unter Verwendung von Anhang A.3:

$$\frac{\partial}{\partial\boldsymbol{K}}\mathrm{Spur}\left(\boldsymbol{P}[k+1|k+1]\right)=2\boldsymbol{K}[k+1]\boldsymbol{P}_{\hat{y}}[k+1|k]-2\boldsymbol{P}_{\hat{x}\hat{y}}[k+1|k]. \tag{5.71}$$

Nullsetzen und Auflösung nach $\boldsymbol{K}$ ergibt dann:

$$\boldsymbol{K}[k+1]=\boldsymbol{P}_{\hat{x}\hat{y}}[k+1|k]\left(\boldsymbol{P}_{\hat{y}}[k+1|k]\right)^{-1}. \tag{5.72}$$

Die gefundene Kalman-Matrix kann dann in (5.62) genutzt werden, um die prädizierte Zustandsschätzung zu korrigieren. Durch Einsetzen und Ausmultiplizieren in (5.70) ergibt sich dann zudem die prädizierte und korrigierte Kovarianzmatrix zu:

$$\boldsymbol{P}[k+1|k+1]=\boldsymbol{P}[k+1|k]-\boldsymbol{K}[k+1]\boldsymbol{P}_{\hat{y}}[k+1|k]\boldsymbol{K}^T[k+1]. \tag{5.73}$$

Somit lässt sich der UKF-Algorithmus wie folgt zusammenfassen:

Unscented Transformation

$$\boldsymbol{x}^{(i)}[k|k]=\overline{\boldsymbol{x}}[k|k]+\tilde{\boldsymbol{x}}^{(i)},\quad i=1,\ldots,2n$$

$$\tilde{\boldsymbol{x}}^{(i)}[k|k]=\left(\sqrt{n\boldsymbol{P}[k|k]}\right)_i^T,\quad i=1,\ldots,n$$

$$\tilde{\boldsymbol{x}}^{(i)}[k|k]=-\left(\sqrt{n\boldsymbol{P}[k|k]}\right)_i^T,\quad i=1,\ldots,n$$

[1] Seien $\boldsymbol{X}$ und $\boldsymbol{Y}$ Zufallsvariablen, $\boldsymbol{A}$ und $\boldsymbol{B}$ konstante Matrizen sowie a und b skalare Konstanten. Dann folgt: $\mathrm{Cov}\left(\boldsymbol{A}\boldsymbol{X}+a,\boldsymbol{B}\boldsymbol{Y}+b\right)=\boldsymbol{A}\mathrm{Cov}\left(\boldsymbol{X},\boldsymbol{Y}\right)\boldsymbol{B}^T$.

[2] Gegeben seien die Zufallsvariablen $\boldsymbol{X}$, $\boldsymbol{Y}$ und $\boldsymbol{Z}$ sowie die Konstanten $\{a,b,c,d\}\in\mathbb{R}$. Dann gilt für die Kovarianz: $\mathrm{Cov}\left(\boldsymbol{X},(a\boldsymbol{Y}+b)+(c\boldsymbol{Z}+d)\right)=a\mathrm{Cov}\left(\boldsymbol{X},\boldsymbol{Y}\right)+c\mathrm{Cov}\left(\boldsymbol{X},\boldsymbol{Z}\right)$.

Prädiktion:

$$\hat{\boldsymbol{x}}^{(i)}[k+1|k] = \boldsymbol{f}\left(\boldsymbol{x}^{(i)}[k|k], \boldsymbol{u}[k]\right)$$

$$\hat{\boldsymbol{x}}[k+1|k] = \frac{1}{2n}\sum_{i=1}^{2n}\hat{\boldsymbol{x}}^{(i)}[k+1|k]$$

$$\boldsymbol{P}_{\hat{x}}[k+1|k] = \frac{1}{2n}\sum_{i=1}^{2n}\left(\hat{\boldsymbol{x}}^{(i)}[k+1|k] - \hat{\boldsymbol{x}}[k+1|k]\right)\left(\hat{\boldsymbol{x}}^{(i)}[k+1|k] - \hat{\boldsymbol{x}}[k+1|k]\right)^T + \boldsymbol{M}[k+1]$$

$$\hat{\boldsymbol{y}}^{(i)}[k+1|k] = \boldsymbol{g}\left(\boldsymbol{x}^{(i)}[k+1|k], \boldsymbol{u}[k+1]\right)$$

$$\hat{\boldsymbol{y}}[k+1|k] = \frac{1}{2n}\sum_{i=1}^{2n}\hat{\boldsymbol{y}}^{(i)}[k+1|k]$$

$$\boldsymbol{P}_{\hat{y}}[k+1|k] = \frac{1}{2n}\sum_{i=1}^{2n}\left(\hat{\boldsymbol{y}}^{(i)}[k+1|k] - \hat{\boldsymbol{y}}[k+1|k]\right)\left(\hat{\boldsymbol{y}}^{(i)}[k+1|k] - \hat{\boldsymbol{y}}[k+1|k]\right)^T + \boldsymbol{N}[k+1]$$

$$\boldsymbol{P}_{\hat{x}\hat{y}}[k+1|k] = \frac{1}{2n}\sum_{i=1}^{2n}\left(\hat{\boldsymbol{x}}^{(i)}[k+1|k] - \hat{\boldsymbol{x}}[k+1|k]\right)\left(\hat{\boldsymbol{y}}^{(i)}[k+1|k] - \hat{\boldsymbol{y}}[k+1|k]\right)^T$$

Korrektur:

$$\boldsymbol{K}[k+1] = \boldsymbol{P}_{\hat{x}\hat{y}}[k+1|k]\left(\boldsymbol{P}_{\hat{y}}[k+1|k]\right)^{-1}$$
$$\hat{\boldsymbol{x}}[k+1|k+1] = \hat{\boldsymbol{x}}[k+1|k] + \boldsymbol{K}[k+1]\left(\boldsymbol{y}[k+1] - \hat{\boldsymbol{y}}[k+1|k]\right)$$
$$\boldsymbol{P}[k+1|k+1] = \boldsymbol{P}_{\hat{x}}[k+1|k] - \boldsymbol{K}[k+1]\boldsymbol{P}_{\hat{y}}[k+1|k]\boldsymbol{K}^T[k+1].$$

Im Vergleich zur EKF-Vorschrift in (5.51) lässt obige UKF-Vorschrift bereits erahnen, dass der Berechnungsaufwand in aller Regel signifikant größer ist. Demnach ist für die konkrete Applikation zu prüfen, ob der Mehraufwand des UKFs lohnenswert ist oder ob mit dem linearisierten Ansatz des EKFs bereits zielführende Ergebnisse erreicht werden können.

Ist hingegen eine analytische Berechnung der Jacobi-Matrizen für das EKF nicht möglich, z. B. im Fall einer Black-Box Modellierung, müssen diese durch finite Differenzen approximiert werden (siehe Kap. 4.1.2). Dies hat auch beim EKF zur Folge, dass die nichtlinearen Zustands- und Ausgangsfunktionen entsprechend häufig ausgewertet werden müssen. Dann erscheint eine direkte Verwendung des UKFs sinnvoller, da bei vergleichbarer Anzahl an Funktionsauswertungen eine bessere Approximation der nichtlinearen Systemcharakteristik resultiert.

5.3.1 Varianten der Unscented-Transformation

Bei der Herleitung des UKFs wurde die *einfache UT* zur Auswahl der Sigma-Punkte verwendet. Hierzu wurden $2n$ Sigma-Punkte erzeugt, wobei n die Anzahl der Zustände darstellt. Im Nachfolgenden werden zwei beliebte Varianten der UT vorgestellt, welche durch einfache Modifikation des Prädiktionsschritt in das UKF integriert werden können. Weitere UT-Varianten können der Literatur entnommen werden (siehe z. B. [JU04][Sch17]).

Verallgemeinerte UT
Die verallgemeinerte UT (*general unscented transformation*) verwendet $(2n+1)$ Sigma-Punkte, wobei der zusätzliche Sigma-Punkt gegenüber der einfachen UT der vorherige Mittelwert der

Schätzung ist. Zudem wird ein Skalierungsfaktor $\kappa \in \mathbb{R}$ eingeführt, um die Spreizung der Sigma-Punkte um den Mittelwert beeinflussen zu können. Die verallgemeinerte UT-Vorschrift lautet in der vereinfachten Nomenklatur des Beispiels (5.57):

$$\begin{aligned}
\boldsymbol{x}^{(0)} &= \overline{\boldsymbol{x}}\,, \\
\boldsymbol{x}^{(i)} &= \overline{\boldsymbol{x}} + \tilde{\boldsymbol{x}}^{(i)}, \quad i = 1, \ldots, 2n\,, \\
\tilde{\boldsymbol{x}}^{(i)} &= \left(\sqrt{(n+\kappa)\boldsymbol{P}_x}\right)_i^T, \quad i = 1, \ldots, n\,, \\
\tilde{\boldsymbol{x}}^{(i)} &= -\left(\sqrt{(n+\kappa)\boldsymbol{P}_x}\right)_{i-n}^T, \quad i = n+1, \ldots, 2n\,.
\end{aligned} \tag{5.75}$$

Für die Rekonstruktion von Mittelwert und Kovarianz werden die Gewichte

$$\begin{aligned}
w^{(0)} &= \frac{\kappa}{n+\kappa}\,, \\
w^{(i)} &= \frac{1}{2(n+\kappa)}, \quad i = 1, \ldots, 2n\,,
\end{aligned} \tag{5.76}$$

genutzt und es folgt

$$\begin{aligned}
\boldsymbol{y}^{(i)} &= \boldsymbol{f}(\boldsymbol{x}^{(i)}), \quad i = 0, \ldots, 2n\,, \\
\hat{\boldsymbol{y}} &= \sum_{i=0}^{2n} w^{(i)} \boldsymbol{y}^{(i)}\,, \\
\hat{\boldsymbol{P}}_y &= \sum_{i=0}^{2n} w^{(i)} \left(\boldsymbol{y}^{(i)} - \hat{\boldsymbol{y}}\right)\left(\boldsymbol{y}^{(i)} - \hat{\boldsymbol{y}}\right)^T.
\end{aligned} \tag{5.77}$$

Für $\kappa = 0$ resultiert die vorherige einfache UT. Demgegenüber kann gezeigt werden, dass im Fall von $\kappa \neq 0$ (unter der Voraussetzung $(n+\kappa) \neq 0$) die Approximationsgüte gegenüber der einfachen UT gesteigert werden kann [JU04][1].

Sphärische UT

Um den Berechnungsaufwand zur Laufzeit gering zu halten, ist es erstrebenswert die Anzahl der Sigma-Punkte zu reduzieren. Die sphärische UT (*spherical unscented transformation*) verwendet lediglich (n+2) Sigma-Punkte. Trotz dieser Reduktion ist die Approximationsgüte gegenüber der einfachen bzw. verallgemeinerten UT i. A. nur geringfügig schlechter. Die entsprechende Vorschrift lautet:

1. Wähle das Gewicht $w^{(0)} \in [0, 1)$.
2. Berechne die restlichen Gewichte zu

$$w^{(i)} = \frac{1 - w^{(0)}}{n+1}, \quad i = 1, \ldots, n+1. \tag{5.78}$$

[1] Beachte: Die gesteigerte Approximationsgüte des nichtlinearen Systems kann für Probleme, welche die Annahmen aus Definition 5.1 ideal erfüllen, vorteilhaft sein. Unter Praxisgesichtspunkten ist demgegenüber kritisch zu prüfen, ob anwendungsspezifische Abweichungen von den Annahmen in Definition 5.1 vorliegen und ob diese nicht derart ausschlaggebend sind, dass folglich der theoretische Zugewinn an Approximationsgüte in der Anwendung nicht genutzt werden kann.

3. Initialisiere die Hilfsvektoren

$$\begin{aligned} \boldsymbol{\sigma}_0^{(1)} &= 0\,, \\ \boldsymbol{\sigma}_1^{(1)} &= \frac{-1}{\sqrt{2w^{(1)}}}\,, \\ \boldsymbol{\sigma}_2^{(1)} &= \frac{1}{\sqrt{2w^{(1)}}}. \end{aligned} \tag{5.79}$$

4. Erweitere die Hilfsvektoren rekursiv für $j = 2, \ldots, n$

$$\boldsymbol{\sigma}_i^{(j)} = \begin{cases} \begin{bmatrix} \boldsymbol{\sigma}_0^{(j-1)} \\ 0 \end{bmatrix}, & i = 0\,, \\ \begin{bmatrix} \boldsymbol{\sigma}_i^{(j-1)} \\ \frac{-1}{\sqrt{j(j+1)w^{(i)}}} \end{bmatrix}, & i = 1, \ldots, j\,, \\ \begin{bmatrix} \mathbf{0}_{j-1} \\ \frac{j}{\sqrt{j(j+1)w^{(i)}}} \end{bmatrix}, & i = j+1\,, \end{cases} \tag{5.80}$$

mit $\mathbf{0}_j$ als Spaltenvektor, der j Nullen enthält. Nach Abschluss der Rekursion existieren $i = 0, \ldots, (n+1)$ Hilfsvektoren $\boldsymbol{\sigma}_i$ der Dimension n.

5. Berechne die Sigma-Punkte

$$\boldsymbol{x}^{(i)} = \overline{\boldsymbol{x}} + \sqrt{\boldsymbol{P}}\boldsymbol{\sigma}_i, \quad i = 0, \ldots, (n+1). \tag{5.81}$$

6. Propagiere die Sigma-Punkte durch das Modell und rekonstruiere den Mittelwert und die Kovarianz

$$\begin{aligned} \boldsymbol{y}^{(i)} &= \boldsymbol{f}(\boldsymbol{x}^{(i)}), \quad i = 0, \ldots, (n+1)\,, \\ \hat{\boldsymbol{y}} &= \sum_{i=0}^{n+1} w^{(i)} \boldsymbol{y}^{(i)}\,, \\ \hat{\boldsymbol{P}}_y &= \sum_{i=0}^{n+1} w^{(i)} \left(\boldsymbol{y}^{(i)} - \hat{\boldsymbol{y}}\right)\left(\boldsymbol{y}^{(i)} - \hat{\boldsymbol{y}}\right)^T. \end{aligned} \tag{5.82}$$

Welche UT-Variante für eine gegebene Anwendung am geeignetsten ist, kann im Vorhinein nicht hinreichend bewertet werden. Die zur Verfügung stehenden Rechenressourcen sowie Rahmenbedingungen der jeweiligen Anwendungen werden sicherlich Hinweise geben, ob eher die Approximationsgenauigkeit oder die Rechenlast im Fokus steht. Häufig, wie auch bei allen anderen Kalman-Filter-Ansätzen, kann diese Frage erst bei der konkreten Auslegung und experimentellen Validierung abschließend bewertet werden.

5.3.2 Vereinfachung für lineare Ausgangsfunktionen

Die vorherige Herleitung des UKFs geht von einer nichtlinearen System- und Ausgangsfunktion aus. In einigen Anwendungen, z. B. wenn die Messgrößen direkt (einen Teil) der Zustände

entsprechen, ist die Ausgangsfunktion allerdings linear. Die Systembeschreibung ist dann

$$\begin{aligned} \boldsymbol{x}[k+1] &= \boldsymbol{f}\left(\boldsymbol{x}[k], \boldsymbol{u}[k]\right) + \boldsymbol{m}[k], \quad \boldsymbol{x}[k=0] = \boldsymbol{x}_0\,, \\ \boldsymbol{y}[k+1] &= \boldsymbol{C}\boldsymbol{x}[k+1] + \boldsymbol{D}\boldsymbol{u}[k+1] + \boldsymbol{n}[k+1] \end{aligned} \tag{5.83}$$

mit den entsprechenden Definitionen und Annahmen gemäß Definition 5.1 bzw. (5.5). In diesem Fall müssen die Sigma-Punkte nur durch die nichtlineare Systemfunktion propagiert werden, während dann für die Berechnung des Mittelwerts und der Kovarianz der Ausgangsgrößen lineare Matrixalgebra verwendet werden kann. Durch Verknüpfung der Herleitungen aus Kap. 5.1 und Kap. 5.3 kann dann das UKF für lineare Ausgangsfunktionen analog erarbeitet werden. Es resultiert zu:

Unscented Transformation

$$\boldsymbol{x}^{(i)}[k|k] = \overline{\boldsymbol{x}}[k|k] + \tilde{\boldsymbol{x}}^{(i)}, \quad i = 1, \ldots, 2n$$

$$\tilde{\boldsymbol{x}}^{(i)}[k|k] = \left(\sqrt{n\boldsymbol{P}[k|k]}\right)_i^T, \quad i = 1, \ldots, n$$

$$\tilde{\boldsymbol{x}}^{(i)}[k|k] = -\left(\sqrt{n\boldsymbol{P}[k|k]}\right)_{i-n}^T, \quad i = n+1, \ldots, 2n$$

Prädiktion:

$$\hat{\boldsymbol{x}}^{(i)}[k+1|k] = \boldsymbol{f}\left(\boldsymbol{x}^{(i)}[k|k], \boldsymbol{u}[k]\right)$$

$$\hat{\boldsymbol{x}}[k+1|k] = \frac{1}{2n}\sum_{i=1}^{2n} \hat{\boldsymbol{x}}^{(i)}[k+1|k]$$

$$\boldsymbol{P}[k+1|k] = \frac{1}{2n}\sum_{i=1}^{2n} \left(\hat{\boldsymbol{x}}^{(i)}[k+1|k] - \hat{\boldsymbol{x}}[k+1|k]\right)\left(\hat{\boldsymbol{x}}^{(i)}[k+1|k] - \hat{\boldsymbol{x}}[k+1|k]\right)^T + \boldsymbol{M}[k+1]$$

Korrektur:

$$\boldsymbol{K}[k+1] = \boldsymbol{P}[k+1|k]\boldsymbol{C}^T[k+1]\left(\boldsymbol{C}[k+1]\boldsymbol{P}[k+1|k]\boldsymbol{C}^T[k+1] + \boldsymbol{N}[k+1]\right)^{-1}$$

$$\hat{\boldsymbol{x}}[k+1|k+1] = \hat{\boldsymbol{x}}[k+1|k] + \boldsymbol{K}[k+1]\left(\boldsymbol{y}[k+1] - \boldsymbol{C}[k+1]\hat{\boldsymbol{x}}[k+1|k] - \boldsymbol{D}[k+1]\boldsymbol{u}[k+1]\right)$$

$$\boldsymbol{P}[k+1|k+1] = \left(\boldsymbol{I} - \boldsymbol{K}[k+1]\boldsymbol{C}[k+1]\right)\boldsymbol{P}[k+1|k].$$

Es ist ersichtlich, dass der Berechnungsaufwand gegenüber dem regulären UKF maßgeblich reduziert werden kann. Analog kann auch eine UKF-Variante für eine lineare Systemgleichung mit nichtlinearer Ausgangsgleichung hergeleitet werden, auf dessen Betrachtung an dieser Stelle allerdings verzichtet wird.

6 Identifikation dynamischer Systeme

In diesem Kapitel werden die Grundlagen zur Identifikation dynamischer Systeme aufgezeigt. Hierbei wird der Fokus auf *zeitdiskrete Modelle* gelegt, da der überwiegende Anteil moderner, ingenieurtechnischer Systeme durch digitale Mess- und Regelplattformen überwacht werden, sodass die Datenbasis zur Systemidentifikation i. d. R. als abgetastete Signalreihen vorliegen. Ferner werden ausschließlich *parametrische Modelle* betrachtet. Für nicht-parametrische Modelle (z. B. Bode-Diagramm) sei auf die weitere Literatur verwiesen (z. B. [IM11][Len17]).

6.1 Methode der kleinsten Quadrate für zeitdiskrete Modelle im Zustandsraum

Um die Methode der kleinsten Quadrate, welche zuvor in Kap. 3.1 für statische Modelle diskutiert wurde, auch für dynamische Systeme einsetzen zu können, wird folgende Systemstruktur angenommen:

Definition 6.1: Vollst. messbares, zeitdiskretes LTI-System im Zustandsraum

Gegeben sei ein zeitdiskretes LTI-Modell im Zustandsraum bei dem alle Zustände unmittelbar messbar sind

$$\begin{aligned} \boldsymbol{x}[k+1] &= \boldsymbol{\Phi}\boldsymbol{x}[k] + \boldsymbol{H}\boldsymbol{u}[k]\,, \\ \boldsymbol{y}[k] &= \boldsymbol{x}[k] \end{aligned} \tag{6.1}$$

mit $\boldsymbol{\Phi} \in \mathbb{R}^{n\times n}$ als zeitdiskrete Transitionsmatrix und $\boldsymbol{H} \in \mathbb{R}^{n\times m}$ als zeitdiskrete Eingangsmatrix. Die entsprechenden Matrizen können durch exakte oder näherungsweise zeitliche Diskretisierung gemäß der Verfahren aus Kap. 2.2 gewonnen werden.

Zunächst werde lediglich ein Zustand bzw. Ausgang

$$\begin{aligned} x_i[k+1] &= \phi_{i1}x_1[k] + \phi_{i2}x_2[k] + \ldots + \phi_{in}x_n[k] + h_{i1}u_1[k] + h_{i2}u_2[k] + \ldots + h_{in}u_m[k] \\ &= \underbrace{\begin{bmatrix}\phi_{i1} & \phi_{i2} & \ldots & \phi_{in}\end{bmatrix}}_{\boldsymbol{\phi}_i^T}\boldsymbol{x}[k] + \underbrace{\begin{bmatrix}h_{i1} & h_{i2} & \ldots & h_{im}\end{bmatrix}}_{\boldsymbol{h}_i^T}\boldsymbol{u}[k] \end{aligned} \tag{6.2}$$

betrachtet. Hierbei beinhalten die Vektoren $\boldsymbol{\phi}_i^T$ und $\boldsymbol{h}_i^T$ die unbekannten Parameter der i-ten

Zeile von $\boldsymbol{\Phi}$ und $\boldsymbol{H}$. Umschreiben ergibt dann[1]:

$$\underbrace{x_i[k+1]}_{\psi_i[k]} = \underbrace{\begin{bmatrix}\boldsymbol{x}^T[k] & \boldsymbol{u}^T[k]\end{bmatrix}}_{\boldsymbol{\xi}_i^T[k]} \underbrace{\begin{bmatrix}\boldsymbol{\phi_i} \\ \boldsymbol{h}_i\end{bmatrix}}_{\boldsymbol{\theta}_i}. \tag{6.3}$$

Liegen Informationen zu $k = 1, \ldots, N$ Transitionsvorgängen vor, welche ein additives Messrauschen $\boldsymbol{e}_i[k]$ aufweisen, folgt unmittelbar

$$\underbrace{\begin{bmatrix} x_i[2] \\ \vdots \\ x_i[N+1] \end{bmatrix}}_{\boldsymbol{\psi}_i} = \underbrace{\begin{bmatrix} x_1[1] & x_2[1] & \ldots & x_n[1] & u_1[1] & u_2[1] & \ldots & u_m[1] \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_1[N] & x_2[N] & \ldots & x_n[N] & u_1[N] & u_2[N] & \ldots & u_m[N] \end{bmatrix}}_{\boldsymbol{\Xi}_i} \underbrace{\begin{bmatrix} \phi_{i1} \\ \vdots \\ \phi_{in} \\ h_{i1} \\ \vdots \\ h_{im} \end{bmatrix}}_{\boldsymbol{\theta}_i} + \underbrace{\begin{bmatrix} e_i[1] \\ \vdots \\ e_i[N] \end{bmatrix}}_{\boldsymbol{e}_i}$$

mit $\boldsymbol{\psi}_i$ als Messvektor, $\boldsymbol{e}_i$ als Residuenvektor und $\boldsymbol{\Xi}_i$ als Regressormatrix[2] bezüglich der i-ten Zeile von (6.1). Dies verlangt, dass der Zustandsvektor $\boldsymbol{x}$ insgesamt (N+1) mal gemessen werden muss. Gegenüber Kap. 3 wird zudem eine leicht geänderte Notation verwendet:

- N ist die Anzahl der Messpunkte,
- n ist die Anzahl der Zustände $\boldsymbol{x}$ und
- m ist die Anzahl der Eingangssignale $\boldsymbol{u}$.

Der Vergleich mit (3.7) zeigt, dass analog zum statischen Modell somit auch für das dynamische LTI System (6.1) ein lineares Gleichungssystem formuliert werden kann, anhand dessen der gesuchte Parametervektor $\boldsymbol{\theta}_i$ gefunden werden kann. Die Lösbarkeit dieses Gleichungssystems richtet sich ebenfalls nach Satz 3.1, wobei im folgenden davon ausgegangen wird, dass hinreichend viele Messpunkte vorliegen, um die Methode der kleinsten Quadrate anwenden zu können. Analog zum statischen Fall soll nun die Kostenfunktion

$$J(\boldsymbol{\theta}_i) = \sum_{k=1}^{N} (e_i[k])^2 = \sum_{k=1}^{N} (\psi_i[k] - \boldsymbol{\xi}_i^T[k]\boldsymbol{\theta}_i)^2 = (\boldsymbol{\psi}_i - \boldsymbol{\Xi}_i\boldsymbol{\theta}_i)^T(\boldsymbol{\psi}_i - \boldsymbol{\Xi}_i\boldsymbol{\theta}_i) \tag{6.4}$$

minimiert werden. Darauf aufbauend wird folgende Problemdefinition formuliert:

---

[1] In (6.1) wurde angenommen, dass für die Anwendung des LS-Verfahrens ein LTI-System mit $\boldsymbol{C} = \boldsymbol{I}$ und $\boldsymbol{D} = 0$ vorliegen muss. Dies ist eine sehr konservative bzw. vereinfachende Einschränkung, da mit Blick auf die Definition der Regressoren in (6.3) lediglich eine Linearität in den Parametern erforderlich ist. In der zeitdiskreten Notation (6.3) können daher sehr wohl nichtlineare Einflüsse hinsichtlich $\boldsymbol{x}[k]$ und $\boldsymbol{u}[k]$ auftreten, sofern ein linearer Zusammenhang mit den gesuchten Parametern verbleibt. Auch sei erwähnt, dass ein Teil der linearen, stochastischen Prozesse aus Tab. 2.1 dieser Form entspricht.

[2] Prinzipiell können zur Identifikation der $i = 1, \ldots, n$ Zeilen der Zustandsraummatrizen die selbe Datenbasis und somit die selbe Regressormatrix verwendet werden. In diesem Fall würde sich nur der Messvektor $\boldsymbol{\psi}_i$ für jede Zeile ändern. Um formal für jede i-te Zeile aber auch eine veränderliche Datenbasis zuzulassen, wird die Notation $\boldsymbol{\Xi}_i$ beibehalten.

Definition 6.2: LS-Problem für zeitdiskrete LTI-Systeme im Zustandsraum

Gegeben sei ein zeitdiskretes LTI-Modell im Zustandsraum entsprechend (6.1). *Für den i-ten Zustand werde eine Regressionsgleichung der Form*

$$\boldsymbol{e}_i = \boldsymbol{\psi}_i - \boldsymbol{\Xi}_i \boldsymbol{\theta}_i \tag{6.5}$$

durch $(N+1)$ Messungen aufgestellt. Hier ist $\boldsymbol{\theta}_i \in \mathbb{R}^{n+m}$ der Parametervektor

$$\boldsymbol{\theta}_i^T = \begin{bmatrix} \boldsymbol{\phi}_i^T & \boldsymbol{h}_i^T \end{bmatrix} \tag{6.6}$$

mit Informationen zur i-ten Zeile der zeitdiskreten Transitions- und Eingangsmatrix. Ferner ist $\boldsymbol{\psi}_i \in \mathbb{R}^N$ der Messvektor, $\boldsymbol{\Xi}_i \in \mathbb{R}^{N \times (n+m)}$ ist die Regressormatrix sowie $\boldsymbol{e}_i \in \mathbb{R}^N$ ist ein Residuenvektor. Es gelte $(m+n) < N$. Der Messdatenvektor $\boldsymbol{\psi}_i$ weise ein additives Messrauschen $\boldsymbol{\nu}_i = \boldsymbol{e}_i$ mit $\mathrm{E}(\boldsymbol{\nu}_i) = 0$ und $\mathrm{Cov}(\boldsymbol{\nu}_i) = \sigma_i^2$ auf, während $\boldsymbol{\Xi}_i$ exakt bekannt sei. Das Auffinden des Parametervektors $\boldsymbol{\theta}_i$ mittels Minimierung der quadratischen Kostenfunktion (6.4) *entsprechend*

$$\boldsymbol{\theta}_i^* = arg\ min\ J(\boldsymbol{\theta}_i) \tag{6.7}$$

wird als LS-Problem für lineare, zeitdiskrete dynamische Systeme bezeichnet.

Die (akademische) Definition des LS-Problems verlangt die exakte Kenntnis der Regressormatrix $\boldsymbol{\Xi}_i$ und erlaubt lediglich ein additives Rauschen auf dem Messvektor $\boldsymbol{\psi}_i$. Dies wirft jedoch zwangsläufig einen Widerspruch auf, da die Einträge im Messvektor als auch in der Regressormatrix von den selben Zuständen $\boldsymbol{x}$ (lediglich zu unterschiedlichen Abtastzeitpunkten) abhängen. Um den Annahmen obiger Problemdefinition gerecht zu werden bedeutet dies daher zwangsläufig, dass das Rauschen auf den Zuständen verschwinden muss. Auch die Eingangssignale $\boldsymbol{u}$ müssen gemäß obigen Annahmen exakt und rauschfrei bekannt sein.

Unter diesen Maßgaben kann die Lösung des LS-Problem analog zum statischen Fall hergeleitet werden[1]. Das Ergebnis lautet:

Satz 6.1: Lösung des LS-Problems für zeitdiskrete LTI-Systeme

Die Lösung des LS-Problems gemäß Definition 6.2 lautet

$$\boldsymbol{\theta}_i^* = \left(\boldsymbol{\Xi}_i^T \boldsymbol{\Xi}_i\right)^{-1} \boldsymbol{\Xi}_i^T \boldsymbol{\psi}_i, \tag{6.8}$$

sofern die Produktsummenmatrix $(\boldsymbol{\Xi}_i^T \boldsymbol{\Xi}_i)$ invertierbar

$$\det(\boldsymbol{\Xi}_i^T \boldsymbol{\Xi}_i) \neq 0 \tag{6.9}$$

und die Hesse-Matrix $\frac{\partial^2 J}{\partial \boldsymbol{\theta}_i^2}$ positiv definit ist, d. h. für Eigenwerte λ_j gilt

$$\lambda_j > 0 \quad j = [1, \ldots, m+n]\,. \tag{6.10}$$

Mit obiger Lösung kann jeweils eine Zeile aus (6.1) identifiziert werden, d. h. die Methodik

[1] Ebenso können die in Kap. 3.2 behandelten Varianten, bspw. RLS oder WLS, auf den dynamischen Fall übertragen werden.

muss n-fach wiederholt angewandt werden, um die zeitdiskrete Transitionsmatrix $\boldsymbol{\Phi}$ sowie die zeitdiskrete Eingangsmatrix $\boldsymbol{H}$ vollständig zu identifizieren.

Das Vorgehen und die Problematik des Rauscheinflusses soll anhand des einfachen SISO-Beispielsystems

$$\begin{aligned} x[k+1] &= ax[k] + b(u[k] + m_u[k])\,, \\ y[k] &= x[k] + n_y[k]. \end{aligned} \tag{6.11}$$

mit m_u und n_y als additivem Eingangs- und Messrauschen bei gegebenem Abtastintervall T_a verdeutlicht werden. Als Prädiktionsmodell wird die unverrauschte Systembeschreibung

$$\begin{aligned} x[k+1] &= ax[k] + bu[k]\,, \\ y[k] &= x[k] \end{aligned} \tag{6.12}$$

genutzt. Das zu identifizierte System entspricht einem zeitdiskreten P-T_1 Element

$$x[k+1] = \underbrace{\left(1 - \frac{T_a}{\tau}\right)}_{a} x[k] + \underbrace{V_s \frac{T_a}{\tau}}_{b} u[k] \tag{6.13}$$

mit der Verstärkung V_s und der Zeitkonstante τ. Werden die unbekannten Parameter des Zustandsraummodells a und b identifiziert, können V_s und τ unmittelbar berechnet werden, was ggf. einen tieferen, physikalischen Einblick in das zu identifizierende System liefert.

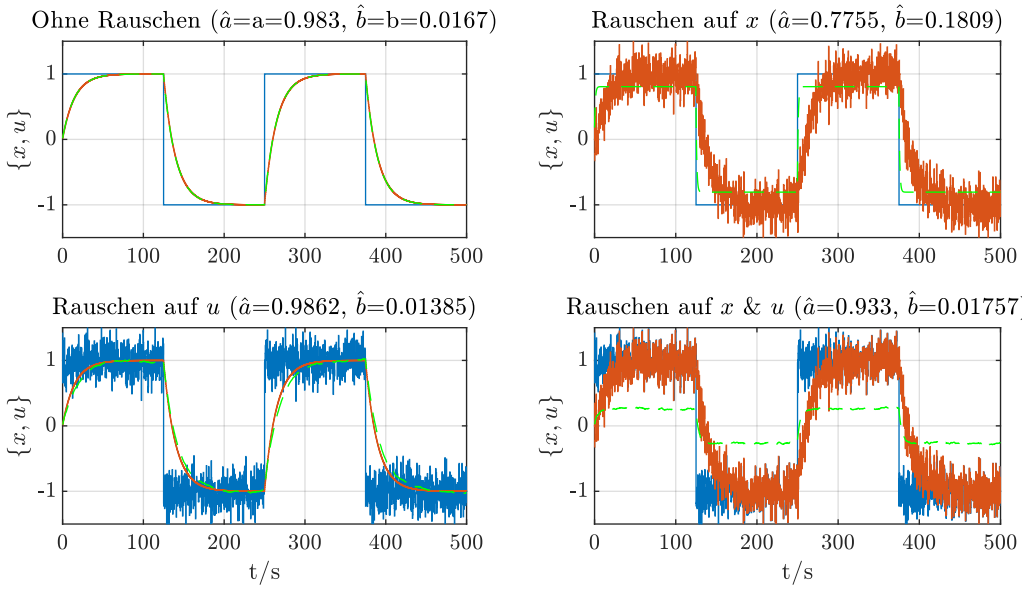


Abb. 6.1: Identifikationsergebnisse für unterschiedliche Rauscheinflüsse (rot = Zustandssignal, blau = Eingangssignal, grün-gestrichelt = Modellvalidierung, normalverteiltes Rauschen $\mathcal{N}(\mu = 0, \sigma^2 = 0,2)$

Für das Beispiel sind verschiedene Identifikationsszenarien in Abb. 6.1 dargestellt, welche sich hinsichtlich der Rauscheinflüsse bei der Erfassung des Zustands x und des Eingangs u unter-

scheiden. Als Modellvalidierung nach erfolgter Identifikation ist ebenfalls die Schätzung

$$\hat{x}[k+1] = \hat{a}\hat{x}[k] + \hat{b}u[k] \tag{6.14}$$

dargestellt. Während sich das System im rauschfreien Fall perfekt identifizieren lässt, resultieren durch Eingangs- und Messrauschen signifikante Unterschiede. Tendenziell ist festzustellen, dass im Beispiel das Zustandsrauschen einen signifikanteren, negativen Einfluss ausübt. Ferner ist eine Verkopplung zu beobachten, d. h. selbst wenn nur eine der beiden Größen verrauscht aufgenommen werden kann, wird der jeweils andere Systemparameter ebenfalls nur fehlerhaft identifiziert. Auch ohne die Residuen zwischen Messung und Modell explizit abzubilden, wird in Abb. 6.1 bereits klar, dass durch das Rauschen systematische Modellierungsfehler resultieren.

6.1.1 Total Least Squares

Abschließend wurde der TLS-Ansatz aus Kap. 3.2.4 auf das vorherige Beispiel (6.13) angewandt. Da das Beispiel dimensionslos ist und die Zustands- und Eingangsgröße in der selben Größenordnung sind, wird auf eine Datenvorverarbeitung verzichtet. Die Daten- und Modellgrundlage ist somit unverändert und die entsprechenden Resultate sind in Abb. 6.2 zusammengefasst. Das Ergebnis ist hierbei äußerst interessant: Im rauschfreien Szenario ist nach wie vor eine ideale Identifikation möglich ist. Auch bei den Szenarien mit einem singulären Rauschen auf $\boldsymbol{x}$ oder $\boldsymbol{u}$ sind die Ergebnisse deutlich besser als mit dem OLS-Verfahren. Allerdings wird ein vollständig unbrauchbares, instabiles System identifiziert, sobald sowohl auf $\boldsymbol{x}$ als auch $\boldsymbol{u}$ Rauschen einwirkt. Zusammenfassend ist somit festzustellen, dass mit dem TLS-Verfahren zwar grundsätzlich ein alternatives Identifikationswerkzeug (sowohl für statische Systeme als auch zeitdisrekte LTI-Systeme) zur Verfügung steht, dessen Nutzen aber stets für das konkrete Problem kritisch zu bewerten ist.

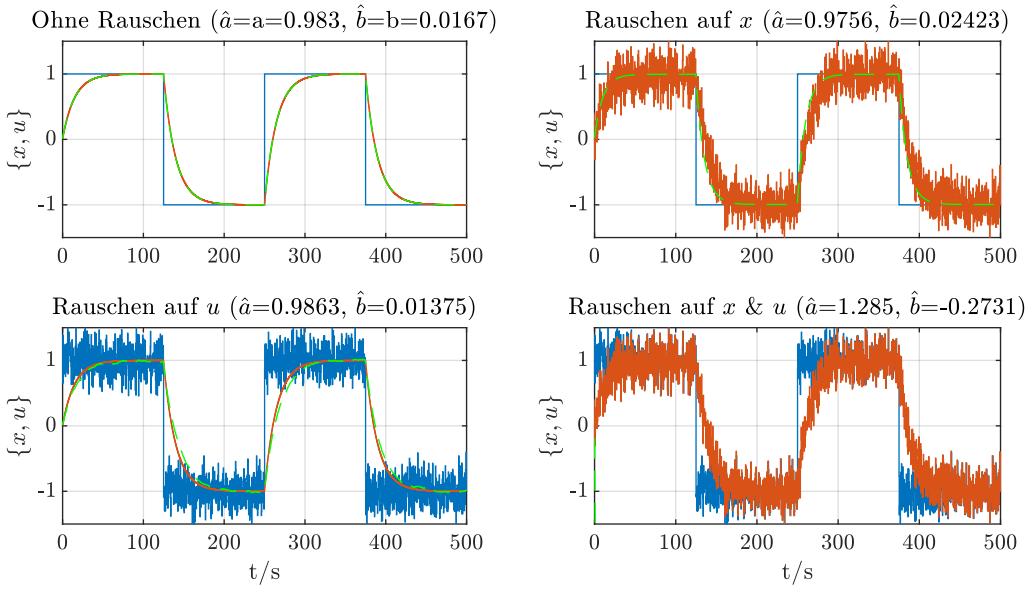


Abb. 6.2: Identifikationsergebnisse für unterschiedliche Rauscheinflüsse (rot = Zustandssignal, blau = Eingangssignal, grün-gestrichelt = Modellvalidierung, normalverteiltes Rauschen $\mathcal{N}(\mu = 0, \sigma^2 = 0,2)$

6.2 Maximum-Likelihood-Schätzung

Die Maximum-Likelihood-Methode[1], kurz ML-Methode, ist ein Schätzverfahren, welches die Parameter derart auswählt, damit die Wahrscheinlichkeit, die beobachteten Daten mit dem zugrundeliegende Modell erklären zu können, maximiert wird.

6.2.1 Grundlagen

Sei

$$p_{\boldsymbol{\theta}}(\boldsymbol{\psi}[k]) \tag{6.15}$$

die bedingte Wahrscheinlichkeit der Messung $\boldsymbol{\psi}[k]$ für einen gegebenen, deterministischen Parametervektor

$$\boldsymbol{\theta} = \begin{bmatrix} \theta_1 & \theta_2 & \cdots & \theta_m \end{bmatrix}^T \tag{6.16}$$

bezüglich eines zu parametrierenden Modells

$$\begin{aligned} \hat{\boldsymbol{x}}[k+1] &= \boldsymbol{f}(\hat{\boldsymbol{x}}[k], \boldsymbol{u}[k], \boldsymbol{\theta}) \,, \\ \hat{\boldsymbol{y}}[k] &= \boldsymbol{g}(\hat{\boldsymbol{x}}[k], \boldsymbol{u}[k], \boldsymbol{\theta}) \,. \end{aligned} \tag{6.17}$$

Obiges Modell kann eine beliebige, nichtlineare und zeitdiskrete Struktur aufweisen. Im Zuge des ML-Ansatzes werden die Parameter $\hat{\boldsymbol{\theta}}$ derart geschätzt, dass die Wahrscheinlichkeit die p-Messgrößen

$$\boldsymbol{\psi}[k] = \begin{bmatrix} \psi_1[k] & \psi_2[k] & \cdots & \psi_p[k] \end{bmatrix}^T . \tag{6.18}$$

zum Zeitpunkt $k = 1, \ldots, N$ mit der resultierenden Datenbasis

$$\boldsymbol{\Psi} = \begin{bmatrix} \boldsymbol{\psi}^T[1] & \boldsymbol{\psi}^T[2] & \cdots & \boldsymbol{\psi}^T[N] \end{bmatrix}^T . \tag{6.19}$$

zu erhalten, maximiert wird, also

$$\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}} p_{\boldsymbol{\theta}}(\boldsymbol{\Psi}). \tag{6.20}$$

Zur Herleitung einer Kostenfunktion, um obige Optimierungsaufgabe lösen zu können, werden folgende Annahmen getätigt:

Definition 6.3: Annahmen für die ML-Schätzung

Es wird angenommen, dass die Messungen $\boldsymbol{\psi}[k]$ unabhängig und identisch verteilte Zufallsvariablen[2] darstellen, d. h.,

- *die Messungen $\boldsymbol{\psi}[k]$ und $\boldsymbol{\psi}[k+1]$ sind zeitlich unkorreliert und*
- *die Zufallsverteilung bleibt konstant ($p_{\boldsymbol{\theta}}(\boldsymbol{\psi}[k]) = p_{\boldsymbol{\theta}}(\boldsymbol{\psi}[j]), \quad \forall\{j,k\} = 1, \ldots, N$).*

[1] Aus dem Englischen: Methode der größten Wahrscheinlichkeit bzw. Plausibilität

[2] Als Abkürzung finden sich in der Literatur unter anderem iid oder i.i.d. als Abkürzung des englischen Ausdrucks *independent identically distributed*.

Wie bereits in (2.40) ausgeführt, ergibt sich dann für die gemeinsame Wahrscheinlichkeit der Datenbasis:

Satz 6.2: Wahrscheinlichkeit der Datenbasis

Gegeben seien $k = 1, \ldots, N$ Messungen $\boldsymbol{\psi}[k]$, welche durch unabhängige und identisch verteilte Zufallsvariablen abgebildet werden können und die Wahrscheinlichkeitsdichteverteilung $p_{\boldsymbol{\theta}}(\boldsymbol{\psi}[k])$ aufweisen. Dann folgt für die Wahrscheinlichkeit aller Messungen (Datenbasis):

$$p_{\boldsymbol{\theta}}(\boldsymbol{\Psi}) = p_{\boldsymbol{\theta}}(\boldsymbol{\psi}[1]) p_{\boldsymbol{\theta}}(\boldsymbol{\psi}[2]) \cdots p_{\boldsymbol{\theta}}(\boldsymbol{\psi}[N]) = \prod_{k=1}^{N} p_{\boldsymbol{\theta}}(\boldsymbol{\psi}[k]). \tag{6.21}$$

Bis hierhin wurden keine Aussagen über die Art und die Eigenschaften der Zufallsverteilung getätigt. Um für den ML-Ansatz allerdings eine konkrete Kostenfunktion bzw. Berechnungsvorschrift ableiten zu können, muss die Zufallsverteilung bekannt sein. Statt die Zufallsverteilung der Messung zu beschreiben, werden die Residuen zwischen Modell und Messung betrachtet, was die nachfolgenden Berechnungen vereinfacht. Hierzu werden folgende Annahmen getroffen[1]:

Definition 6.4: Weitere Annahmen für die ML-Schätzung

Die Residuen

$$\boldsymbol{e}[k] = \boldsymbol{\psi}[k] - \hat{\boldsymbol{y}}[k] = \boldsymbol{\psi}[k] - \boldsymbol{g}(\boldsymbol{x}[k], \boldsymbol{u}[k], \boldsymbol{\theta}) \tag{6.22}$$

zwischen Modell und Messung seien unabhängig und identisch verteilte Zufallsvariablen. Diese folgen der Normalverteilung, seien mittelwerfrei und weisen eine konstante Kovarianzmatrix auf:

$$\mathrm{E}(\boldsymbol{e}[k]) = \boldsymbol{0}, \quad \mathrm{Cov}(\boldsymbol{e}[k], \boldsymbol{e}[k]) = \boldsymbol{R}. \tag{6.23}$$

Hieraus folgt unmittelbar, dass die Messung

$$\boldsymbol{\psi}[k] = \boldsymbol{g}(\boldsymbol{x}[k], \boldsymbol{u}[k], \boldsymbol{\theta}) + \boldsymbol{e}[k] \tag{6.24}$$

ebenfalls normalverteilt ist, da $\boldsymbol{g}$ ein deterministisches Modell ist, welches lediglich den Mittelwert der Verteilung beeinflusst. Folglich ergibt sich die Wahrscheinlichkeitsdichtefunktion von $\boldsymbol{\psi}[k]$ als mehrdimensionale Normalverteilung:

$$\begin{aligned} p_{\boldsymbol{\theta}}(\boldsymbol{\psi}[k]) &= \frac{1}{\sqrt{(2\pi)^p \det(\boldsymbol{R})}} e^{\left(-\frac{1}{2}(\boldsymbol{\psi}[k]-\boldsymbol{\mu}_\psi)^T \boldsymbol{R}^{-1}(\boldsymbol{\psi}[k]-\boldsymbol{\mu}_\psi)\right)} \\ &= \frac{1}{\sqrt{(2\pi)^p \det(\boldsymbol{R})}} e^{\left(-\frac{1}{2}(\boldsymbol{\psi}[k]-\boldsymbol{g}[k])^T \boldsymbol{R}^{-1}(\boldsymbol{\psi}[k]-\boldsymbol{g}[k])\right)} = p_{\boldsymbol{\theta}}(\boldsymbol{e}[k]). \end{aligned} \tag{6.25}$$

Auf Basis der vorherigen Annahmen und Definitionen kann dann die Likelihood-Funktion gebildet werden:

[1] Eine alternative Betrachtungsweise ist, dass die gemachten Annahmen als Idealwunsch an die Residuen nach erfolgter Identifikation interpretiert werden können.

Definition 6.5: Likelihood-Funktion

Unter den Annahmen aus Satz 6.2 sowie Definition 6.3 ist die Likelihood-Funktion definiert als

$$\begin{aligned}\mathbb{L}(\boldsymbol{\Psi},\boldsymbol{\theta}) = p_{\boldsymbol{\theta}}(\boldsymbol{\Psi}) &= \prod_{k=1}^{N} p_{\boldsymbol{\theta}}(\boldsymbol{\psi}[k]) \\ &= \left((2\pi)^p \det(\boldsymbol{R})\right)^{-\frac{N}{2}} e^{\left(-\frac{1}{2}\sum_{k=1}^{N}(\boldsymbol{\psi}[k]-\boldsymbol{g}[k])^T \boldsymbol{R}^{-1}(\boldsymbol{\psi}[k]-\boldsymbol{g}[k])\right)}.\end{aligned} \tag{6.26}$$

Die Likelihood-Funktion dient als Kostenfunktion zum Auffinden eines optimalen Parametersatzes in der nachfolgenden Optimierung aus Kap. 6.2.2. Hierzu wird die Ableitung der Kostenfunktion benötigt, welche sich zu

$$\begin{aligned}\frac{\partial \mathbb{L}(\boldsymbol{\Psi},\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} &= \frac{\partial p_{\boldsymbol{\theta}}(\boldsymbol{\psi}[1])}{\partial \boldsymbol{\theta}} \prod_{k=2}^{N} p_{\boldsymbol{\theta}}(\boldsymbol{\psi}[k]) + \cdots + \frac{\partial p_{\boldsymbol{\theta}}(\boldsymbol{\psi}[i])}{\partial \boldsymbol{\theta}} \prod_{k=1,k\neq i}^{N} p_{\boldsymbol{\theta}}(\boldsymbol{\psi}[k]) + \cdots \\ &= \sum_{i=1}^{m} \frac{\partial p_{\boldsymbol{\theta}}(\boldsymbol{\psi}[i])}{\partial \boldsymbol{\theta}} \prod_{k=1,k\neq i}^{N} p_{\boldsymbol{\theta}}(\boldsymbol{\psi}[k])\end{aligned} \tag{6.27}$$

ergibt. Diese Berechnung ist sehr aufwendig, weshalb i. d. R. der negative Logarithmus der Likelihood-Funktion verwendet, um die Berechnungsaufwand zu reduzieren. Die Optimierungsaufgabe kann dann umgeformt werden zu

$$\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}} \mathbb{L}(\boldsymbol{\Psi},\boldsymbol{\theta}) \quad \Leftrightarrow \quad \boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} -\ln\left(\mathbb{L}(\boldsymbol{\Psi},\boldsymbol{\theta})\right). \tag{6.28}$$

Anwenden des negativen Logarithmus auf (6.26) führt dann zu

$$\begin{aligned}-\ln\left(\mathbb{L}(\boldsymbol{\Psi},\boldsymbol{\theta})\right) &= -\ln\left(\prod_{k=1}^{N} p_{\boldsymbol{\theta}}(\boldsymbol{\psi}[k])\right) \\ &= -\ln\left(\left((2\pi)^p \det(\boldsymbol{R})\right)^{-\frac{N}{2}}\right) \\ &+ \frac{1}{2}\sum_{k=1}^{N} \left(\boldsymbol{\psi}[k] - \boldsymbol{g}(\boldsymbol{x}[k],\boldsymbol{u}[k],\boldsymbol{\theta})\right)^T \boldsymbol{R}^{-1} \left(\boldsymbol{\psi}[k] - \boldsymbol{g}(\boldsymbol{x}[k],\boldsymbol{u}[k],\boldsymbol{\theta})\right)\end{aligned} \tag{6.29}$$

bzw. durch weiteres Ausmultiplizieren ergibt sich dann:

Definition 6.6: Negative Log-Likelihood-Funktion

Unter den Annahmen aus Satz 6.2 sowie Definition 6.3 ist die negative Log-Likelihood-Funktion definiert als

$$\begin{aligned}-\ln\left(\mathbb{L}(\boldsymbol{\Psi},\boldsymbol{\theta})\right) &= \frac{1}{2}\sum_{k=1}^{N} \left(\boldsymbol{\psi}[k] - \boldsymbol{g}(\boldsymbol{x}[k],\boldsymbol{u}[k],\boldsymbol{\theta})\right)^T \boldsymbol{R}^{-1} \left(\boldsymbol{\psi}[k] - \boldsymbol{g}(\boldsymbol{x}[k],\boldsymbol{u}[k],\boldsymbol{\theta})\right) \\ &+ \frac{N}{2}\ln(\det(\boldsymbol{R})) + \frac{pN}{2}\ln(2\pi).\end{aligned} \tag{6.30}$$

Es sei darauf hingewiesen, dass der letzte Term

$$\frac{pN}{2}\ln(2\pi)$$

der negativen Log-Likelihood-Funktion (6.30) für eine gegebene Modellstruktur (Anzahl Messgrößen p gegeben) sowie ein gegebenes Datenset (Anzahl Abtastungen N gegeben) eine Konstante ist. In diesem Fall kann der Term im Sinne eine Maximierungsaufgabe vernachlässigt werden. Auch vereinfacht sich die Ableitung nach $\boldsymbol{\theta}$ zu

$$\begin{aligned}\frac{\partial - \ln\left(\mathbb{L}(\boldsymbol{\Psi},\boldsymbol{\theta})\right)}{\partial\boldsymbol{\theta}} &= -\frac{\partial}{\partial\boldsymbol{\theta}}\left(\sum_{k=1}^{N}\ln\left(p_{\boldsymbol{\theta}}(\boldsymbol{\psi}[k])\right)\right)\\ &= -\sum_{k=1}^{N}\frac{\partial\ln\left(p_{\boldsymbol{\theta}}(\boldsymbol{\psi}[k])\right)}{\partial\boldsymbol{\theta}}\end{aligned} \tag{6.31}$$

mit

$$\ln\left(p_{\boldsymbol{\theta}}(\boldsymbol{\psi}[k])\right) = -\frac{1}{2}\ln\left(\sqrt{(2\pi)^p\det(\boldsymbol{R})}\right) - \frac{1}{2}\left(\boldsymbol{\psi}[k]-\boldsymbol{g}[k]\right)^T\boldsymbol{R}^{-1}\left(\boldsymbol{\psi}[k]-\boldsymbol{g}[k]\right).$$

6.2.2 Grundsätzliche Lösung des ML-Problems

Unter der Annahme einer gegebenen Modellstruktur sowie eines gegebenen Datensatzes ergibt sich die Kostenfunktion zu:

$$J(\boldsymbol{\theta}) = \frac{1}{2}\sum_{k=1}^{N}\left(\boldsymbol{\psi}[k]-\boldsymbol{g}(\boldsymbol{x}[k],\boldsymbol{u}[k],\boldsymbol{\theta})\right)^T\boldsymbol{R}^{-1}\left(\boldsymbol{\psi}[k]-\boldsymbol{g}(\boldsymbol{x}[k],\boldsymbol{u}[k],\boldsymbol{\theta})\right) + \frac{N}{2}\ln(\det(\boldsymbol{R})). \tag{6.32}$$

Unter Verwendung der Residuen

$$\boldsymbol{e}[k] = \boldsymbol{\psi}[k] - \hat{\boldsymbol{y}}[k] = \boldsymbol{\psi}[k] - \boldsymbol{g}(\boldsymbol{x}[k],\boldsymbol{u}[k],\boldsymbol{\theta})$$

kann (6.32) zusammengefasst werden als:

$$J(\boldsymbol{\theta}) = \frac{1}{2}\sum_{k=1}^{N}\left(\boldsymbol{e}^T[k]\boldsymbol{R}^{-1}\boldsymbol{e}[k]\right) + \frac{N}{2}\ln(\det(\boldsymbol{R})). \tag{6.33}$$

Im Idealfall repräsentiert die Kovarianzmatrix $\boldsymbol{R}$ das Messrauschen, welches nach erfolgter Identifikation als Residuen zwischen Modell und Messung verbleibt. Für eine optimale Schätzung müsste $\boldsymbol{R}$ daher a-priori angegeben werden, was in vielen Applikationen aufgrund von unzureichenden Informationen i. d. R. nicht möglich sein wird. Während des (iterativen) Identifikationsprozesses, spiegelt $\boldsymbol{R}$ darüber hinaus noch folgende Effekte wieder:

- Systematische Modellierungsfehler (Modellstrukturabweichungen, nicht-ideale Messungen etc.) sowie
- Abweichungen im iterativen Optimierungsprozess, welche durch $\boldsymbol{\theta}[i]$ im i-ten Iterationsschritt[1] (noch nicht) abgebildet werden.

[1] Im Folgenden wird $x[i]$ zur Anzeige des i-ten Iterationsschrittes in einem Optimierungsproblem genutzt, während $x[k]$ den k-ten Abtastschritt in einem Datenset darstellt.

Daher ist es angezeigt im Lösungsprozess nicht nur die eigentlichen Parameter $\boldsymbol{\theta}$, sondern auch die Kovarianzmatrix $\boldsymbol{R}$ zu identifizieren. Formal würde dies bedeuten, dass dann das erweiterte Parameterset

$$\tilde{\boldsymbol{\theta}} = \begin{bmatrix} \boldsymbol{\theta}^T & \boldsymbol{\theta}_{\boldsymbol{R}}^T \end{bmatrix}^T \tag{6.34}$$

bestimmt werden muss, was folglich die Anzahl der zu identifizierenden Parameter signifikant erhöhen würde. Auch die 1. Ableitung

$$\nabla J(\tilde{\boldsymbol{\theta}}) = \begin{bmatrix} \frac{\partial J(\tilde{\boldsymbol{\theta}})}{\partial \boldsymbol{\theta}} & \frac{\partial J(\tilde{\boldsymbol{\theta}})}{\partial \boldsymbol{\theta}_{\boldsymbol{R}}} \end{bmatrix} \tag{6.35}$$

steigt somit in ihrer Komplexität an. Zur Reduktion des damit einhergehenden Berechnungsaufwands wird statt einer simultanen Lösung zum Auffinden von (6.34) ein relaxierter, zweistufiger Ansatz entsprechend Abb. 6.3 verfolgt:

1. Berechne $\boldsymbol{R}[i]$ für ein konstantes $\boldsymbol{\theta}$.
2. Berechne $\boldsymbol{\theta}[i]$ für ein $\boldsymbol{R}$.

Ausgehend von einer allgemeinen, nichtlinearen Modellstruktur, muss für $\boldsymbol{\theta}[i]$ ein geeigneter Optimierungsalgorithmus aus Kap. 4 herangezogen werden. Für $\boldsymbol{R}[i]$ existiert hingegen eine analytisch geschlossene Lösung, welche nachfolgend hergeleitet wird.

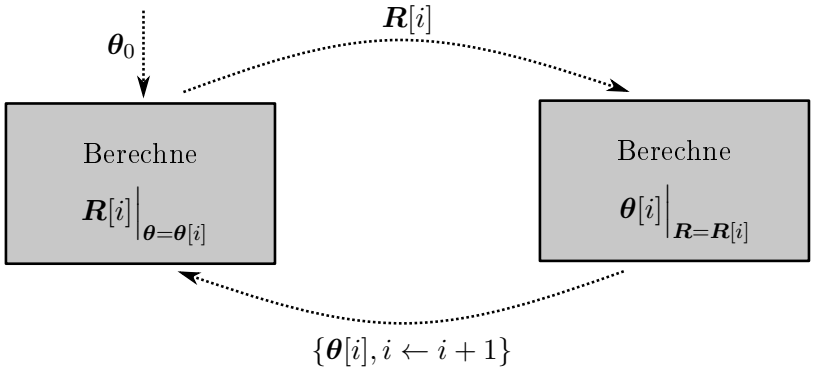


Abb. 6.3: Zweistufiger Optimierungsprozess für die ML-Parameteridentifkation

Berechnung der Kovarianzmatrix
Zum Auffinden eines optimalen $\boldsymbol{R}$ muss

$$\frac{\partial J}{\partial \boldsymbol{R}} = \frac{\partial}{\partial \boldsymbol{R}} \left(\frac{1}{2} \sum_{k=1}^{N} \left(\boldsymbol{e}^T[k] \boldsymbol{R}^{-1} \boldsymbol{e}[k] \right) \right) + \frac{\partial}{\partial \boldsymbol{R}} \left(\frac{N}{2} \ln(\det(\boldsymbol{R})) \right) = 0 \tag{6.36}$$

gelten. Der erste Term kann unter Verwendung der Matrix-Spur umgeschrieben werden zu

$$\frac{\partial}{\partial \boldsymbol{R}} \left(\frac{1}{2} \sum_{k=1}^{N} \left(\boldsymbol{e}^T[k] \boldsymbol{R}^{-1} \boldsymbol{e}[k] \right) \right) = \frac{\partial}{\partial \boldsymbol{R}} \text{Spur} \left(\frac{1}{2} \sum_{k=1}^{N} \left(\boldsymbol{e}^T[k] \boldsymbol{R}^{-1} \boldsymbol{e}[k] \right) \right), \tag{6.37}$$

da die Teilkostenfunktion ein Skalar ist. Da die Matrix-Spur zudem eine lineare Operation ist, gilt:

$$\frac{\partial}{\partial \boldsymbol{R}} \text{Spur} \left(\frac{1}{2} \sum_{k=1}^{N} \left(\boldsymbol{e}^T[k] \boldsymbol{R}^{-1} \boldsymbol{e}[k] \right) \right) = \frac{1}{2} \frac{\partial}{\partial \boldsymbol{R}} \sum_{k=1}^{N} \text{Spur} \left(\boldsymbol{e}^T[k] \boldsymbol{R}^{-1} \boldsymbol{e}[k] \right). \tag{6.38}$$

Unter Verwendung der Rechenregel

$$\text{Spur}(\boldsymbol{ABC}) = \text{Spur}(\boldsymbol{CAB})$$

folgt

$$\frac{1}{2}\frac{\partial}{\partial \boldsymbol{R}}\sum_{k=1}^{N}\text{Spur}\left(\boldsymbol{e}^T[k]\boldsymbol{R}^{-1}\boldsymbol{e}[k]\right) = \frac{1}{2}\frac{\partial}{\partial \boldsymbol{R}}\sum_{k=1}^{N}\text{Spur}\left(\boldsymbol{R}^{-1}\boldsymbol{e}[k]\boldsymbol{e}^T[k]\right) \tag{6.39}$$

welches wiederum unter Anwendung der Rechenregeln aus Anhang A.3 zu

$$\frac{1}{2}\frac{\partial}{\partial \boldsymbol{R}}\sum_{k=1}^{N}\text{Spur}\left(\boldsymbol{R}^{-1}\boldsymbol{e}[k]\boldsymbol{e}^T[k]\right) = -\frac{1}{2}\sum_{k=1}^{N}\left(\boldsymbol{R}^{-1}\boldsymbol{e}[k]\boldsymbol{e}^T[k]\boldsymbol{R}^{-1}\right) \tag{6.40}$$

umgeschrieben werden kann. Für den zweiten Term in (6.36) ergibt sich unter Verwendung von

$$\frac{\partial}{\partial \boldsymbol{X}}\ln(\det(\boldsymbol{X})) = \boldsymbol{X}^{-1}$$

der Ausdruck

$$\frac{\partial}{\partial \boldsymbol{R}}\left(\frac{N}{2}\ln(\det(\boldsymbol{R}))\right) = \frac{N}{2}\boldsymbol{R}^{-1}. \tag{6.41}$$

Einsetzen der vorherigen Berechnungen in (6.36) ergibt dann

$$-\frac{1}{2}\sum_{k=1}^{N}\left(\boldsymbol{R}^{-1}\boldsymbol{e}[k]\boldsymbol{e}^T[k]\boldsymbol{R}^{-1}\right) + \frac{N}{2}\boldsymbol{R}^{-1} = 0. \tag{6.42}$$

Auflösen nach $\boldsymbol{R}$ ergibt dann die geschlossene Lösung

$$\boldsymbol{R} = \frac{1}{N}\sum_{k=1}^{N}\left(\boldsymbol{e}[k]\boldsymbol{e}^T[k]\right). \tag{6.43}$$

Der gefundene Ausdruck entspricht somit dem ML-Schätzer der Kovarianzmatrix $\boldsymbol{R}$, welcher vergleichsweise einfach aus den Residuen zwischen Modell und Messung gewonnen werden kann.

Berechnung des Parametervektors
Für die Berechnung von $\boldsymbol{\theta}[i]$ gemäß Abb. 6.3 ist $\boldsymbol{R}$ eine Konstante. Somit ist der Modellausgang $\boldsymbol{g}(\boldsymbol{x}[k], \boldsymbol{u}[k], \boldsymbol{\theta})$ der einzige Teilterm der Kostenfunktion, welcher von $\boldsymbol{\theta}$ abhängt. Dies reduziert den Berechnungsaufwand hinsichtlich des Gradienten bzw. der Hesse-Matrix, sofern ein ableitungsbehaftetes Optimierungsverfahren angewandt wird. Alternativ können ableitungsfreie Verfahren, wie beispielsweise die Partikelschwarmoptimierung, herangezogen werden. Sofern der Gradient benötigt wird, ergibt sich diese unter Anwendung der Produktregel zu:

$$\begin{aligned}\frac{\partial J}{\partial \boldsymbol{\theta}} &= -\frac{1}{2}\sum_{k=1}^{N}\left(\frac{\partial \boldsymbol{g}(\boldsymbol{x}[k], \boldsymbol{u}[k], \boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\right)^T \boldsymbol{R}^{-1}\boldsymbol{e}[k] - \frac{1}{2}\sum_{k=1}^{N}\left(\frac{\partial \boldsymbol{g}(\boldsymbol{x}[k], \boldsymbol{u}[k], \boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\right)^T \boldsymbol{R}^{-1}\boldsymbol{e}[k] \\ &= -\sum_{k=1}^{N}\left(\frac{\partial \boldsymbol{g}(\boldsymbol{x}[k], \boldsymbol{u}[k], \boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\right)^T \boldsymbol{R}^{-1}\boldsymbol{e}[k].\end{aligned} \tag{6.44}$$

Hierbei entspricht

$$\frac{\partial \boldsymbol{g}(\boldsymbol{x}[k], \boldsymbol{u}[k], \boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = \begin{bmatrix} \frac{\partial g_1[k]}{\partial \theta_1} & \cdots & \frac{\partial g_1[k]}{\partial \theta_m} \\ \vdots & \ddots & \vdots \\ \frac{\partial g_p[k]}{\partial \theta_1} & \cdots & \frac{\partial g_p[k]}{\partial \theta_m} \end{bmatrix} \tag{6.45}$$

der sogenannten *Ausgangssensitivität*. Diese zeigt an, wie stark die i-te Ausgangsgröße $\hat{y}_i$ auf eine Veränderung des j-ten Parameters θ_j reagiert. Hierbei gilt es zu beachten, dass die Zustände $\boldsymbol{x} = \boldsymbol{x}(\boldsymbol{\theta})$ über die Zustandsfunktion $\boldsymbol{f}$ ebenfalls von den Parametern abhängen, sodass mittels verallgemeinerter Kettenregel folgt:

$$\begin{aligned} \frac{\partial \boldsymbol{g}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}} &= \frac{\partial \boldsymbol{g}}{\partial \boldsymbol{x}} \frac{\partial \boldsymbol{x}}{\partial \boldsymbol{\theta}} + \frac{\partial \boldsymbol{g}}{\partial \boldsymbol{u}} \underbrace{\frac{\partial \boldsymbol{u}}{\partial \boldsymbol{\theta}}}_{=0} + \frac{\partial \boldsymbol{g}}{\partial \boldsymbol{\theta}} \\ &= \frac{\partial \boldsymbol{g}}{\partial \boldsymbol{x}} \frac{\partial \boldsymbol{x}}{\partial \boldsymbol{\theta}} + \frac{\partial \boldsymbol{g}}{\partial \boldsymbol{\theta}} \end{aligned} \tag{6.46}$$

Der zweite Term in obiger Gleichung kann ggf. direkt analytisch ausgerechnet werden, sofern ein expliziter Zusammenhang zwischen der Ausgangsgleichung und den Parametern besteht. Für den ersten Term kann unter Einbeziehung der Systemdynamik aus (6.17) folgende Beschreibung gefunden werden:

$$\begin{aligned} \frac{\partial \boldsymbol{x}[k+1]}{\partial \boldsymbol{\theta}} &= \frac{\partial}{\partial \boldsymbol{\theta}} \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{\theta}) \\ &= \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}} \frac{\partial \boldsymbol{x}}{\partial \boldsymbol{\theta}} + \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{u}} \underbrace{\frac{\partial \boldsymbol{u}}{\partial \boldsymbol{\theta}}}_{=0} + \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{\theta}}. \end{aligned} \tag{6.47}$$

Unter Verwendung der Definition der Zustandssensitvität

$$\boldsymbol{S} = \frac{\partial \boldsymbol{x}}{\partial \boldsymbol{\theta}} \tag{6.48}$$

kann die Ausgangssensitivität aus folgender Systemdarstellung gewonnen werden:

$$\begin{aligned} \boldsymbol{S}[k+1] &= \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}} \boldsymbol{S}[k] + \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{\theta}}, \quad \boldsymbol{S}[k=0] = \frac{\boldsymbol{x}[k=0]}{\partial \boldsymbol{\theta}}, \\ \frac{\partial \boldsymbol{g}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}}[k] &= \frac{\partial \boldsymbol{g}}{\partial \boldsymbol{x}} \boldsymbol{S}[k] + \frac{\partial \boldsymbol{g}}{\partial \boldsymbol{\theta}}. \end{aligned} \tag{6.49}$$

Obiges Differenzengleichungssystem kann parallel zur Berechnung des eigentlichen Modells (6.17) gelöst werden, sofern die Ableitungen $\partial \boldsymbol{f}/\partial \boldsymbol{x}$, $\partial \boldsymbol{f}/\partial \boldsymbol{\theta}$, $\partial \boldsymbol{g}/\partial \boldsymbol{x}$, $\partial \boldsymbol{g}/\partial \boldsymbol{\theta}$ zuvor analytisch berechnet werden können. Demgegenüber besteht die Möglichkeit den Gradienten numerisch über die in Kap. 4.1.2 behandelten Verfahren zu approximieren. Die Vor- und Nachteile beider Ansätze sind in Tab. 6.1 zusammengefasst.

Für viele Optimierungsverfahren wird zudem noch die Hesse-Matrix benötigt. Diese kann unter erneuter Anwendung der Produktregel gewonnen werden und es ergibt sich zu:

$$\frac{\partial^2 J}{\partial \boldsymbol{\theta}^2} = \sum_{k=1}^{N} \left(\frac{\partial \boldsymbol{g}[k]}{\partial \boldsymbol{\theta}} \right)^T \boldsymbol{R}^{-1} \frac{\partial \boldsymbol{g}[k]}{\partial \boldsymbol{\theta}} + \sum_{j=1}^{p} \sum_{k=1}^{N} \left(\frac{\partial^2 (\boldsymbol{g}[k])_j}{\partial \boldsymbol{\theta}^2} \right)^T \left(\boldsymbol{R}^{-1} \boldsymbol{e}[k] \right)_j. \tag{6.50}$$

Sensitivitätsansatz	Numerische Approximation
Vorteile: • Präzise, da analytische Ableitungen genutzt werden. • Schnell, aufgrund der Einbindung vorab analytisch berechneter Ableitungen.	**Vorteile:** • Flexibel, da direkt das Ein-/Ausgangsverhalten modelliert wird. • Einfach, da keine analytischen Ableitungen notwendig.
Nachteile: • Nicht universell, da Ableitungen ggf. nicht berechenbar (z. B. aufgrund nicht differenzierbarer Funktionen) oder nur mit enormen Aufwand. • Nicht flexibel, da $\partial \boldsymbol{f}/\partial \boldsymbol{x}$, $\partial \boldsymbol{f}/\partial \boldsymbol{\theta}$, $\partial \boldsymbol{g}/\partial \boldsymbol{x}$, $\partial \boldsymbol{g}/\partial \boldsymbol{\theta}$ bei Strukturänderungen neu berechnet werden müssen.	**Nachteile:** • Hoher Berechnungsaufwand, insbesondere bei Systemen mit vielen Parametern und folglich vielen notwendigen Funktionsauswertungen. • Potentiell unpräzise, aufgrund von Rundungsfehlern sowie der Schrittweitenwahl in θ_i Richtung

Tab. 6.1: Vor- und Nachteile der analytischen und numerischen Berechnung des Gradienten

Hier ist insbesondere die Berechnung der 2. Ableitung des Modellausgangs

$$\frac{\partial^2 \boldsymbol{g}[k]}{\partial \boldsymbol{\theta}^2}$$

mit hohem numerischen Aufwand verbunden, sofern finite Differenzen zu dessen Approximation genutzt werden. Allerdings wird dieser Ausdruck in (6.50) mit den Residuen $\boldsymbol{e}[k]$ multipliziert, welche nach erfolgreicher Identifikation mittelwertfrei sind bzw. seien sollten. Analog zum *Gauss-Newton-Verfahren* aus Kap. 4.2.3 kann daher durch die Vernachlässigung des zweiten Terms in (6.50) der Berechnungsaufwand signifikant reduziert werden und es folgt:

$$\frac{\partial^2 J}{\partial \boldsymbol{\theta}^2} \approx \sum_{k=1}^{N} \left(\frac{\partial \boldsymbol{g}[k]}{\partial \boldsymbol{\theta}} \right)^T \boldsymbol{R}^{-1} \frac{\partial \boldsymbol{g}[k]}{\partial \boldsymbol{\theta}}. \tag{6.51}$$

Vorteilhaft an dieser Näherung ist zudem, dass die Ausgangssensitivität $\partial \boldsymbol{g}/\partial \boldsymbol{\theta}$ bereits nach der Berechnung des Gradienten vorliegt und die Berechnung der approximierten Hesse-Matrix daher mit vergleichsweise geringem Aufwand möglich ist.

6.2.3 Parameteridentifikation mittels Minimierung des Ausgangsfehlers

Der obige Optimierungsansatz für das ML-Problem soll nun zur Parameteridentifikation mittels Minimierung des Ausgangsfehlers (*output error model* - OEM) eingesetzt werden. Das entsprechende Blockdiagramm ist in Abb. 6.4 dargestellt. Hinsichtlich der ML-OEM-Methodik seien nochmal kurz die wesentlichen Annahmen zusammengefasst:

- Es liegt ein additives, unkorreliertes sowie normalverteiltes Messrauschen vor.
- Ein Systemrauschen liegt nicht vor.
- Die Systemanregung ist ausreichend, um das dynamische Verhalten vollständig abbilden zu können.

- Alle Eingangsgrößen sind exakt bekannt und unabhängig von den Systemausgängen (d. h. es liegt keine geschlossene Regelschleife vor).

Die letzte Annahme basiert auf der Forderung, dass sowohl die Eingangsgrößen und die Systemausgänge unkorreliert sind als auch das der Systemausgang unkorreliert mit sich selbst ist. Liegt eine geschlossene Regelschleife vor, resultiert hieraus unmittelbar eine Korrelation zwischen diesen Größen, welche von der Reglerauslegung, den Störeinwirkungen sowie der Solltrajektorie abhängt. Zudem kann ein Teil der Systemdynamik durch einen Regler verschleiert werden, hier sein beispielsweise die Auslegung nach dem Betragsoptimum oder der *internal model control*(IMC) Ansatz genannt, bei dem der Strecke jeweils eine definierte Dynamik aufgeprägt wird.

Da manche Systeme allerdings prinzipbedingt oder aus praktischen Gesichtspunkten ohne Regler nicht stabil sind, ist in diesem Fall die letzte Annahme eher als Anreiz zu verstehen einen möglichst hohen *Nutzsignalanteil* sicherzustellen, also hohe Sollwertaussteuerungen verglichen mit der vorliegenden Rauschamplitude. Hierzu kann entweder das Referenzsignal selbst oder ein additives Anregesignal entsprechend manipuliert werden. Ein klassisches Signal ist die *pseudo-random binary sequence* (PRBS)[1], welches das Spektrum von weißem Rauschen approximiert und durch einen deterministischen Zufallsgenerator erzeugt wird. Weitere Information zur Systemanregung erfolgen in Kap. 6.3 und Kap. 6.4.

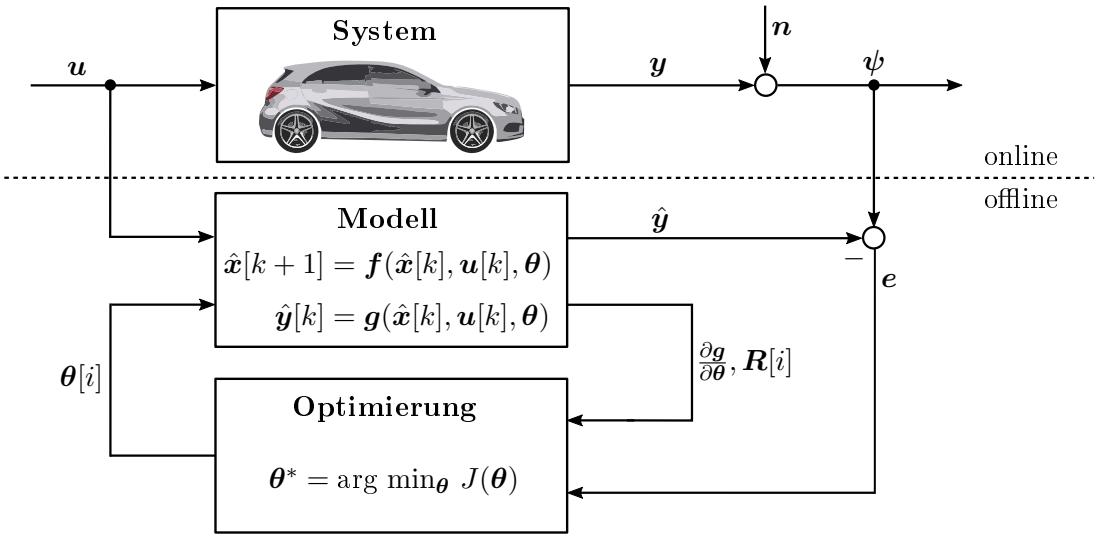


Abb. 6.4: Blockdiagramm zur Parameteridentifikation mittels Minimierung des Ausgangsfehlers (*output error model* - OEM)

Das Identifikationsvorgehen bei einem vorliegenden Datensatz ist in Algorithmus 6.6 zusammengefasst. Die dort dargestellte Abbruchbedingung ist lediglich als exemplarisch zu verstehen, da in vielen Anwendungen eine sinnvolle Kostenschranke im Vorhinein nur selten definiert werden kann. Typischerweise werden stattdessen Gütekriterien bezüglich der Residuen oder das Konvergenzverhalten des Algorithmus (relative Verbesserung, Anzahl Iterationen,...) zur Terminierung herangezogen. Je nach gewählten Optimierungsalgorithmus im 7. Schritt (ableitungsfrei oder

[1] In vielen Anwendungen wird das Aufschalten tatsächlicher PRBS-Signale aus technischen Gründen nicht möglich sein, beispielsweise weil die Stellgröße nicht sprungartig verändert werden kann. Insofern ist das PRBS-Signal als Idealwunsch einer gleichmäßigen Anregung in einem möglichst großen Frequenzband zu verstehen.

Algorithmus 6.6 Algorithmische Umsetzung zur Lösung des ML-OEM-Problems

Initialisierung:

1: $\boldsymbol{\theta}_0 = \boldsymbol{\theta}[i=0]$ ▷ Startlösung

2: $i = 1$ ▷ Iterationszähler

3: Konfiguriere unterlagerten Optimierungsalgorithmus

Iterieren:

4: **while** $J \leq \varepsilon$ and $i < N_i$ **do**

5: Berechne die Systemantwort $\boldsymbol{y}[k](\boldsymbol{\theta}[i])$.

6: Berechne die Residuen $\boldsymbol{e}[k]$ sowie die Kovarianzmatrix $\boldsymbol{R}$.

7: Berechne einen aktualisierten Parametersatz $\boldsymbol{\theta}[i]$ durch Minimierung von (6.30).

8: $i \leftarrow i + 1$

9: **end while**

ableitungsbehaftet), muss die Ausgangssensitivität $\partial \boldsymbol{g}/\partial \boldsymbol{\theta}$ noch vorab ermittelt werden, um Gradient und Hesse-Matrix zu berechnen.

Beispiel: P-T_1 Element entsprechend Kap. 6.1
Das in Kap. 6.1 vorgestellte SISO-LTI-System (6.13) unter Berücksichtigung verschiedener Rauscheinflüsse wird auch an dieser Stelle als Einstiegsbeispiel verwendet. Zur Initialisierung des ML-OEM-Algorithmus wurde die zuvor ermittelte OLS-Lösung herangezogen, als Optimierungsalgorithmus wurde die Quasi-Newton-Methode genutzt. Die entsprechenden Ergebnisse sind in Abb. 6.5 dargestellt. Es ist ersichtlich, dass trotz eines z. T. signifikanten Rauschens in allen Fällen eine zielführende Identifikation gelingt. Die quantitativen Abweichungen zwischen den Systemparametern und den identifizierten Werten sind gering, auch die Modellvalidierung ist zufriedenstellend.

Hierbei gilt es zu bedenken, dass für den ML-OEM-Algorithmus eigentlich die perfekte Kenntnis der Eingangsgrößen voraussetzt wurde, was in zwei der vier Szenarien durch Modellierung eines entsprechende Eingangsrauschen nicht der Fall ist. Dieses Eingangsrauschen wird über die Faktor b in ein Systemrauschen überführt, also einen Teil der Systemdynamik, welcher durch das eigentliche Modell nicht abgedeckt wird. Da im vorliegenden Beispiel $b << 1$ ist, führt das augenscheinlich signifikante Eingangsrauschen zu einem vergleichsweise geringen Systemrauschen, was sich im vorliegenden Fall nicht auswirkt. Um das Sytemrauschen noch einmal gesondert betrachten zu können, wird die Systembeschreibung des P-T_1 Elements wie folgt verändert:

$$\begin{aligned} x[k+1] &= ax[k] + b(u[k] + m_u[k]) + m_x[k] \\ y[k] &= x[k] + n_y[k]. \end{aligned} \tag{6.52}$$

Hier sind m_u, m_x und n_y separate Eingangs-, Zustands-, und Messrauschterme. Wie bereits erwähnt wirken dass Eingangs- und das Systemrauschen in gleicherweise, wobei das Messrauschen über den Parameter b skaliert wird. In Abb. 6.6 wurde das Beispiel daher erneut durchgeführt, wobei in allen Szenarien ein Systemrauschen $\mathcal{N}_x(\mu_x = 0, \sigma_x^2 = 0,02)$ modelliert wurde. Es ist ersichtlich, dass in allen Fällen signifikante Abweichungen zu den wahren Parameterwerten

$$a = 0,983, \quad b = 0,0167$$

auftreten. Dies ist insofern beachtlich, da das Systemrauschen eine Größenordnung kleiner ist als das Mess- und Eingangsrauschen und es trotzdem zu nennenswerten Abweichungen kommt. Der ML-OEM-Algorithmus berücksichtigt diese Art des Störungseinflusses nicht und versucht daher durch entsprechende Variation der gesuchten Parameterwerte ausschließlich den Ausgangsfehler zu minimieren.

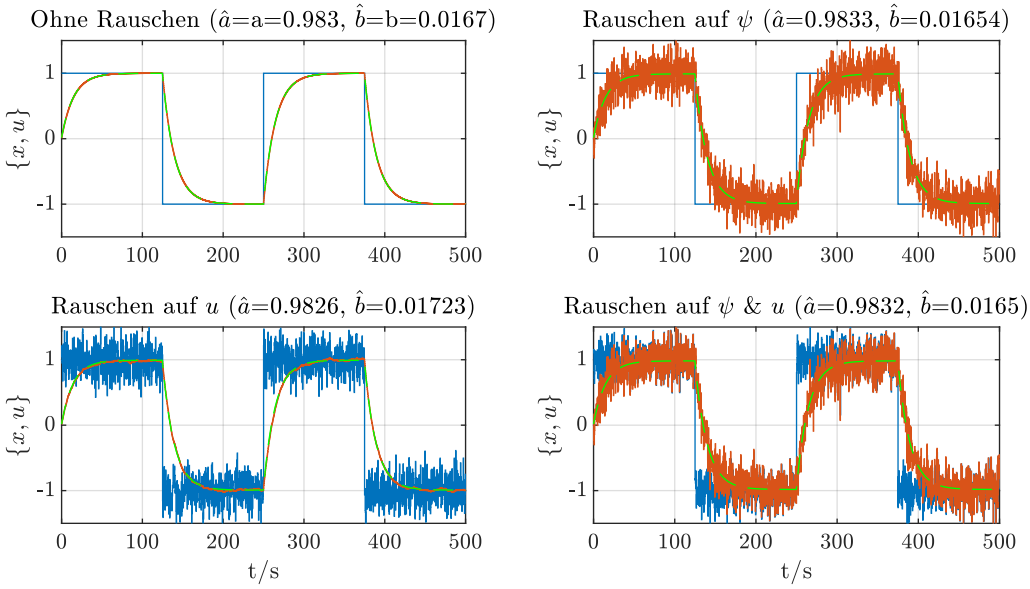


Abb. 6.5: Identifikationsergebnisse für unterschiedliche Rauscheinflüsse (rot = Zustandssignal, blau = Eingangssignal, grün-gestrichelt = Modellvalidierung, normalverteiltes Mess- und Eingangsrauschen $\mathcal{N}(\mu = 0, \sigma^2 = 0,2)$

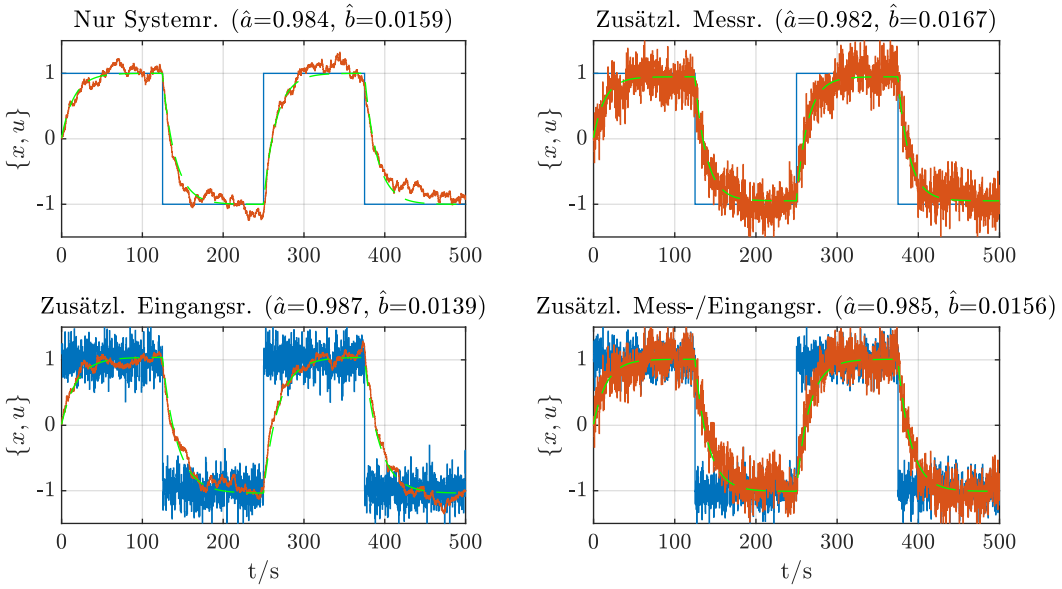


Abb. 6.6: Identifikationsergebnisse für unterschiedl. Rauschen (rot = Zustandssignal, blau = Eingangssignal, grün-gestrichelt = Modellvalidierung, normalverteiltes Eingangs-/Messrauschen $\mathcal{N}(\mu = 0, \sigma^2 = 0,2)$ bzw. Systemrauschen $\mathcal{N}_x(\mu_x = 0, \sigma_x^2 = 0,02)$

Beispiel: RLC-Schwingkreis
Als weiteres Anwendungsbeispiel wird ein RLC-Schwingkreis mit externer, idealer Stromquelle gemäß Abb. 6.7 diskutiert. Die Systemzustände sind der Spulenstrom $x_1 = i_L$ und die Konden-

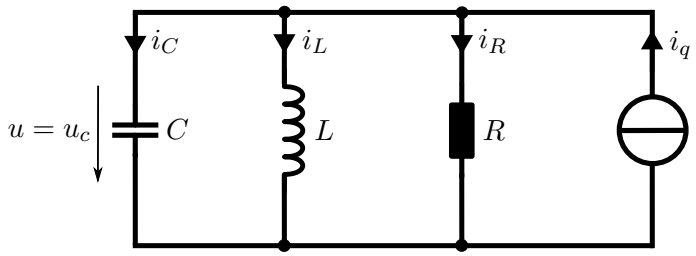


Abb. 6.7: RLC-Schwingkreis mit externer Stromquelle

satorspannung $x_2 = u_C$. Das zeitkontinuierliche Zustandsraummodell ergibt sich entsprechend zu

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t}\begin{bmatrix} i_L \\ u_C \end{bmatrix} &= \begin{bmatrix} 0 & \frac{1}{L} \\ \frac{-1}{C} & \frac{-1}{RC} \end{bmatrix}\begin{bmatrix} i_L \\ u_C \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{C} \end{bmatrix} i_q \,, \\ \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} i_L \\ u_C \end{bmatrix} . \end{aligned} \tag{6.53}$$

Der Quellenstrom i_q entspricht hierbei dem Systemeingang durch den eine Anregung erfolgt. Ferner ist der Anfangszustand

$$\boldsymbol{x}_0 = \begin{bmatrix} 0\ \mathrm{A} & 0\ \mathrm{V} \end{bmatrix}^T \tag{6.54}$$

für alle nachfolgenden Betrachtungen gegeben und bekannt[1]. Die wahren Systemparameter sind gegeben durch

$$R = 2{,}0\ \Omega\,, \quad L = 0{,}6\ \mathrm{H}\,, \quad C = 0{,}4\ \mathrm{F}\,. \tag{6.55}$$

Typische Kenngrößen für den RLC-Schwingkreis sind die Kennfreuenz f_0, der Kennwiderstand Z_0 und die Dämpfung d:

$$f_0 = \frac{1}{2\pi\sqrt{LC}} = 0{,}325\ \mathrm{Hz}, \quad Z_0 = \sqrt{\frac{L}{C}} = 1{,}23\ \Omega, \quad d = \frac{1}{2}\frac{Z_0}{R} = 0{,}3\,. \tag{6.56}$$

Um das hier vorgestellte ML-OEM-Verfahren anwenden zu können, muss das System zeitlich disktretisiert werden. Hierfür wird das explizite Euler-Verfahren gemäß (2.27) angewandt:

$$\begin{bmatrix} i_L[k+1] \\ u_C[k+1] \end{bmatrix} = \begin{bmatrix} 1 & \frac{T_a}{L} \\ \frac{-T_a}{C} & 1 + \frac{-T_a}{RC} \end{bmatrix}\begin{bmatrix} i_L[k] \\ u_C[k] \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{T_a}{C} \end{bmatrix} i_q\,. \tag{6.57}$$

Hier entspricht T_a der Abtastzeit des Systems. Um nachfolgend die Ergebnisse verschiedener Identifikationsszenarien vergleichen zu können, werden neben der Kap. 3.1.3 eingeführten Kenngrößen zur Genauigkeitsanalyse, noch die mittlere, normierte euklidische Distanz D zwischen

[1] Natürlich ist es ohne weiteres möglich auch den Anfangszustand $\boldsymbol{x}_0$ als unbekannten Parameter zu interpretieren und im Zuge der Identifikation zu ermitteln. Hierauf wird an dieser Stelle allerdings verzichtet.

geschätzten und wahren Parametern eingeführt:

$$D = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\frac{\theta_i - \hat{\theta}_i}{\theta_i}\right)^2}\,. \tag{6.58}$$

Hierzu ist in Abb. 6.8 ein Referenzbeispiel dargestellt. Die Anregung des Systems erfolgt über ein als ideal angenommenes PRBS, dessen Bandbreite auf 5 % der Nyquist-Frequenz für die gegebene Abtastrate gesetzt wurde. Zudem wird ein additives, mittelwertfreies, unkorreliertes und normalverteiltes Rauschen auf den beiden Messgrößen angenommen. Wie der Darstellung zu entnehmen ist, führt die Anwendung des ML-Verfahrens zu einer zufriedenstellenden Identifikation. Das Bestimmtheitsmaß R^2 beider Zustände liegt über 98 %, die Autokorrelation der Residuen liegt (näherungsweise) im 95 % Konfidenzintervall und der mittlere, normierte euklidische Parameterabstand liegt unterhalb von 1 %.

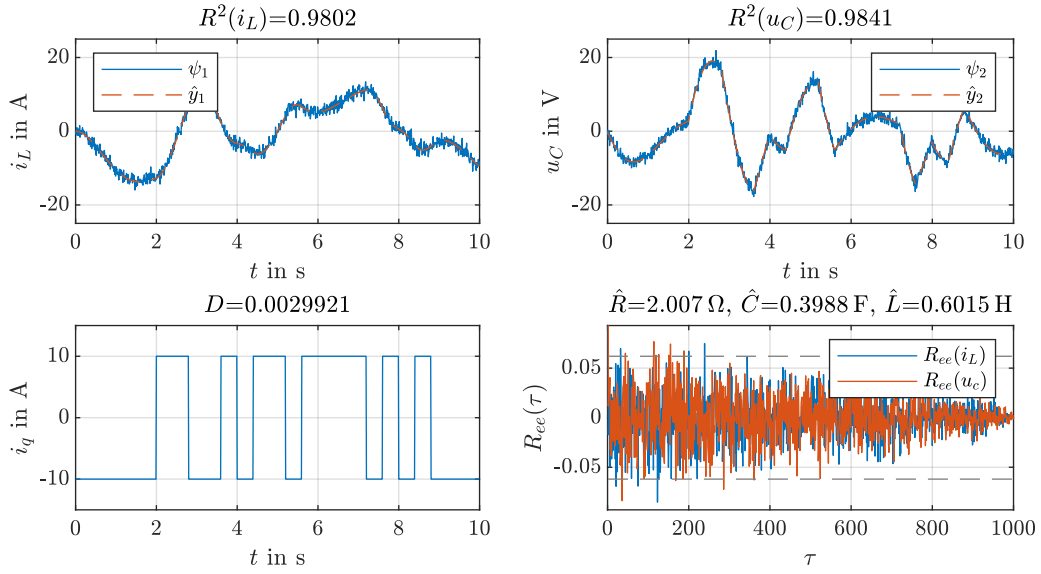


Abb. 6.8: Identifikationsergebnis mit normalverteilten Messrauschen $\mathcal{N}(\mu_{\psi_1} = 0, \sigma_{\psi_1} = 1\,\text{A})$, $\mathcal{N}(\mu_{\psi_2} = 0, \sigma_{\psi_2} = 1\,\text{V})$, ohne Eingangsrauschen und $T_a = 10\,\text{ms}$

Des Weiteren ist in Abb. 6.9 das gleiche Beispiel unter Modellierung eines zusätzlichen Systemrauschens dargestellt. Im Kontext des RLC-Schwingkreises kann das Systemrauschen beispielsweise als parasitärer, hochfrequenter Anteil der Leitungsimpedanz interpretiert werden, welcher nicht im einfachen Modell (6.57) berücksichtigt wird. Obwohl auch hier das Systemrauschen hinsichtlich seiner Varianz deutlich unterhalb des Messrauschen angesetzt wurde, ist der Effekt auf die Parametertreue enorm, da der mittlere, normierte euklidische Parameterabstand nun oberhalb von 12 % liegt. Auch die weiteren Kennzahlen zur Genauigkeitsbewertung der Ausgangsgrößen deuten auf ein signifikant schlechteres Ergebnis hin.

Der ML-OEM-Ansatz geht von einer perfekten Kenntnis der Modellstruktur aus. Jegliche strukturelle Abweichung zwischen realem System und dem mathematischen Modell zur Parameteridentifikation führt zu einer Verzerrung der identifizierten Parameterwerte, selbst dann, wenn die Modellabweichung nur als mittelwertfreies, normalverteiltes Rauschen dargestellt werden kann. Für eine zielführende Anwendung des ML-OEM-Algorithmus muss daher sichergestellt

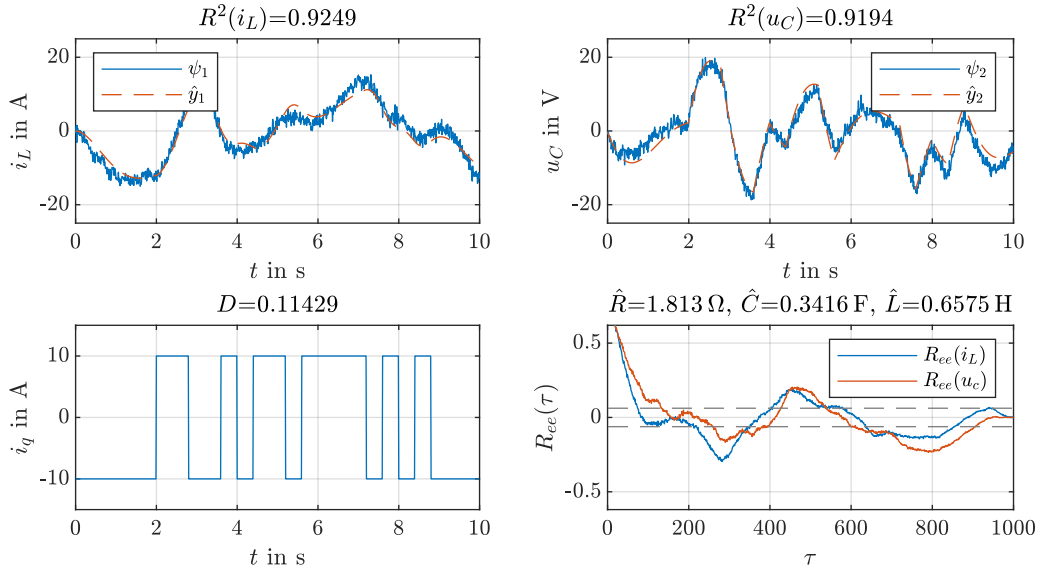


Abb. 6.9: Identifikationsergebnis basierend auf dem Szenario in Abb. 6.8 mit zusätzlichen Systemrauschen $\mathcal{N}(\mu_{x_1} = 0, \sigma_{x_1} = 0,25\ \text{A})$, $\mathcal{N}(\mu_{x_2} = 0, \sigma_{x_2} = 0,25\ \text{V})$

sein, dass das Modell das reale System äußerst genau abbildet und das das Eingangssignal hinreichend genau bekannt ist, was ggf. mit empfindlichen Modellierungs- und Messaufwand (auch bezüglich der Aktuatoren) verbunden ist.

6.2.4 Parameteridentifikation mittels Minimierung des Filter-Ausgangsfehlers

Ist ein signifikantes Systemrauschen nicht auszuschließen oder nur mit nicht vertretbarem Aufwand zu eliminieren, kann die Berechnung des Modellausgangs durch Integration in ein Kalman-Filter gestützt werden (*filter error model* - FEM). Die Kostenfunktion der Maximum-Likelihood-Methode wird dann auf den prädizierten Ausgangswert des Kalman-Filters bezogen. Analog zum Kalman-Ansatz kann für das Modell im ML-Kontext nun ein System- und ein Messrauschen angenommen werden:

$$\begin{aligned} \boldsymbol{x}[k+1] &= \boldsymbol{f}\left(\boldsymbol{x}[k], \boldsymbol{u}[k]\right) + \boldsymbol{m}[k], \\ \boldsymbol{y}[k+1] &= \boldsymbol{g}\left(\boldsymbol{x}[k+1], \boldsymbol{u}[k+1]\right) + \boldsymbol{n}[k+1]. \end{aligned} \tag{6.59}$$

Der entsprechende Integrationsansatz ist vereinfacht in Abb. 6.10 dargestellt. In Folge der Korrektur des internen Modells in jedem Abtastschritt durch das KF werden zusätzliche Stützstellen in die Ausgangsprädiktion eingebracht, sodass das Modell nicht wie beim OEM-Ansatz als vollkommen offener Schätzer betrieben werden muss. Es ist allerdings zu beachten, dass die 1-Schrittprädiktion $\hat{\boldsymbol{y}}[k|k-1]$ des Modellausgangs für die Identifikation herangezogen wird und nicht die korrigierte Schätzung $\hat{\boldsymbol{y}}[k|k]$ (siehe Abb. 6.10 am rechten Bildrand). Andernfalls würde die Messung $\boldsymbol{y}[k]$ genutzt werden, um im gleichen Schritt sowohl das KF intern zu korrigieren als auch, um das Residuum $\boldsymbol{e}[k]$ für die Modelloptimierung zu berechnen, was zu einer Verzerrung der Modellgüte führen würde. Auch würde die Wahl einer beliebig kleinen Messrauschmatrix in diesem Fall $\hat{\boldsymbol{y}}[k|k]$ und $\boldsymbol{y}[k]$ künstlich angleichen und das interne Prädiktionsmodell wäre ohne Einfluss auf $\boldsymbol{e}[k]$.

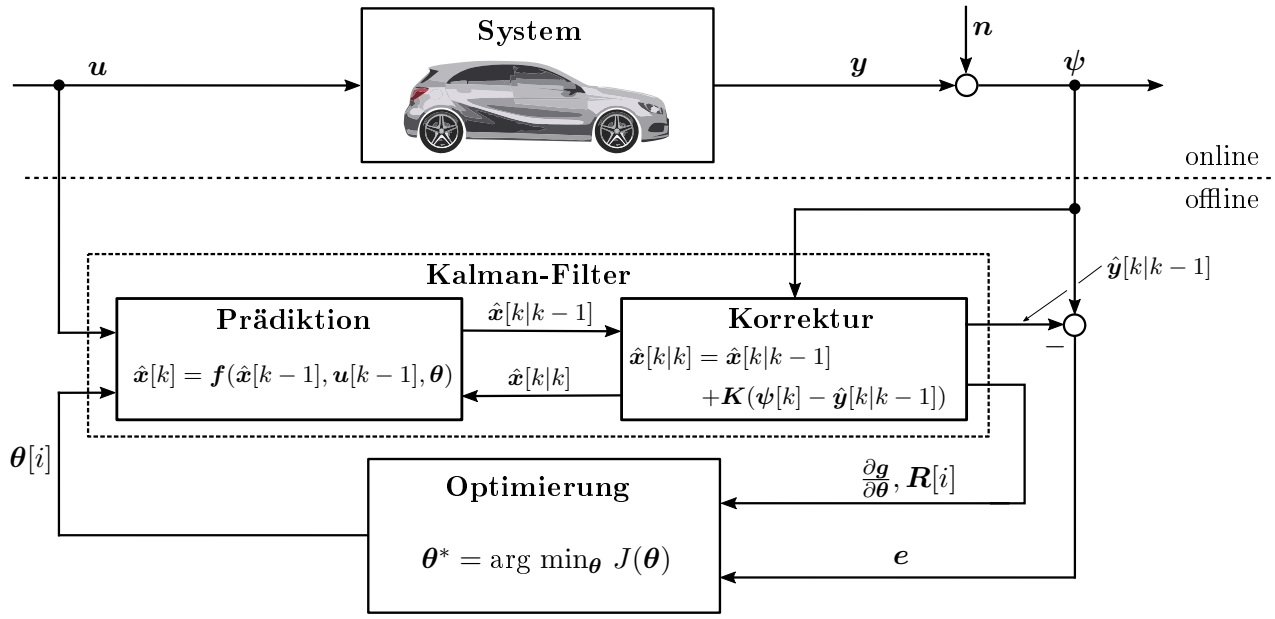


Abb. 6.10: Blockdiagramm zur Parameteridentifikation mittels Minimierung des gefilterten Ausgangsfehlers (*filter error model* - FEM)

Je nach zugrundeliegendem Modell, kann eine der in Kap. 5 vorgestellten Kalman-Filter-Varianten herangezogen werden, wobei hier im Wesentlichen zwischen den linearen und den nichtlinearen KF-Ansätzen (EKF, UKF) zu differenzieren ist. Unabhängig hiervon ist jedes Kalman-Filter geeignet zu parametrieren, d. h. die Rauschenmatrizen sowie die inertiale Kovarianzmatrix

$$\boldsymbol{M}[k], \quad \boldsymbol{N}[k], \quad \boldsymbol{P}[0|0] = \boldsymbol{P}_0 \tag{6.60}$$

müssen bedatet werden[1]. In einigen Anwendungen werden diese Kenngrößen a priori unbekannt sein, sodass diese als einstellbare Freiheitsgrade des FEM-ML-Ansatzes zu interpretieren sind. Um die Anzahl der einstellbaren Parameter im Rahmen zu halten, ist es zielführend, die Rauschmatrizen als konstant

$$\boldsymbol{M}[k] = \boldsymbol{M}, \quad \boldsymbol{N}[k] = \boldsymbol{N}$$

anzunehmen[2]. Hinsichtlich alternativer KF-Auslegungen können im Wesentlichen zwei Ansätze verfolgt werden:

Erweiterung des Optimierungsproblems
Sind die Einträge in $\boldsymbol{M}$ und $\boldsymbol{N}$ gänzlich unbekannt, können diese als zu identifizierende Parameter interpretiert werden. Der gesuchte Parametervektor vergrößert sich somit zu

$$\tilde{\boldsymbol{\theta}} = \begin{bmatrix} \boldsymbol{\theta}^T & M_{ij} & N_{st} \end{bmatrix}^T, \quad \forall\{i,j\} = 1\ldots,n\,, \quad \forall\{s,t\} = 1\ldots,p\,. \tag{6.61}$$

Dieser Ansatz ist sehr direkt und einfach anwendbar, hat allerdings die folgenden Nachteile:

[1] Im Falle des UKFs sind ggf. noch weitere Freiheitsgrade hinsichtlich der Unscented-Transformation zu wählen. Hierauf wird an dieser Stelle allerdings nicht weiter eingegangen

[2] Ebenfalls aus praktischen Gründen kann es zielführend sein, die Rauschmatrizen als Diagonalmatrizen anzunehmen, um die Anzahl der freien Parameter weiter einzugrenzen.

- *Konvergenz*: Der Parameterraum wird massiv ausgeweitet und $\boldsymbol{\theta}$ ist zudem abhängig von $\boldsymbol{M}$ und $\boldsymbol{N}$. Folglich liegt ein schwieriges Optimierungsproblem vor, welches insbesondere bei nichtlinearen Modellen die Gefahr vielfältiger lokaler Minima aufweist. Die Lösung des Problems hängt daher stark von der Initialisierung ab bzw. benötigt wohl möglich globale Optimierungsansätze.
- *Berechnungsaufwand*: Gegenüber dem OEM-ML-Ansatz liegt bereits i. A. beim FEM-ML-Verfahren ein höherer Berechnungsaufwand vor, da zusätzlich die KF-Vorschrift berechnet werden muss. Durch die gesteigerte Parameteranzahl resultieren zudem bei ableitungsbehafteten Optimierungsverfahren mehr Funktionsauswertungen zur Approximation von Gradient und Hesse-Matrix. Auch meta-heuristische Verfahren benötigen dann i. d. R. mehr Funktionsauswertungen, um zielführend zu konvergieren.
- *Verzerrung*: Durch obige Freiheitsgrade bezüglich der Auslegung des Kalman-Filters könnten $\boldsymbol{M}$ und $\boldsymbol{N}$ derart gewählt werden, dass durch eine starke Filterkorrektur die Schätzung $\hat{\boldsymbol{y}}$ künstlich an die Messwerte $\boldsymbol{\psi}$ angepasst werden, da dies zur Minimierung der Kostenfunktion (6.33) beiträgt. Die Auswirkung von $\boldsymbol{\theta}$ auf das Ergebnis wird in diesem Fall verzerrt. Es ist daher ratsam im Zuge der Optimierung konservative Grenzen für $\boldsymbol{M}$ und $\boldsymbol{N}$ zu wählen.

Stationäres Kalman-Filter

Je nach Anwendung mag es ggf. ausreichend sein mit einer konstanten Korrektur-Matrix $\boldsymbol{K}$ zu arbeiten. Dies hat zum einen den Vorteil, dass ein Teil der rechenintensive KF-Schritte nicht benötigt werden und somit ein Modelldurchlauf deutlich schneller abläuft. Zum anderen sinkt i. d. R. die Anzahl der zu identifizierenden Parameter, da der augmentierte Parametervektor sich zu

$$\tilde{\boldsymbol{\theta}} = \begin{bmatrix} \boldsymbol{\theta}^T & K_{ij} \end{bmatrix}^T, \quad \forall \{i, = 1 \ldots, n \quad j = 1 \ldots, p\}. \tag{6.62}$$

ergibt. Gegenüber dem vorherigen Ansatz ergibt sich folgender Nachteil[1]:

- *Interpretierbarkeit*: Während $\boldsymbol{M}$ und $\boldsymbol{N}$ physikalische Größen repräsentieren und auf Basis der vermuteten Unsicherheit der entsprechenden System- und Messgrößen zumindest hinsichtlich ihres Wertebereich grob eingegrenzt werden können, ist dies bei der direkten Ermittlung von $\boldsymbol{K}$ weniger intuitiv möglich. Daher müssen bei der Definition des (beschränkten) Optimierungsproblems größere Wertebereiche berücksichtigt werden, was die Wahrscheinlichkeit in ein lokales Nebenminimum zu konvergieren erhöht.
- *Verzerrung*: Dieser Aspekt bleibt ähnlich problematisch, da auch hier durch eine starke Filterkorrektur die Schätzung $\hat{\boldsymbol{y}}$ künstlich an die Messwerte $\boldsymbol{\psi}$ angepasst werden kann und so die Modellgüte verschleiert.

6.3 Identifikation im geschlossenen Regelkreis

Damit die Anregung u und Messung ψ unkorreliert sind, ist eine Datenaufnahme außerhalb geschlossener Regelkreise empfehlenswert. In einigen Fällen lässt sich dies allerdings nicht realisieren, z. B. wenn das System instabil ist oder ohne Regelung sicherheitsrelevante Systemgrenzen

[1] Die Verzerrungsproblematik durch die interne Filterkorrektur bleibt natürlich bestehen, sodass auch bei der direkten Optimierung von $\boldsymbol{K}$ konservative Grenzen gewählt werden sollten.

leicht überschritten werden. In diesem Fall muss eine Identifikation im geschlossenen Regelkreis erfolgen. Hierzu werden in folgenden zwei Ansätze gegenübergestellt[1]. Zuvor sei allerdings ein vereinfachtes Beispiel gegeben, welches die Problematik verdeutlicht.

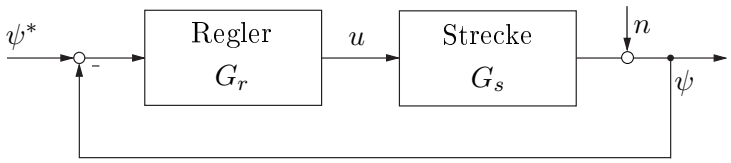


Abb. 6.11: Einfacher, abstrahierter SISO-Regelkreis

Beispiel
Um die Korrelationsproblematik im geschlossenen Regelkreis zu verdeutlichen, sei folgendes Beispiel angeführt (entlehnt aus [Len17]): Die Parameter des Systems

$$y[k] = ay[k-1] + bu[k-1] \tag{6.63}$$

sollen ermittelt werden, während dieses mittels eines P-Glieds

$$u[k] = k_p \left(\psi^*[k] - \psi[k]\right) \tag{6.64}$$

mit Verstärkung k_p geregelt wird. Für die Sollgröße $\psi^*[k] = \psi^* = 0$ folgt für den rauschfreien Fall ($\psi = y$) unmittelbar

$$y[k] = ay[k-1] - bk_p y[k-1]. \tag{6.65}$$

Für die k-te Regressorzeile des LS-Ansatzes resultiert:

$$\boldsymbol{\xi}^T[k] = \begin{bmatrix} y[k-1] & k_p y[k-1] \end{bmatrix}. \tag{6.66}$$

Die beiden Regressoren sind somit vollständig linear abhängig und die über mehrere Abtastschritte gewonnene Produktsummenmatrix ($\boldsymbol{\Xi}^T\boldsymbol{\Xi}$) ist daher nicht invertierbar. Das LS-Verfahren kann daher prinzipbedingt nicht angewandt werden und auch die ML-Methodik ist numerisch derart schlecht konditioniert, dass diese keine sinnvollen Ergebnisse liefert.

Ein möglicher Einwand, dass in einer realen Applikation das Messrauschen eine gewisse Regler-Anregung hervorruft und obiges Beispiel daher zurückzuweisen sei, ist häufig nicht stichhaltig. Typischerweise werden Regelkreise derart ausgelegt, dass die Bandbreite der Regler begrenzt werden, um nicht auf das Rauschen zu reagieren, da hierdurch die gesamte harmonische Verzerrung (*total harmonic distortion* - THD) der Regelgröße reduziert wird (insbesondere in mechatronischen Anwendungen mit Aktoren begrenzter Dynamik). Und auch für den Fall, dass der Regler auf das Rauschen reagiert, ist der hierdurch gewonnene Nutzsignalanteil derart gering, dass im Identifikationsprozess i. A. numerische Probleme auftreten.

Des Weiteren spielt die Regler-Auslegung und dessen Dynamik im Vergleich zur Strecke natürlich auch eine wichtige Rolle. Kompensationsregler (beispielsweise nach dem Betragsoptimum) oder auch Regler nach dem *internal model control* (IMC) Prinzip prägen der Strecke ein definiertes dynamisches Verhalten auf, welches mitunter die Dynamik des eigentlichen Systems

[1] In [Lju99] wird noch ein dritter Ansatz zur gemeinsamen Schätzung der Regler- und der Streckenparameter vorgestellt, sofern der Regler ebenfalls unbekannt ist. Notwendige Voraussetzung ist die Kenntnis über ψ^*, u und ψ. Da es sich hierbei um einer eher exotisches Problem handelt, wird es an dieser Stelle nicht vertieft.

maßgeblich schattiert. Eine derart gewonnene Datenbasis wird tendenziell wenig Informationen bezüglich der für die Strecke relevanten Frequenzbereichen aufweisen und zu einer schlechten Identifikation führen. Um dies zu verhindern bietet es sich an, entweder Regler mit maximal möglicher Bandbreite heranzuziehen und das Spektrum dann breitbandig über ψ^* anzuregen oder eine Datenbasis unter Verwendung unterschiedlich ausgelegter Regler zu erstellen, um den Einflusses einen speziellen Reglers zu reduzieren. Nachfolgend werden in diesem Rahmen zwei mögliche Lösungsansätze vorgestellt.

Indirekte Methode
Bei der indirekten Methode wird der gesamte geschlossene Regelkreis von der Sollgröße ψ^* bis zur Messgröße ψ betrachtet. Unter Annahme einer weiterhin zeitdiskreten Modellierung für ein SISO-System folgt:

$$\frac{\psi(z)}{\psi^*(z)} = G_g(z) = \frac{G_r(z)G_s(z)}{1 + G_r(z)G_s(z)}. \tag{6.67}$$

Hier sind $G_s(z)$, $G_r(z)$ und $G_g(z)$ die z-Übertragungsfunktionen der Strecke, des Reglers und der geschlossenen Regelschleife. Mittels des Rechts-Verschiebesatzes der z-Transformation gemäß (2.34) kann $G_g(z)$ zunächst in den Zeitbereich überführt werden, sofern ein LS- und ML-Ansatz zur Identifikation herangezogen werden soll – alternativ kann natürlich auch eine Identifikation im Frequenzbereich erfolgen (siehe z. B. [IM11]). Ist $G_g(z)$ identifiziert, kann dann auf $G_s(z)$ geschlossen werden:

$$G_s(z) = \frac{G_g(z)}{G_r(z)(1 - G_g(z))}. \tag{6.68}$$

Bei diesem indirekten Ansatz sind folgende Aspekte als kritisch zu bewerten:

- Annahme, dass System sei linear (d. h. es liegen auch keinerlei Stellgrößenbegrenzungen oder Anti-Windup-Maßnahmen vor).
- Annahme, dass $G_r(z)$ exakt bekannt sei.

Diese Einschränkungen begrenzen den Praxiseinsatz der Methodik deutlich.

Direkte Methode
Bei der direkten Methode wird über die Kenntnis der Stellgröße u und der Messung ψ direkt das Streckenverhalten ermittelt:

$$\frac{\psi(z)}{u(z)} = G_s(z). \tag{6.69}$$

Hierdurch korreliert ψ mit u bzw. das Rauschen n mit u (der Regler reagiert ggf. auf Rauscheinflüsse), sodass der Nutzsignalanteil gegenüber eine Identifikation außerhalb geschlossener Regelschleifen gemindert wird. Um trotzdem eine zielführende Identifikation zu gewährleisten, muss eine Anregung gefunden werden, welche innerhalb der zulässigen Systemgrenzen den Nutzsignalanteil maximiert. Mögliche Ansatzpunkte hierzu sind:

- PRBS-artige Solltrajektorien ψ^* mit variierenden Amplituden[1] (sofern möglich),
- PRBS-artige, additive zusätzliche Anregung auf u (innerhalb der Stabilitätsgrenzen).

[1] Unterschiedliche Amplituden sind besonders für nichtlineare Systeme relevant, da diese ein arbeitspunktabhängiges Verhalten aufzeigen (z. B. Sättigung bei magnetischen Bauteilen). Eine durchgängig große Amplitude resultiert zwar in einem wünschenswert hohen Nutzsignalanteil, allerdings wird das nichtlineare Systemverhalten ggf. nicht im gesamten Arbeitsbereich zielführend erfasst.

Neben einer PRBS-basierten (additiven) Anregung können natürlich auch weitere Signalformen (siehe Kap. 6.4.2) berücksichtigt werden.

6.4 Weitere praktische Aspekte der Identifikation dynamischer Systene

6.4.1 Wahl der Abtastzeit

Die Wahl der Abtastrate zur Aufnahme einer Datenbasis kann entscheidend für Erfolg oder Misserfolg eines Identifikationsvorgangs sein. Die triviale Wahl könnte auf die technisch höchst mögliche Abtastrate fallen. Diese ist begrenzt durch:

- *Bandbreite Sensor*: Ein Sensor kann vereinfacht als P-T_1-Filter mit Bandbreite bzw. Grenzfrequenz f_b aufgefasst werden. Folglich ist davon auszugehen, dass Signalanteile einer Frequenz oberhalb von f_b signifikant gedämpft werden. Gemäß Nyquist-Shannon--Abtasttheorem sollte die Abtastrate $f_a > 2f_b$ betragen. Für $f_a >> 2f_b$ wird durch die Sensor-Dämpfung allerdings kein wesentlicher, zusätzlicher Nutzsignalanteil aufgenommen, sodass die Wahl einer Abtastrate deutlich oberhalb der Empfehlung des Abtasttheorems nicht gewinnbringend ist.
- *Analog-Digital-Umsetzung:* Der bzw. die eingesetzten A/D-Wandler verfügen über eine minimale Wandlungszeit, welche die Abtastrate nach oben hin begrenzt. Eine mit dem A/D-Wandler verknüpfe Auswerte- bzw. Verarbeitungselektronik (z. B. Microprozessor) kann darüber hinaus die Abtastrate aufgrund der anfallenden Rechenlast begrenzen, beispielsweise wenn eine größere Anzahl an Messwerten gleichzeitig aufgenommen und gespeichert werden soll oder zeitgleich noch ein Regelungsalgorithmus berechnet werden muss.

Was spricht demgegenüber gegen die Wahl der technisch maximal möglichen Abtastrate:

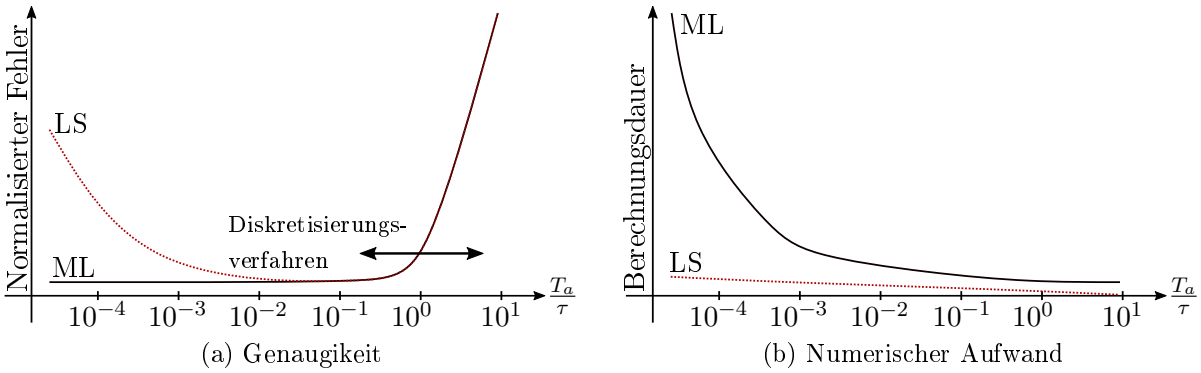
- *Messrauschen*: Hochfrequente Rauschanteile werden in die Messdaten aufgenommen[1].
- *Numerische Probleme*: Bei sehr geringer Abtastzeit ist die Änderung im Messsignal zwischen zwei Abtastschritten ebenfalls äußerst gering. Auf Basis der Messdaten durchgeführte numerische Operationen, beispielsweise bezogen auf die Produktsummenmatrix[2] $\boldsymbol{\Xi}^T\boldsymbol{\Xi}$, sind daher schlecht konditioniert und werden somit potentiell ungenau.
- *Datenmenge*: Eine hohe Abtastrate führt bei gegebener Messdauer zu einer steigenden Anzahl an Messpunkten und somit zunehmenden Datenmenge. Der hierfür notwendige Speicherplatzbedarf kann ggf. beschränkend wirken.
- *Verarbeitungsaufwand*: Jedes numerische Lösungsverfahren zum Auffinden von Parametern eines (dynamischen) Systems wird i. A. mit der Anzahl der Messpunkte ebenfalls länger für die Berechnung eines Parametersatzes benötigen. Je nach verwendeten Verfahren ist häufig ein näherungsweise linearer oder sogar superlinearer Zusammenhang beobachtbar.

[1] Typischerweise wird vor der AD-Wandlung ein Anti-Aliasing-Filter (analoges Tiefpass-Filter) eingesetzt, um störende Frequenzanteile im Sinne des Alias-Effekts zu bedämpfen. Unterhalb der Grenzfrequenz dieses Filters können natürlich weiterhin Rauschanteile in das gewandelte Signal eingehen.

[2] Hier sind konkret die Zeilen in $\boldsymbol{\Xi}$ nahezu linear abhängig, was die Berechnung der Inversen bzw. die Nutzung einer äquivalenten Zerlegung erschwert.

Obige Argumente motivieren somit eine möglichst geringe Abtastrate bzw. ein großes Abtastintervall T_a zu wählen. Allerdings kann auch keine beliebig geringe Abtastrate gewählt werden, da dies die Dynamik des zu identifizierenden Systems begrenzt (der Informationsgehalt im relevanten Frequenzbereich wird durch das Anti-Aliasing-Filter begrenzt). Die genannten Zusammenhänge werden in Abb. 6.12 bildlich für das LS- sowie ML-Verfahren dargestellt[1]:

- Ist $T_a > \tau$ wird die Systemdynamik nicht mehr ausreichend in den Messdaten widergespiegelt[2]. Es folgen systematische Fehler. Der Berechnungsaufwand ist hingegen für beide Verfahren sehr gering.
  - Die Wahl des zeitlichen Diskretisierungsverfahren (siehe Kap. 2.2) hat ebenfalls Einfluss auf die Identifikationsgenauigkeit, da der numerische Diskretisierungsfehler die Systemdynamik überdeckt.
  - So führt das Euler-Vorwärtsverfahren mit einer Abtastzeit in der selben Größenordnung wie die kleinste Systemzeitkonstante i. d. R. bereits zu signifikanten Diskretisierungsfehlern.
- Für $T_a << \tau$ tritt beim LS-Verfahren zwangsweise verstärkte Multikollinearität in der Produktsummenmatrix auf, sodass numerische Berechnungsprobleme zu Parameterabweichungen führen. Das ML-Verfahren ist hier i. d. R. deutlich robuster, wobei der eingesetzte Optimierungsalgorithmus und dessen Konfiguration maßgeblichen Einfluss auf die Genauigkeit haben können. Der Berechnungsaufwand beider Verfahren ist für ein sinkendes T_a für beide Verfahren näherungsweise linear, wobei der Berechnungsaufwand für das ML-Verfahren besonders stark zunehmen kann (z. B. aufgrund numerischer Probleme bei der Berechnung der Differenzenquotienten).



(a) Genaugikeit (b) Numerischer Aufwand

Abb. 6.12: Einfluss der Abtastzeit T_a auf Genauigkeit und den numerischen Aufwand für den Least-Squares- sowie den Maximum-Likelikehood-Ansatz für eine charakteristische Zeitkonstante τ eines zu identifizierenden Systems (vereinfachte Darstellung)

Obige Aspekte zusammenfassend, kann für die Wahl der Abtastrate T_a bei gegebener, kleinster Systemzeitkonstante[3] τ folgende Empfehlung gegeben werden (vgl. [IM11]): Liegt eine hinrei-

[1] Dort ist mit Zeitkonstante vereinfacht die Inverse der charakteristischen Systemfrequenz gemeint. Bei einem linearem Modell im Zustandsraum somit die Inverse der Eigenwerte der Systemmatrix.

[2] Für Systeme mit mehreren Zeitkonstanten ist hier die kleinste Zeitkonstante maßgeblich. Für Systeme mit stark unterschiedlichen Zeitkonstanten kann ggf. eine iterative Identifikation von Teilsystemen ähnlicher Dynamik sinnvoll sein.

[3] Dies impliziert, dass im Vorfeld der Messdatenaufnahme τ bereits hinreichend genau bekannt ist, beispielsweise durch applikationsspezifisches Domänenwissen oder durch Vortests.

chend genaue, zeitliche Diskretisierung des Systems vor, ist der Bereich

$$\frac{1}{5}\tau < T_a < \frac{3}{5}\tau \tag{6.70}$$

eine zielführende Wahl. Ist die zeitliche Diskretisierung hingegen als tendenziell einfach zu beschreiben (z. B. Euler-Vorwärts), sollte die Abtastzeit zur Vermeidung numerischer Abweichungen weiter (deutlich) reduziert werden. Daher empfiehlt [Hol18] pauschal

$$T_a \approx \frac{1}{25}\tau \tag{6.71}$$

zu wählen. In der weiteren Literatur sind darüber hinaus ähnliche Empfehlungen zu finden, welche eher als empirische Richtlinien und weniger als Ergebnis ausgedehnter Herleitungen zu verstehen sind. Sind die charakteristischen Eigenfrequenzen eines Systems hingegen a-priori (grob) bekannt, bietet sich ggf. eine simulative Voruntersuchung an, um einen zielführenden, applikationsspezifischen Kompromiss bezüglich der Güte des Diskretisierungsverfahrens und der Schrittweite zu finden.

Soll das zu identifizierende Modell nachfolgend für eine digitale Regelung (ggf. inklusive Beobachter) genutzt werden, bietet es sich an, das Abtastintervall für die Identifikation konsistent zum geplanten Regelungstakt zu wählen. Dies hat den Vorteil das etwaig verbleibende numerische Fehler aufgrund der Zeitdiskretisierung implizit durch das identifizierte Modell abgebildet werden. Ist T_a während der Identifikation und dem späteren Einsatz in der Regelung hingegen unterschiedlich, können numerische Fehler die Regelung ggf. negativ beeinflussen.

6.4.2 Bewertung und Wahl der Systemanregung

Die Systemanregung $\boldsymbol{u}$ hat maßgeblichen Einfluss auf das Identifikationsergebnis. Im Idealfall kann $\boldsymbol{u}$ innerhalb der Systemgrenzen[1] frei gewählt werden. Hier stellt sich die Frage, wie die Systemanregung zu gestalten ist, um eine bestmögliche Identifikation zu erhalten. Für einige Anwendungen ist $\boldsymbol{u}$ allerdings nicht oder nur geringfügig veränderbar, beispielsweise innerhalb laufender Produktionsvorgänge in der chemischen Prozessindustrie oder im Kontext makroökonomischer Modelle. In diesen Fällen gilt es zumindest vorhandene Datenreihen zu bewerten und diejenigen für die Identifikation auszuwählen, welche vergleichsweise zielführend erscheinen.

Informative Experimente

Ein *informatives Experiment* führt zu einem Datenset, auf dessen Basis verschiedene Modelle bezüglich eines zu identifizierenden Systems eindeutig unterschieden werden können (vgl. [Lju99]). Dies bedeutet auch, dass die Parameterschätzung konsistent ist, falls Modell und System strukturell identisch sind (vgl. Satz 3.7). Hierzu sei folgende Definition herangezogen, wobei zunächst Systeme mit lediglich einem Eingang $\boldsymbol{u} = u$ betrachtet werden:

[1] Typische Begrenzungen bestehen hinsichtlich der maximalen Amplitude sowie Änderungsrate von $\boldsymbol{u}$ sowie dem resultierenden Verlauf von $\boldsymbol{x}$ bzw. $\boldsymbol{y}$ (insbesondere hinsichtlich ihrer Amplitude, beispielsweise im Kontext der Überlastvermeidung und daraus resultierender Sicherheitsabschaltungen).

Definition 6.7: Beständige Anregung (persistence of excitation)

Sei $u[k]$ eine quasi-stationäre Signalfolge[1]. Diese wird als beständige Anregung (persistently exciting) der Ordnung $m \in \mathbb{N}$ (in N-Abtastschritten) bezeichnet, falls die Folge

$$R_{uu}(\tau) = \frac{1}{N}\sum_{k=1}^{N}(u[k]u[k+\tau]), \quad \tau = \{0, 1, \ldots, m-1\} \tag{6.72}$$

existiert und die Matrix

$$\overline{R}_m = \begin{bmatrix} R_{uu}(0) & R_{uu}(1) & \cdots & R_{uu}(m-1) \\ R_{uu}(1) & R_{uu}(0) & \cdots & R_{uu}(m-2) \\ \vdots & \vdots & \ddots & \vdots \\ R_{uu}(m-1) & R_{uu}(m-2) & \cdots & R_{uu}(0) \end{bmatrix} \tag{6.73}$$

positiv definit ist.

Anschaulich interpretiert bedeutet obige Definition, dass ein beständig angeregtes Signal der Ordnung m nicht durch ein digitales, gleitendes Mittelwert-Filter der Ordnung $(m-1)$ zu Null gefiltert werden kann. Im Frequenzbereich bedeutet dies, dass das diskrete Spektrum von u an mindestens m diskreten Frequenzen größer Null ist. Für eine möglichst gute, beständige Anregung (hoher Ordnung) folgt somit der Wunsch, dass u eine große Bandbreite aufweisen solle. Für den Fall eines Eingangsvektors $\boldsymbol{u}$ wird obige Definition erweitert zu (vgl. [BS85]):

Definition 6.8: Beständige Anregung für vektorielle Signale

Sei $\boldsymbol{u}[k]$ eine vektorielle, quasi-stationäre Signalfolge. Diese wird als beständige Anregung der Ordnung m (in N-Abtastschritten) bezeichnet, falls ein $\alpha \in \{\mathbb{R}|\alpha > 0\}$ existiert, sodass

$$\sum_{k=1}^{N} \begin{bmatrix} \boldsymbol{u}^T[k+1] \\ \boldsymbol{u}^T[k+2] \\ \vdots \\ \boldsymbol{u}^T[k+m] \end{bmatrix} \begin{bmatrix} \boldsymbol{u}[k+1] & \boldsymbol{u}[k+2] & \cdots & \boldsymbol{u}[k+m] \end{bmatrix} \geq \alpha \boldsymbol{I} \tag{6.74}$$

mit $\boldsymbol{I}$ als Einheitsmatrix gilt.

Darauf aufbauend kann folgender Satz bewiesen werden (vgl. [Lju99]):

[2]Ein (quasi-)stationäres Signal entspricht der Realisierung eines (quasi-)stationären stochastischen Prozess. Der stochastische Prozess heißt stationär, falls dessen zugrundeliegende Zufallsverteilung zeitinvariant ist.

Satz 6.3: Informative Experimente für LTI-Modelle

Gegeben sei ein dynamisches, zeitdiskretes, lineares und zeitinvariantes Modell. Das Modell habe $p \in \mathbb{N}$ unbekannte Parameter. Das zugrundeliegende Anregungssignal $\boldsymbol{u}$ führt genau dann zu einem informativen Experiment, falls dieses eine beständige Anregung der Ordnung m mit

$$m \geq p \tag{6.75}$$

aufweist.

Mit obigen Satz steht somit ein hinreichendes Kriterium zur Verfügung, um LTI-Systeme derart anzuregen, dass eine konsistente Parameterschätzung möglich ist. Aufgrund der angenommenen Linearität ist für die Amplitude der Anregung lediglich sicherzustellen, dass ein möglichst hoher Signal-Nutzanteil hinsichtlich der relevanten Messgrößen resultiert. Das Rauschen bestimmt somit allein, welche Anregungsamplituden sinnvollerweise gewählt werden sollten.

Für nichtlineare Systeme ist hingegen die ausschließliche Betrachtung des Spektrums von $\boldsymbol{u}$ nicht ausreichend. Dies kann man sich relativ leicht an einem System mit Sättigungscharakteristik, beispielsweise einer elektrischen Spule mit sättigendem Eisenkern, verdeutlichen: Ist die Anregungsamplitude (in diesem Fall die Spannung über der Spule) zu gering, wird die Sättigungskennlinie (also der resultierende Strom bzw. der magnetische Fluss) nicht vollständig erfasst. Ein Modell zur Abbildung der Sättigungskennlinie wird daher mit großer Wahrscheinlichkeit einen systematischen Fehler aufweisen, selbst wenn eine beständige Anregung erfolgte.

Für allgemeine, nichtlineare Systeme ist es daher nicht ohne weiteres möglich ein universelles Gütekriterium zu formulieren, anhand dessen eine Anregung abgeleitet werden kann, welche zu einer konsistenten Schätzung führt. Als grobe Richtlinie kann allerdings folgende Empfehlung herangezogen werden: um ein nichtlineares System vollständig anzuregen (und damit zu identifizieren zu können), muss der Systemeingang $\boldsymbol{u}$ sowohl hinsichtlich der Amplitude als auch bezüglich der Frequenz möglichst reichhaltig sein [Now02]. Darüber hinaus ist die Gewinnung zielführender Anregungsprofile für (nicht-)lineare Systeme eine anspruchsvolle Aufgabe, welche häufig situationsspezifische Optimierungsprobleme nach sich ziehen (siehe z. B. [Meh74][SGT+97]).

Typische Anregungssignale

Im nachfolgenden werden einige, typische Anregungssignale für einen skalaren Eingang vorgestellt.

Multi-Sinus

Wie der Name bereits vermuten lässt, werden bei der Multi-Sinus-Anregung eine diskrete Anzahl von Sinus-Signalen unterschiedlicher Frequenz und ggf. unterschiedlicher Amplitude überlagert:

$$u(t) = \sum_{j=1}^{m} \hat{u}_j \sin\left(2\pi f_j t + \varphi_j\right) \ . \tag{6.76}$$

Die Wahl des Phasenversatzes φ_j stellt einen weiteren Freiheitsgrad dar. Diese Art der Anregung ist vergleichsweise einfach zu generieren und regt das zu identifizierende System an den gewählten Frequenzen signifikant an. Es ist daher davon auszugehen, dass das resultierende Modell an den gewählten Frequenzen eine gute Performanz liefern wird. Das Anregungsspektrum

ist allerdings nicht kontinuierlich, sodass eine verlässliche Identifikation in den Zwischenbereichen nicht garantiert ist. Siehe hierzu das Beispiel in Abb. 6.13.

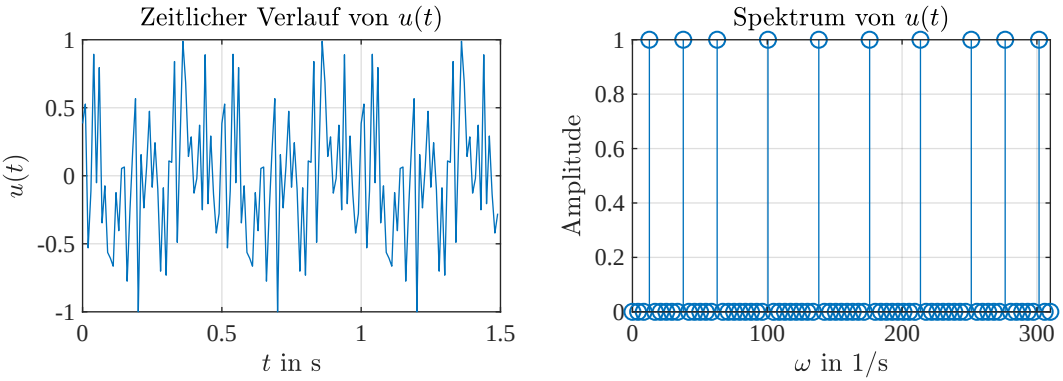


Abb. 6.13: Beispielhafte Anregung mit einem Multi-Sinus gleicher Amplitude im Zeit- und Frequenzbereich

Pseudorandom Binary Sequence (PRBS)
PRBS-Anregungen sind binäre Signale (häufig alternierend zwischen ±1 oder 0/1), welche das Spektrum von weißem Rauschen approximieren und somit eine breitbandige Anregung ermöglichen. Obwohl PRBS-Signalabfolgen eine zufällig anmutende Erscheinung aufweisen, können diese durch einen deterministischen Zufallsgenerator erzeugt werden. Somit ist die PRBS-Erstellung zudem reproduzierbar.

Die Realisierung des PRBS-Signals erfolgt typischerweise über ein linear rückgekoppeltes Schieberegister (*linear-feedback shift register* – LFSR). Die am häufigsten verwendete lineare Funktion zur Rückkopplung ist exklusiv-oder (XOR). Somit ist ein LFSR meist ein Schieberegister, dessen Eingangsbit durch das XOR einiger Bits des gesamten Schieberegisterwertes gesteuert wird. Der Anfangswert des LFSR wird als *Seed* bezeichnet. Da die Vorgänge im Register deterministisch sind, wird der vom Register erzeugte Bit-Strom vollständig durch seinen aktuellen (oder vorherigen) Zustand bestimmt. Eine beispielhaftes LFSR ist in Abb. 6.14 dargestellt.

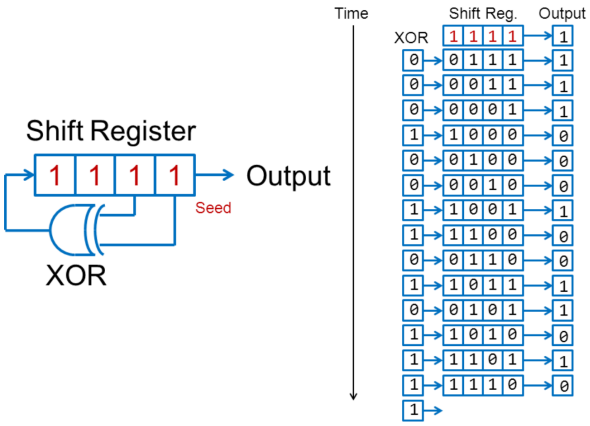


Abb. 6.14: Abfolge eines 4-Bit linear rückgekoppelten Schieberegisters (Quelle: [Oka13])

Die im LFSR rückgeführten Bit-Positionen (so-genannte *taps*) bestimmen den Wert des Bits am Eingang des LFSR. In der Literatur findet sich häufig eine Polynom-Darstellung der Form

$$p(\boldsymbol{u}) = a_n u^n + a_{n-1} u^{n-1} + \cdots + a_1 u^1 + a_0 u^0 = a_n u^n + a_{n-1} u^{n-1} + \cdots + a_1 u^1 + 1, \quad (6.77)$$

um die LFSR-Vorschrift abzubilden. Hier repräsentiert u^n das Bit an der n-ten Position und $a_n \in \{a_n \in \mathbb{N}, 0 \leq a_n \leq 1\}$ ist der entsprechende Koeffizient, welcher angibt, ob das jeweilige Bit zum Eingang des LFSR rückgeführt wird oder nicht. Das zu setzende Eingangs-Bit ergibt sich dann aus der Betrachtung

$$\text{Eingang} = p(\boldsymbol{u}) \bmod 2. \tag{6.78}$$

Alternativ zur polynomialen Darstellung kann eine Reihe genutzt werden, um das LFSR abzubilden. Es gilt:

$$u[1] = (a_n u[n] + a_{n-1} u[n-1] + \cdots + a_1 u[1] + 1) \bmod 2. \tag{6.79}$$

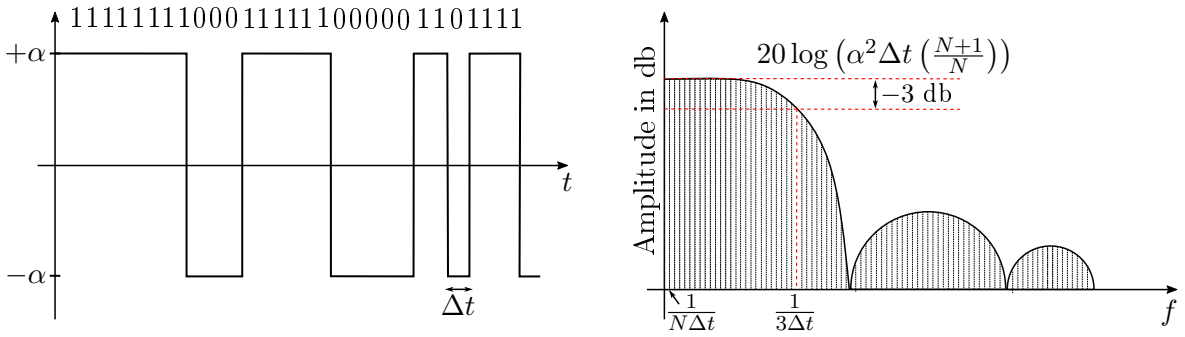
Liegt ein n-Bit LFSR vor, kann damit eine PRBS-n Sequenz gebildet werden, dessen Länge maximal

$$N = 2^n - 1 \tag{6.80}$$

Bits enthält, bevor diese sich wiederholen. Dies wird auch Folge maximaler Länge (*maximum length sequence* - MLS) genannt. Nur spezielle LFSR-Konfigurationen erzeugen ein MLS-PRBS, nämlich solche, dessen Rückführung entsprechend (6.77) ein *primitives Polynom* darstellen. Die MLS-PRBS-Konfiguration für LFSR geringer Ordnung sind in Tab. 6.2 zusammengefasst.

Ordnung	$N = 2^n - 1$	Indizes $a_n \neq 0$
2	3	1,2
3	7	2,3
4	15	1,4
5	31	2,5
6	63	1,6
7	127	3,7
9	255	1,2,6,8
9	511	4,9
10	1023	7,10
...	...	...

Tab. 6.2: LFSR-Konfigurationen mit MLS-Eigenschaft



(a) Zeitbereich (b) Frequenzbereich

Abb. 6.15: Veranschaulichung zu den Eigenschaften einer PRBS-Anregung im Frequenzbereich

Die Eigenschaften der PRBS-Anregung im Frequenzbereich sind in Abb. 6.15 qualitativ veranschaulicht. Unter der Annahme, dass das PRBS-Signal symmetrisch um den Nullpunkt eine

alternierende Amplitude α aufweise und dessen Taktrate $f_s = 1/\Delta t$ entspricht ergibt sich das in Abb. 6.15b dargestellte Spektrum. Die minimale und maximale Grenzfrequenz (-3 db) beträgt:

$$f_{min} = \frac{1}{N\Delta t}, \quad f_{max} = \frac{1}{3\Delta t}. \tag{6.81}$$

Für besonders große Werte für α und N sowie kleine Werte für Δt resultiert eine breitbandige Anregung mit hoher Amplitude. Soll hingegen eine Anregung mit verringerter Bandbreite erzeugt werden, um beispielsweise ein PRBS-Signal mit besonderer Anregung bei kleiner Frequenz zu erhalten, kann eine vorhandene PRBS-Signalfolge P-fach überabgetastet werden. Dies führt zu einer Verlängerung der konstanten Anteile im Signal und hat den selben Effekt im Frequenzbereich, wie die Anwendung eines gleitenden Mittelwertfilters der Ordnung P:

$$\tilde{u}[k] = \frac{1}{P}\left(u[k] + u[n+1] + \cdots + u[k+P]\right). \tag{6.82}$$

Ferner sei ein Hinweis auf die Verwendung von PRBS-Folgen für mehrdimensionale Probleme gegeben: Durch die Periodizität des Signals sind unterschiedliche initialisierte PRBS-Folgen korreliert. Folglich eigenen sich PRBS-Folgen für Systeme mit mehreren Eingängen nur bedingt.

Gaußverteilte Zufallsfolge

Bei der gaußverteilten Zufallsfolge wird der Systemeingang aus einer Normalverteilung

$$u[k] \sim \mathcal{N}(\mu, \sigma^2) \tag{6.83}$$

mit einstellbarem Mittelwert μ sowie Varianz σ^2 fortlaufend gezogen, welches i. d. R. über einen in Software definierten Zufallszahlengenerator approximiert wird. Je nach spezifischer Implementierung und Initialisierung des Zufallszahlengenerators resultieren variierende Anregungssequenzen mit entsprechenden Frequenzspektren. Ein exemplarisches Beispiel hierzu ist in Abb. 6.16 dargestellt.

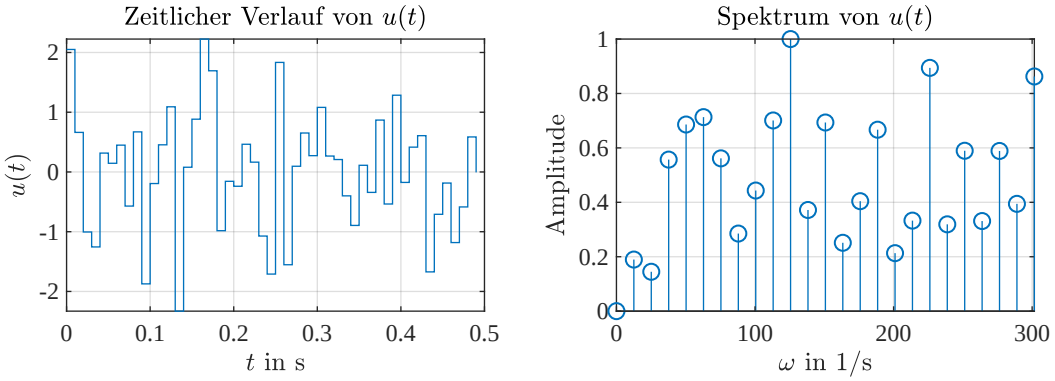


Abb. 6.16: Beispielhafte Anregung aus einer gaußverteilten Zufallsfolge $\mathcal{N}(\mu = 0, \sigma^2 = 1)$ mit Abtastintervall $T_a = 10$ ms

Ornstein-Uhlenbeck-Prozess

Der Ornstein-Uhlenbeck-Prozess (OU-Prozess) ist ein elementarer stochastischer Prozess, wel-

cher über folgende Differentialgleichung (skalare Darstellung) beschrieben wird:

$$\frac{\mathrm{d}u}{\mathrm{d}t}(t) = \theta(\mu - u(t)) + \sigma\eta(t) \tag{6.84}$$

Hierbei ist $\eta(t)$ ein weißes Rauschen, welches den Prozess selbst anregt. Die weiteren Parameter des OU-Prozesses sind:

- μ: Beschreibt das Gleichgewicht des OU-Prozesses, also den zeitlichen Mittelwert zu dem der Prozess mittels des sog. Driftterm $\theta(\mu - u(t))$ gezogen wird.
- θ: Beschreibt die Anziehungskraft von μ, sprich wie weit sich der Prozess $u(t)$ von μ entfernen kann. Für besonders kleine Werte von θ verhält sich der OU-Prozess näherungsweise wie weißes Rauschen ohne Eigendynamik, für große Werte von θ spricht man von einem steifen Prozess, da dieser sich kaum von μ entfernen kann.
- σ: Gibt die Stärke des Zufalleinflusses über η an.

Durch entsprechende Parametrierung von $\{\mu, \theta, \sigma\}$ kann unterschiedliches OU-Prozessverhalten generiert werden. Die Lösung der obigen Differentialgleichung erfolgt i.d.R. mittels numerischer Integration (z. B. Euler-Vorwärts).

Gegenüber der reinen gaußverteilten Zufallsfolge hat der OU-Prozess eine Eigendynamik, sodass auch über mehrere Abtastpunkte das Eingangssignal zur Systemanregung maßgeblich verschieden von Null sein kann. Zur Verdeutlichung ist in Abb. 6.17 eine gaußverteilte Anregung (links) gegen ein OU-Anregungsprozess (rechts) dargestellt. Die Systemantwort entspricht einem Tiefpass erster Ordnung

$$\tau\frac{\mathrm{d}x}{\mathrm{d}t} = x(t) + u(t)$$

mit Zeitkonstante τ. Charakteristisch ist zu beobachten, dass durch die OU-Anregung eine signifikant größere Zustandsabdeckung erzielt wird, was für nichtlineare bzw. parametervariante Systeme interessant sein kann.

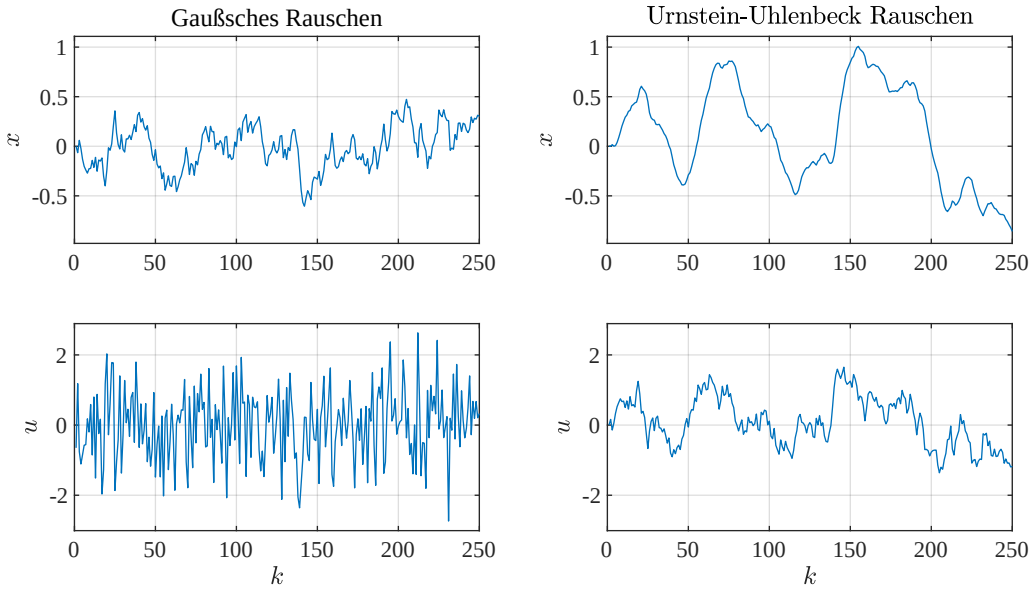


Abb. 6.17: Anregung auf einen Tiefpass mit $\tau = 1$ ms und den OU-Parametern $\mu = 0$, $\theta = 0,05$ und $\sigma = 0,25$ (Diskretisierung mit Euler-Vorwärts, Abtastzeit $T_s = 0,1$ ms)

Toolboxen zur Erstellung von Anregungsprofilen
Neben dem eigenhändigen gestalten entsprechender Anregungsprofile sei darauf hingewiesen, dass es in verfügbaren Softwarelösungen ebenfalls Werkzeuge zum automatisierten Erstellen dieser gibt. Dies sind beispielsweise:

- idinput: Anregungsgenerierung der Matlab System Identification Toolbox
  `https://de.mathworks.com/help/ident/ref/idinput.html`
- SIPPY (Verschiedene Identifikationsalgorithmen inkl. Anregungsgenerierung für Python)
  `https://github.com/CPCLAB-UNIPI/SIPPY`

Zusammenfassende Bewertung

Die bisherige Diskussion zur Anregung sei auf die wesentlichen Punkte zusammengefasst:

- *Keine Universallösung*: Während für LTI-Systeme zwar eindeutige Bedingungen an die notwendige Systemanregung gestellt werden können (s. Satz 6.3), ist dies für nichtlineare Systeme nicht möglich. Nur die wenigsten realen technischen Systeme zeigen allerdings im gesamten Betriebsbereich ein LTI-Verhalten, sodass ein identifiziertes LTI-Modell häufig nur lokal gültig ist. Zudem stellen Systemgrenzen bezüglich $\{\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{y}\}$ Einschränkungen an die Anregung dar.
- *Vorwissen nutzen*: Liegt Expertenwissen über das System vor, ist ggf. analytisch eine erste Systembeschreibung möglich. Diese kann genutzt werden, um eine Analyse zielführender Anregungen durchzuführen, z. B. zur Bestimmung charakteristischer Eigenfrequenzen, welche dann im Experiment besonders angeregt werden.
- *Betriebsbereich abdecken*: Typischerweise sollte für ein System (näherungsweise) bekannt sein, in welchem Bereich bezüglich $\{\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{y}\}$ dieses betrieben wird. Unter Umständen existieren hierzu klare Definitionen hinsichtlich zu erwartender stationärer Arbeitspunkte und/oder transienter Systemtrajektorien. Diese Vorgaben sollte gemäß Abb. 1.4 genutzt werden, um eine Anregung zu erstellen bzw. Datenbasis zu extrahieren, welche im besonderen Maße den relevanten Betriebsbereich repräsentiert.
- *Datenschieflagen vermeiden*: Es ist sicherzustellen, dass in den Trainings- und Testprofilen keine Teilbereiche des kombinierten Zustands- und Eingangsraum über- oder unter repräsentiert sind (*bias*). Als allgemeine Richtlinie ist eine multivariate Gleichverteilung hinsichtlich der relevanten Zustands- und Eingangsgrößen anzustreben.

Literaturverzeichnis

[Agg18] Aggarwal, Charu C.: *Neural Networks and Deep Learning: a Textbook*. Springer, 2018

[BH18] Bücker, Martin ; Hovland, Paul: *autodiff*. `http://www.autodiff.org/`. Version: 12.11.2018

[BLR04] Biethahn, Jörg ; Lackner, Andreas ; Range, Michael: *Optimierung und Simulation*. Oldenbourg, München, 2004

[Bro18] Brooks, M.: *The Matrix Reference Manual*. `http://www.ee.ic.ac.uk/hp/staff/dmb/matrix/intro.html`. Version: 17.12.2018

[BS85] Bai, E.W. ; Sastry, S.S.: Persistency of Excitation, Sufficient Richness and Parameter Convergence in Discrete Time Adaptive Control. In: *Systems & Control Letters* (1985), S. 153–163

[BSMM05] Bronstein, I.N. ; Semendjajew, K.A. ; Musiol, G. ; Mühlig, H.: *Taschenbuch der Mathematik*. Verlag Harri Deutsch, 2005

[Cle06] Clerc, Maurice: *Particle Swarm Optimization*. ISTE, 2006

[Föl13] Föllinger, Otto: *Regelungstechnik: Einführung in die Methoden und ihre Anwendung*. VDE Verlag, 2013

[Gon17] Gonzalez, Javier: *Introduction to Bayesian Optimization*. 2017

[Gra13] Graichen, Knut: *Systemtheorie (Vorlesungsskript)*. Universität Ulm, 2013

[GW08] Griewank, Andreas ; Walther, Andrea: *Evaluating Derivatives : Principles and Techniques of Algorithmic Differentiation*. SIAM, 2008

[Hen13] Henze, Norbert: *Stochastik für Einsteiger*. Springer Fachmedien Wiesbaden, 2013

[Hol18] Holzapfel, Florian: *Skriptum: System Identification*. Technische Universität München, 2018

[HP95] Horst, Reiner (Hrsg.) ; Pardalos, Panos M. (Hrsg.): *Handbook of Global Optimization*. Springer US, Boston, 1995

[HTF17] Hastie, T. ; Tibshirani, R. ; Friedman, J.: *The Elements of Statistical Learning : Data Mining, Inference, and Prediction*. Springer, 2017

[IM11] Isermann, Rolf ; Münchhof, Marco: *Identifcation of Dynamic Systems*. Springer-Verlag Berlin Heidelberg, 2011

[JGD06] Janschek, K. ; Giebler, E. ; Dyblenko, S.: *Simulationstechnik (Vorlesungsskript)*. Technische Universität Dresden, 2006

[JU04] Julier, S.J. ; Uhlmann, J.K.: Unscented Filtering and Nonlinear Estimation. In: *Proceedings of the IEEE* 92 (2004), Nr. 3, S. 401–422

[KE95] Kennedy, J. ; Eberhart, R.: Particle Swarm Optimization. In: *IEEE International Conference on Neural Networks*, 1995

[Kug17] Kugi, Andreas: *Automatisierung (Vorlesungsskript)*. Technische Universität Wien, 2017

[Len17] Lenz, Eric: *Skriptum: Identfikation dynamischer Systeme*. Technische Universität Darmstadt, 2017

[Lju99] Ljung, Lennart: *System Identification: Theory for the User*. Prentice Hall Inc., 1999

[Lun13] Lunze, Jan: *Regelungstechnik 1: Systemtheoretische Grundlagen, Analyse und Entwurf einschleifiger Regelungen*. Springer Verlag, 2013

[Lun16] Lunze, Jan: *Regelungstechnik 2: Mehrgrößensysteme, Digitale Regelung*. Springer-Verlag Berlin Heidelberg, 2016

[LW14] Lutz, Holger ; Wendt, Wolfgang: *Taschenbuch der Regelungstechnik: mit MATLAB und Simulink*. Verlag Europa Lehrmittel, 2014

[Meh74] Mehra, Raman K.: Optimal Input Signals for Parameter Estimation in Dynamic Systems: Survey and New Results. In: *IEEE Transactions on Automatic Control* 19 (1974), Nr. 6, S. 753–768

[Now02] Nowak, Robert D.: Nonlinear System Identification. In: *Circuits, Systems and Signal Processing* 21 (2002), Nr. 1, S. 109–122

[NW06] Nocedal, J. ; Wright, S.J.: *Numerical Optimization*. Springer-Verlag New York, 2006

[Oka13] Okawara, Hideo: *Mixed Signal Lecture Series: PRBS (Pseudo Random Binary Sequence)*, 2013

[PLB12] Papageorgiou, Markos ; Leibold, Marion ; Buss, Martin: *Optimierung*. Springer Berlin Heidelberg, 2012

[PR02] Pardalos, P. M. (Hrsg.) ; Romeijn, H. E. (Hrsg.): *Handbook of Global Optimization - Volume 2*. Springer US, Boston, 2002

[Prü18] Prüfert, Uwe: *Skriptum: Einführung in das Algorithmische Differenzieren*. Technische Universität Bergakademie Freiberg, 2018

[Ras18] Raschka, Sebastian: Model Evaluation, Model Selection, and Algorithm Selection in Machine Learning. In: *CoRR* abs/1811.12808 (2018)

[Sch17] Schweers, Christoph: *Adaptive Sigma-Punkte-Filter-Auslegung zur Zustands- und Parameterschätzung an Black-Box-Modellen*, Diss., 2017

[SEBB18] Stork, Jörg ; Eiben, A. E. ; Bartz-Beielstein, Thomas: A new Taxonomy of Continuous Global Optimization Algorithms. In: *ArXiv* abs/1808.08818 (2018)

[SGT+97] Swevers, J. ; Ganseman, C. ; Tükel, D. ; De Schutter, J. ; Van Brussel, H.: Optimal Robot Excitation and Identification. In: *IEEE Transactions on Robotics and Automation* 13 (1997), Nr. 5, S. 730–741

[Sim06] Simon, Dan: *Optimal State Estimation*. John Wiley & Sons, 2006

[SK06] Schwarz, Hans R. ; Köckler, Norbert: *Numerische Mathematik*. Teubner Verlag, 2006

[SL03] Seber, George A. ; Lee, Alan J.: *Linear Regression Analysis*. Wiley-Interscience, 2003

[Ste18] Steinböck, Andreas: *Skritpum: Optimierung*. Technische Universität Wien, 2018

[SWP12] Strehmel, Karl ; Weiner, Rüdiger ; Podhaisky, Helmut: *Numerik gewöhnlicher Differentialgleichungen*. Vieweg+Teubner Verlag, 2012

[Tal09] Talbi, El-Ghazali: *Metaheuristics : From Design to Implementation*. Wiley, Hoboken, 2009

[Tar10] Taragna, Michele: *Skritpum: System Identification, Estimation and Filtering*. Universität Turin, 2010

[Uni18] University of Wisconsin-Madison: *NEOS (Network-Enabled Optimization System)*. `https://neos-server.org/neos/`. Version: 10.12.2018

[Wik18a] Wikipedia: *Systemidentifikation*. `https://de.wikipedia.org/wiki/Systemidentifikation`. Version: 01.10.2018

[Wik18b] WIKIPEDIA: *List of Optimization Software*. https://en.wikipedia.org/wiki/List_of_optimization_software. Version: 10.12.2018

[Wik18c] WIKIPEDIA: *Kalman Filter*. https://en.wikipedia.org/wiki/Kalman_filter. Version: 14.12.2018

[Wik18d] WIKIPEDIA: *Wahrscheinlichkeitsraum*. https://de.wikipedia.org/wiki/Wahrscheinlichkeitsraum. Version: 15.08.2018

A Zusammenstellung von Rechenregeln

A.1 Eigenschaften transponierter Matrizen

Gegeben sei eine Matrix $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ mit

$$A = (a_{ij})_{ij} = \begin{bmatrix} a_{11} & \dots & a_{1n} \\ \vdots & & \vdots \\ a_{m1} & \dots & a_{mn} \end{bmatrix} \in \mathbb{R}^{m \times n} \tag{A.1}$$

dann ist die transponierte Matrix definiert als

$$A^T = (a_{ij})_{ji} = \begin{bmatrix} a_{11} & \dots & a_{m1} \\ \vdots & & \vdots \\ a_{1n} & \dots & a_{mn} \end{bmatrix} \in \mathbb{R}^{n \times m}. \tag{A.2}$$

Nachfolgend werden einige wichtige Eigenschaften zum Rechnen mit transponierten Matrizen wiedergegeben[1]:

Summe:
Allgemein ergibt sich die Summe von k Matrizen gleicher Größe zu

$$(A_1 + A_2 + \dots + A_k)^T = A_1^T + A_2^T + \dots + A_k^T. \tag{A.3}$$

Die Transponierte einer Summe von Matrizen ist demnach gleich der Summe der Transponierten.

Mehrfache Transposition:
Für die Transponierte der Transponierten einer Matrix gilt

$$\left(A^T\right)^T = A. \tag{A.4}$$

Durch zweifache Transposition ergibt sich demnach stets wieder die Ausgangsmatrix.

Produkt:
Für die Transponierte des Produkts einer Matrix $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ mit einer Matrix $\boldsymbol{B} \in \mathbb{R}^{m \times n}$ gilt

$$(A \cdot B)^T = \left(\sum_{j=1}^{n} a_{ij} \cdot b_{jk} \right)_{ki} = \left(\sum_{j=1}^{n} b_{jk} \cdot a_{ij} \right)_{ki} = B^T \cdot A^T \tag{A.5}$$

[1] Der Einfachheit halber wird eine reellwertige Matrix angenommen. Die Rechenregeln für transponierte Matrizen gelten allerdings auch für Matrizen mit Einträgen aus den komplexen Zahlen.

bzw. allgemein für das Produkt von k Matrizen gleicher Größe (also hier: $m = n$)

$$(A_1 \cdot A_2 \cdots A_k)^T = A_k^T \cdots A_2^T \cdot A_1^T. \tag{A.6}$$

Die Transponierte eines Produkts von Matrizen ist demnach gleich dem Produkt der Transponierten, jedoch in umgekehrter Reihenfolge.

Inverse:
Sei die Matrix A regulär und somit invertierbar. Dann gilt:

$$\left(A^{-1}\right)^T = \left(A^T\right)^{-1}. \tag{A.7}$$

Die Transponierte der inversen Matrix ist demnach gleich der Inversen der transponierten Matrix.

A.2 Ableitung einer Funktion nach einem Vektor von Variablen

Gegeben sei eine Funktion

$$f(\boldsymbol{x}) : \mathbb{R}^m \to \mathbb{R} \quad \text{mit} \quad \boldsymbol{x} = \begin{bmatrix} x_1 & x_2 & \dots & x_m \end{bmatrix}^T. \tag{A.8}$$

Die Ableitungen der Funktion f nach dem Vektor $\boldsymbol{x}$

$$\nabla f(\boldsymbol{x}) = \begin{bmatrix} \frac{\partial f}{\partial x_1} & \frac{\partial f}{\partial x_2} & \dots & \frac{\partial f}{\partial x_m} \end{bmatrix}^T = \frac{\partial f}{\partial \boldsymbol{x}}, \tag{A.9}$$

$$(\nabla f(\boldsymbol{x}))^T = \begin{bmatrix} \frac{\partial f}{\partial x_1} & \frac{\partial f}{\partial x_2} & \dots & \frac{\partial f}{\partial x_m} \end{bmatrix} = \frac{\partial f}{\partial \boldsymbol{x}^T} \tag{A.10}$$

führt somit zu einem Vektor dessen Komponenten die partiellen Ableitungen umfasst. In Kombination mit einer weiteren Funktion $g(\boldsymbol{x}) : \mathbb{R}^n \to \mathbb{R}$ gilt die Summen- und Produktregel:

$$\frac{\partial \left(f(\boldsymbol{x}) \pm g(\boldsymbol{x})\right)}{\partial \boldsymbol{x}} = \frac{\partial f(\boldsymbol{x})}{\partial \boldsymbol{x}} \pm \frac{\partial g(\boldsymbol{x})}{\partial \boldsymbol{x}}, \tag{A.11}$$

$$\frac{\partial \left(f(\boldsymbol{x}) \cdot g(\boldsymbol{x})\right)}{\partial \boldsymbol{x}} = \frac{\partial f(\boldsymbol{x})}{\partial \boldsymbol{x}} g(\boldsymbol{x}) + \frac{\partial g(\boldsymbol{x})}{\partial \boldsymbol{x}} f(\boldsymbol{x}). \tag{A.12}$$

Sei ferner ein Vektor $\boldsymbol{b} \in \mathbb{R}^m$ mit

$$\boldsymbol{b} = \begin{bmatrix} b_1 & b_2 & \dots & b_m \end{bmatrix}^T \tag{A.13}$$

gegeben. Dann folgt für die Ableitung der skalaren Funktion $\boldsymbol{x}^T\boldsymbol{b}$ nach $\boldsymbol{x}$

$$\frac{\partial(\boldsymbol{x}^T\boldsymbol{b})}{\partial \boldsymbol{x}} = \begin{bmatrix} \frac{\partial \sum_{i=1}^m x_i b_i}{\partial x_1} & \frac{\partial \sum_{i=1}^m x_i b_i}{\partial x_2} & \dots & \frac{\partial \sum_{i=1}^m x_i b_i}{\partial x_m} \end{bmatrix} = \begin{bmatrix} b_1 & b_2 & \dots & b_m \end{bmatrix} = \boldsymbol{b}^T \tag{A.14}$$

bzw. analog

$$\frac{\partial(\boldsymbol{b}^T\boldsymbol{x})}{\partial \boldsymbol{x}} = \begin{bmatrix} \frac{\partial \sum_{i=1}^m x_i b_i}{\partial x_1} & \frac{\partial \sum_{i=1}^m x_i b_i}{\partial x_2} & \dots & \frac{\partial \sum_{i=1}^m x_i b_i}{\partial x_m} \end{bmatrix} = \begin{bmatrix} b_1 & b_2 & \dots & b_m \end{bmatrix} = \boldsymbol{b}^T \tag{A.15}$$

für $\boldsymbol{b}^T\boldsymbol{x}$. Sei weiterhin eine Matrix $\boldsymbol{A} \in \mathbb{R}^{m\times m}$ gegeben und es soll die Ableitung der skalaren Funktion

$$\boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x} = \sum_{j=1}^{m}\sum_{i=1}^{m} x_j a_{ji} x_i \tag{A.16}$$

berechnet werden. Zwar ist es auch dann möglich, analog zum vorherigen Vorgehen, die einzelnen partiellen Ableitungen zu bilden – allerdings fällt es dann schwer Gesetzmäßigkeiten in der Zerlegung aufzufinden. Stattdessen wird ein alternativer Zugang über die Taylorreihe gewählt. Hierzu wird $\boldsymbol{x}$ aufgespalten in

$$\boldsymbol{x} = \boldsymbol{x}_0 + \boldsymbol{x}_\delta. \tag{A.17}$$

Einsetzen in (A.16) und ausmultiplizieren führt dann zu:

$$\begin{aligned}(\boldsymbol{x}_0^T + \boldsymbol{x}_\delta^T)\boldsymbol{A}(\boldsymbol{x}_0 + \boldsymbol{x}_\delta) &= \boldsymbol{x}_0^T\boldsymbol{A}\boldsymbol{x}_0 + \boldsymbol{x}_0^T\boldsymbol{A}\boldsymbol{x}_\delta + \boldsymbol{x}_\delta^T\boldsymbol{A}\boldsymbol{x}_0 + \boldsymbol{x}_\delta^T\boldsymbol{A}\boldsymbol{x}_\delta \\ &= \boldsymbol{x}_0^T\boldsymbol{A}\boldsymbol{x}_0 + (\boldsymbol{x}_0^T\boldsymbol{A} + \boldsymbol{x}_0^T\boldsymbol{A}^T)\boldsymbol{x}_\delta + \boldsymbol{x}_\delta^T\boldsymbol{A}\boldsymbol{x}_\delta\end{aligned} \tag{A.18}$$

Dies kann als Taylorreihenentwicklung der Funktion $\boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x}$ um den Punkt x_0 interpretiert werden. Der erste Summand stellt den Wert am Punkt x_0 dar. Der zweite Summand beschreibt die Änderung des Funktionswert, die linear von der Abweichung zu x_0 abhängt und der dritte Summand beschreibt eine quadratisch abhängige Änderung. Weitere Terme existieren hier nicht, da es sich um eine quadratische Funktion handelt. Die linear eingehende Änderung (2. Term) beschreibt bei gegebenem $\boldsymbol{x}_\delta$ gerade die (erste) Ableitung, sodass

$$\frac{\partial \boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x}}{\partial \boldsymbol{x}} = \boldsymbol{x}^T(\boldsymbol{A} + \boldsymbol{A}^T) \tag{A.19}$$

gilt. Sofern $\boldsymbol{A}$ zudem eine symmetrische Matrix, also $\boldsymbol{A}^T = \boldsymbol{A}$, ist, dann vereinfacht sich der Term zu:

$$\frac{\partial \boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x}}{\partial \boldsymbol{x}} = 2\boldsymbol{x}^T\boldsymbol{A} \quad \text{falls} \quad \boldsymbol{A}^T = \boldsymbol{A}. \tag{A.20}$$

A.3 Ableitung einer Matrix-Spur

Gegeben seien drei beliebige Matrizen $\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{C}$ und $\boldsymbol{X}$. Dann gelten folgende Rechenregeln für die Ableitungen unter Verwendung der Matrix-Spur [Bro18]:

$$\frac{\partial}{\partial \boldsymbol{X}}\text{Spur}\left(\boldsymbol{A}\boldsymbol{X}\boldsymbol{B}\right) = \boldsymbol{A}^T\boldsymbol{B}^T, \tag{A.21}$$

$$\frac{\partial}{\partial \boldsymbol{X}}\text{Spur}\left(\boldsymbol{A}\boldsymbol{X}^T\boldsymbol{B}\right) = \boldsymbol{B}\boldsymbol{A}, \tag{A.22}$$

$$\frac{\partial}{\partial \boldsymbol{X}}\text{Spur}\left(\boldsymbol{A}\boldsymbol{X}\boldsymbol{B}\boldsymbol{X}^T\boldsymbol{C}\right) = \boldsymbol{A}^T\boldsymbol{C}^T\boldsymbol{X}\boldsymbol{B}^T + \boldsymbol{C}\boldsymbol{A}\boldsymbol{X}\boldsymbol{B}, \tag{A.23}$$

$$\frac{\partial}{\partial \boldsymbol{X}}\text{Spur}\left(\boldsymbol{X}\boldsymbol{A}\boldsymbol{X}^T\right) = \boldsymbol{X}\boldsymbol{A}^T + \boldsymbol{X}\boldsymbol{A}, \tag{A.24}$$

$$\frac{\partial}{\partial \boldsymbol{X}}\text{Spur}\left(\boldsymbol{X}^{-1}\boldsymbol{A}\right) = -(\boldsymbol{X}^{-1})^T\boldsymbol{A}(\boldsymbol{X}^{-1})^T. \tag{A.25}$$

B Singulärwertzerlegung und Total Least Squares

Im Folgenden wird zunächst eine anschauliche Einleitung in die Singulärwertzerlegung (*singular value decomposition* - SVD) gegeben, um darauf aufbauend ihre Anwendung zur Lösung TLS-Problems aus Definition 3.5 zu erläutern.

Jeder Vektor $\boldsymbol{x}$ kann durch eine Menge von Basisvektoren $\boldsymbol{v}$ und der Länge des Vektors in die jeweilige Projektionsrichtung s_i abgebildet werden. Dies ist für den zweidimensionalen Raum in Abb. B.1 dargestellt und kann durch folgende Gleichung ausgedrückt werden:

$$\boldsymbol{AV} = \begin{bmatrix} a_x & a_y \\ b_x & b_y \end{bmatrix} \begin{bmatrix} v_{1x} & v_{2x} \\ v_{1y} & v_{2y} \end{bmatrix} = \begin{bmatrix} s_{a1} & s_{a2} \\ s_{b1} & s_{b2} \end{bmatrix} = \boldsymbol{S}\,. \tag{B.1}$$

Hier sind $\boldsymbol{a}$ und $\boldsymbol{b}$ beliebige Vektoren, welche im kartesischen Koordinatensystem in $\boldsymbol{A}$ zusammengefasst sind und mittels der Projektion $\boldsymbol{V}$ entlang $\boldsymbol{v}_1$ und $\boldsymbol{v}_2$ abgebildet werden.

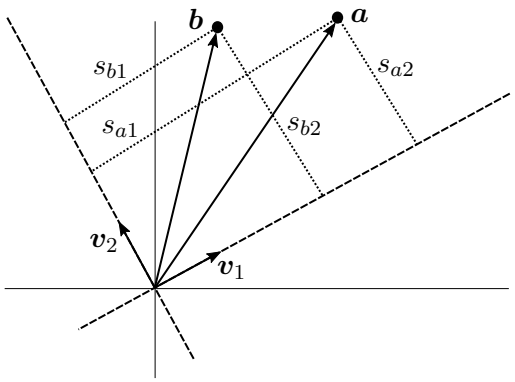


Abb. B.1: Beispielhafte Zerlegung der Vektoren $\boldsymbol{a}$ und $\boldsymbol{b}$ entlang $\boldsymbol{v}_1$ und $\boldsymbol{v}_2$

Dies kann auf eine beliebige Anzahl von n Punkten bzw. m-Projektionsachsen erweitert werden

$$\boldsymbol{AV} = \begin{bmatrix} a_x & a_y & \cdots \\ b_x & b_y & \cdots \\ \vdots & \vdots & \ddots \end{bmatrix} \begin{bmatrix} v_{1x} & v_{2x} & \cdots \\ v_{1y} & v_{2y} & \cdots \\ \vdots & \vdots & \ddots \end{bmatrix} = \begin{bmatrix} s_{a1} & s_{a2} & \cdots \\ s_{b1} & s_{b2} & \cdots \\ \vdots & \vdots & \ddots \end{bmatrix} = \boldsymbol{S} \tag{B.2}$$

mit $\boldsymbol{A} \in \mathbb{R}^{n\times m}$ als Matrix mit gegebenen Zeilenvektoren (Datenpunkte), $\boldsymbol{V} \in \mathbb{R}^{m\times m}$ als Projektion- bzw. Zerlegungsmatrix und $\boldsymbol{S} \in \mathbb{R}^{n\times m}$ als Matrix, dessen Zeilen die Länge der projizierten Vektoren im neuen Koordinatensystem wiedergeben. Umstellen ergibt dann:

$$\boldsymbol{A} = \boldsymbol{S}\boldsymbol{V}^{-1}. \tag{B.3}$$

Wird angenommen, dass die Projektion $\boldsymbol{V}$ eine unitäre Matrix sei und somit eine Orthonomal-

basis besitze, d. h. die Projektionsachsen in $\boldsymbol{V}$ sind orthogonal zueinander, folgt:

$$\boldsymbol{A} = \boldsymbol{S}\boldsymbol{V}^{-1} = \boldsymbol{S}\boldsymbol{V}^{T}. \tag{B.4}$$

Der obige Ausdruck spiegelt die zentrale Aussage der SVD wieder: Jeder Satz von Vektoren kann durch die Länge dessen Projektion in einem orthogonalen Koordinatensystem ausgedrückt werden. Allerdings ist gegenüber (B.4) für die klassische SVD-Definition noch eine Erweiterung zu berücksichtigen, nämlich:

$$\boldsymbol{A} = \boldsymbol{S}\boldsymbol{V}^{T} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^{T}. \tag{B.5}$$

Hier ist $\boldsymbol{U} \in \mathbb{R}^{n\times n}$ eine Matrix, dessen Spaltenvektoren die sog. Links-Singulärvektoren von $\boldsymbol{A}$ enthalten und $\boldsymbol{\Sigma} \in \mathbb{R}^{n\times m}$ ist eine Diagonalmatrix, dessen Elemente die sog. Singulärwerte von $\boldsymbol{A}$ repräsentieren. Obige Umschreibung stellt somit eine normierte Darstellung von $\boldsymbol{S}$ dar, wobei σ_i die geometrischen Summe aller Ausgangsvektoren auf die i-te Projektionsachse

$$\sigma_i = \sqrt{s_{ai}^2 + s_{bi}^2 + s_{ci}^2 + \cdots} \tag{B.6}$$

darstellt und $\boldsymbol{U}$ somit die normierte Projektion wiedergibt. Die Größe σ_i gibt somit an, wie nah die Ausgangsvektoren an der Projektionsachse $\boldsymbol{u}_i$ bzw. $\boldsymbol{v}_i$ sind. Für $\sigma_i > \sigma_j$ bedeutet dies somit, dass die gegebenen Ausgangsvektoren (Datenpunkte) stärker in Richtung $\boldsymbol{v}_i$ als $\boldsymbol{v}_j$ zeigen[1]. Die SVD soll in folgender Definition zusammengefasst werden:

Definition B.1: Singulärwertzerlegung

Gegeben sei die Matrix $\boldsymbol{A} \in \mathbb{R}^{n\times m}$ mit Rang r. Dann existiert eine Zerlegung der Form

$$\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^{T} \tag{B.7}$$

mit $\boldsymbol{U} \in \mathbb{R}^{n\times n}$ und $\boldsymbol{V} \in \mathbb{R}^{m\times m}$, dessen Spaltenvektoren $\boldsymbol{u}_i$ bzw. $\boldsymbol{v}_j$ die Links- bzw. Rechts-Singulärvektoren von $\boldsymbol{A}$ darstellen sowie $\boldsymbol{\Sigma} \in \mathbb{R}^{n\times m}$ mit

$$\boldsymbol{\Sigma} = \begin{bmatrix} \sigma_1 & & & & \vdots & \\ & \ddots & & \cdots & 0 & \cdots \\ & & \sigma_r & & \vdots & \\ & \vdots & & & \vdots & \\ \cdots & 0 & \cdots & \cdots & 0 & \cdots \end{bmatrix} \tag{B.8}$$

und den entsprechenden Singulärwerten $\sigma_1 \geq \cdots \geq \sigma_r > 0$ von $\boldsymbol{A}$.

Entsprechend der obigen Sortierung der Singulärwerte σ_i in $\boldsymbol{\Sigma}$ nimmt der Informationsgehalt über ein gegebenes Datenset pro Singulärwert mit steigendem Index ab. Es sei zudem angemerkt, dass $\boldsymbol{V}$ und $\boldsymbol{U}$ in obiger Transformation nicht eindeutig sind und sich je nach verwen-

[1] Diese Eigenschaft wird ebenfalls bei der Hauptkomponentenanalyse (*principal component analysis* – PCA) genutzt, um die diejenigen Projektionsachsen zu finden, welche einen gegebenen Datensatz bezüglich seiner Varianz bestmöglich erklären. Die Projektionsachsen sind hier als Regressoren zu interpretieren. Die PCA kann dann sowohl genutzt werden, um Regressoren mit geringer Varianz und somit Wichtigkeit zu verwerfen oder um eine lineare Transformation des Regressorraums auf eine Darstellung mit weniger Dimensionen vorzunehmen.

detem numerischen Algorithmus ergeben können[1].

Nachfolgend ein kleines Zahlenbeispiel: Gegeben sei

$$\boldsymbol{A} = \begin{bmatrix} 2 & 2 \\ -1 & 1 \end{bmatrix} = \begin{bmatrix} \boldsymbol{a} \\ \boldsymbol{b} \end{bmatrix} \tag{B.9}$$

also eine Datenmatrix mit den Vektoren $\boldsymbol{a} = \begin{bmatrix} 2 & 2 \end{bmatrix}$ und $\boldsymbol{b} = \begin{bmatrix} -1 & 1 \end{bmatrix}$. Entsprechend obiger Definition stellt

$$\boldsymbol{A} = \begin{bmatrix} 2 & 2 \\ -1 & 1 \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}}_{\boldsymbol{U}} \underbrace{\begin{bmatrix} 2\sqrt{2} & 0 \\ 0 & \sqrt{2} \end{bmatrix}}_{\boldsymbol{\Sigma}} \underbrace{\begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix}}_{\boldsymbol{V}^T} \tag{B.10}$$

eine gültige SVD dar. Mit Blick auf Abb. B.2 kann diese Zerlegung wie folgt interpretiert werden:

- Das Produkt $(\boldsymbol{\Sigma}\boldsymbol{V}^T)$ ist eine zweidimensionale Rotationsmatrix, welche eine Drehung von $\pi/4$ erzeugt.
- $\boldsymbol{U}$ beinhaltet zwei Einheitsvektoren und sagt somit aus, dass die Ausgangsvektoren im projizierten (also hier um 45 ° rotierten) Koordinatensystem genau auf dessen Achsen liegen.

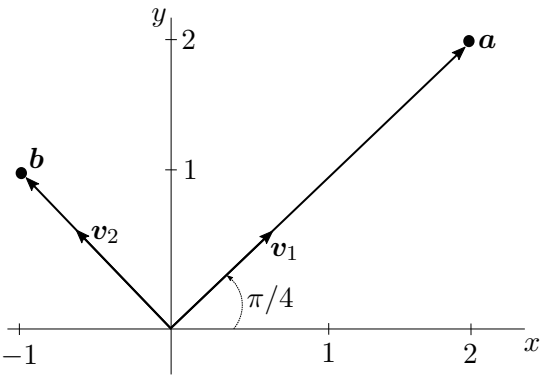


Abb. B.2: Beispielhafte Zerlegung mittels SVD im zweidimensionalem Raum

SVD-Anwendung auf das TLS-Problem
Die TLS-Nebenbedingung aus Definition 3.5

$$\left(\begin{bmatrix} \boldsymbol{\Xi} & \boldsymbol{\psi} \end{bmatrix} + \begin{bmatrix} \boldsymbol{\Pi} & \boldsymbol{e} \end{bmatrix}\right) \begin{bmatrix} \boldsymbol{\theta} \\ -1 \end{bmatrix} = \boldsymbol{0} \tag{B.11}$$

ist ein Gleichungssystem mit n Zeilen und $m+1$ Spalten. Damit allerdings eine eindeutige Lösung für $\boldsymbol{\theta} \in \mathbb{R}^m$ vorliegt, muss $\begin{bmatrix} \boldsymbol{\Xi} & \boldsymbol{\psi} \end{bmatrix} + \begin{bmatrix} \boldsymbol{\Pi} & \boldsymbol{e} \end{bmatrix}$ lediglich m unabhängige Spalten aufweisen. Die TLS-Aufgabe aus Definition 3.5 kann daher auch derart umformuliert werden, dass eine Matrix $\boldsymbol{\Pi}_{TLS} = \begin{bmatrix} \boldsymbol{\Pi} & \boldsymbol{e} \end{bmatrix}$ gesucht wird, welche den

$$\operatorname{rang}\left(\begin{bmatrix} \boldsymbol{\Xi} & \boldsymbol{\psi} \end{bmatrix}\right) = m + 1 \tag{B.12}$$

[1] Beispielsweise *svd(A)* in MATLAB oder *numpy.linalg.svd(A)* mittels numpy in Python.

entsprechend auf

$$\operatorname{rang}\left(\begin{bmatrix}\boldsymbol{\Xi} & \boldsymbol{\psi}\end{bmatrix} + \begin{bmatrix}\boldsymbol{\Pi} & \boldsymbol{e}\end{bmatrix}\right) = m \tag{B.13}$$

reduziert und dabei minimal im Sinne der Frobenius Norm $\|\boldsymbol{\Pi}_{TLS}\|_2^2$ ist. Eine Lösung zu diesem Problem liefert der folgende Satz:

Satz B.1: Matrix-Approximation nach Eckart–Young–Mirsky

Sei $\boldsymbol{A} \in \mathbb{R}^{n\times m}$ eine Matrix mit $n \leq m$. Die Aufgabe

$$\min_{\tilde{\boldsymbol{A}}} \quad \left\|\boldsymbol{A} - \tilde{\boldsymbol{A}}\right\|_2^2, \quad \text{u. d. Nb.} \quad \operatorname{rang}(\tilde{\boldsymbol{A}}) \leq r \tag{B.14}$$

kann mittels der SVD von $\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$ entsprechend der Definition

$$\boldsymbol{U} = \begin{bmatrix}\boldsymbol{U}_1 & \boldsymbol{U}_2\end{bmatrix}, \quad \boldsymbol{\Sigma} = \begin{bmatrix}\boldsymbol{\Sigma}_1 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{\Sigma}_2\end{bmatrix}, \quad \boldsymbol{V} = \begin{bmatrix}\boldsymbol{V}_1 & \boldsymbol{V}_2\end{bmatrix} \tag{B.15}$$

mit $\boldsymbol{U}_1 \in \mathbb{R}^{n\times r}$, $\boldsymbol{\Sigma}_1 \in \mathbb{R}^{r\times r}$ und $\boldsymbol{V}_1 \in \mathbb{R}^{m\times r}$ gelöst werden, konkret durch

$$\tilde{\boldsymbol{A}} = \boldsymbol{U}_1\boldsymbol{\Sigma}_1\boldsymbol{V}_1^T. \tag{B.16}$$

Die Lösung ist zudem eindeutig, falls $\sigma_r \neq \sigma_{r+1}$.

Gemäß der SVD-Definition werden bei obigem Satz somit alle Singulärwerte oberhalb des gewünschten Rangs r verworfen. Bezogen auf (B.12) bedeutet dies, dass der kleinste Singulärwert σ_q entsprechend der SVD-Zerlegung

$$\begin{bmatrix}\boldsymbol{\Xi} & \boldsymbol{\psi}\end{bmatrix} = \begin{bmatrix}\boldsymbol{U}_p & \boldsymbol{u}_q\end{bmatrix}\begin{bmatrix}\boldsymbol{\Sigma}_p & \boldsymbol{0} \\ \boldsymbol{0} & \sigma_q\end{bmatrix}\begin{bmatrix}\boldsymbol{V}_{pp} & \boldsymbol{v}_{pq} \\ \boldsymbol{v}_{qp} & v_{qq}\end{bmatrix}^T \tag{B.17}$$

verworfen wird. Beidseitiges Multiplizieren mit $\boldsymbol{V}$ führt zu[1]

$$\begin{aligned}\begin{bmatrix}\boldsymbol{\Xi} & \boldsymbol{\psi}\end{bmatrix}\begin{bmatrix}\boldsymbol{V}_{pp} & \boldsymbol{v}_{pq} \\ \boldsymbol{v}_{qp} & v_{qq}\end{bmatrix} &= \begin{bmatrix}\boldsymbol{U}_p & \boldsymbol{u}_q\end{bmatrix}\begin{bmatrix}\boldsymbol{\Sigma}_p & \boldsymbol{0} \\ \boldsymbol{0} & \sigma_q\end{bmatrix}\begin{bmatrix}\boldsymbol{V}_{pp} & \boldsymbol{v}_{pq} \\ \boldsymbol{v}_{qp} & v_{qq}\end{bmatrix}^T\begin{bmatrix}\boldsymbol{V}_{pp} & \boldsymbol{v}_{pq} \\ \boldsymbol{v}_{qp} & v_{qq}\end{bmatrix}, \\ \begin{bmatrix}\boldsymbol{\Xi} & \boldsymbol{\psi}\end{bmatrix}\begin{bmatrix}\boldsymbol{V}_{pp} & \boldsymbol{v}_{pq} \\ \boldsymbol{v}_{qp} & v_{qq}\end{bmatrix} &= \begin{bmatrix}\boldsymbol{U}_p & \boldsymbol{u}_q\end{bmatrix}\begin{bmatrix}\boldsymbol{\Sigma}_p & \boldsymbol{0} \\ \boldsymbol{0} & \sigma_q\end{bmatrix}.\end{aligned} \tag{B.18}$$

Ausmultiplizieren lediglich der letzten Spalte bezogen auf σ_q ergibt dann

$$\begin{bmatrix}\boldsymbol{\Xi} & \boldsymbol{\psi}\end{bmatrix}\begin{bmatrix}\boldsymbol{v}_{pq} \\ v_{qq}\end{bmatrix} = \boldsymbol{u}_q\sigma_q. \tag{B.19}$$

Sei $\begin{bmatrix}\boldsymbol{\Xi} & \boldsymbol{\psi}\end{bmatrix} + \begin{bmatrix}\boldsymbol{\Pi} & \boldsymbol{e}\end{bmatrix}$ eine Rang m Approximation von $\begin{bmatrix}\boldsymbol{\Xi} & \boldsymbol{\psi}\end{bmatrix}$ im Sinne von Satz B.1 sei, dann hat die approximierte Matrix die gleichen Singulärvektoren wie die Ursprungsmatrix,

[1] Gemäß der SVD-Definition hat $\boldsymbol{V}$ eine Orthonomalbasis und es gilt $\boldsymbol{V}^T = \boldsymbol{V}^{-1}$.

aber lediglich m Singulärwerte in $\mathbf{\Sigma}_p$. Für die SVD folgt somit:

$$\begin{bmatrix}\mathbf{\Xi} & \boldsymbol{\psi}\end{bmatrix} + \begin{bmatrix}\mathbf{\Pi} & \boldsymbol{e}\end{bmatrix} = \begin{bmatrix}\boldsymbol{U}_p & \boldsymbol{u}_q\end{bmatrix} \begin{bmatrix}\mathbf{\Sigma}_p & \mathbf{0} \\ \mathbf{0} & 0\end{bmatrix} \begin{bmatrix}\boldsymbol{V}_{pp} & \boldsymbol{v}_{pq} \\ \boldsymbol{v}_{qp} & v_{qq}\end{bmatrix}^T. \tag{B.20}$$

Unter Verwendung von (B.17) und (B.20) kann folgende Gleichung umgeschrieben werden:

$$\begin{aligned}\begin{bmatrix}\mathbf{\Pi} & \boldsymbol{e}\end{bmatrix} &= \begin{bmatrix}\mathbf{\Xi} & \boldsymbol{\psi}\end{bmatrix} + \begin{bmatrix}\mathbf{\Pi} & \boldsymbol{e}\end{bmatrix} - \begin{bmatrix}\mathbf{\Xi} & \boldsymbol{\psi}\end{bmatrix} \\ &= -\begin{bmatrix}\boldsymbol{U}_p & \boldsymbol{u}_q\end{bmatrix} \begin{bmatrix}\mathbf{0} & \mathbf{0} \\ \mathbf{0} & \sigma_q\end{bmatrix} \begin{bmatrix}\boldsymbol{V}_{pp} & \boldsymbol{v}_{pq} \\ \boldsymbol{v}_{qp} & v_{qq}\end{bmatrix}^T \\ &= -\begin{bmatrix}\mathbf{0} & \boldsymbol{u}_q\sigma_q\end{bmatrix} \begin{bmatrix}\boldsymbol{V}_{pp} & \boldsymbol{v}_{pq} \\ \boldsymbol{v}_{qp} & v_{qq}\end{bmatrix}^T \\ &= -\boldsymbol{u}_q\sigma_q \begin{bmatrix}\boldsymbol{v}_{pq} \\ v_{qq}\end{bmatrix}^T \\ &= -\begin{bmatrix}\mathbf{\Xi} & \boldsymbol{\psi}\end{bmatrix} \begin{bmatrix}\boldsymbol{v}_{pq} \\ v_{qq}\end{bmatrix} \begin{bmatrix}\boldsymbol{v}_{pq} \\ v_{qq}\end{bmatrix}^T.\end{aligned} \tag{B.21}$$

Einsetzen liefert dann:

$$\begin{aligned}\begin{bmatrix}\mathbf{\Xi} & \boldsymbol{\psi}\end{bmatrix} + \begin{bmatrix}\mathbf{\Pi} & \boldsymbol{e}\end{bmatrix} &= \begin{bmatrix}\mathbf{\Xi} & \boldsymbol{\psi}\end{bmatrix} - \begin{bmatrix}\mathbf{\Xi} & \boldsymbol{\psi}\end{bmatrix} \begin{bmatrix}\boldsymbol{v}_{pq} \\ v_{qq}\end{bmatrix} \begin{bmatrix}\boldsymbol{v}_{pq} \\ v_{qq}\end{bmatrix}^T \\ \left(\begin{bmatrix}\mathbf{\Xi} & \boldsymbol{\psi}\end{bmatrix} + \begin{bmatrix}\mathbf{\Pi} & \boldsymbol{e}\end{bmatrix}\right) \begin{bmatrix}\boldsymbol{v}_{pq} \\ v_{qq}\end{bmatrix} &= \begin{bmatrix}\mathbf{\Xi} & \boldsymbol{\psi}\end{bmatrix} \begin{bmatrix}\boldsymbol{v}_{pq} \\ v_{qq}\end{bmatrix} - \begin{bmatrix}\mathbf{\Xi} & \boldsymbol{\psi}\end{bmatrix} \begin{bmatrix}\boldsymbol{v}_{pq} \\ v_{qq}\end{bmatrix} \begin{bmatrix}\boldsymbol{v}_{pq} \\ v_{qq}\end{bmatrix}^T \begin{bmatrix}\boldsymbol{v}_{pq} \\ v_{qq}\end{bmatrix} \\ \left(\begin{bmatrix}\mathbf{\Xi} & \boldsymbol{\psi}\end{bmatrix} + \begin{bmatrix}\mathbf{\Pi} & \boldsymbol{e}\end{bmatrix}\right) \begin{bmatrix}\boldsymbol{v}_{pq} \\ v_{qq}\end{bmatrix} &= \begin{bmatrix}\mathbf{\Xi} & \boldsymbol{\psi}\end{bmatrix} \begin{bmatrix}\boldsymbol{v}_{pq} \\ v_{qq}\end{bmatrix} - \begin{bmatrix}\mathbf{\Xi} & \boldsymbol{\psi}\end{bmatrix} \begin{bmatrix}\boldsymbol{v}_{pq} \\ v_{qq}\end{bmatrix} \\ \left(\begin{bmatrix}\mathbf{\Xi} & \boldsymbol{\psi}\end{bmatrix} + \begin{bmatrix}\mathbf{\Pi} & \boldsymbol{e}\end{bmatrix}\right) \begin{bmatrix}\boldsymbol{v}_{pq} \\ v_{qq}\end{bmatrix} &= \mathbf{0}.\end{aligned} \tag{B.22}$$

Ein Koeffizientenvergleich zur Ausgangsgleichung (B.11) ergibt dann die Lösung des TLS-Problems:

$$\boldsymbol{\theta}^* = \frac{-\boldsymbol{v}_{pq}}{v_{qq}}. \tag{B.23}$$

Abkürzungen und Formelzeichen

Nachfolgend werden die wichtigsten Abkürzungen und Formelzeichen zusammengefasst. Einige der Formelzeichen haben je nach Kapitel unterschiedliche Bedeutungen, sodass nachfolgende Darstellung z. T. kontextbezogen zu interpretieren ist.

Abkürzungen/Indizes

ARMAX	*autoregressive-moving average model with exogenous inputs*
BLUE	*best linear unbiased estimator*
Corr	Korrelationsmatrix
Cov	Kovarianz
EKF	Erweitertes KF
FEM	filter error model
KF	Kalman-Filter
LASSO	*least absolute shrinkage and selection operator*
LPV	*linear parameter-variant*
LS	*least squares*
LTI	*linear time-invariant*
MAE	*mean absolute error*
MIMO	*multiple-input multiple-output*
ML	*maximum likelihood*
MSE	*mean squared error*
OEM	*output error model*
OLS	*ordinary least squares*
PRBS	*pseudo-random binary sequence*
PS	Partikelschwarmoptimierung
RLS	*recursive least squares*
SGD	*stochastic gradient descent*
SISO	*single-input single-output*
SQE	Erklärte Abweichungsquadratsumme
SQP	Sequentielle quadratische Programmierung
SQR	Residuenquadratsumme
SQT	Gesamtabweichungsquadratsumme
SVD	*singular value decomposition*
TLS	*total least squares*
UB	Ungleichungsnebenbeschränkung
UKF	*unscented Kalman-filter*
Var	Varianz
WLS	*weighted least squares*

Formelzeichen

$\boldsymbol{A}$	Systemmatrix
$\mathcal{A}$	Aktive Menge
$\boldsymbol{B}$	Eingangsmatrix
$\boldsymbol{C}$	Ausgangsmatrix
C_{XX}	Autokovarianz
C_{XY}	Kreuzkovarianz
d	Modellfreiheitsgrade
$\boldsymbol{D}$	Durchgangsmatrix / Datensatz (Kreuz-Validierung)
e	Residuum
E	Erwartungswert
f	(Zustands-)Funktion / Modellfunktion
$\mathcal{F}$	Ereignissystem
g	Ausgangsfunktion / Gleichungsnebenbedingung
$\boldsymbol{G}$	Übertragungsmatrix
h	Ungleichungsnebenbedingung
$\boldsymbol{H}$	Zeitdiskrete Eingangsmatrix
$\boldsymbol{I}$	Einheitsmatrix
I	Intervall
J	Kostenfunktion
k	Lauf-, Abtast- oder Iterationsindex
$\boldsymbol{K}$	Kalman-Matrix
L	Lagrange-Funktion
$\mathbb{L}$	Likelihood-Funktion
m	Systemrauschen
$\boldsymbol{M}$	Messrauschen
n	Systemrauschen
N	Anzahl Messpunkte / Stichprobenumfang
$\boldsymbol{N}$	Messrauschenmatrix
p	Wahrscheinlichkeitsdichtefuntkion / Konvergenzordnung
P	Wahrscheinlichkeit
$\boldsymbol{P}$	Kovarianzmatrix
$\boldsymbol{Q}_B$	Beobachtbarkeitsmatrix
$\boldsymbol{Q}_S$	Steuerbarkeitsmatrix
$\boldsymbol{R}$	Messrauschmatrix / Rauschmatrix des ML-Verfahrens
R_{XX}	Autokorrelation
R_{XY}	Kreuzkorrelation
R^2	Bestimmtheitsmaß
s	Parameter im Laplace-Bereich
$\boldsymbol{s}$	Suchrichtungsvektor
t	Zeit(punkt)
T	Zeitintervall
u	Eingangsgröße
v	Partikelgeschwindigkeit (PSO-Verfahren)
$\boldsymbol{W}$	Gewichtungsmatrix
x	Zustandsgröße / Optimierungsgröße
X	Zufallsvariable
$\mathcal{X}$	Zulässiges Optimierungsgebiet

y	Ausgangsgröße
z	Parameter im z-Bereich
α	Schrittweite / Allgemeiner Skalierungsfaktor
γ	Konfidenzniveau / Verstärkungsterm (RLS-Verfahren)
$\boldsymbol{\theta}$	Parametervektor
κ	Kondition (einer Matrix)
λ	Vergessensfaktor (RLS-Verfahren) / Lagrange-Multiplikator
μ	Erwartungswert / Konvergenzrate / Lagrange-Multiplikator
ν	Rauschterm
$\mathbf{\Pi}$	Regressorrauschmatrix
ρ	Korrelationskoeffizient
ϱ	Verschiebeoperator
σ	Standardabweichung
σ^2	Varianz
$\mathbf{\Phi}$	(Zeitdiskrete) Transitionsmatrix
$\boldsymbol{\xi}$	Daten-/Regressorvektor
$\mathbf{\Xi}$	Daten-/Regressormatrix
$\boldsymbol{\psi}$	Messvektor
$\mathbf{\Psi}$	Datenbasis
ω	Ergebnis (Realisierung) eines Zufallsexperiments
Ω	Ergebnismenge

Allgemeine Notation

x, X	Skalere Größen
x^*, X^*	Soll- bzw. Referenz-Größe / Optimum
$\hat{x}, \widehat{X}$	Geschätzte Größe
$\overline{x}, \overline{X}$	Arithmetischer Mittelwert
$\boldsymbol{x}^T, \boldsymbol{X}^T$	Transponierte
$\mathcal{L}$	Laplace-Transformation
$\mathcal{Z}$	z-Transformation
∇	Nabla-Operator
Δ	(Numerische) Differenz