

Masterarbeit

Zustandsrekonstruktion mit wenigen Messungen durch
Kombination von Kalman-Filterung und Compressive-Sensing

ZHAO, Haibin

15. November 2020

Referent: Prof. Dr.-Ing. Uwe D. Hanebeck

Betreuer: Dr.-Ing. Benjamin Noack, Christopher Funk, M.Sc.

Zusammenfassung

Die Signalrekonstruktion ist immer ein wichtiges Thema, wobei das Kalman-Filter eine am weitesten verbreitete Methode ist. Aber wenn es viele Zustandsgrößen gibt, brauchen wir viele Messungen. Das Compressive-Sensing (CS) ist in letzter Zeit ein sehr beliebtes Thema. Man kann mit dem Compressive-Sensing ein Signal mit sehr wenigen Messungen zu rekonstruieren, aber das Compressive-Sensing hat viele Einschränkungen. Daher kombiniert diese Arbeit das Kalman-Filter und das Compressive-Sensing, um die Anzahl der benötigten Messungen zur Zustandsschätzung zu reduzieren. Erstens schlägt diese Arbeit drei Methoden vor, um die Anforderungen von dem Compressive-Sensing abzuschwächen: dünnbesetzte Änderung, Dimensionreduzierung und Sparse-Coding-Representation. Außerdem schlägt diese Arbeit zwei Methoden zur Fusion der Informationen aus dem Kalman-Filter und dem Compressive-Sensing vor: durch die Kalman-Gleichung und durch die Pseudo-Messungen. Darüber hinaus schlägt diese Arbeit ein iteratives Verfahren zur Verbesserung der Schätzung. Im idealen Fall kann die Schätzung mit dem neuen Verfahren im Vergleich zum traditionellen Kalman-Filter um 70 Prozent verbessert werden. Dann wendet diese Arbeit das Verfahren auf rauschbehaftete und zweidimensionale Signale an und stellt fest, dass das Verfahren die Schätzung im Vergleich zum traditionellen Kalman-Filter um 25 Prozent verbessert. Schließlich analysiert diese Arbeit den Zusammenhang zwischen Fehler und Systemrauschen sowie Messunsicherheiten und kommt zu dem Schluss, dass das Verfahren bei guter Modellierung und schlechten Sensoren deutlich gute Ergebnisse liefert.

Eidesstattliche Erklärung

Hiermit erkläre ich, die vorliegende Masterarbeit selbstständig angefertigt zu haben.
Die verwendeten Quellen sind im Text gekennzeichnet und im Literaturverzeichnis
aufgeführt.

Karlsruhe, 15. November 2020



ZHAO, Haibin

Inhaltsverzeichnis

| | |
|--|-----------|
| Abbildungsverzeichnis | V |
| Notation | IX |
| 1 Einleitung | 1 |
| 1.1 Motivation | 1 |
| 1.2 Anwendungen | 2 |
| 1.3 Ziel der Arbeit | 3 |
| 1.4 Aufbau der Arbeit | 3 |
| 2 Temperaturverteilung | 5 |
| 2.1 Modellierung der Wärmeleitung | 5 |
| 2.1.1 Wärmeleitung in einem Draht | 5 |
| 2.1.2 Wärmeleitung in Platte | 6 |
| 2.2 Numerische Lösung | 6 |
| 2.2.1 Modalanalyse | 6 |
| 2.2.2 Finite-Differenzen-Verfahren | 10 |
| 2.3 Simulationen | 11 |
| 2.3.1 1D-Fall | 11 |
| 2.3.2 2D-Fall | 13 |
| 2.4 Zusammenfassung | 16 |
| 3 Kalman-Filter | 17 |
| 3.1 Kalman-Filter für die Modalanalyse | 18 |
| 3.2 Kalman-Filter für die FDM | 19 |
| 3.3 Simulation | 21 |
| 3.3.1 1D-Fall | 22 |
| 3.3.2 2D-Fall | 24 |
| 3.4 Fehlermaß und Bewertung | 24 |
| 3.5 Zusammenfassung | 25 |

| | | |
|----------|---|-----------|
| 4 | Compressive-Sensing | 27 |
| 4.1 | Einleitung | 27 |
| 4.2 | ℓ_p -Norm | 28 |
| 4.2.1 | Definition der ℓ_p -Norm | 28 |
| 4.2.2 | Vergleich von ℓ_0 -, ℓ_1 - und ℓ_2 -Norm | 28 |
| 4.3 | Dünnbesetzte und komprimierbare Signale | 30 |
| 4.4 | Messbasis | 30 |
| 4.4.1 | Anforderung auf der Messbasis | 30 |
| 4.4.2 | Anzahl der Messungen | 31 |
| 4.4.3 | Kohärenz der Matrix | 32 |
| 4.5 | Basisfunktionen | 32 |
| 4.5.1 | Fourier-Basis | 33 |
| 4.5.2 | DCT-Basis | 33 |
| 4.5.3 | Wavelet-Basis | 33 |
| 4.6 | Beispiele | 36 |
| 4.6.1 | 1D-Signal-Rekonstruktion | 36 |
| 4.6.2 | 2D-Signal-Rekonstruktion | 37 |
| 4.7 | Zusammenfassung | 39 |
| 5 | Kalman-Filtered Compressive-Sensing | 41 |
| 5.1 | Einleitung | 41 |
| 5.2 | Stand der Technik | 41 |
| 5.3 | Kalman-Filter mit zufälligen Messungen | 42 |
| 5.3.1 | Systemmodell | 42 |
| 5.3.2 | Simulation | 43 |
| 5.3.3 | Diskussion | 44 |
| 5.4 | Signal-Analyse | 45 |
| 5.4.1 | Hauptprobleme | 45 |
| 5.4.2 | Komprimierbare Änderungen | 46 |
| 5.4.3 | Dimensionreduzierung | 48 |
| 5.4.4 | Sparse-Coding-Representation | 49 |
| 5.5 | Kalman-Gleichung mit Compressive-Sensing | 50 |
| 5.6 | Kalman-Filter mit Pseudo-Messungen | 52 |
| 5.6.1 | Pseudo-Messungen | 53 |
| 5.6.2 | Dynamische Gewichtung | 53 |
| 5.6.3 | Fehler-Akkumulation | 54 |
| 5.7 | Koeffizientenaktualisierung | 54 |
| 5.7.1 | Aktualisierung aus der Schätzung | 56 |
| 5.7.2 | Aktualisierung aus der Prädiktion | 56 |

| | | |
|-----------------------------|---|-----------|
| 5.8 | Iteratives Verfahren | 57 |
| 5.9 | Simulation | 57 |
| 5.10 | Zusammenfassung | 58 |
| 6 | Erweiterungen des Verfahrens | 61 |
| 6.1 | KFCS für rauschenbahaftete Signale | 61 |
| 6.2 | KFCS für 2D-Signale | 63 |
| 6.3 | Fehlerbewertung | 64 |
| 6.4 | Zusammenfassung | 65 |
| 7 | Zusammenfassung und Ausblick | 67 |
| 7.1 | Zusammenfassung | 67 |
| 7.2 | Ausblick | 68 |
| Anhang | | 69 |
| A.1 | Herleitung der Temperaturverteilung | 69 |
| A.1.1 | Modalanalyse | 69 |
| A.1.2 | Finite-Differenzen-Verfahren | 77 |
| A.2 | Herleitung des Kalman-Filters | 80 |
| Literaturverzeichnis | | 83 |

Abbildungsverzeichnis

| | | |
|------|---|----|
| 1.1 | Zu messende Temperaturverteilung. | 1 |
| 2.1 | Nummerierung. | 9 |
| 2.2 | Temperaturverteilungen mit sinusförmiger Anfangsbedingung | 12 |
| 2.3 | Temperaturverteilungen mit treppenförmiger Anfangsbedingung | 12 |
| 2.4 | Temperaturverteilungen mit sinusförmiger Anfangsbedingung | 12 |
| 2.5 | Temperaturverteilungen mit sinusförmiger Anfangsbedingung | 13 |
| 2.6 | 2D-Temperaturverteilung aus der MA | 13 |
| 2.7 | 2D-Temperaturverteilung aus der FDM | 14 |
| 2.8 | 2D-Temperaturverteilung aus der MA | 14 |
| 2.9 | 2D-Temperaturverteilung aus der FDM | 15 |
| 2.10 | 2D-Temperaturverteilung aus der MA | 15 |
| 2.11 | 2D-Temperaturverteilung aus der FDM | 16 |
| 3.1 | Kalman-Filter für die Modalanalyse mit 12 äquidistanten Sensoren. | 22 |
| 3.2 | Kalman-Filter für die Modalanalyse mit 24 äquidistanten Sensoren. | 22 |
| 3.3 | Kalman-Filter für die Modalanalyse mit 36 äquidistanten Sensoren. | 22 |
| 3.4 | Kalman-Filter für die FDM mit 12 äquidistanten Sensoren. | 23 |
| 3.5 | Kalman-Filter für die FDM mit 24 äquidistanten Sensoren. | 23 |
| 3.6 | Kalman-Filter für die FDM mit 36 äquidistanten Sensoren. | 23 |
| 3.7 | 2D-Kalman-Filter für die Modalanalyse mit 64 äquidistante Sensoren. | 24 |
| 3.8 | 2D-Kalman-Filter für die FDM mit 64 äquidistante Sensoren. | 24 |
| 3.9 | Vergleich aller Schätzungen. | 25 |
| 4.1 | Ablauf traditioneller Signalverarbeitung. | 27 |
| 4.2 | Ablauf des Compressive-Sensing. | 27 |
| 4.3 | Lineare Approximationen. | 29 |
| 4.4 | Vergleich ℓ_p Norm. | 29 |
| 4.5 | Verlauf der Wavelet-Transformation. | 34 |
| 4.6 | Koeffizienten mit Wavelet-Basis | 35 |
| 4.7 | Signal und seine Spektrum. | 36 |

| | | |
|------|---|----|
| 4.8 | Messstellen | 36 |
| 4.9 | Rekonstruiertes Signal. | 37 |
| 4.10 | Originales Bild. | 37 |
| 4.11 | Bilder | 38 |
| 4.12 | Rekonstruiertes Bild. | 38 |
| | | |
| 5.1 | Eingestellte Sensoren. | 42 |
| 5.2 | rsKF aus der FDM mit 12 zufällige Messungen. | 43 |
| 5.3 | rsKF aus der Modalanalyse mit 12 zufällige Messungen. | 44 |
| 5.4 | Vergleich rsKF und KF. | 44 |
| 5.5 | Nachteil des rsKF | 45 |
| 5.6 | K -Terme-Approximation | 46 |
| 5.7 | Rekonstruiertes Signal | 46 |
| 5.8 | K -Terme-Approximation für die Änderung der Koeffizienten | 47 |
| 5.9 | Rekonstruiertes Signal für die Änderung der Koeffizienten | 47 |
| 5.10 | K -Terme-Approximation für die Änderung der Koeffizienten | 48 |
| 5.11 | Rekonstruiertes Signal für die Änderung der Koeffizienten | 48 |
| 5.12 | Visualisierung von Θ | 49 |
| 5.13 | Rekonstruiertes Signal mit Kalman-Gleichung | 51 |
| 5.14 | Rekonstruiertes Signal mit Kalman-Gleichung | 51 |
| 5.15 | Rekonstruiertes Signal mit Kalman-Gleichung | 52 |
| 5.16 | Fehler mit verschiedenen G_2/G_1 | 52 |
| 5.17 | Pseudo-Messungen. | 53 |
| 5.18 | Vergleich ohne und mit dynamischer Gewichtung. | 54 |
| 5.19 | Pseudo-Messungen mit Wavelet-Basis. | 54 |
| 5.20 | Verlauf des vorherigen KFCS. | 55 |
| 5.21 | Verlauf des vorherigen KFCS. | 55 |
| 5.22 | Strukturbild des KFCS mit Rückkoppelung. | 55 |
| 5.23 | Verlauf für Koeffizientenaktualisierung aus der Schätzung. | 56 |
| 5.24 | Verlauf für Koeffizientenaktualisierung aus der Prädiktion. | 56 |
| 5.25 | Verlauf für iteratives Verfahren. | 57 |
| 5.26 | Vergleich aller Verfahren. | 57 |
| 5.27 | Vollständiger Gedankengang zu diesem Kapitel. | 58 |
| | | |
| 6.1 | Temperaturverteilung mit Rauschen. | 61 |
| 6.2 | Kalman-Filter für rauschbehaftetes Signal | 62 |
| 6.3 | rsKF für rauschbehaftetes Signal | 62 |
| 6.4 | KFCS für rauschbehaftetes Signal | 62 |
| 6.5 | Vergleich für rauschbehaftetes Signal. | 63 |
| 6.6 | 2D-Kalman-Filter mit 25 Sensoren. | 63 |

| | | |
|------|---|----|
| 6.7 | 2D-KFCS mit 25 Messungen. | 64 |
| 6.8 | Vergleich für 2D-Signal. | 64 |
| 6.9 | Fehler | 65 |
| 6.10 | Differenzfehler von verschiedenen Aspekten. | 65 |
| A.1 | Nummerierung | 76 |
| A.2 | Nummerierung | 79 |

Notation

Konventionen

| | |
|------------------|--|
| N | Anzahl der räumlichen Punkte |
| M | Anzahl der Messungen |
| G | Grad der Modalanalyse |
| S | Anzahl der möglichen Sensoren |
| f | Temperatur, Zustandsgröße im Finite-Differenzen-Verfahren |
| x | räumliche Position |
| t | Zeit |
| k | Wärmeleitfähigkeit |
| T | Zustandsgröße in der Modalanalyse |
| \mathbf{A} | Systemmodell in der Modalanalyse |
| φ | Eigenfunktionen in x -Richtung |
| ψ | Eigenfunktionen in y -Richtung |
| \mathcal{U} | Anregungen |
| w | Systemrauschen |
| v | Messrauschen |
| Φ | Messabbildung im Compressive-Sensing |
| Φ_S | Abbildung von allen Knoten zu allen Stellen zu möglichen Sensorstellen |
| Ψ | Messabbildung in der Modalanalyse |
| $\tilde{\Psi}$ | komprimierende Messabbildung in der Modalanalyse, $\tilde{\Psi} = \Phi \cdot \Psi$ |
| y | Messungen |
| \tilde{y} | komprimierte Messungen, $\tilde{y} = \Phi \cdot y$ |
| \mathbf{D} | Systemmodell im Finite-Differenzen-Verfahren |
| \mathbf{K} | Verstärkungsmatrix des Kalman-Filters |
| \mathbf{C} | Kovarianzmatrix in der Schätzung des Kalman-Filters |
| Θ | Basisfunktionsmatrix |
| $\tilde{\Theta}$ | Basisfunktionsmatrix im Compressive-Sensing, $\tilde{\Theta} = \Phi \cdot \Theta$ |

Notation

| | |
|------------|---|
| z^{CS} | Koeffizienten der Basisfunktionen |
| Δz | Änderung der Koeffizienten |
| z^{KF} | aktualisierte z^{CS} aus der Schätzung des Kalman-Filters |
| z^P | prädiziertes z^{CS} aus Systemmodell |
| ϵ | Fehler bei der Signalrekonstruktion |

Abkürzungen

| | |
|------|-------------------------------------|
| KF | Kalman-Filter |
| CS | Compressive-Sensing |
| KFCS | Kalman-Filtered Compressive-Sensing |
| MA | Modalanalyse |
| FDM | Finite-Differenzen-Verfahren |
| WT | Wavelet-Transformation |
| DCT | Diskrete Cosinus-Transformtion |
| SCR | Sparse-Coding-Representation |
| PM | Pseudo-Messungen |

KAPITEL 1

Einleitung

1.1 Motivation

Das Kalman-Filter wird häufig zur Zustandsschätzung verwendet, aber hat es den Nachteil, dass es viele Messungen erfordert. Wenn wir zum Beispiel die Temperaturverteilung wie unten dargestellt überwachen wollen, brauchen wir 100 Sensoren, wenn wir das traditionelle Kalman-Filter verwenden. Dies verbraucht viele Energie für die Messungen und Kommunikation in einem Sensornetzwerk. Deshalb möchten wir das Signal mit weniger Messungen abschätzen und damit Energie einsparen.

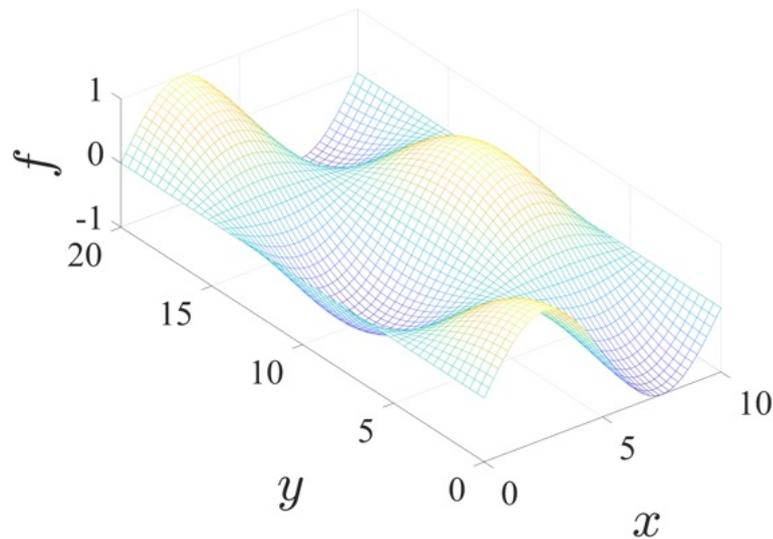


Abbildung 1.1: Zu messende Temperaturverteilung.

In den letzten Jahren wird Compressive-Sensing auf dem Gebiet der Signalgewinnung immer mehr geforscht, bei dem man ein Signal durch viel weniger Messungen rekonstruieren kann, als mit der traditionellen Abtastung, wenn das Signal und das System einige bestimmte Eigenschaften erfüllen. Deshalb möchten wir das Kalman-Filter mit dem Compressive-Sensing kombinieren, um die Zustandsschätzung mit

weniger Messungen durchzuführen, sodass wir die Energie für Messungen und Kommunikation sparen können.

1.2 Anwendungen

Compressive-Sensing hat ein breites Anwendungsspektrum in vielen kostengünstigen Digitalkameras und Audioerfassungsgeräten [2; 3], stromsparenden Lautsprechern und Bildaufnahmegeräten [4], astronomischen Beobachtungen [5], militärischen Karten aus Radarsignalen [6] usw. Folgende sind einige Beispiele:

- Datenkomprimierung

Beim Compressive-Sensing werden die Daten normalerweise mit einer Zufallmatrix komprimiert und gemessen. Deswegen kann man die Daten gleichzeitig messen und komprimieren. Dies ist vor allem bei der Distributed-Source-Coding (z. B. Sensornetzwerken) besonders nützlich.

- Kanal-Kodierung

Die Dünnbesetztheit, die Zufälligkeit und die Konvexität vom Compressive-Sensing können auf den Entwurf schneller Fehlerkorrekturcodes angewendet werden, um Fehler bei der Übertragung zu vermeiden.

- Inverse Probleme

In anderen Fällen besteht die einzige Möglichkeit, das Signal zu erfassen, darin, ein spezifisches Messsystem anzuwenden. Allerdings mithilfe vom Compressive-Sensing kann das Signal effektiv aufgenommen werden, z. B. bei der Herzspinnresonanztomografie.

- Datenerfassung

In einigen kritischen Situationen ist es schwierig, analoge Signale vollständig zu erfassen und zu komprimieren, aber mithilfe des Compressive-Sensing können physische Erfassungsgeräte so entworfen werden, dass sie analoge Signale direkt aufnehmen.

Es gibt auch viele Einrichtungen, die schon entwickelt wurden. Z. B. eine Ein-Pixel-Kamera und ein A/I-Wandler wurden an Rice Universität entwickelt [7; 8], eine Kamera mit codierter Blende [11] und ein MRI-RF-Impulsgerät [9] wurden am MIT entwickelt und ein DNA-Mikroarray-Sensor [10] wurde an Universität Illinois entwickelt.

1.3 Ziel der Arbeit

Das Ziel dieser Arbeit ist, die Anzahl der Messungen, die zur Schätzung des Signals erforderlich sind, durch die Kombination vom Kalman-Filter und dem Compressive-Sensing zu reduzieren. Es ist daher wichtig, in dieser Arbeit ein neues Verfahren zur Fusion von den beiden Ansätzen vorzuschlagen, das verallgemeinerbar sein sollte und das die Informationen aus den beiden Ansätzen vollständig fusionieren sollte. Zweitens, da das Compressive-Sensing viele Anforderungen an das Signal und das System hat, die in der Praxis nicht immer erfüllt werden, müssen wir einige Maßnahmen treffen, um die Anforderungen vom Compressive-Sensing abzuschwächen. Mit diesen beiden Aspekten können wir ein effizientes und allgemein anwendbares Verfahren zur Signalrekonstruktion entwerfen.

1.4 Aufbau der Arbeit

Um das Ziel zu erreichen, wird in dieser Arbeit zunächst die Wärmeleitung als Beispiel hergeleitet, dann werden zwei wichtige Grundlagen, nämlich das Kalman-Filter und das Compressive-Sensing vorgestellt. Dann werden drei Verbesserungsmethoden und zwei Fusionsverfahren vorgeschlagen. Schließlich wird das neue Verfahren auf weitere Fällen angewendet. Um das neue Verfahren zu bewerten, wird am Ende noch eine Fehlerbewertung durchgeführt.

- Temperaturverteilung

In dieser Arbeit nehmen wir die Temperaturverteilung als Beispiel, um die Wirksamkeit des Verfahrens zu prüfen. Deswegen leiten wir zunächst die Temperaturverteilung aus dem Diffusionsmodell her. Wir werden nicht nur die analytischen Lösungen herleiten, sondern auch die numerischen Lösungen. Schließlich werden wir die Simulationen für verschiedene Fälle durchführen.

- Kalman-Filter

Wir werden die Formeln für das Kalman-Filter herleiten. Und dann werden wir das Kalman-Filter verwenden, um die Temperaturverteilung im ersten Kapitel zu rekonstruieren.

- Compressive-Sensing

Zuerst werden wir das Modell, die Eigenschaften und die Anforderungen vom Compressive-Sensing vorstellen. Dann werden wir das Compressive-Sensing mit der traditionellen Abtastung vergleichen. Schließlich werden wir einige Beispiele vom Compressive-Sensing im 1D- und 2D-Fall zeigen.

- Abschwächen der Anforderungen des Compressive-Sensing

In diesem Teil werden wir drei Methoden vorschlagen, die die Anforderungen vom CS an die Signale abzuschwächen. Dabei handelt es sich um dünnbesetzte Änderung, Reduktion der Dimension und Sparse-Coding-Representation. Durch diese drei Methoden werden die Anforderungen des CS stark abgeschwächt.

- Fusionsverfahren

Dies ist der Hauptteil dieser Arbeit. Wir werden zwei Fusionsverfahren vorstellen und die Vor- und Nachteile von beiden Ansätzen analysieren. Und schließlich werden wir eine vollständige Struktur erstellen. Diese Struktur ist universell, vielseitig und effizient. Durch dieses Verfahren können wir ein besseres Ergebnis erzielen als traditionelle Kalman-Filter.

- Erweiterungen

Nachdem wir das neue Verfahren vorgestellt haben, werden wir es auf viele andere Fälle, wie z. B. rauschbehaftete Fälle und zweidimensionale Fälle, anwenden und deren Fehler berechnen. Schließlich werden wir die Fehler vom Kalman-Filter und dem neuen Verfahren unter verschiedenen Systemrauschen und Messunsicherheiten analysieren und damit die Vor- und Nachteile des neuen Verfahrens analysieren. Dann kommen wir zu dem Ergebnis, zu welchen Fällen das neue Verfahren am meisten passt.

KAPITEL 2

Temperaturverteilung

2.1 Modellierung der Wärmeleitung

In der Arbeit wird die Wärmeleitung als Beispiel genommen, weil die Temperaturüberwachung eine wichtige Aufgabe in unserem täglichen Leben ist. Vom Automobil bis zur Luft- und Raumfahrt, von Wäldern bis zu den Ozeanen, von unseren Körpern bis zum Klima, ist die Temperaturmessung von großer Bedeutung. Die mathematische Beschreibung der Wärmeleitung ist das Diffusionsmodell. Das Diffusionsmodell ist eine partielle Differentialgleichung. Im Folgenden werden wir zwei Ansätze zur Lösung dieser Gleichung verwenden, nämlich das Finite-Differenzen-Verfahren (FDM) und die Modalanalyse (MA).

2.1.1 Wärmeleitung in einem Draht

Das Diffusionsmodell für den 1D-Fall ist

$$\underbrace{\frac{\partial f}{\partial t}(x, t) = k \frac{\partial^2 f}{\partial x^2}(x, t)}_{\text{Diffusionsgleichung}} + \underbrace{u(x, t)}_{\text{Anregung}},$$

wobei f für die Temperatur und k für die Wärmeleitfähigkeit steht. Die Randbedingungen beschreiben eine Isolation an den Enden (Neumann-Randbedingung), also

$$\left. \frac{\partial f}{\partial x}(x, t) \right|_{x=0} = \left. \frac{\partial f}{\partial x}(x, t) \right|_{x=L} = 0.$$

Die Isolation bedeutet, dass der Wärmefluss an den Punkten auf der Oberfläche des Objekts null ist. Die Anfangsbedingung ist eine bekannte Temperaturverteilung $f(x, t_0)$ zum Zeitpunkt t_0 .

2.1.2 Wärmeleitung in Platte

Das Diffusionsmodell für den 2D-Fall ist

$$\frac{\partial f}{\partial t}(x, y, t) = k \left(\frac{\partial^2 f}{\partial x^2}(x, y, t) + \frac{\partial^2 f}{\partial y^2}(x, y, t) \right) + u(x, y, t) .$$

Die Randbedingungen sind Isolation an den Rändern (Neumann-Randbedingung), also

$$\begin{aligned} \frac{\partial f}{\partial x}(x, y, t) \Big|_{x=0} &= \frac{\partial f}{\partial x}(x, y, t) \Big|_{x=L_x} = 0 , \\ \frac{\partial f}{\partial y}(x, y, t) \Big|_{y=0} &= \frac{\partial f}{\partial y}(x, y, t) \Big|_{y=L_y} = 0 . \end{aligned}$$

Die Anfangsbedingung ist eine bekannte Temperaturverteilung $f(x, y, t_0)$ zum Zeitpunkt t_0 .

2.2 Numerische Lösung

Da der Computer die Differenzialgleichung nicht direkt lösen kann, müssen wir die Diffusionsgleichung numerisch lösen.

2.2.1 Modalanalyse

Die Modalanalyse wird zur Bestimmung der Eigenfrequenzen (Eigenfunktionen) und der Eigenformen (Gewichtung zu den Eigenfunktionen) in der Strukturdynamik verwendet. Die Modalanalyse wird auch Eigenwertanalyse genannt [12]. Bei der Modalanalyse werden die Variablen getrennt und separat gelöst. Z. B. beim Diffusionsmodell wird die Variable x als Eigenfunktionen und die andere Variable t als die Gewichtungen bezeichnet.

Zuerst betrachten wir den 1D-Fall ohne Anregung. Jetzt trennen wir die Temperaturverteilung $f(x, t)$ durch $f(x, t) =: X(x)T(t)$. Dann separieren wir die Variablen

$$\frac{1}{k} \frac{\dot{T}(t)}{T(t)} = \frac{\ddot{X}(x)}{X(x)} \stackrel{!}{=} C .$$

Wenn wir die Variable $X(x)$ mit den Nebenbedingungen lösen, können wir finden, dass die Lösung viele Cosinusfunktionen mit unterschiedlichen Frequenzen sind. D. h. die Lösung der Variable $X(x)$ hat viele Möglichkeiten, also:

$$X_i(x) = D_i \cos(\lambda_i x), \quad \lambda_i = i \frac{\pi}{L} \quad i = 0, 1, 2, \dots ,$$

wobei D_i Konstanten sind. Offensichtlich erfüllen $X_i(x)$ die Orthogonalitätsbedingung, d. h. nach einer Normalisierung bilden sie eine Basis (hier wird auch als Eigenfunktion bezeichnet), also

$$\begin{aligned}\psi_0(x) &= \sqrt{\frac{1}{L}}, \\ \psi_i(x) &= \sqrt{\frac{2}{L}} \cos(\lambda_i x), \quad i = 1, 2, \dots\end{aligned}$$

Wir können auch so verstehen, dass wir durch die Wahl von bestimmten D_i die $X_i(x)$ in $\psi_i(x)$ normalisiert haben, was bedeutet, dass $X_i(x)$ und $\psi_i(x)$ äquivalent sind. Deswegen erhalten wir

$$f(x, t) = \sum_{i=0}^{\infty} T_i(t) \psi_i(x).$$

Betrachtet man $T_i(t)$ aus der obigen Gleichung als Zustandsgrößen, dann haben wir die Messabbildung, was in Matrixform

$$\underbrace{\begin{bmatrix} f_1[k+1] \\ \vdots \\ f_N[k+1] \end{bmatrix}}_{=: \underline{f}_{k+1}} = \underbrace{\begin{bmatrix} \psi_1[1] & \cdots & \psi_G[1] \\ \vdots & \ddots & \vdots \\ \psi_1[N] & \cdots & \psi_G[N] \end{bmatrix}}_{=: \Psi} \underbrace{\begin{bmatrix} T_1[k+1] \\ \vdots \\ T_G[k+1] \end{bmatrix}}_{=: \underline{T}_{k+1}}$$

ist, wobei N für die Anzahl der räumlichen Knoten und k für die Zeitpunkt stehen. Für jeder $\psi_i(x)$ können wir eine entsprechende $T_i(t)$ durch

$$\begin{aligned}\frac{1}{k} \frac{\dot{T}(t)}{T(t)} &= C = -\lambda^2 \\ \Rightarrow \dot{T}_i(t) &= -k\lambda_i^2 T_i(t)\end{aligned}$$

berechnen. Dann bekommen wir das Systemmodell

$$\underbrace{\begin{bmatrix} T_1[k+1] \\ \vdots \\ T_G[k+1] \end{bmatrix}}_{=: \underline{T}_{k+1}} = \underbrace{\begin{bmatrix} 1 - \Delta t k \lambda_1^2 & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & 1 - \Delta t k \lambda_G^2 \end{bmatrix}}_{=: \mathbf{A}} \underbrace{\begin{bmatrix} T_1[k] \\ \vdots \\ T_G[k] \end{bmatrix}}_{=: \underline{T}_k}.$$

Für den inhomogenen Fall können wir die Anregungen auch durch die Eigenfunktionen beschreiben, dann bekommen wir das Systemmodell in den 1D-Fall mit An-

regungen, nämlich

$$\begin{aligned}\underline{T}_{k+1} &= \mathbf{A} \cdot \underline{T}_k + \Delta t \underbrace{\begin{bmatrix} U_1[k] \\ \vdots \\ U_G[k] \end{bmatrix}}_{=: \underline{U}_k} \\ &= \mathbf{A} \cdot \underline{T}_k + \underline{U}_k, \\ \underline{f}_{k+1} &= \mathbf{\Psi} \cdot \underline{T}_{k+1}.\end{aligned}$$

Es gibt zwei Anmerkungen zur Herleitung des Systemmodells in den 2D-Fall.

Erstens, es gibt zwei Gruppen Eigenfunktionen jeweils für x - und y -Richtung, also

- die Eigenfunktionen in x -Richtung

$$\psi_i(x) = \begin{cases} \sqrt{\frac{1}{L_x}}, & i = 0, \\ \sqrt{\frac{2}{L_x}} \cos(\lambda_i x), & i = 1, 2, \dots \end{cases}$$

- und die Eigenfunktionen in y -Richtung

$$\varphi_j(y) = \begin{cases} \sqrt{\frac{1}{L_y}}, & j = 0, \\ \sqrt{\frac{2}{L_y}} \cos(\lambda_j y), & j = 1, 2, \dots \end{cases}$$

Zweitens, da die Zustandsgröße $\underline{T}_{i,j}[k]$ und die Temperaturverteilung $f_k(x, y)$ in den 2D-Fall Matrizen sind, müssen wir eine Nummerierung durchführen, um die Matrizen in Vektoren umzuformen, also

$$\begin{aligned}\mathbf{T} \in \mathbb{R}^{G_x \times G_y} &\mapsto \underline{T} \in \mathbb{R}^{G_x G_y \times 1}, \\ \mathbf{f} \in \mathbb{R}^{N_x \times N_y} &\mapsto \underline{f} \in \mathbb{R}^{N_x N_y \times 1},\end{aligned}$$

durch

$$\begin{aligned}\underline{T}_{index} &= \mathbf{T}_{i,j}, \text{ mit } index := i + (j - 1)G_x, \\ \underline{f}_{index} &= \mathbf{f}_{i,j}, \text{ mit } index := i + (j - 1)N_x,\end{aligned}$$

wie in der folgende Abbildung gezeigt.

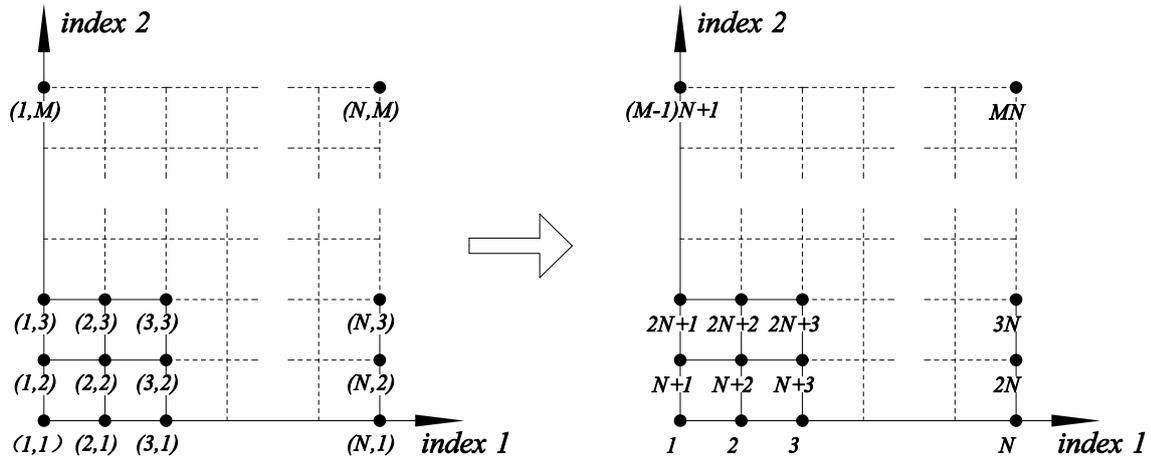


Abbildung 2.1: Nummerierung.

Dann können mit analogem Verfahren das Systemmodell in den 2D-Fall herleiten, also

$$\begin{aligned}\underline{T}_{k+1} &= \mathbf{A} \cdot \underline{T}_k + \underline{U}_k, \\ \underline{f}_{k+1} &= \mathbf{\Psi} \cdot \underline{T}_{k+1},\end{aligned}$$

wobei

$$\begin{aligned}\underline{T}_k &= \left[T_1[K] \quad \cdots \quad T_{G_x G_y}[K] \right]^T \\ &= \left[T_{1,1}[K] \quad \cdots \quad T_{G_x,1}[K] \quad \cdots \quad T_{1,G_y}[K] \quad \cdots \quad T_{G_x,G_y}[K] \right]^T, \\ \mathbf{A} &= \begin{bmatrix} 1 - \kappa_{1,1} & & \underline{0} \\ & \ddots & \\ \underline{0} & & 1 - \kappa_{G_x, G_y} \end{bmatrix}, \\ \underline{U}_k &= \Delta t \cdot \left[U_1[K] \quad \cdots \quad U_{G_x G_y}[K] \right]^T \\ &= \Delta t \cdot \left[U_{1,1}[K] \quad \cdots \quad U_{G_x,1}[K] \quad \cdots \quad U_{1,G_y}[K] \quad \cdots \quad U_{G_x,G_y}[K] \right]^T, \\ \mathbf{\Psi} &= \begin{bmatrix} \psi_1[1]\varphi_1[1] & \cdots & \psi_{G_x}[1]\varphi_1[1] & \cdots & \psi_1[1]\varphi_{G_y}[1] & \cdots & \psi_{G_x}[1]\varphi_{G_y}[1] \\ \vdots & \ddots & \vdots & & \vdots & \ddots & \vdots \\ \psi_1[N_x]\varphi_1[1] & \cdots & \psi_{G_x}[N_x]\varphi_1[1] & \cdots & \psi_1[N_x]\varphi_{G_y}[1] & \cdots & \psi_{G_x}[N_x]\varphi_{G_y}[1] \\ \vdots & & \vdots & \ddots & \vdots & & \vdots \\ \psi_1[1]\varphi_1[N_y] & \cdots & \psi_{G_x}[1]\varphi_1[N_y] & \cdots & \psi_1[1]\varphi_{G_y}[N_y] & \cdots & \psi_{G_x}[1]\varphi_{G_y}[N_y] \\ \vdots & \ddots & \vdots & & \vdots & \ddots & \vdots \\ \psi_1[N_x]\varphi_1[N_y] & \cdots & \psi_{G_x}[N_x]\varphi_1[N_y] & \cdots & \psi_1[N_x]\varphi_{G_y}[N_y] & \cdots & \psi_{G_x}[N_x]\varphi_{G_y}[N_y] \end{bmatrix}\end{aligned}$$

$$= \underbrace{\begin{bmatrix} \varphi_1[1] & \cdots & \varphi_{G_y}[1] \\ \vdots & \ddots & \vdots \\ \varphi_1[N_y] & \cdots & \varphi_{G_y}[N_y] \end{bmatrix}}_{\Phi_y} \otimes \underbrace{\begin{bmatrix} \psi_1[1] & \cdots & \psi_{G_x}[1] \\ \vdots & \ddots & \vdots \\ \psi_1[N_x] & \cdots & \psi_{G_x}[N_x] \end{bmatrix}}_{\Psi_x} .$$

Hierbei steht das Zeichen „ \otimes “ für das Kronecker-Produkt, mit dem wir die Messabbildung Ψ klarer und einfacher beschreiben können.

2.2.2 Finite-Differenzen-Verfahren

Bei dem Finite-Differenzen-Verfahren wandeln wir die partielle Differenzialgleichung mithilfe von der Taylorentwicklung in ein lineares Gleichungssystem (LGS) um. Die Differenzialgleichungen werden dann durch das Lösen des linearen Gleichungssystems gelöst. Die Approximationen sind

$$\begin{aligned} \frac{\partial f_n^k}{\partial t} &\approx \frac{f_n^k - f_n^{k-1}}{\Delta t} , \\ \frac{\partial^2 f_n^k}{\partial x^2} &\approx \frac{f_{n-1}^k - 2f_n^k + f_{n+1}^k}{\Delta x^2} . \end{aligned}$$

Dann können wir für jeden Knoten eine lineare Gleichung

$$-f_n^k + \underbrace{\frac{\Delta x^2}{\Delta x^2 + 2k\Delta t}}_{=:A} f_n^{k-1} + \underbrace{\frac{k\Delta t}{\Delta x^2 + 2k\Delta t}}_{=:B} f_{n-1}^k + \underbrace{\frac{k\Delta t}{\Delta x^2 + 2k\Delta t}}_{=:B} f_{n+1}^k = \underbrace{\frac{\Delta x^2 \Delta t}{\Delta x^2 + 2k\Delta t}}_{=:C} u_n^k$$

stellen. Und schließlich bekommen wir ein LGS

$$\underbrace{\begin{bmatrix} 1 & & & & & & & & & & \\ & \ddots & & & & & & & & & \\ & & 1 & & & & & & & & \\ & & & -1 & 1 & & & & & & \\ & A & & B & -1 & B & & & & & \\ & & \ddots & & \ddots & \ddots & \ddots & & & & \\ & & & A & & B & -1 & B & & & \\ & & & & & & 1 & -1 & & & \end{bmatrix}}_{=:H} \cdot \underbrace{\begin{bmatrix} f_1^1 \\ \vdots \\ f_N^1 \\ f_1^2 \\ f_2^2 \\ \vdots \\ f_{N-1}^K \\ f_N^K \end{bmatrix}}_f = \underbrace{\begin{bmatrix} f_1^1 \\ \vdots \\ f_N^1 \\ 0 \\ Cu_2^2 \\ \vdots \\ Cu_{N-1}^{K-1} \\ 0 \end{bmatrix}}_{=:b} .$$

Nachdem wir das LGS gelöst haben, bekommen wir die Temperaturverteilung. Und im 2D-Fall ist das Verfahren auch so. Die Approximationen für den 2D-Fall sind

also

$$\begin{aligned}\frac{\partial f_{m,n}^k}{\partial t} &\approx \frac{f_{m,n}^k - f_{m,n}^{k-1}}{\Delta t}, \\ \frac{\partial^2 f_{m,n}^k}{\partial x^2} &\approx \frac{f_{m-1,n}^k - 2f_{m,n}^k + f_{m+1,n}^k}{\Delta x^2}, \\ \frac{\partial^2 f_{m,n}^k}{\partial y^2} &\approx \frac{f_{m,n-1}^k - 2f_{m,n}^k + f_{m,n+1}^k}{\Delta y^2}.\end{aligned}$$

Dann können wir für jeden Knoten eine lineare Gleichung

$$-f_{m,n}^k + Af_{m,n}^{k-1} + Bf_{m,n-1}^k + Bf_{m,n+1}^k + Cf_{m-1,n}^k + Cf_{m+1,n}^k = Du_{m,n}^k,$$

mit

$$\begin{aligned}A &= \frac{\Delta x^2 \Delta y^2}{\Delta x^2 \Delta y^2 + 2k \Delta t \Delta x^2 + 2k \Delta t \Delta y^2}, \quad B = \frac{k \Delta t \Delta x^2}{\Delta x^2 \Delta y^2 + 2k \Delta t \Delta x^2 + 2k \Delta t \Delta y^2}, \\ C &= \frac{k \Delta t \Delta y^2}{\Delta x^2 \Delta y^2 + 2k \Delta t \Delta x^2 + 2k \Delta t \Delta y^2}, \quad D = -\frac{\Delta t \Delta x^2 \Delta y^2}{\Delta x^2 \Delta y^2 + 2k \Delta t \Delta x^2 + 2k \Delta t \Delta y^2}.\end{aligned}$$

erstellen. Nach der Nummerierung und nachdem die Gleichungen in Matrizenform umgeformt und gelöst werden, erhalten wir schließlich die Temperaturverteilung in den 2D-Fall.

2.3 Simulationen

Nach der theoretische Herleitung simulieren wir die Temperaturverteilung in einigen Fällen.

2.3.1 1D-Fall

Als das erste Beispiel für den homogenen Fall wird eine sinusförmige Anfangsbedingung genommen, wobei die Länge des Stabs $L = 10$ cm und die Wärmeleitfähigkeit $k = 0.1$ cm²/s sind. Die Ergebnisse aus der Modalanalyse und der FDM werden in der Abbildung 2.2 gezeigt. Der Grund für die unterschiedliche Aussehen besteht darin, dass die Abstände der Knoten von der MA und der FDM unterschiedlich sind.

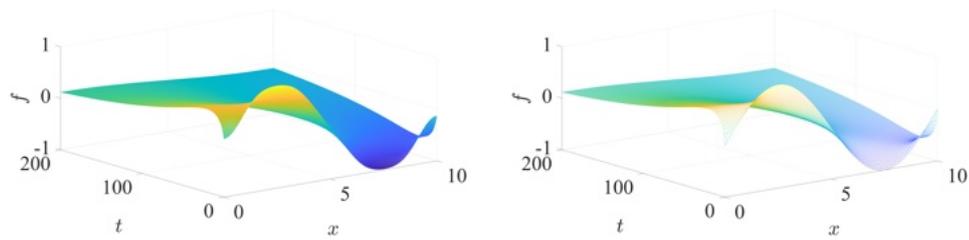


Abbildung 2.2: Temperaturverteilungen mit sinusförmiger Anfangsbedingung, links: MA mit $\Delta x = 0.01$ cm, $\Delta t = 0.1$ s, rechts: FDM mit $\Delta x = 0.1$ cm, $\Delta t = 1$ s.

Als das zweite Beispiel für den homogenen Fall wird eine treppenförmige Anfangsbedingung genommen. Die Ergebnisse aus der Modalanalyse und der FDM werden in der Abbildung 2.3 gezeigt.

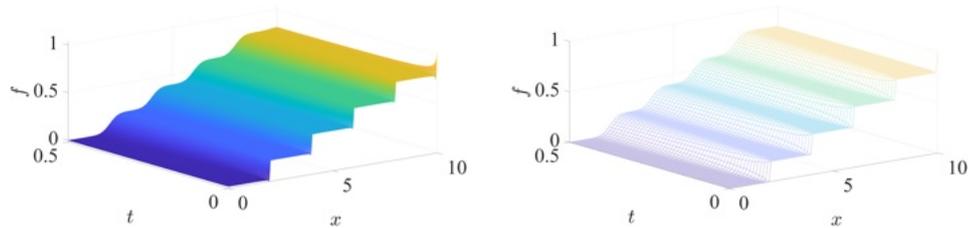


Abbildung 2.3: Temperaturverteilungen mit treppenförmiger Anfangsbedingung, links: MA mit $\Delta x = 0.001$ cm, $\Delta t = 2.5 \times 10^{-4}$ s, rechts: FDM mit $\Delta x = 0.1$ cm, $\Delta t = 0.01$ s.

Für den inhomogenen Fall haben werden auch zwei Fälle simuliert, jeweils mit Wärmeabstrahlung

$$u_{WA}(x, t) = -a \cdot f(x, t) ,$$

und punktförmiger Anregung

$$u_{PA}(x, t) = \sum_i u_i(t) \cdot \delta(x - x_i) .$$

Die Anfangsbedingung wird auch als die sinusförmige Anfangsbedingung angenommen. Die Ergebnisse aus der Modalanalyse und der FDM mit $a = 0.1$ werden in der Abbildung 2.4 gezeigt.

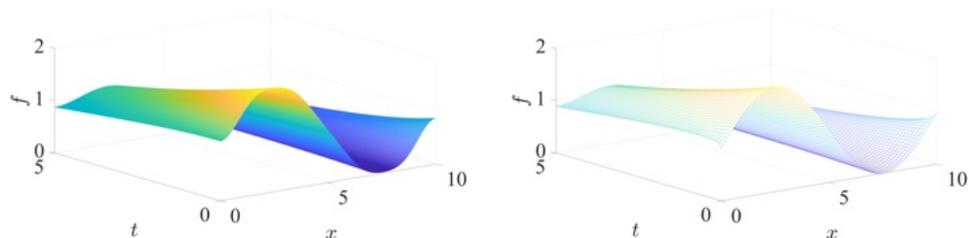


Abbildung 2.4: Temperaturverteilungen mit sinusförmiger Anfangsbedingung, links: MA mit $\Delta x = 0.01$ cm, $\Delta t = 0.01$ s, rechts: FDM mit $\Delta x = 0.1$ cm, $\Delta t = 0.1$ s.

Als das zweite Beispiel wird punktförmige Anregung genommen. Die Ergebnisse aus der Modalanalyse und der FDM mit

$$u(x, t) = 0.1 \sin\left(t - \frac{\pi}{4}\right) \cdot \delta(x - 3) - 0.2 \sin(t) \cdot \delta(x - 5) + 0.01t \cdot \delta(x - 7)$$

werden in der Abbildung 2.5 gezeigt.

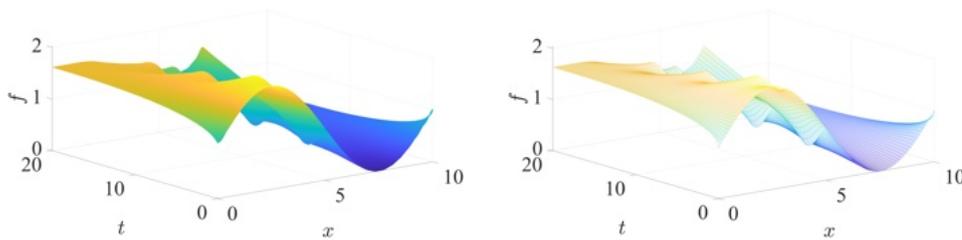


Abbildung 2.5: Temperaturverteilungen mit sinusförmiger Anfangsbedingung, links: MA mit $\Delta x = 0.01$ cm, $\Delta t = 0.01$ s, rechts: FDM mit $\Delta x = 0.1$ cm, $\Delta t = 0.1$ s.

2.3.2 2D-Fall

In den 2D-Fall wird sinusförmige Anfangsbedingung mit $L_x = 10$ cm, $L_y = 20$ cm, $k = 0.1$ cm²/s genommen. Die Ergebnisse werden in folgenden Abbildungen gezeigt.

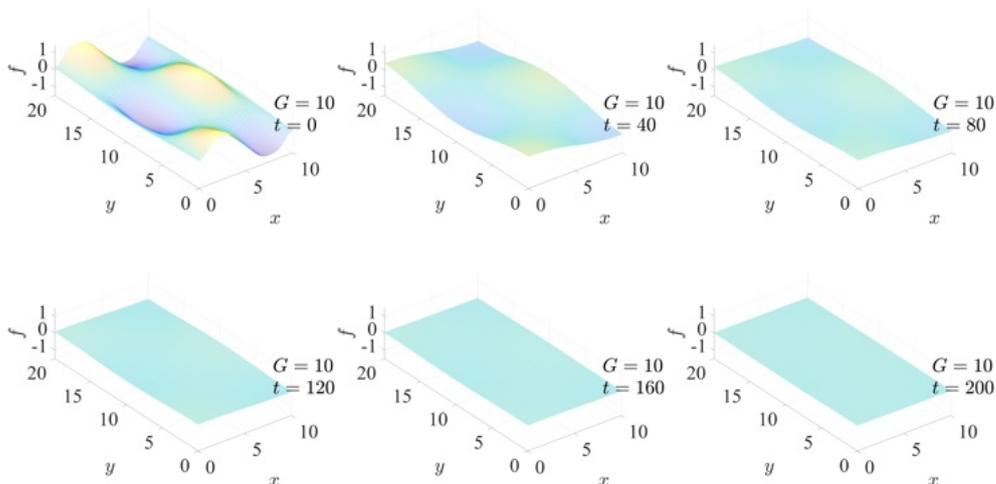


Abbildung 2.6: 2D-Temperaturverteilung aus der MA mit $\Delta x = 0.1$ cm, $\Delta y = 0.2$ cm, $\Delta t = 1$ s.

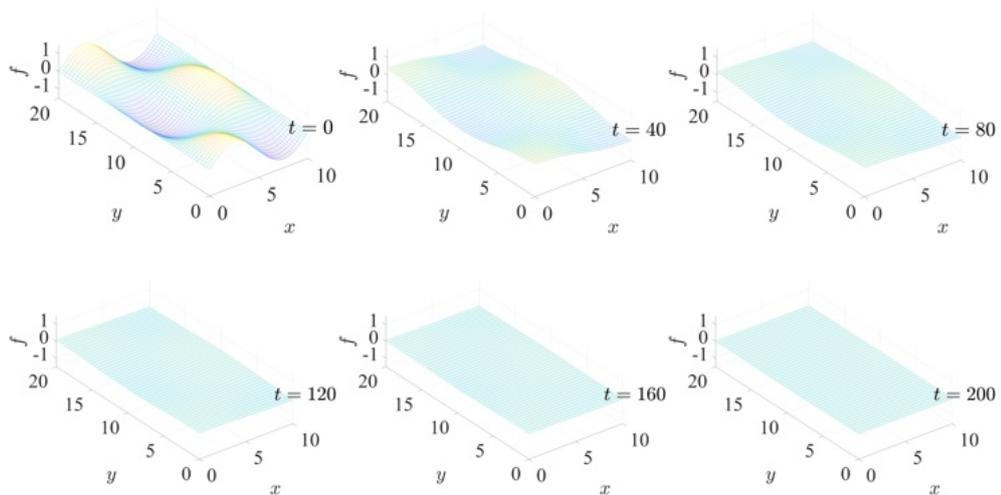


Abbildung 2.7: 2D-Temperaturverteilung aus der FDM mit $\Delta x = 0.2$ cm, $\Delta y = 0.4$ cm, $\Delta t = 5$ s.

Für den inhomogenen Fall werden auch zwei Fälle simuliert, jeweils mit Wärmeabstrahlung

$$u_{WA}(x, y, t) = -a \cdot f(x, y, t) ,$$

und punktförmiger Anregung

$$u_{PA}(x, y, t) = \sum_i u_{i,j}(t) \cdot \delta(x - x_i) \cdot \delta(y - y_j) .$$

Die Ergebnisse aus der Modalanalyse und der FDM mit $a = 0.1$ werden in den folgenden Abbildungen gezeigt.

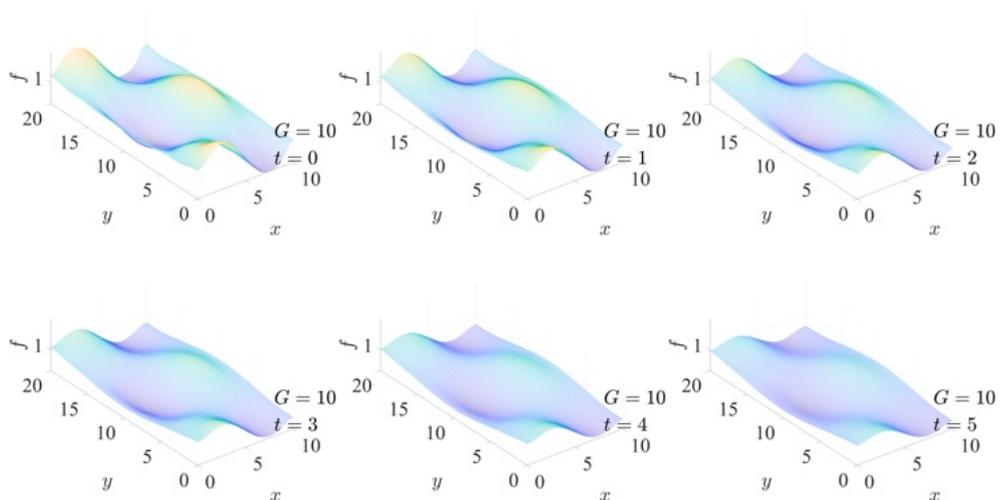


Abbildung 2.8: 2D-Temperaturverteilung aus der MA mit $\Delta x = 0.1$ cm, $\Delta y = 0.2$ cm, $\Delta t = 0.1$ s.

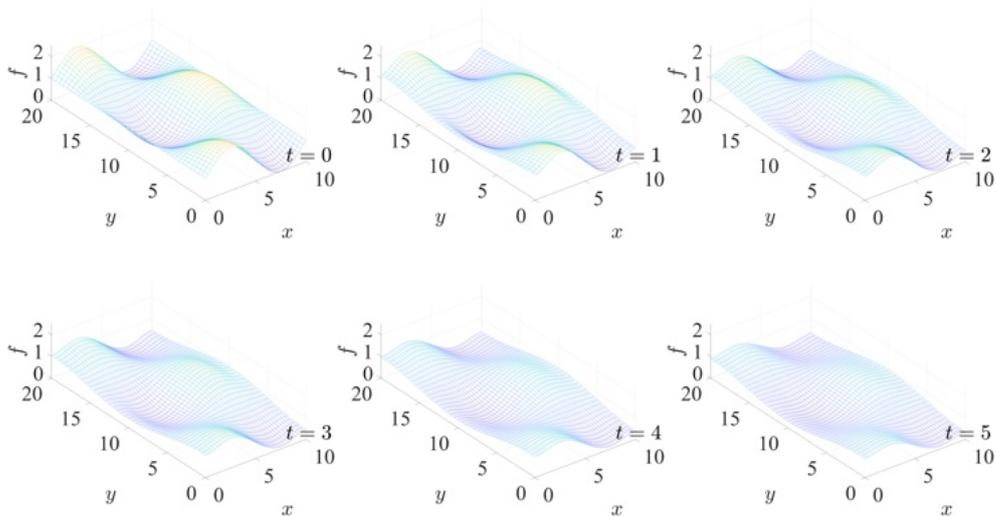


Abbildung 2.9: 2D-Temperaturverteilung aus der FDM mit $\Delta x = 0.25$ cm, $\Delta y = 0.5$ cm, $\Delta t = 0.1$ s.

Als das zweite Beispiel werden punktförmige Anregungen angenommen, wobei

$$\begin{aligned}
 u(x, y, t) = & \sin\left(t - \frac{\pi}{4}\right) \cdot \delta(x - 3) \cdot \delta(y - 3) \\
 & - 2 \sin(t) \cdot \delta(x - 5) \cdot \delta(y - 10) \\
 & + 0.1t \cdot \delta(x - 7) \cdot \delta(y - 13) \\
 & + 0.3t \cdot \delta(x - 8) \cdot \delta(y - 18) .
 \end{aligned}$$

Die Ergebnisse aus der Modalanalyse und der FDM werden in den folgenden Abbildungen gezeigt.

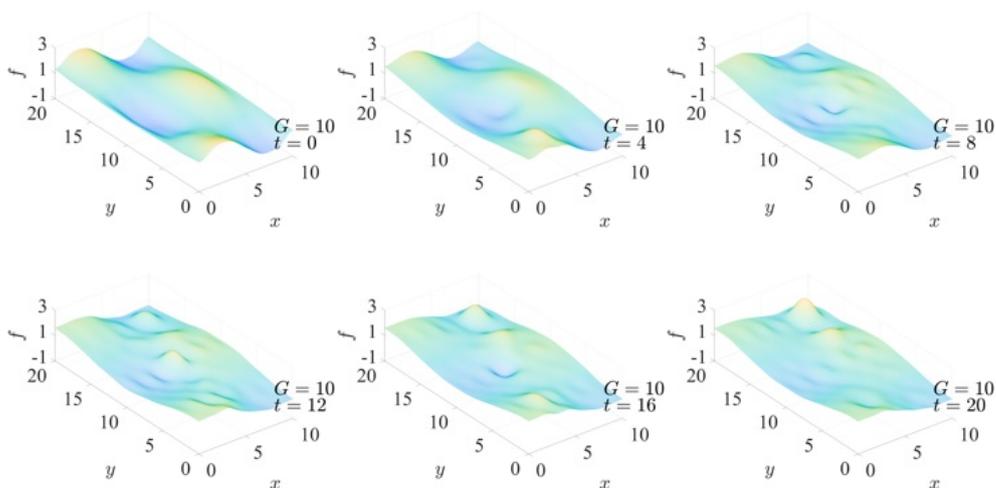


Abbildung 2.10: 2D-Temperaturverteilung aus der MA mit $\Delta x = 0.1$ cm, $\Delta y = 0.2$ cm, $\Delta t = 0.4$ s.

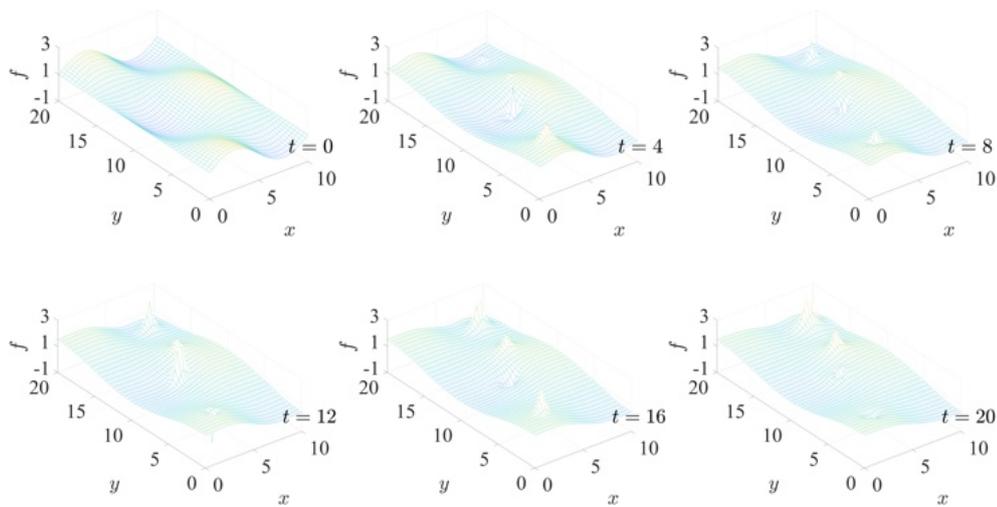


Abbildung 2.11: 2D-Temperaturverteilung aus der FDM mit $\Delta x = 0.25$ cm, $\Delta y = 0.5$ cm, $\Delta t = 0.4$ s.

2.4 Zusammenfassung

In diesem Kapitel haben wir das Modell für die Wärmeleitung vorgestellt, die numerische Lösung aus zwei Ansätzen (Modalanalyse und FDM) hergeleitet und die Temperaturverteilung in den 1D- und 2D-Fällen mit verschiedenen Anfangsbedingungen und Anregungen simuliert. Im Vergleich zur FDM hat die Modalanalyse den Vorteil, dass sie weniger Rechenaufwand erfordert. Der Nachteil ist jedoch, dass sie an spitz-förmigen Teilen nicht gut funktioniert. Umgekehrt präsentiert die FDM die Spitzen sehr klar, weil es nicht von den Eigenfunktionen beeinflusst wird. Da die FDM viel mehr Rechenaufwand erfordert als die Modalanalyse, muss die FDM einen großen Knotenabstand haben, dann ist die Auflösung bei der FDM schlechter als die MA.

Nach dem Erstellen der Testfunktionen, also die Temperaturverteilung, werden wir im nächsten Kapitel das Kalman-Filter vorstellen, und wir werden auch mit dem Kalman-Filter die Temperaturverteilung, die wir in diesem Kapitel erstellt, rekonstruieren, weil das Kalman-Filter einer der wichtigsten Grundlagen für unsere Arbeit (Kalman-Filtered-Compressive-Sensing) ist.

KAPITEL 3

Kalman-Filter

Das Kalman-Filter ist ein Zustandsschätzer [13; 14]. Es arbeitet mit einer Prozess- und Signaldarstellung im Zustandsraum und bildet den Prozess- und Signalzustand über eines lineares Systemmodell nach. Dabei wird ein neuer Schätzwert aus dem alten Schätzwert und den aktuell gemessenen Ausgangswerten rekursiv berechnet. Als Optimierungskriterium wird die Minimierung des Erwartungswertes der quadratischen Norm des Schätzfehlers verwendet. Ein Kalman-Filter besteht aus zwei Schritten, Prädiktion und Filterung.

Es wird so angenommen, dass das Systemmodell

$$\begin{aligned}\underline{x}_{k+1} &= \mathbf{A} \cdot \underline{x}_k + \mathbf{B} \cdot u_k + \underline{w}_k , \\ \underline{y}_{k+1} &= \mathbf{H}_{k+1} \cdot \underline{x}_{k+1} + \underline{v}_{k+1} ,\end{aligned}$$

bekannt ist. Die Systemrauschen \underline{w}_k und die Messrauschen \underline{v}_k sind Gauß'sche Verteilungen mit

$$\begin{aligned}E \{ \underline{w}_k \} &= 0, & \text{Cov} \{ \underline{w}_i, \underline{w}_j \} &= \mathbf{C}_k^w \cdot \delta_{i,j} , \\ E \{ \underline{v}_k \} &= 0, & \text{Cov} \{ \underline{v}_i, \underline{v}_j \} &= \mathbf{C}_k^v \cdot \delta_{i,j} .\end{aligned}$$

Dann ist der Prädiktionsschritt

$$\begin{aligned}\hat{\underline{x}}_{k+1}^p &= \mathbf{A}_k \cdot \hat{\underline{x}}_k^e + \mathbf{B}_k \cdot u_k , \\ \mathbf{C}_{k+1}^p &= \mathbf{A}_k \cdot \mathbf{C}_k^e \cdot \mathbf{A}_k^T + \mathbf{C}_k^w .\end{aligned}$$

Und der Filterungsschritt

$$\begin{aligned}\hat{\underline{x}}_{k+1}^e &= (\mathbf{I} - \mathbf{K}_{k+1} \cdot \mathbf{H}_{k+1}) \cdot \hat{\underline{x}}_{k+1}^p + \mathbf{K}_{k+1} \cdot \underline{y}_{k+1} , \\ \mathbf{C}_{k+1}^e &= (\mathbf{I} - \mathbf{K}_{k+1} \cdot \mathbf{H}_{k+1}) \cdot \mathbf{C}_{k+1}^p ,\end{aligned}$$

mit

$$\mathbf{K}_{k+1} = \mathbf{C}_{k+1}^p \cdot \mathbf{H}_{k+1}^T \cdot (\mathbf{H}_{k+1} \cdot \mathbf{C}_{k+1}^p \cdot \mathbf{H}_{k+1}^T + \mathbf{C}_{k+1}^v)^{-1} .$$

Die konkrete Herleitung wird in den Anhang aufgeführt.

3.1 Kalman-Filter für die Modalanalyse

In vielen Fällen ist die Anregung unbekannt, z. B. um die Temperaturverteilung eines Waldes zu erfassen, um einen Brand zu verhindern. In diesem Fall können wir die Anregung als Systemrauschen behandeln. In den folgenden Simulationen werden wir die folgenden Annahmen haben, dass

- die Anregung unbekannt ist,
- das System rauschfrei ist,
- die Messungen rauschenbehaftet sind.

Im Kapitel 2 haben wir das Systemmodell

$$\begin{aligned}\underline{T}_{k+1} &= \mathbf{A} \cdot \underline{T}_k + \underline{w}_k, \\ \underline{f}_{k+1} &= \mathbf{\Psi} \cdot \underline{T}_{k+1} + \underline{v}_k\end{aligned}$$

mit

$$\begin{aligned}E\{\underline{w}_k\} &= \underline{0}, & \text{Cov}\{\underline{w}_i, \underline{w}_j\} &= \mathbf{C}_k^w \cdot \delta_{i,j} \\ E\{\underline{v}_k\} &= \underline{0}, & \text{Cov}\{\underline{v}_i, \underline{v}_j\} &= \mathbf{C}_k^v \cdot \delta_{i,j}\end{aligned}$$

hergeleitet. Aber es ist unmöglich, dass wir die Temperatur an allen Knoten messen. Wir messen die Temperatur nur an M äquidistanten Stellen, also

$$x_{\lfloor i \frac{N-1}{M} + 1 \rfloor}, \quad i = 1, \dots, M.$$

Somit erstellen wir eine Delta-Matrix $\mathbf{\Phi} \in \mathbb{R}^{M \times N}$, wobei

$$\mathbf{\Phi}_{i,j} = \begin{cases} 1, & j = \lfloor i \frac{N-1}{M} + 1 \rfloor, \\ 0, & \text{sonst.} \end{cases}$$

Deswegen ist die Messabbildung im Kalman-Filter

$$\underbrace{\mathbf{\Phi} \underline{f}_{k+1}}_{=: \underline{y}_{k+1}} = \underbrace{\mathbf{\Phi} \mathbf{\Psi}}_{=: \mathbf{H}} \cdot \underline{T}_{k+1} + \underbrace{\mathbf{\Phi} \underline{v}_{k+1}}_{=: \tilde{\underline{v}}_{k+1}}.$$

Dann ergibt sich das Modell für das Kalman-Filter im 2D-Fall aus der Modalanalyse, also

$$\begin{aligned}\underline{T}_{k+1} &= \mathbf{A} \cdot \underline{T}_k + \underline{w}_k, \\ \underline{y}_{k+1} &= \mathbf{H} \cdot \underline{T}_{k+1} + \tilde{\underline{v}}_{k+1}.\end{aligned}$$

Dann ist das Verfahren des Kalman-Filters.

- Prädiktion

$$\begin{aligned}\hat{\underline{T}}_{k+1}^p &= \mathbf{A} \cdot \hat{\underline{T}}_k^e, \\ \mathbf{C}_{k+1}^p &= \mathbf{A} \cdot \mathbf{C}_k^e \cdot \mathbf{A}^T + \mathbf{C}_k^w.\end{aligned}$$

- Filterung

$$\begin{aligned}\hat{\underline{T}}_{k+1}^e &= (\mathbf{I} - \mathbf{K}_{k+1} \cdot \mathbf{H}) \cdot \hat{\underline{T}}_{k+1}^p + \mathbf{K}_{k+1} \cdot \underline{y}_{k+1}, \\ \mathbf{C}_{k+1}^e &= (\mathbf{I} - \mathbf{K}_{k+1} \cdot \mathbf{H}) \cdot \mathbf{C}_{k+1}^p,\end{aligned}$$

mit der Verstärkungsmatrix

$$\mathbf{K}_{k+1} = \mathbf{C}_{k+1}^p \cdot \mathbf{H}^T \cdot (\mathbf{H} \cdot \mathbf{C}_{k+1}^p \cdot \mathbf{H}^T + \mathbf{C}_{k+1}^{\tilde{v}})^{-1}.$$

Analog können wir das Kalman-Filter für den 2D-Fall herleiten. Wählen wir zuerst $M_x \times M_y$ äquidistanten Sensorstellen, dann nummerieren wir sie in einen Vektor \underline{S} mit $\underline{S} \in \mathbb{R}^{M_x M_y \times 1}$. Dann können wir mit gleichem Verfahren eine Delta-Matrix $\Phi \in \mathbb{R}^{M_x M_y \times N_x N_y}$ erstellen. Danach ist das Kalman-Filter in demselben Form wie das Kalman-Filter in den 1D-Fall.

3.2 Kalman-Filter für die FDM

Zunächst wird das explizite Verfahren betrachtet. Laut Diffusionsmodell und Taylor-Entwicklung erhalten wir

$$\begin{aligned}\frac{f_n^{k+1} - f_n^k}{\Delta t} &= k \frac{f_{n-1}^k - 2f_n^k + f_{n+1}^k}{\Delta x^2} \\ f_n^{k+1} &= \underbrace{\frac{k\Delta t}{\Delta x^2} f_{n-1}^k}_{=:P} + \underbrace{\left(1 - 2\frac{k\Delta t}{\Delta x^2}\right) f_n^k}_{=:Q} + \underbrace{\frac{k\Delta t}{\Delta x^2} f_{n+1}^k}_{=:P} \\ &= P f_{n-1}^k + Q f_n^k + P f_{n+1}^k.\end{aligned}$$

Laut Neumann-Bedingung gilt

$$\begin{aligned}f_1^{k+1} &= f_2^{k+1} = P f_1^k + Q f_2^k + P f_3^k, \\ f_N^{k+1} &= f_{N-1}^{k+1} = P f_{N-2}^k + Q f_{N-1}^k + P f_N^k.\end{aligned}$$

In Matrixschreibweise ist es

$$\underbrace{\begin{bmatrix} f_1^{k+1} \\ f_2^{k+1} \\ \vdots \\ f_{N-1}^{k+1} \\ f_N^{k+1} \end{bmatrix}}_{=: \underline{f}_{k+1}} = \underbrace{\begin{bmatrix} P & Q & P & & \\ P & Q & P & & \\ & \ddots & \ddots & \ddots & \\ & & P & Q & P \\ & & P & Q & P \end{bmatrix}}_{=: \mathbf{D}} \cdot \underbrace{\begin{bmatrix} f_1^k \\ f_2^k \\ \vdots \\ f_{N-1}^k \\ f_N^k \end{bmatrix}}_{=: \underline{f}_k}.$$

Für die Messabbildung ist es analog wie in der Modalanalyse, also

$$\underline{y}_{k+1} = \Phi \cdot \underline{f}_{k+1} + \underbrace{\Phi \cdot \underline{v}_{k+1}}_{=: \tilde{\underline{v}}_{k+1}} .$$

Allerdings muss Q die folgende Eigenschaft erfüllen, um die Stabilitätsbedingung [15] zu halten. D. h.

$$Q = 1 - 2 \frac{k \Delta t}{\Delta x^2} > \frac{1}{2} ,$$

also

$$\Delta t < \frac{\Delta x^2}{4k} .$$

Bei z. B. $\Delta x = 0.01$, $k = 0.1$ muss

$$\Delta t < 2.5 \times 10^{-4}$$

halten. Das ist aber nicht günstig und manchmal unmöglich. Deswegen ist ein implizites Verfahren nötig. Ein implizites Verfahren ist

$$\frac{f_n^{k+1} - f_n^k}{\Delta t} = k \frac{f_{n-1}^{k+1} - 2f_n^{k+1} + f_{n+1}^{k+1}}{\Delta x^2} ,$$

$$-P f_{n-1}^{k+1} + \underbrace{(1 + 2P)}_{=: Q} f_n^{k+1} - P f_{n+1}^{k+1} = f_n^k .$$

Laut Neumann-Bedingung gilt

$$f_1^k = f_2^k = -P f_1^{k+1} + (1 + 2P) f_2^{k+1} - P f_3^{k+1} = (1 + P) f_1^{k+1} - P f_3^{k+1} ,$$

$$f_N^k = f_{N-1}^k = -P f_{N-2}^{k+1} + (1 + 2P) f_{N-1}^{k+1} - P f_N^{k+1} = -P f_{N-2}^{k+1} + (1 + P) f_N^{k+1} .$$

In Matrixschreibweise ist es

$$\underbrace{\begin{bmatrix} 1 + P & 0 & -P & & & \\ -P & Q & -P & & & \\ & \ddots & \ddots & \ddots & & \\ & & -P & Q & -P & \\ & & -P & 0 & 1 + P & \end{bmatrix}}_{=: \mathcal{D}^{-1}} \cdot \underbrace{\begin{bmatrix} f_1^{k+1} \\ f_2^{k+1} \\ \vdots \\ f_{N-1}^{k+1} \\ f_N^{k+1} \end{bmatrix}}_{=: \underline{f}_{k+1}} = \underbrace{\begin{bmatrix} f_1^k \\ f_2^k \\ \vdots \\ f_{N-1}^k \\ f_N^k \end{bmatrix}}_{=: \underline{f}_k} .$$

Dann ergibt sich das Modell aus der implizite FDM, also

$$\underline{f}_{k+1} = \mathcal{D} \cdot \underline{f}_k + \underline{w}_k ,$$

$$\underline{y}_{k+1} = \Phi \cdot \underline{f}_k + \tilde{\underline{v}}_{k+1} .$$

Das Kalman-Filter läuft dann die folgende Schritte durch, nämlich

- Prädiktion

$$\begin{aligned}\hat{\underline{f}}_{k+1}^p &= \mathcal{D} \cdot \hat{\underline{f}}_k^e, \\ \mathbf{C}_{k+1}^p &= \mathcal{D} \cdot \mathbf{C}_k^e \cdot \mathcal{D}^T + \mathbf{C}_k^w,\end{aligned}$$

- und Filterung

$$\begin{aligned}\hat{\underline{f}}_{k+1}^e &= (\mathbf{I} - \mathbf{K}_{k+1} \cdot \Phi) \cdot \hat{\underline{f}}_{k+1}^p + \mathbf{K}_{k+1} \cdot \underline{y}_{k+1}, \\ \mathbf{C}_{k+1}^e &= (\mathbf{I} - \mathbf{K}_{k+1} \cdot \Phi) \cdot \mathbf{C}_{k+1}^p,\end{aligned}$$

mit der Verstärkungsmatrix

$$\mathbf{K}_{k+1} = \mathbf{C}_{k+1}^p \cdot \Phi^T \cdot (\Phi \cdot \mathbf{C}_{k+1}^p \cdot \Phi^T + \mathbf{C}_{k+1}^{\tilde{v}})^{-1}.$$

Analog können wir das Modell für den 2D-Fall herleiten. Beim expliziten Verfahren gilt

$$f_{m,n}^{k+1} = Q f_{m,n}^k + A f_{m-1,n}^k + A f_{m+1,n}^k + B f_{m,n-1}^k + B f_{m,n+1}^k.$$

Und beim impliziten Verfahren gilt

$$Q f_{m,n}^{k+1} - A f_{m-1,n}^{k+1} - A f_{m+1,n}^{k+1} - B f_{m,n-1}^{k+1} - B f_{m,n+1}^{k+1} = f_{m,n}^k,$$

wobei

$$\begin{aligned}A &= \frac{k\Delta t}{\Delta x^2}, & B &= \frac{k\Delta t}{\Delta y^2}, \\ Q &= 1 - 2A - 2B, & Q &= 1 + 2A + 2B.\end{aligned}$$

Nach der Nummerierung

$$\underline{f}_{m,n}^k \in \mathbb{R}^{N_x \times N_y} \mapsto \underline{f}_k \in \mathbb{R}^{N_x N_y \times 1}$$

können wir dann das Systemmodell in Matrixform schreiben, und das Kalman-Filter verwenden.

3.3 Simulation

Als die Anregung nehmen wir die punktförmige Anregung im Kapitel 1 für den 1D- und 2D-Fall. Außerdem wird die Anfangsbedingung gegeben, d. h. wir nehmen an, dass die sinusförmige Anfangsbedingung bekannt ist. Bei der Simulation wird $\Delta t = 0.1$ s gewählt.

3.3.1 1D-Fall

Wir haben 3 Fälle simuliert, jeweils mit 12, 24, und 36 äquidistanten Sensoren. Beim Kalman-Filter für Modalanalyse werden $\Delta x = 2^{-10}$ cm, $G = M$ gewählt, dann werden die Ergebnisse in folgenden Abbildungen gezeigt.

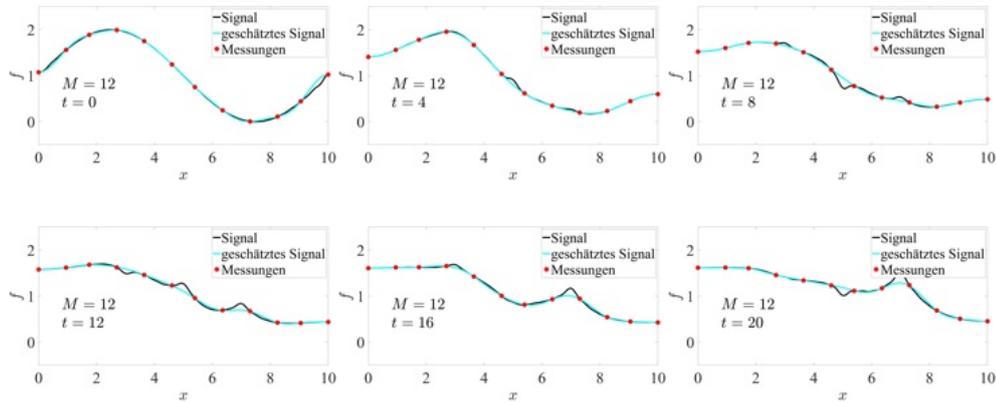


Abbildung 3.1: Kalman-Filter für die Modalanalyse mit 12 äquidistanten Sensoren.

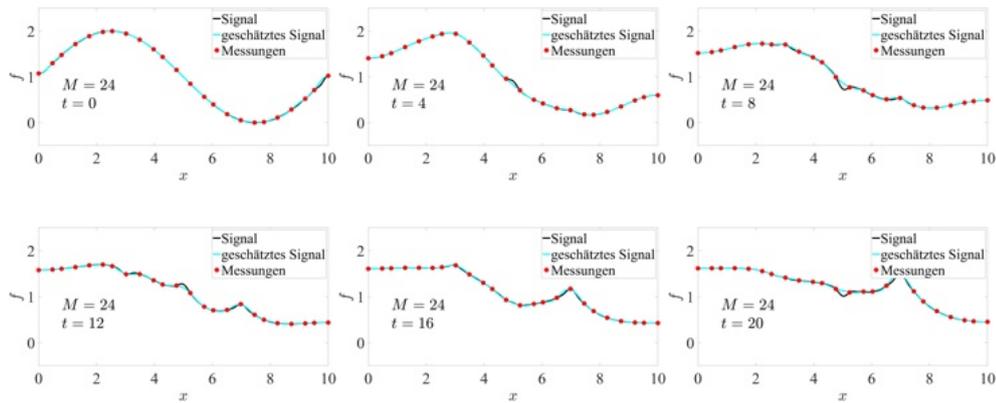


Abbildung 3.2: Kalman-Filter für die Modalanalyse mit 24 äquidistanten Sensoren.

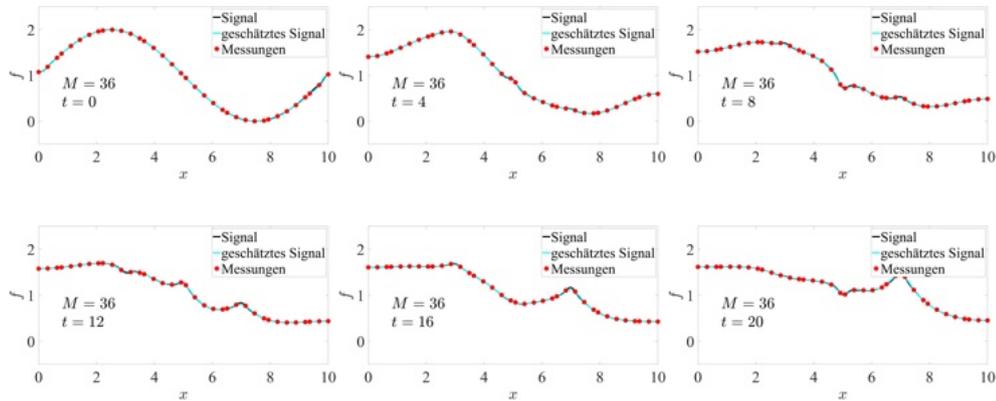


Abbildung 3.3: Kalman-Filter für die Modalanalyse mit 36 äquidistanten Sensoren.

Bei der Signalrekonstruktion aus dem Finite-Differenzen-Verfahren werden $\Delta x = 2^{-10}$ gewählt. Dann ergeben sich die folgenden Ergebnisse.

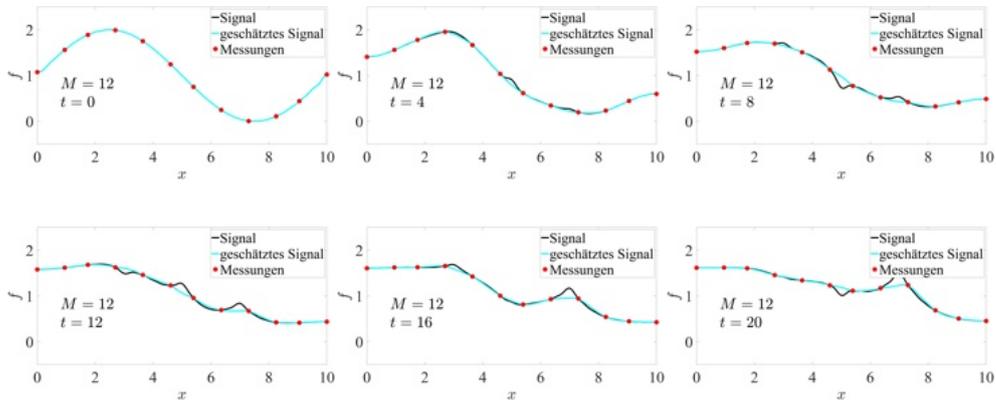


Abbildung 3.4: Kalman-Filter für die FDM mit 12 äquidistanten Sensoren.

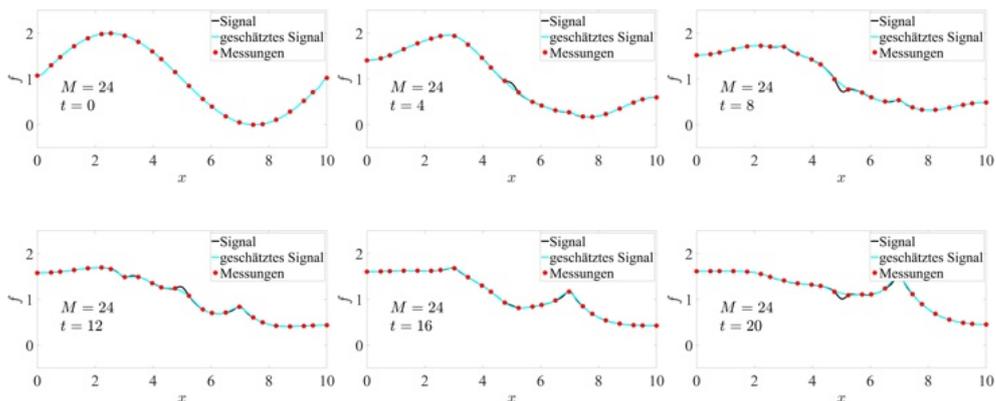


Abbildung 3.5: Kalman-Filter für die FDM mit 24 äquidistanten Sensoren.

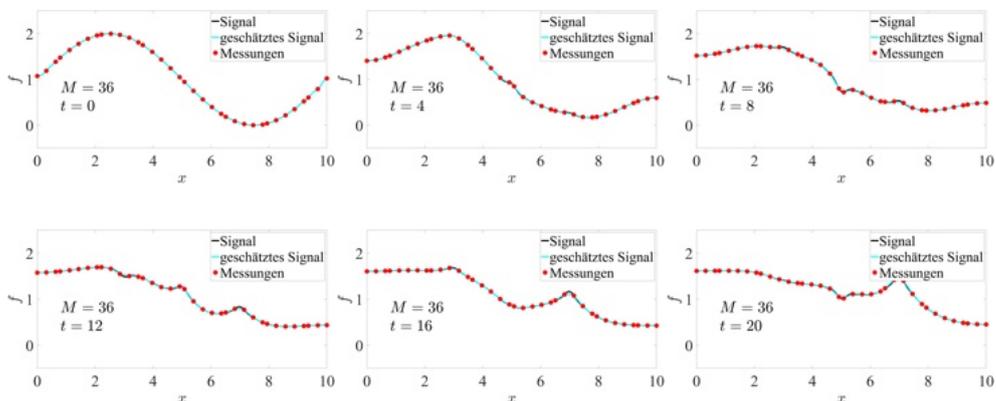


Abbildung 3.6: Kalman-Filter für die FDM mit 36 äquidistanten Sensoren.

3.3.2 2D-Fall

Für 2D-Fall werden $8 \times 8 = 64$ äquidistante Sensoren gewählt. Bei der Signalrekonstruktion aus der Modalanalyse werden $\Delta x = 2^{-7}$ cm, $\Delta y = 2^{-6}$ cm, $G_x = G_y = 8$ gewählt. Dann erhalten wir die Ergebnisse wie folgend.

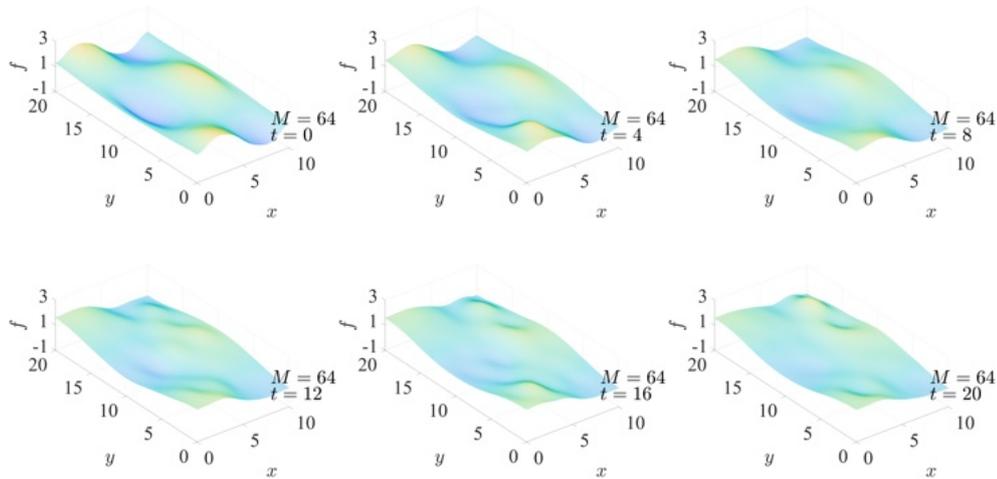


Abbildung 3.7: 2D-alman-Filter für die Modalanalyse mit 64 äquidistante Sensoren.

Und beim Finite-Differenzen-Verfahren wählen wir wegen des Rechenaufwands $\Delta x = 2^{-6}$ cm und $\Delta y = 2^{-5}$ cm.

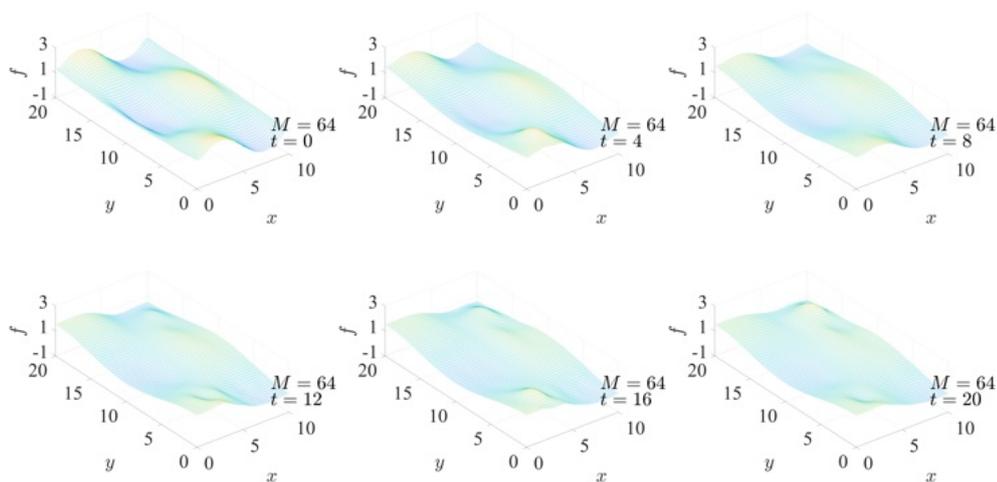


Abbildung 3.8: 2D-Kalman-Filter für die FDM mit 64 äquidistante Sensoren.

3.4 Fehlermaß und Bewertung

Um den Fehler quantitativ zu bewerten, müssen wir ein Fehlermaß bilden. Wir verwenden den relativen RMSE (Engl. Root Mean Square Error) als das Fehlermaß,

also

$$\epsilon(t) = \frac{\|\underline{f}(t) - \hat{\underline{f}}^e(t)\|_2}{\|\underline{f}(t)\|_2},$$

wobei $\underline{f}(t)$ die echte Temperaturverteilung zum Zeitpunkt t ist und $\hat{\underline{f}}^e(t)$ die geschätzte Temperaturverteilung aus dem Kalman-Filter zum Zeitpunkt t ist.

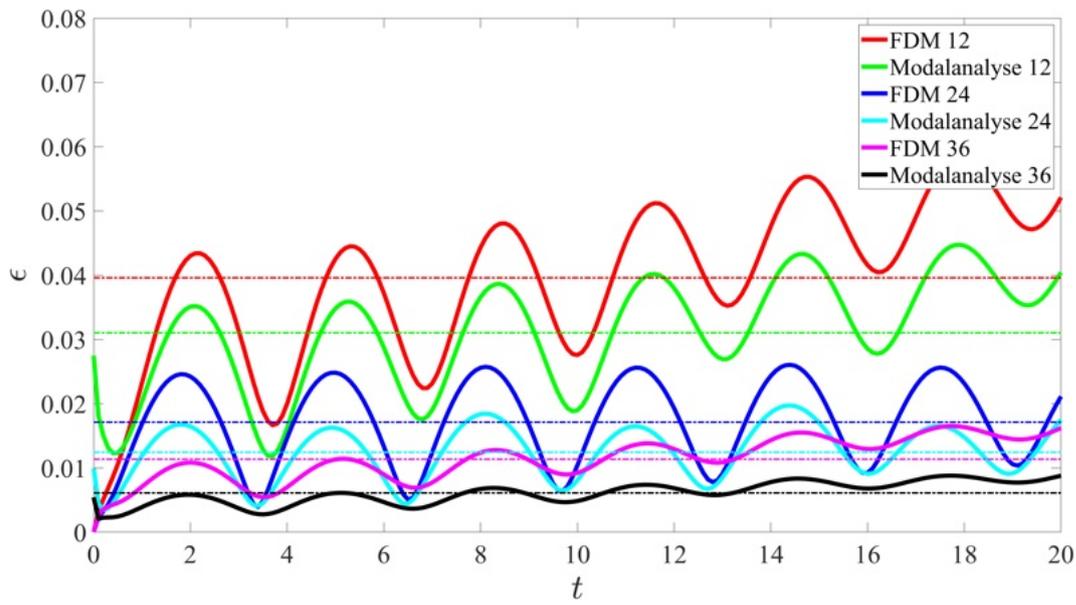


Abbildung 3.9: Vergleich aller Schätzungen.

Nach der Abbildung 3.9 können wir davon ausgehen, dass die Modalanalyse in diesem Fall besser als die FDM funktioniert. Je mehr Sensoren wir benutzen, desto besser ist das Ergebnis. Es ist wichtig zu beachten, dass die FDM eine weitverbreitete Methode ist, um partielle Differenzialgleichungen zu lösen. Daher werden wir in der folgenden Analyse die Ergebnisse des Kalman-Filters mit der FDM mit anderen Verfahren vergleichen.

3.5 Zusammenfassung

In diesem Kapitel haben wir das Kalman-Filter vorgestellt. Es besteht aus zwei Teilen: Prädiktion und Filterung. Im Wesentlichen handelt es sich um eine Informationsfusion durch die Kombination vom Systemmodell und den Messungen zur Zustandsschätzung. Dann haben wir das entsprechenden 1D- und 2D-Kalman-Filter durch das Systemmodell aus der Modalanalyse und der FDM hergeleitet. Nachdem wir sie simuliert haben, analysieren wir den Fehler des Kalman-Filters in diesen Fällen, indem wir ein Fehlermaß festlegen. Wir stellen aus der Abbildung 3.9 fest,

dass die Schätzungen umso genauer sind, je mehr Sensoren verwendet werden. Aufgrund der breiten Anwendung der FDM fokussieren wir uns nach wie vor auf die Ergebnisse aus der FDM statt der Modalanalyse, obwohl die Ergebnisse aus der Modalanalyse ein bisschen besser sind. Im nächsten Kapitel werden wir über das Compressive-Sensing besprechen, das eine der wichtigsten Grundlagen dieser Arbeit darstellt.

Compressive-Sensing

4.1 Einleitung

In der Vergangenheit spielte das Shannon-Abtasttheorem [16] eine Hauptrolle bei der Signalgewinnung. Laut des Theorems muss die Abtastfrequenz größer als das Doppelte der höchsten Signalfrequenz sein. Nach der Signalgewinnung komprimieren wir die Signale mithilfe der Transformation z. B. der Fourier- oder Wavelet-Transformation in verschiedene Basisfunktionen sowie ihre Gewichtungen. Wie in der Abbildung 4.1 gezeigt.



Abbildung 4.1: Ablauf traditioneller Signalverarbeitung.

Mit der kontinuierlichen Erhöhung der Signalbandbreite stehen Abtastung und Speicherung vor großen Herausforderungen. In den letzten Jahren beschäftigen sich immer mehr Wissenschaftler mit dem Compressive-Sensing (CS) [17]. Die grundlegende Idee ist, die Koeffizienten statt nach der Abtastung zu komprimieren, sondern aus der komprimierten Signale direkt zu gewinnen.



Abbildung 4.2: Ablauf des Compressive-Sensing.

Bei dem Compressive-Sensing ist eine Erfüllung der Nyquist-Frequenz nicht mehr nötig, und man macht normalerweise zufällige Abtastungen [18]. Nach der Abtastung bekommt man ein LGS

$$\tilde{y} = \Phi \cdot x + e, \tag{4.1}$$

In der Formel (4.1) steht $\tilde{\underline{y}}$ für die komprimierte Messungen, Φ für die komprimierende Messabbildung, und \underline{e} für den Fehler. Außerdem gilt $\Phi \in \mathbb{R}^M$ mit $M \ll N$, d. h. das ist ein unterbestimmtes Problem. Normalerweise kann man eine eindeutige Lösung dieses Problems nicht finden. Aber wir konvertieren das LGS in ein Optimierungsproblem [21], also

$$\begin{aligned} \underline{x} &= \arg \min_{\underline{x}} \|\underline{x}\|_0 \\ \text{NB : } &\|\tilde{\underline{y}} - \Phi \cdot \underline{x}\|_2 \leq \varepsilon . \end{aligned} \quad (4.2)$$

Wenn die Messabbildung Φ einige Eigenschaften erfüllt, dann gibt es große Möglichkeit, dass man eine eindeutige Lösung finden kann [18]. Diese Eigenschaften werden im Kapitel 4.4 erklärt.

4.2 ℓ_p -Norm

Die Norm beschreibt die Länge eines mathematischen Objekts. Wir können die Norm des Fehlers zu verwenden, um die Größe des Fehlers zu beschreiben. Die ℓ_1 -, ℓ_2 - und ℓ_∞ -Normen werden häufig benutzt. Im CS führt man auch die ℓ_0 -Norm ein, wobei es wertvoll zu beachten ist, dass die ℓ_0 -Norm keine Norm ist, sondern sie die ähnliche Form wie eine Norm hat.

4.2.1 Definition der ℓ_p -Norm

Die Definition der ℓ_p -Norm eines Vektors [24] ist

$$\|\underline{x}\|_p = \left(\sum_{i=1}^n |x_i|^p \right)^{\frac{1}{p}}, \quad p \geq 1 .$$

Die Definition der ℓ_0 -Norm eines Vektors ist

$$\|\underline{x}\|_0 = \lim_{p \rightarrow 0} \sum_{i=1}^n |x_i|^p .$$

Also die ℓ_0 -Norm eines Vektors entspricht der Anzahl der Nicht-Null-Elemente. Es ist offensichtlich, dass ℓ_0 -Norm keine Norm ist, weil sie die Homogenität nicht erfüllt, also $\|2\underline{x}\|_0 = \|\underline{x}\|_0 \neq 2\|\underline{x}\|_0$.

4.2.2 Vergleich von ℓ_0 -, ℓ_1 - und ℓ_2 -Norm

Als Beispiel approximieren wir nun einigen Datenpunkte $([x_i, y_i]^T, i = 1, \dots, N)$ mit Linien nach jeweils ℓ_1 - und ℓ_2 -Norm. D. h. wir lösen die folgenden Optimierungsprobleme.

Bei ℓ_1 -Norm lösen wir

$$\min_{a_1, b_1} \sum_{i=1}^N |a_1 x_i + b_1 - y_i| .$$

Und bei ℓ_2 -Norm lösen wir

$$\min_{a_2, b_2} \sum_{i=1}^N (a_2 x_i + b_2 - y_i)^2 .$$

Die Ergebnisse sind in Abbildung 4.3 gezeigt.

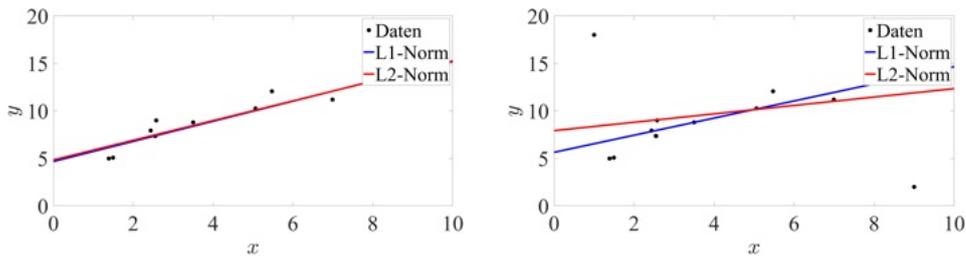


Abbildung 4.3: Lineare Approximationen.

Es ist anschaulich, dass ℓ_1 -Norm trifft mehr Datenpunkte während ℓ_2 -Norm nur wenige Datenpunkte trifft. Laut der Definition kann man davon ausgehen, dass bei ℓ_0 -Norm trifft die Linie die meisten Punkten. D. Donoho beweist im Paper [31], dass ℓ_0 -Norm durch den ℓ_1 -Norm gut approximiert werden kann. Als ein einfaches Beispiel stellen wir das folgende 2D-Optimierungsproblem

$$\underline{x} = \arg \min_{\underline{x}} \|\underline{x}\|_p$$

$$\text{NB} : \|x_1 + 3x_2 - 3\|_2 = 0 .$$

Die Nebenbedingung und der ℓ_p -Norm wird in der folgende Abbildung gezeigt, wobei $p = 0, 1, 2$.

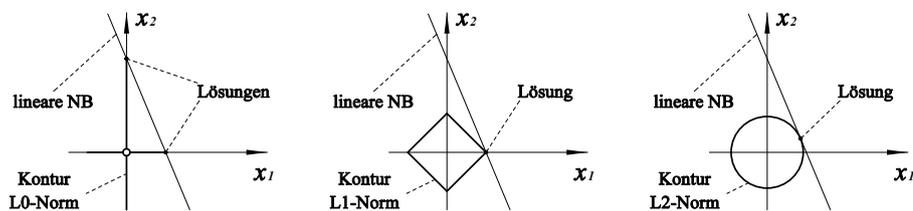


Abbildung 4.4: Vergleich ℓ_p Norm.

In der Abbildung 4.4 ist es offensichtlich, dass unter der linearen Nebenbedingung $y = \Phi \cdot \underline{x}$ eine Lösung von ℓ_0 -Norm identisch wie die Lösung von ℓ_1 -Norm ist. D. h.

bei den ℓ_0 -Norm-Optimierungsproblemen kann man als Alternative die Lösung von ℓ_1 -Norm-Optimierungsproblemen verwenden. Deswegen können wir den ℓ_0 -Norm in (4.2) durch ℓ_1 -Norm ersetzen [35; 36; 39], also

$$\underline{x} = \arg \min_{\underline{x}} \|\underline{x}\|_1 ,$$

$$\text{NB : } \|\tilde{\underline{y}} - \Phi \cdot \underline{x}\|_2 \leq \varepsilon .$$

Der Grund für diese Alternative ist, dass ℓ_0 -Norm-Optimierungsproblem NP-Schwer und nicht-konvex ist [31].

4.3 Dünnbesetzte und komprimierbare Signale

Ein Signal ist dünnbesetzt (Eng. sparse), wenn $\|\underline{x}\|_0 \ll \dim(\underline{x})$ gilt. Eine schwächere Bedingung der Dünnbesetztheit ist die Komprimierbarkeit. Ein Signal ist komprimierbar, wenn $|x_s| < C_1 \cdot s^q$ für nach absteigende Größe sortierte x_s und bestimmte Konstanten $C_1, q > 0$ gilt [24; 25].

Wir können K -Terme-Approximation verwenden [24; 26], um zu beurteilen, ob ein Signal auf einer bestimmter Basis komprimierbar ist. Die Definition der „beste K -Terme-Approximation“ ist

$$\underline{y}_K := \Phi \cdot \underline{x}_K ,$$

wobei \underline{x}_K der Vektor aus \underline{x} ist, dessen K größte Elemente gehalten und die restlichen zu null gesetzt werden. Dabei ist der Fehler zwischen dem echten Signal und dem approximierten Signal $\alpha_K(\underline{y})_p = \arg \min_{\underline{x} \in \Sigma_K} \|\underline{y} - \Phi \cdot \underline{x}\|_p$, wobei Σ_K die Menge aller Vektoren, deren ℓ_0 -Norm kleiner oder gleich K sind. Für komprimierbare Signale gilt $\sigma_K(\underline{y})_2 \leq C_2 K^{\frac{1}{2}-s}$, wobei C_2 eine bestimmte Konstante ist.

4.4 Messbasis

Messbasis ist eine Matrix, die die Zustandsgrößen in die Messungen abbilden. Normalerweise gilt für eine Messbasis $\Phi \in \mathbb{R}^{M \times N}$ $M \geq N$, damit man das Problem mit dem LS-Verfahren lösen kann. Aber beim CS gilt $M \leq N$, d. h. das ist ein unterbestimmtes Problem. Um das Problem zu lösen, muss die Messbasis einige Eigenschaften erfüllen.

4.4.1 Anforderung auf der Messbasis

Die Anforderung an Φ für eine eindeutige Lösung der Signalrekonstruktion aus dem Compressive-Sensing werden in drei Situationen vorgestellt, jeweils für exakt dünn-

besetzte Signale, komprimierbare Signale und rauschbehaftete Signale [28].

Für exakt dünnbesetzte Signale gilt das Theorem [29], dass für allen Vektoren $\underline{y} \in \mathbb{R}^M$ am meisten ein Signal $\underline{x} \in \Sigma_K$ existiert, das $\underline{y} = \Phi \underline{x}$ erfüllt, wenn $\text{spark}(\Phi) > 2K$ gilt. Die Definition von Spark ist die minimale Anzahl der linearen abhängig Spalten einer Matrix Φ [29].

Für komprimierbare Signale, wenn NSP (Eng. Null Space Property) erfüllt wird, gibt es die Rekonstruktionen Δ [30], wobei

$$\|\Delta(\Phi \underline{x}) - \underline{x}\|_2 \leq C \frac{\sigma_K(\underline{x})_1}{\sqrt{K}} \quad (4.3)$$

gilt. $\Delta(\cdot)$ ist eine Funktion zur Rekonstruktion des Signals und C ist eine bestimmte Konstante. Formel (4.3) garantiert eine obere Grenze des Rekonstruktionsfehlers [30]. Eine Matrix erfüllt K -Grad-NSP, wenn eine Konstante $C > 0$ existiert, mit $\|h_\Lambda\|_2 \leq C \frac{\|h_{\Lambda^c}\|_1}{\sqrt{K}}$ für alle $h \in \mathcal{N}(\Phi)$ und $|\Lambda| < K$ halten, wobei $\Lambda \subset \{1, \dots, N\}$ und h_Λ ein Vektor ist, dessen Λ^c -te Elemente null gesetzt werden, mit $\Lambda^c = \{1, \dots, N\} \setminus \Lambda$. $\mathcal{N}(\Phi)$ ist Null-Space (Kern) von Φ , also $\mathcal{N}(\Phi) = \{\underline{x} : \Phi \underline{x} = 0\}$.

Für rauschbehaftete Signale brauchen wir eine stärkere Bedingung: die RIP (Eng. Restricted Isometry Property) um (4.3) zu garantieren [23]. Die Definition der $\text{RIP}(K, \delta_K)$ ist folgend. Eine Matrix Φ erfüllt die $\text{RIP}(K, \delta_K)$, wenn für alle $\underline{x} \in \Sigma_K$ gilt

$$(1 - \delta_K) \|\underline{x}\|_2^2 \leq \|\Phi \underline{x}\|_2^2 \leq (1 + \delta_K) \|\underline{x}\|_2^2 .$$

Wenn die Messbasis Φ die $\text{RIP}(2K, \delta_{2K})$ Eigenschaft erfüllt, dann gibt es eine eindeutige Lösung für das Compressive-Sensing.

4.4.2 Anzahl der Messungen

Obwohl durch das CS das Signal mit sehr wenigen Anzahl der Messungen rekonstruiert werden kann, ist es klar, dass die Anzahl der Messungen größer oder gleich der Anzahl der von Null verschiedenen Elemente in den Koeffizienten \underline{x} sein muss, sonst ist es ein unterbestimmtes Problem, falls wir kein weiteres Vorwissen dazu haben. In der Praxis erfordert das CS jedoch noch mehr Messungen als die Anzahl der Nicht-Null-Elementen in \underline{x} . Sie hängt auch von Faktoren wie der Dimension des Signals ab. R. Baraniuk hat in [24] bewiesen, wenn eine Matrix $\Phi \in \mathbb{R}^M$ die $\text{RIP}(2K, \delta_{2K})$ mit der Konstante $\delta \in (0, \frac{1}{2}]$ erfüllt, dann soll die Anzahl der Messungen $M \geq C \cdot K \log \frac{N}{K}$ erfüllen, wobei $C \approx 0.28$.

Wenn die Elemente in Φ sub-Gauß'sche Verteilungen sind, also

$$E(\theta_{i,j}^2) = \frac{1}{M},$$

$$E(e^{\theta_{i,j}t}) \leq e^{\frac{t^2}{2}}.$$

z. B. Normalverteilung, Bernoulli-Verteilung usw., ist die Wahrscheinlichkeit, dass Φ die RIP($2K, \delta_{2K}$) erfüllt, höher als $1 - 2e^{-\kappa_2 M}$, wenn $M \geq \kappa_1 K \log \frac{N}{K}$ gilt, wobei $\delta \in (0, 1)$ gegeben wird, κ_1 beliebig und $\kappa_2 = \frac{\delta^2}{2\kappa_1}$ ist [27].

4.4.3 Kohärenz der Matrix

Es ist schwierig, durch die oben genannten Eigenschaften zu überprüfen, ob eine Matrix die RIP erfüllt, weil es NP-Schwer ist. Deswegen prüfen wir die RIP durch die Kohärenz der Matrix [29]. Die Definition der Kohärenz einer Matrix ist

$$\mu(\Phi) = \max_{i \neq j} \frac{|\langle \varphi_i, \varphi_j \rangle|}{\|\varphi_i\|_2 \|\varphi_j\|_2}.$$

Es gilt das Theorem im Paper [29], dass für alle Vektoren $\tilde{y} \in \mathbb{R}^M$ am meisten ein Signal $x \in \Sigma_K$ existiert, das $\tilde{y} = \Phi \cdot x$ erfüllt, wenn

$$K < \frac{1}{2} \left(1 + \frac{1}{\mu(\Phi)} \right),$$

d. h. die Kohärenz der Matrix soll möglichst klein sein.

4.5 Basisfunktionen

Wenn ein Signal x selbst nicht dünnbesetzt ist, aber es mit einer Basis Φ sowie dem entsprechenden dünnbesetzten Koeffizienten z dargestellt werden kann, also

$$\tilde{y} = \Phi \cdot x = \underbrace{\Phi \cdot \Theta}_{=: \tilde{\Theta}} z = \tilde{\Theta} \cdot z, \quad (4.4)$$

mit z dünnbesetzt ist, und $\tilde{\Theta}$ die RIP erfüllt, können wir das Problem auch mit dem Compressive-Sensing lösen. Die Matrix Θ ist eine Basis, z. B. Fourier-, DCT-, Wavelet-Basis usw. Wie in Kapitel 4.4.3 diskutiert wurde, ist die RIP von der $\tilde{\Theta}$ nicht einfach zu überprüfen, deswegen verwenden wir auch die Kohärenz als Alternative, die Definition der Kohärenz zweier Matrizen [29] ist

$$\mu(\Phi, \Theta) = \max_{i \neq j} \frac{|\langle \varphi_i, \theta_j \rangle|}{\|\varphi_i\|_2 \|\theta_j\|_2}.$$

Außerdem gibt es hohe Möglichkeit, dass der Produkt von einer orthogonalen Matrix und einer Matrix, die die RIP erfüllt, die RIP auch erfüllt, z. B. Fourier-Matrix mit Delta-Matrix, Haar-Wavelet mit Noiselet-Matrix, usw [24; 28].

4.5.1 Fourier-Basis

Bei der Fourier-Basis werden Fourierfunktionen als Basisfunktionen gewählt, also

$$\Theta_{m,n} = \exp\left(j2\pi \frac{mn}{N}\right), \quad m, n = 0, 1, \dots, N-1.$$

Nach der Normalisierung gilt

$$\Theta_{m,n} = \frac{1}{\sqrt{n}} \exp\left(j2\pi \frac{mn}{N}\right), \quad m, n = 0, 1, \dots, N-1.$$

z. B. für $N = 2$ gilt

$$\Theta_2 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}.$$

für $N = 4$ gilt

$$\Theta_4 = \frac{1}{\sqrt{4}} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & i & -1 & -i \\ 1 & -1 & 1 & -i \\ 1 & -i & -1 & i \end{bmatrix}.$$

4.5.2 DCT-Basis

Diskrete Cosinus-Transformation bracht im Prinzip weniger Punkte als Fourier-Basis, weil bei der DCT keine negative Frequenz gibt.

Basisvektoren der DCT sind

$$\theta_i[m] = \sqrt{\frac{2}{N}} \gamma_i \cos\left(\left(m + \frac{1}{2}\right) \pi \frac{i}{N}\right), \quad \gamma_i = \begin{cases} \frac{1}{\sqrt{2}}, & i = 0, N \\ 1, & \text{sonst} \end{cases},$$

dann gilt DCT-Matrix

$$\Theta = \frac{1}{\sqrt{N}} \begin{bmatrix} 1 & \dots & 1 \\ \sqrt{2} \cos \frac{\pi}{2N} & \dots & \sqrt{2} \cos \frac{(2N-1)\pi}{2N} \\ \vdots & \dots & \vdots \\ \sqrt{2} \cos \frac{(N-1)\pi}{2N} & \dots & \sqrt{2} \cos \frac{(2N-1)(N-1)\pi}{2N} \end{bmatrix}.$$

4.5.3 Wavelet-Basis

Wavelet-Transformation (WT) ist unterschiedlich von DCT oder FT, weil WT viele Filterungen in vielen Schichten hat, wie in der Abbildung 4.5 gezeigt.

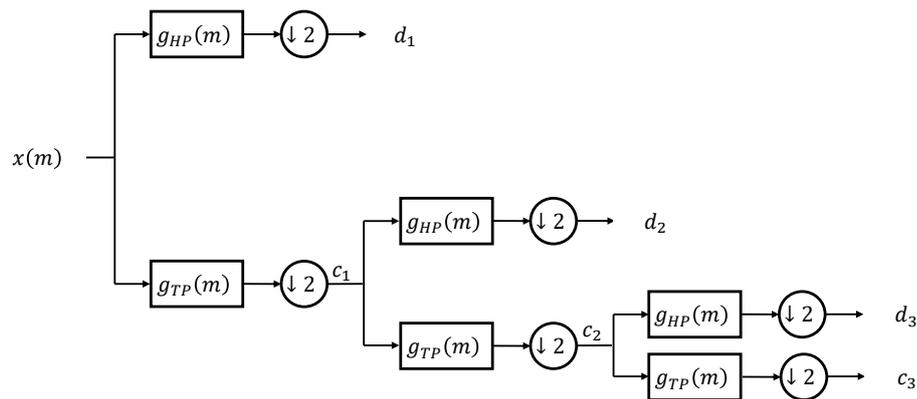


Abbildung 4.5: Verlauf der Wavelet-Transformation.

Wir müssen eine Matrix zu entwerfen, die die WT approximieren, um WT in Form von (4.4) zu bilden.

Erstens, wir berechnen die maximale Anzahl der Filterungsschichten. Nach jeder Filterung wird das Signal in zwei Teile gefiltert, also

$$c_i \rightarrow \begin{bmatrix} d_{i+1} \\ c_{i+1} \end{bmatrix} .$$

Die Länge wird auch halbiert, d. h. die maximale Anzahl der Filterungsschichten ist $Z = \lfloor \log_2 N \rfloor$.

Zweitens, wir machen die Filterung, also $d_1[n] = g_{HP} * x = \sum_{k=0}^{N-1} g_{HP}[n-k] \cdot x[k]$. In Matrixschreibweise ist es

$$d_1 = \begin{bmatrix} d_1[0] \\ \vdots \\ d_1[N+M-1] \end{bmatrix} = \begin{bmatrix} h[0] & \cdots & h[-N+1] \\ \vdots & & \vdots \\ h[N+M-2] & \cdots & h[M-1] \end{bmatrix} \cdot \begin{bmatrix} x[0] \\ \vdots \\ x[N-1] \end{bmatrix} .$$

Nach jeder Filterung geben wir die erste und letzte $\frac{M}{2}$ Zeile auf, und dann machen wir ein Down-Sampling.

Drittens, wir machen das iterative Verfahren. Nach der ersten Filterung erhalten wir

$$\begin{bmatrix} [c_1]_{\frac{N}{2} \times 1} \\ [d_1]_{\frac{N}{2} \times 1} \end{bmatrix} =: [\mathbf{W}_1]_{N \times N} \cdot [x]_{N \times 1} . \quad (4.5)$$

Nach der zweite Filterung

$$\begin{bmatrix} [c_2]_{\frac{N}{4} \times 1} \\ [d_2]_{\frac{N}{4} \times 1} \end{bmatrix} =: [\mathbf{W}_2]_{\frac{N}{2} \times \frac{N}{2}} \cdot [c_1]_{\frac{N}{2} \times 1} . \quad (4.6)$$

Setzen wir (4.5) und (4.6) zusammen, erhalten wir

$$\begin{aligned}
 \begin{bmatrix} [c_2]_{\frac{N}{4} \times 1} \\ [d_2]_{\frac{N}{4} \times 1} \\ [d_1]_{\frac{N}{2} \times 1} \end{bmatrix} &= \begin{bmatrix} [\mathbf{W}_2]_{\frac{N}{2} \times \frac{N}{2}} \cdot [c_1]_{\frac{N}{2} \times 1} \\ [d_1]_{\frac{N}{2} \times 1} \end{bmatrix} \\
 &= \begin{bmatrix} [\mathbf{W}_2]_{\frac{N}{2} \times \frac{N}{2}} & [\mathbf{0}]_{\frac{N}{2} \times \frac{N}{2}} \\ [\mathbf{0}]_{\frac{N}{2} \times \frac{N}{2}} & [\mathbf{I}]_{\frac{N}{2} \times \frac{N}{2}} \end{bmatrix} \cdot \begin{bmatrix} [c_1]_{\frac{N}{2} \times 1} \\ [d_1]_{\frac{N}{2} \times 1} \end{bmatrix} \\
 &= \underbrace{\begin{bmatrix} [\mathbf{W}_2]_{\frac{N}{2} \times \frac{N}{2}} & [\mathbf{0}]_{\frac{N}{2} \times \frac{N}{2}} \\ [\mathbf{0}]_{\frac{N}{2} \times \frac{N}{2}} & [\mathbf{I}]_{\frac{N}{2} \times \frac{N}{2}} \end{bmatrix}}_{=:[\mathbf{W}_2]_{N \times N}} \cdot [\mathbf{W}_1]_{N \times N} \cdot [x]_{N \times 1} \\
 &= [\mathbf{W}_2]_{N \times N} \cdot [\mathbf{W}_1]_{N \times N} \cdot [x]_{N \times 1} .
 \end{aligned}$$

Schließlich erhalten wir

$$\begin{bmatrix} [c_Z]_{1 \times 1} \\ [d_Z]_{1 \times 1} \\ \vdots \\ [d_1]_{\frac{N}{2} \times 1} \end{bmatrix} = \underbrace{[\mathbf{W}_Z]_{N \times N} \cdots [\mathbf{W}_1]_{N \times N}}_{=:[\mathbf{W}]_{N \times N}} \cdot [x]_{N \times 1} = [\mathbf{W}]_{N \times N} \cdot [x]_{N \times 1} .$$

Dementsprechend bekommen wir die Wavelet-Basis durch Inversion (oder Transposition) von $[\mathbf{W}]_{N \times N}$, also

$$\begin{aligned}
 \underline{z} &= [\mathbf{W}]_{N \times N} \cdot [x]_{N \times 1} =: \mathbf{W} \cdot \underline{x} , \\
 \underline{x} &= \mathbf{W}^{-1} \cdot \underline{z} =: \mathbf{\Theta} \cdot \underline{z} .
 \end{aligned}$$

In der Abbildung 4.6 werden die Ergebnisse eines Signal $x(t) = \sin 2\pi t$ aus Wavelet-Matrix und aus Wavelet-Transformation im MATLAB verglichen. Die Ergebnisse stimmen überein.

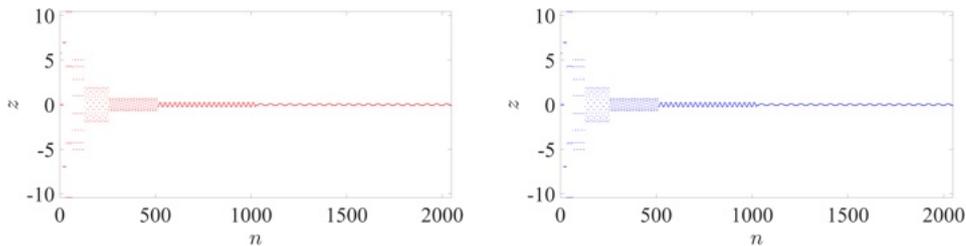


Abbildung 4.6: Koeffizienten, links: aus Wavelet-Matrix, rechts: aus Wavelet-Transformation.

Nach der Beschreibung von der WT mittels einer Matrix ist es möglich, die arithmetischen und inversen Prozesse günstiger in ein lineares Modell zu führen. Aber der Nachteil dabei ist, dass die Anzahl der Elemente zweier-Potenz sein muss.

4.6 Beispiele

In Folgenden werden zwei Beispiele gestellt, jeweils für 1D- und 2D-Signale. In den Beispielen werden wir die Signale erstellen, zufällig abtasten und schließlich durch das CS das Signal rekonstruieren.

4.6.1 1D-Signal-Rekonstruktion

Beispielsweise stellen wir nun eine Test-Funktion

$$x = 2 \sin(2\pi 10t) + \sin(2\pi 25t), \quad t \in [0, 3],$$

mit der Zeitdiskretisierung $\Delta t = 0.001$, wie in der Abbildung 4.7 gezeigt.

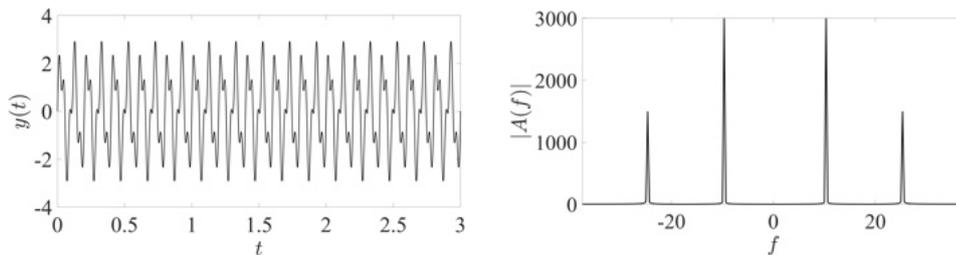


Abbildung 4.7: Signal und seine Spektrum.

Es ist anschaulich, dass das Signal mit der Fourier-Basis dünnbesetzt ist. Die Frequenzen liegen bei 10 Hz und 25 Hz. Laut Abtasttheorems muss die Abtastfrequenz f_A über $2f_G = 2 \times 25 = 50$ Hz sein, d. h. man muss mindestens $M = f_A \cdot (t_{\max} - t_{\min}) = 50 \times 3 = 150$ Messungen abtasten, wie in der folgenden Abbildung gezeigt.

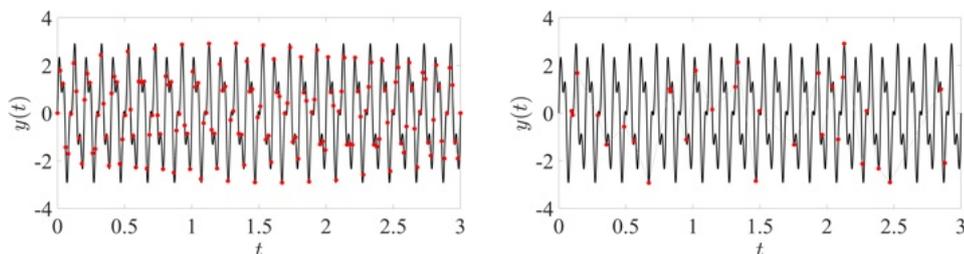


Abbildung 4.8: Messstellen, links: traditionelle Abtastung, rechts: CS-Abtastung.

Aber mithilfe des Compressive-Sensing kann man die Anzahl der Abtastungspunkte stark senken. Z. B. wählen wir nun Fourier-Basis und Delta-Messbasis, und tasten $M = 30$ Punkte an t_a^1, \dots, t_a^M zufällig ab, wie in der Abbildung gezeigt 4.8. Dann

erstellen wir die Basis

$$\Theta = \begin{bmatrix} \sin(2\pi f_0 t_0) & \cdots & \sin(2\pi f_N t_0) \\ \vdots & \ddots & \vdots \\ \sin(2\pi f_0 t_N) & \cdots & \sin(2\pi f_N t_N) \end{bmatrix},$$

und Messbasis Φ

$$\Phi_{i,j} = \begin{cases} 1, & j = t_a^i, \\ 0, & \text{sonst}, \end{cases}$$

also $\underline{y}_a = \Phi \cdot \Theta \cdot \underline{z} =: \tilde{\Theta} \cdot \underline{z}$. Nach der Lösung des Optimierungsproblem s

$$\underline{z} = \arg \min_{\underline{z}} \|\underline{z}\|_1,$$

$$\text{NB: } \|\underline{y}_a - \tilde{\Theta} \cdot \underline{z}\|_2 \leq \varepsilon,$$

rekonstruieren wir das Signal.

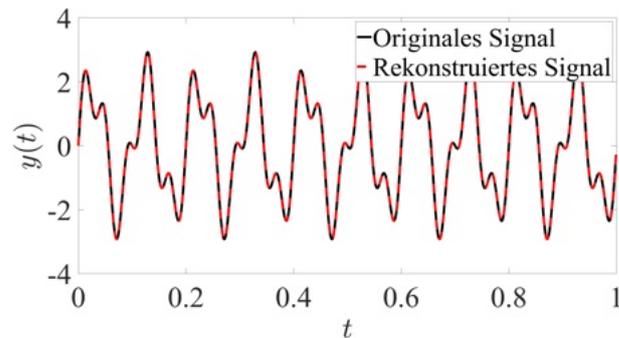


Abbildung 4.9: Rekonstruiertes Signal.

4.6.2 2D-Signal-Rekonstruktion

Nehmen wir nun ein Bild in Größe von 3024×4032 als Beispiel.



Abbildung 4.10: Originales Bild.

Um den Rechenaufwand zu sparen, machen wir zunächst ein Down-Sampling zu $N_x \times N_y = 48 \times 63 = N = 3024$, Und wählen wir zufällig 50% Punkte, also $M = \lfloor 48 \times 63 \times 50\% \rfloor = 1512$ Punkte $[x_a^1, y_a^1]^T, \dots, [x_a^M, y_a^M]^T$, sowie ihre Grauwerte g^1, \dots, g^M .



Abbildung 4.11: links: Bild nach Down-Sampling, rechts: Abtastung.

Nach der Nummerierung transformieren wir das 2D-Signal ins 1D-Signal, und wählen wir DCT-Funktionen in jeweils x und y Richtungen als Basisfunktionen, also

$$\Theta = \Theta_y \otimes \Theta_x .$$

Analog können die Delta-Messbasis Φ erstellen. Nach dem Lösen des Optimierungsproblems können wir das Bild rekonstruieren.



Abbildung 4.12: Rekonstruiertes Bild.

Wir können finden, dass das Ergebnis nicht perfekt ist. Der Grund dafür ist, dass die Dünnbesetztheit des Bilds auf DCT nicht stark ist. Wenn wir eine bessere Basis

finden können, auf der das Bild dünnbesetzt dargestellt wird, wird das Ergebnis besser als die Abbildung 4.12.

4.7 Zusammenfassung

In diesem Kapitel stellen wir das Modell für das Compressive-Sensing, ℓ_1 -Norm Optimierung und die Anforderungen an die Messbasis vor. Dann folgt die Einführung einiger grundlegender Basisfunktionen, und die Ableitung der Wavelet-Transformation in Matrixform. Schließlich zeigen wir zwei Beispiele des CS in Signalreduktion für jeweils 1D- und 2D-Signale. Wir können sehen, dass bei den dünnbesetzten Signalen die Anzahl der Messungen viel weniger benötigt werden. Im nächsten Kapitel, das der Hauptteil dieser Arbeit darstellt, werden wir die Anwendung vom CS auf nicht-dünnbesetzte Signale diskutieren. Mithilfe des Systemmodells können wir die Anforderung nach Dünnbesetztheit stark abschwächen.

KAPITEL 5

Kalman-Filtered Compressive-Sensing

5.1 Einleitung

Aus dem Kapitel 4 wissen wir, wenn ein Signal dünnbesetzt ist, kann man das Compressive-Sensing (CS) anwenden, um das Signal durch weniger Messungen zu rekonstruieren, als nach dem Nyquist-Shannon-Theorem gefordert sind. Das CS ist jedoch nicht für nicht-dünnbesetzte Signale geeignet. Daher muss man ein neues Verfahren aufstellen, um das CS anwenden zu können. Durch die Kombination vom CS mit dem Systemmodell (Kalman-Filter) kann man das neue Verfahren realisieren. Im Folgenden werden wir die Temperaturverteilung mit punktförmigen Anregungen aus Kapitel 2 als das Beispiel zur Simulation und Bewertung unseres Algorithmus nehmen.

5.2 Stand der Technik

Es gibt bereits einige Forscher, die an der Kombination vom Compressive-Sensing und dem Kalman-Filter oder anderen Verfahren arbeiten [45]. Z. B.

M. Salman Asif [46; 47] hat den Begriff Homotopie-Parameter in das Compressive-Sensing eingeführt, um eine schnellere und genauere Lösungsmethode des CS zu finden, nämlich die dynamische Aktualisierung für ℓ_1 -Minimierung [47]. Als Anwendungsbeispiel hat er die Methode auf Streaming-Signale und Videos angewendet [46].

S. Ji. et. al. [51] haben Bayes'sche Compressive-Sensing vorgeschlagen, wobei die Messungen adaptiv ausgewählt werden und die Varianz der Fehler auch gleichzeitig geschätzt werden. Dadurch ist der Fehler deutlich kleiner, besonders bei wenigen Messungen.

Im Jahr 2008 hat N. Vaswani [48] erstmals ein Verfahren für die Kombination von Kalman-Filter und Compressive-Sensing entwickelt, bei dem das Compressive-Sensing in der Innovationsschritt (Filterschritt) des Kalman-Filters durchgeführt wird. Das Kalman-Filter schätzt nur die Zustandsgröße in "Support-Set". Ein Support-Set ist die Menge der Indices C , für die x_C die Nicht-Null-Elemente sind. Wenn der Fehler größer als der Schwellwert ist, wird das Compressive-Sensing benutzt, um die Support-Set zu aktualisieren. Im [49] hat N. Vaswani das Verfahren verbessert. Das verbesserte Verfahren wird als modifiziertes KFCS bezeichnet, bei dem ein vorläufiges Kalman-Filter eingeführt wird, um die Rechenaufwand zu reduzieren. Das Verfahren funktioniert besonders gut für die sich langsam veränderte Signale.

Aber ihre Arbeit basiert immer auf der Dünnbesetztheit des Signals [52] und versucht, die Genauigkeit zu verbessern und die Algorithmen schneller zu machen. Es gibt noch keine Arbeit sich damit beschäftigt, die Anforderung der Dünnbesetztheit durch die Kombination des Systemmodells abzuschwächen.

5.3 Kalman-Filter mit zufälligen Messungen

Beim CS macht man zufällige Messungen. Deswegen kombinieren wir zuerst die zufällige Messungen mit dem Kalman-Filter, um eine bessere Schätzung zu erhalten.

5.3.1 Systemmodell

Bevor das CS und das KF kombiniert werden, probieren wir zunächst das Kalman-Filter mit zufälligen Messungen, d. h. die Messwerte werden nicht an fest Stellen gewählt, sondern sich zufällig verändern. Um diese Funktion zu realisieren, können wir vorab viele Sensoren ausbringen, und zu jedem Zeitpunkt aktivieren wir einige Sensoren davon zufällig.

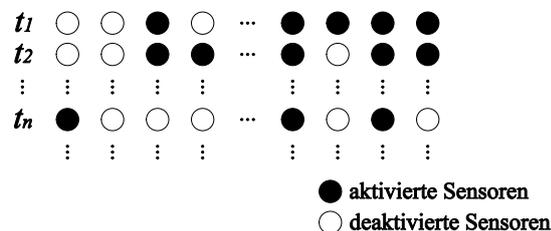


Abbildung 5.1: Eingestellte Sensoren.

Es kann durch eine Delta-Matrix Φ_k beschrieben werden, welche Sensoren zum Zeitpunkt k aktiviert oder deaktiviert werden. Dann können wir das Systemmodell und Messmodell für das rsKF (Engl. random sampling Kalman Filter) herleiten:

- Bei der Modalanalyse gilt

$$\begin{aligned} \underline{T}_{k+1} &= \mathbf{A} \cdot \underline{T}_k + \underline{w}_k, \\ \underbrace{\Phi_{k+1} \cdot \underline{y}_{k+1}}_{=:\tilde{\underline{y}}_{k+1}} &= \underbrace{\Phi_{k+1} \cdot \Psi}_{=:\tilde{\Psi}} \cdot \underline{T}_{k+1} + \underbrace{\Phi_{k+1} \cdot \underline{v}_{k+1}}_{=:\tilde{\underline{v}}_{k+1}}, \end{aligned}$$

also

$$\begin{aligned} \underline{T}_{k+1} &= \mathbf{A} \cdot \underline{T}_k + \underline{w}_k, \\ \tilde{\underline{y}}_{k+1} &= \tilde{\Psi}_{k+1} \cdot \underline{T}_{k+1} + \tilde{\underline{v}}_{k+1}, \end{aligned}$$

wobei

$$\begin{aligned} \mathbf{C}_{k+1}^{\tilde{\underline{v}}} &= E \{ \tilde{\underline{v}}_{k+1} \cdot \tilde{\underline{v}}_{k+1}^T \} \\ &= E \{ \Phi_{k+1} \underline{v} \cdot \underline{v}^T \Phi_{k+1}^T \} \\ &= \Phi_{k+1} \mathbf{C}_{k+1}^{\underline{v}} \Phi_{k+1}^T. \end{aligned}$$

- Bei dem Finite-Differenzen-Verfahren gilt

$$\begin{aligned} \underline{f}_{k+1} &= \mathcal{D} \cdot \underline{f}_k + \underline{w}_k, \\ \tilde{\underline{y}}_{k+1} &= \tilde{\Phi}_{k+1} \cdot \underline{f}_{k+1} + \tilde{\underline{v}}_{k+1}. \end{aligned}$$

Wir nennen $\tilde{\underline{y}}_{k+1}$ die komprimierte Messung.

5.3.2 Simulation

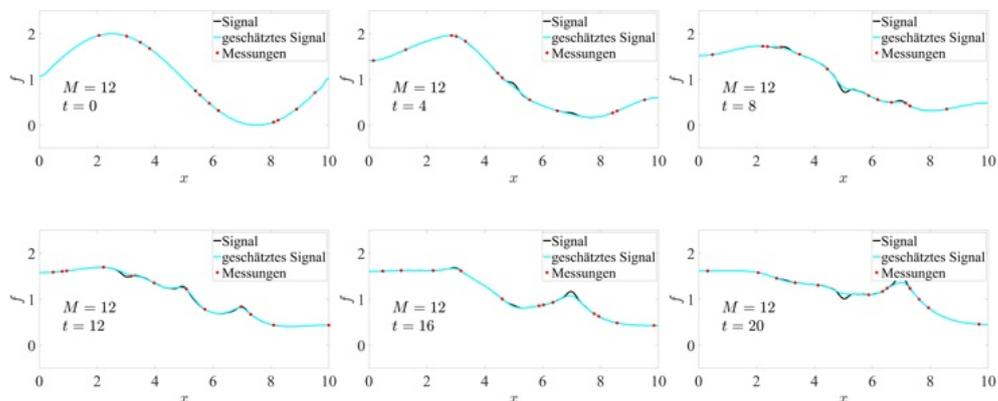


Abbildung 5.2: rsKF aus der FDM mit 12 zufällige Messungen.

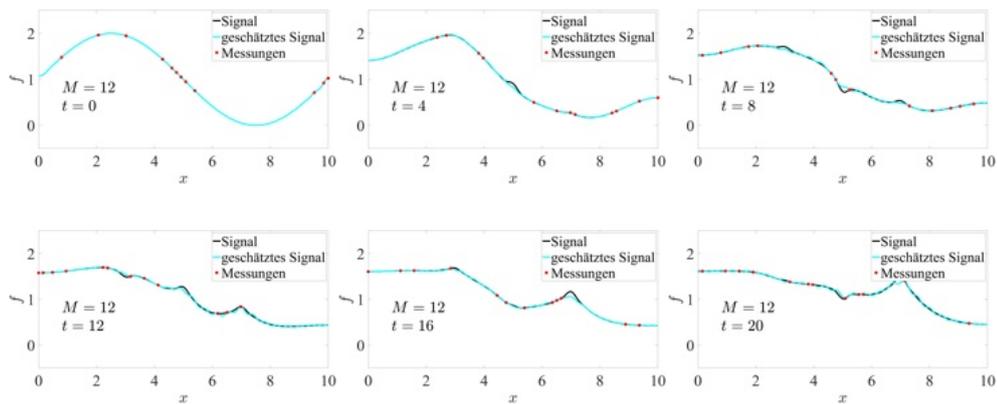


Abbildung 5.3: rsKF aus der Modalanalyse mit 12 zufällige Messungen.

Es ist anschaulich, dass es viele Schwingungen in der Simulation aus der Modalanalyse gibt, weil der Grad G der Modalanalyse größer ist als die Anzahl der Messungen. Wenn der Grad G jedoch zu klein gewählt wird, führt dies dazu, dass das Modell zu der Temperaturverteilung nicht gut passen kann. Daher ist die Wahl des richtigen Grads ein wichtiges Problem. Außerdem ist die FDM, wie in Kapitel 1 dargelegt, breiter verwendet als die Modalanalyse, sodass wir uns in den zukünftigen Diskussionen auf der FDM konzentrieren werden.

5.3.3 Diskussion

Der Vergleich zwischen dem rsKF und dem traditionellen Kalman-Filter (mit festen Sensorstellen) wird in der Abbildung 5.4 gezeigt.

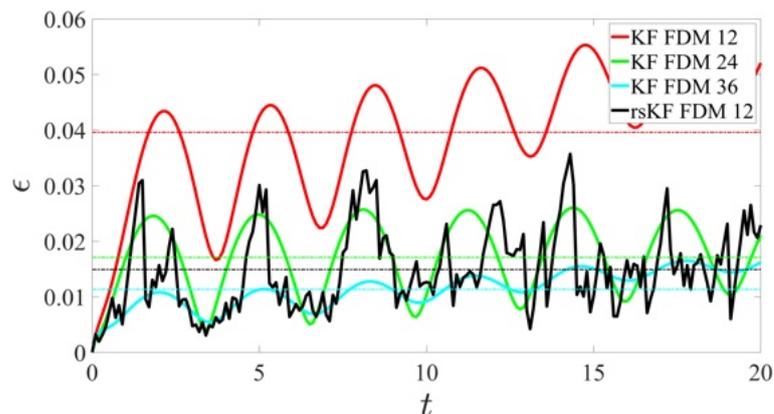


Abbildung 5.4: Vergleich rsKF und KF.

Das Ergebnis aus dem rsKF mit 12 Messungen ist viel besser als das traditionellen KF mit 12 Sensoren. Es ist sogar besser als das KF mit 24 Sensoren. Aber der rsKF hat noch viele Nachteile. Z. B. er wird in einigen Fällen die Position der

Spitzen nicht gemessen, was zu großen Fehlern führen kann, wie in der Abbildung 5.5 gezeigt. Deshalb wollen wir die Genauigkeit der Zustandsschätzung durch die Kombination vom rsKF und dem CS weiter verbessern.

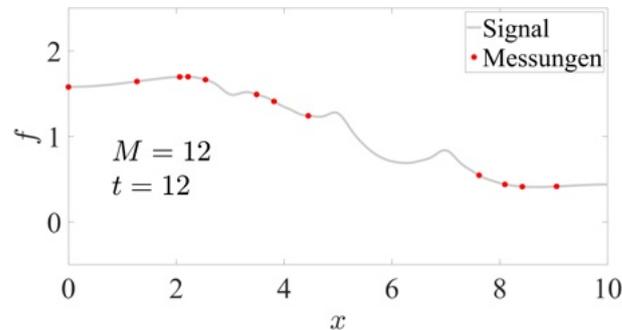


Abbildung 5.5: Nachteil des rsKF, keine Messung fällt an den Spitzen.

5.4 Signal-Analyse

Wie in Kapitel 3 dargelegt, sollten, wenn wir das Compressive-Sensing verwenden wollen, die geschätzten Koeffizienten dünnbesetzt oder komprimierbar sein und gleichzeitig die Basis $\tilde{\Theta}$ die RIP erfüllen muss. Deswegen müssen wir zuerst die Eigenschaften des Signals analysieren, und wenn das Signal nicht komprimierbar ist, müssen wir das Signal durch einige Methoden so dünnbesetzt wie möglich machen, um das Compressive-Sensing verwenden zu können. Und danach treffen wir einige Maßnahmen, um die RIP von $\tilde{\Theta}$ zu garantieren.

5.4.1 Hauptprobleme

Zuerst wird die Komprimierbarkeit des Signals betrachtet. Die beste Methode, um festzustellen, ob ein Signal komprimierbar ist, ist die K -Term-Approximation. Wenn ein Signal mit K Termen approximiert werden kann, bedeutet das, dass das Signal durch ein K -dünnbesetztes Signal approximiert werden kann, mithilfe des Compressive-Sensing können wir dieses K -dünnbesetzte Signal berechnen. Da ε in der Formel (4.2) die Differenz zwischen dem approximierten Signal und dem Originalsignal darstellt, ist es nötig zu beachten, dass ε bei exakt dünnbesetzten Signalen gleich 0 sein kann, da das approximierte Signal genau das Originalsignal ist. Wenn das Signal jedoch nur komprimierbar ist, gibt es immer einen Fehler zwischen dem approximierten Signal und dem ursprünglichen Signal, sodass ε nicht zu klein sein darf. Wenn ε zu klein ist, führt dies zu einer Überanpassung.

Im Folgenden wählen wir $K = 10$, weil wir 12 Messungen zur Rekonstruktion der

Temperaturverteilung verwenden wollen. Daher muss K mindestens kleiner als 12 sein. Außerdem wählen wir den Zeitpunkt $t = 20$ als Beispiel:

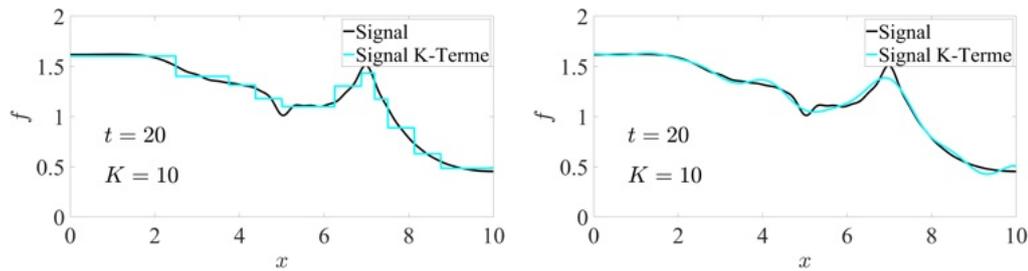


Abbildung 5.6: K -Terme-Approximation links: Waveletbasis, rechts: DCT-Basis.

Von der Abbildung 5.6 können wir ausgehen, dass das Signal nicht K -dünnbesetzt ist, weder mit der Wavelet-Basis noch mit der DCT-Basis. Trotzdem versuchen wir, dieses Signal mit dem Compressive-Sensing zu rekonstruieren.

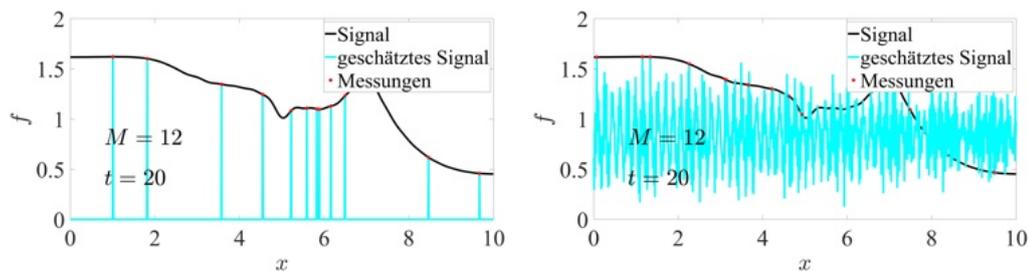


Abbildung 5.7: Rekonstruiertes Signal links: Waveletbasis, rechts: DCT-Basis.

Es ist aus der Abbildung 5.7 anschaulich, dass das rekonstruiertes Signal aus dem Compressive-Sensing zu der K -Terme-Approximation überhaupt nicht passt. Der Grund dafür ist, dass die Basis die RIP nicht erfüllt.

Zusammenfassen, wir haben zwei Hauptprobleme getroffen. Erstens, das Signal nicht dünnbesetzt oder komprimierbar ist. Zweitens, die Basis erfüllt die RIP nicht. Im Folgenden werden wir auf diese zwei Probleme fokussieren.

5.4.2 Komprimierbare Änderungen

In dem Beispiel können die manuell gegebenen Anfangsbedingungen beliebig sein, aber nachdem die Anfangsbedingung festgelegt ist, ist die Wärmeleitung mit dem Systemmodell konsistent. So spekuliert man unweigerlich, dass vielleicht die Koeffizienten z_{k+1} selbst nicht komprimierbar ist (weil die Anfangsbedingung beliebig gegeben sein kann), die Änderung der Koeffizienten $\Delta z_k := z_{k+1} - z_k$ jedoch komprimierbar sein könnte, weil die Änderung des Signals einem bestimmten Gesetz

(dem Systemmodell) entspricht. Daher werden wir die Änderung der Koeffizienten betrachten. Folgende Grafik zeigt die K -Terme-Approximation der Änderung der Koeffizienten Δz mit $\Delta t = 0.5$ s, nämlich $t_1 = 19.5$ s und $t_2 = 20$ s, d. h. die Koeffizienten zum Zeitpunkt $t_1 = 19.5$ s wird als bekannt angenommen, während die Koeffizienten zum Zeitpunkt $t_2 = 20$ s zu schätzen sind.

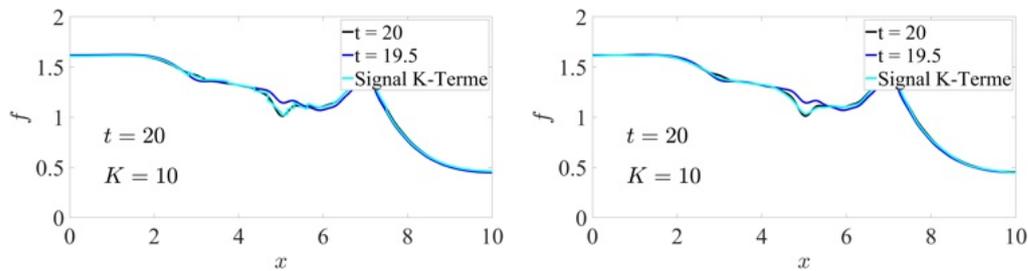


Abbildung 5.8: K -Terme-Approximation für die Änderung der Koeffizienten links: Waveletbasis, rechts: DCT-Basis.

Wie wir aus der obigen Grafik ersehen können, sind die Ergebnisse der K -Terme-Approximation von der Änderung der Koeffizienten viel besser als die von der Koeffizienten selbst, insbesondere mit DCT-Basis. Daher können wir die Änderungen der Koeffizienten als komprimierbar bezeichnen. Aber wir dürfen eine weitere Voraussetzung für das Compressive-Sensing nicht außer Acht lassen, nämlich dass die RIP erfüllt sein muss. Zuerst versuchen wir durch das Compressive-Sensing das Signal zu rekonstruieren, und wir erhalten die Ergebnisse in der Abbildung 5.9.

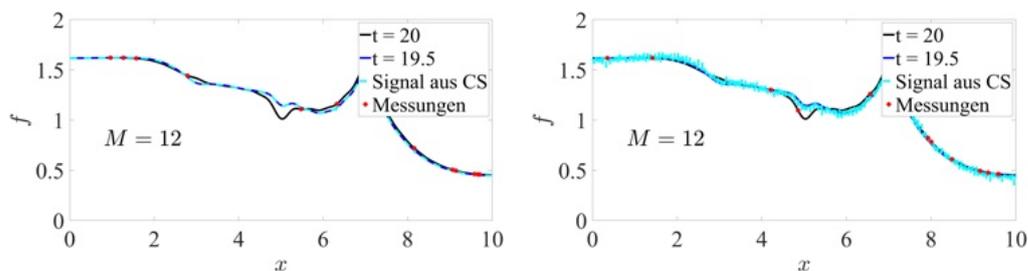


Abbildung 5.9: Rekonstruiertes Signal für die Änderung der Koeffizienten links: Waveletbasis, rechts: DCT-Basis.

Leider können wir durch die ℓ_1 -Norm-Optimierung die richtigen Ergebnisse nicht erzielen. Denn obwohl die Änderung der Koeffizienten Δz komprimierbar ist, wird die andere Bedingung des Compressive-Sensing, die RIP, nicht erfüllt, sodass wir das Problem der RIP noch lösen müssen. Im nächsten Abschnitt werden wir uns auf die Lösung des RIP-Problems konzentrieren, damit wir es ermöglichen, das Kalman-Filter und das Compressive-Sensing zu kombinieren.

5.4.3 Dimensionreduzierung

Aus der Definition der RIP

$$(1 - \delta_K) \|\underline{z}\|_2^2 \leq \left\| \tilde{\Theta} \cdot \underline{z} \right\|_2^2 \leq (1 + \delta_K) \|\underline{z}\|_2^2 ,$$

wissen wir, dass die RIP den Grad der Orthogonalität der Basis $\tilde{\Theta} \in \mathbb{R}^{M \times N}$ unter K -dünnbesetzten Signalen garantiert. Mit anderen Worten, je kleiner die $\frac{M}{N}$ ist, desto kleiner Möglichkeit es gibt, dass die Basis $\tilde{\Theta}$ die RIP erfüllt. Das heißt, wir sollten N so weit wie möglich reduzieren, um eine höhere Wahrscheinlichkeit zu haben, dass die RIP erfüllt. Deswegen rekonstruieren wir nicht alle Werte an den Knoten, sondern nur eine bestimmte Anzahl der Werte, die ausreichend für die Schätzung der Zustandsgrößen ist. Beispielsweise können wir die Werte nur an den möglichen Sensorstellen rekonstruieren, somit reduziert sich N in diesem Fall von $N = 1024$ auf $N = 64$. Auf diese Weise simulieren wir nun die Fälle im letzten Kapitel:

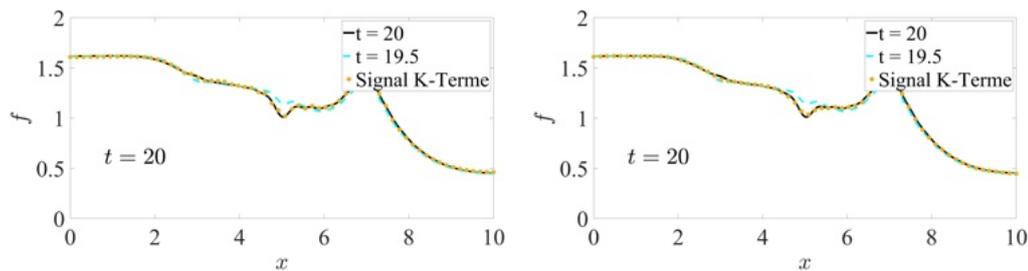


Abbildung 5.10: K -Terme-Approximation für die Änderung der Koeffizienten mit $N = 64$ links: Waveletbasis, rechts: DCT-Basis.

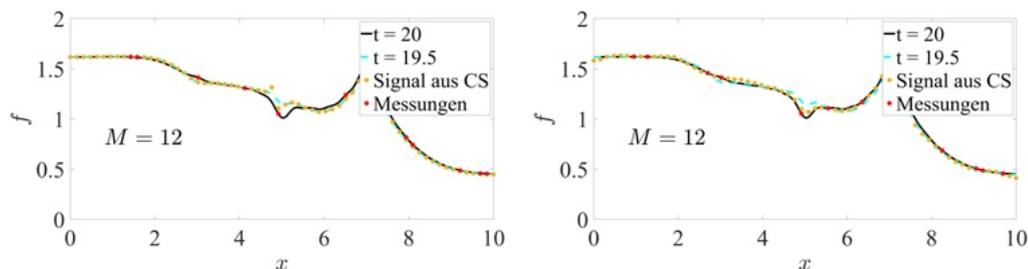


Abbildung 5.11: Rekonstruiertes Signal für die Änderung der Koeffizienten mit $N = 64$ links: Waveletbasis, rechts: DCT-Basis.

Wie aus Abbildung 5.11 ersichtlich ist, haben wir nach der Schätzung von der Änderung der Koeffizienten und der Reduzierung der Dimension relativ genauere Ergebnisse erhalten, und das Ergebnis mit der DCT-Basis ist im Vergleich zu dem Ergebnis mit Wavelet-Basis besser. Aus der Modalanalyse können wir auch erkennen, dass die Temperaturverteilung durch einige glatte Cosinusfunktionen gut approximiert werden kann. Die Haar-Wavelets sind jedoch schärfer und mit vielen Sprüngen.

5.4.4 Sparse-Coding-Representation

Aus dem obigen Beispiel ist ersichtlich, dass verschiedene Basen großen Einfluss auf die Ergebnisse haben, da das Signal mit verschiedenen Basen eine unterschiedliche Dünnbesetztheit aufweist. Deshalb hoffen wir, eine Basis zu finden, mit der das Signal so dünnbesetzt wie möglich dargestellt werden kann.

Sparse-Coding-Representation (SCR) war in den letzten Jahren ein großes Thema im Bereich des maschinellen Lernens [34; 35; 36]. Das Ziel der SCR besteht darin, den größten Teil oder das gesamte ursprüngliche Signal in einer Linearkombination aus weniger Basissignalen darzustellen. Die SCR wurde erstmals im Jahr 1996 von Olshausen und Field vorgeschlagen [33], um die visuelle Verarbeitung im Gehirn zu erklären. Wir können die SCR verwenden, um eine Basis zu finden, auf der das Signal dünnbesetzt dargestellt werden kann. Eine Realisierung des SCR-Problems ist Lösung des das Optimierungsproblems (5.1).

$$\min_{\underline{\alpha}_i, \Theta} \sum_{i=1}^K \left\{ \left\| y_i - \Theta \cdot \underline{\alpha}_i \right\|_2^2 + \lambda_i \|\underline{\alpha}_i\|_0 \right\}$$

$$\text{NB } \Theta^T \Theta = \mathbf{I} \quad (5.1)$$

Der erste Term steht für den Fehler der Approximation von der Basis Θ und seiner Koeffizienten $\underline{\alpha}_i$ zu allen Zeitpunkten $i = 1, \dots, K$. Der zweite Term garantiert die Dünnbesetztheit der Koeffizienten $\underline{\alpha}_i$ zu jedem Zeitpunkt und die Nebenbedingung bedeutet die Orthogonalität der Basis Θ . Die Abbildung 5.12 zeigt eine Visualisierung der Basis Θ :

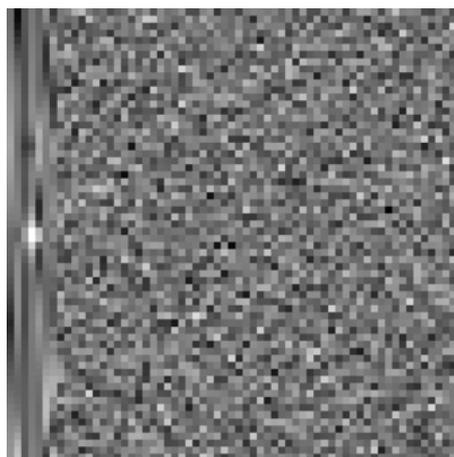


Abbildung 5.12: Visualisierung von Θ .

In der Abbildung 5.12 beschreibt jede Spalte eine Basisfunktion $\underline{\theta}_i$.

5.5 Kalman-Gleichung mit Compressive-Sensing

Um die Informationen aus dem Kalman-Filter und aus dem Compressive-Sensing zu fusionieren, brauchen wir dazu noch ein Verfahren. Zunächst stellen wir eine weitverbreitete Methode vor, die die ℓ_1 -Norm-Optimierung direkt in der Kalman-Gleichung einsetzen.

Die Kalman-Gleichung, z. B. aus der Modalanalyse, ist

$$\underline{T}_{k+1}^e = \arg \min_{\underline{T}_{k+1}} \left\{ \frac{1}{\sigma_S^2} \|\mathbf{A} \cdot \underline{T}_k - \underline{T}_{k+1}\|_2^2 + \frac{1}{\sigma_M^2} \left\| \tilde{\underline{y}}_{k+1} - \tilde{\Psi} \cdot \underline{T}_{k+1} \right\|_2^2 \right\}, \quad (5.2)$$

wobei σ_S für die Varianz der Systemrauschen und σ_M für die Varianz der Messunsicherheit stehen. Bei der FDM kann man die Zustandsgrößen und die Systemmatrix entsprechend wecheln. Für das Compressive-Sensing gilt

$$\tilde{\underline{y}}_{k+1} = \tilde{\Theta} \cdot \underline{z}_{k+1},$$

und

$$\begin{aligned} \underline{z}_{k+1}^e &= \arg \min_{\underline{z}_{k+1}} \|\underline{z}_{k+1}\|_1, \\ \text{NB} \quad &\left\| \tilde{\underline{y}}_{k+1} - \tilde{\Theta} \cdot \underline{z}_{k+1} \right\|_2^2 \leq \varepsilon. \end{aligned}$$

Man kann die oben genannten zwei Formeln auch in eine Formel umschreiben [24], also

$$\underline{z}_{k+1}^e = \arg \min_{\underline{z}_{k+1}} \left\{ \|\underline{z}\|_1 + \left\| \tilde{\underline{y}}_{k+1} - \tilde{\Theta} \cdot \underline{z}_{k+1} \right\|_2^2 \right\}. \quad (5.3)$$

Es ist zu betrachten, dass

$$\begin{aligned} \tilde{\Psi} \cdot \underline{T}_{k+1}^e &= \tilde{\Theta} \cdot \underline{z}_{k+1}^e \\ \underline{T}_{k+1}^e &\approx \left(\tilde{\Psi}^T \tilde{\Psi} \right)^{-1} \tilde{\Psi}^T \tilde{\Theta} \cdot \underline{z}_{k+1}^e \\ &= \left(\tilde{\Psi}^T \tilde{\Psi} \right)^{-1} \tilde{\Psi}^T \tilde{\Theta} \cdot (\underline{z}_k^e + \Delta \underline{z}_k^e) \end{aligned}$$

gilt, wobei $\tilde{\Psi}$ die komprimierende Messabbildung in der Modalanalyse und $\tilde{\Theta}$ die komprimierende Messabbildung im Compressive-Sensing sind. Dann kombiniert man

(5.2) und (5.3) in Form von

$$\Delta \underline{z}_k^c = \arg \min_{\Delta \underline{z}_k} \left\{ G_1 \|\Delta \underline{z}_k\|_1 + G_2 \left(\frac{1}{\sigma_S^2} \left\| \mathbf{A} \cdot \underline{T}_k - \left(\tilde{\Psi}^T \tilde{\Psi} \right)^{-1} \tilde{\Psi}^T \tilde{\Theta} \cdot (\underline{z}_k + \Delta \underline{z}_k) \right\|_2^2 + \frac{1}{\sigma_M^2} \left\| \tilde{y}_{k+1} - \tilde{\Theta} \cdot (\underline{z}_k + \Delta \underline{z}_k) \right\|_2^2 \right) \right\}$$

$$= \arg \min_{\Delta \underline{z}_k} \left\{ G_1 \|\Delta \underline{z}_k\|_1 + G_2 \left\| \begin{bmatrix} \frac{1}{\sigma_S} \left(\mathbf{A} \cdot \underline{T}_k - \left(\tilde{\Psi}^T \tilde{\Psi} \right)^{-1} \tilde{\Psi}^T \tilde{\Theta} \cdot (\underline{z}_k + \Delta \underline{z}_k) \right) \\ \frac{1}{\sigma_M} \left(\tilde{y}_{k+1} - \tilde{\Theta} \cdot (\underline{z}_k + \Delta \underline{z}_k) \right) \end{bmatrix} \right\|_2^2 \right\}.$$

G_1 und G_2 bezeichnen die Gewichtungen für den ℓ_p -Norm-Term und den zweiten Term. Bei den Simulationen werden mehrerer Werte von $\frac{G_2}{G_1}$ gewählt, und zur Vereinfachung wählen wir $\sigma_S = \sigma_M = 1$.

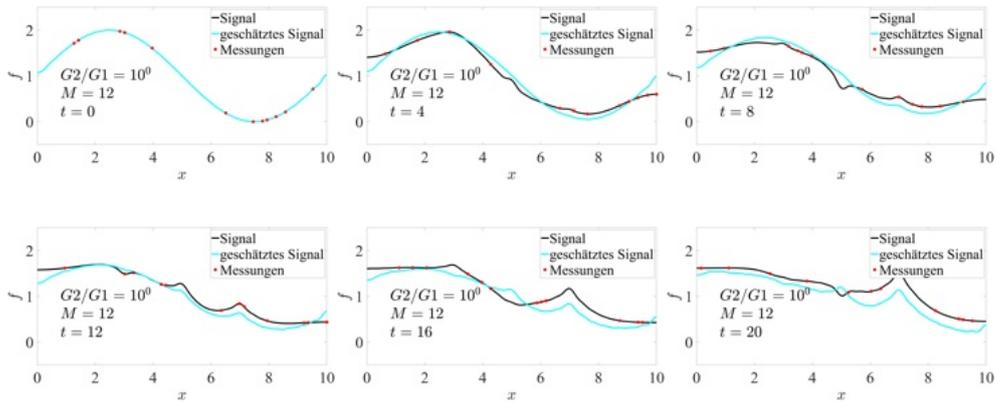


Abbildung 5.13: Rekonstruiertes Signal mit Kalman-Gleichung mit $G_2/G_1 = 10^0$.

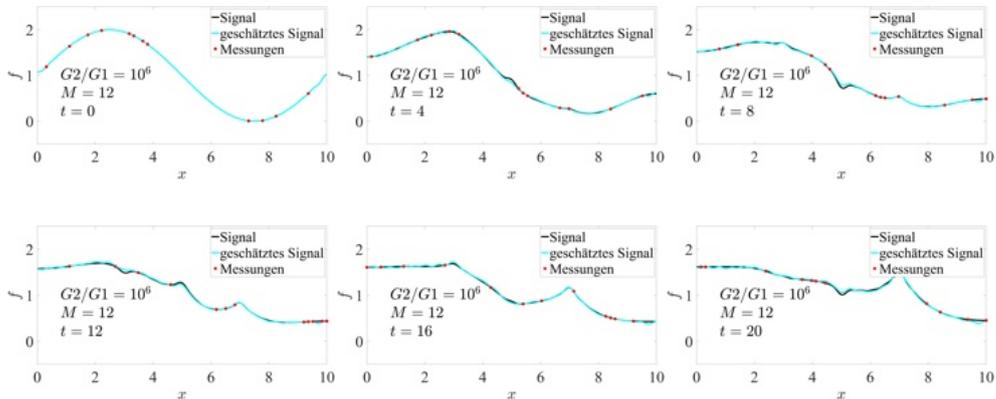


Abbildung 5.14: Rekonstruiertes Signal mit Kalman-Gleichung mit $G_2/G_1 = 10^6$.

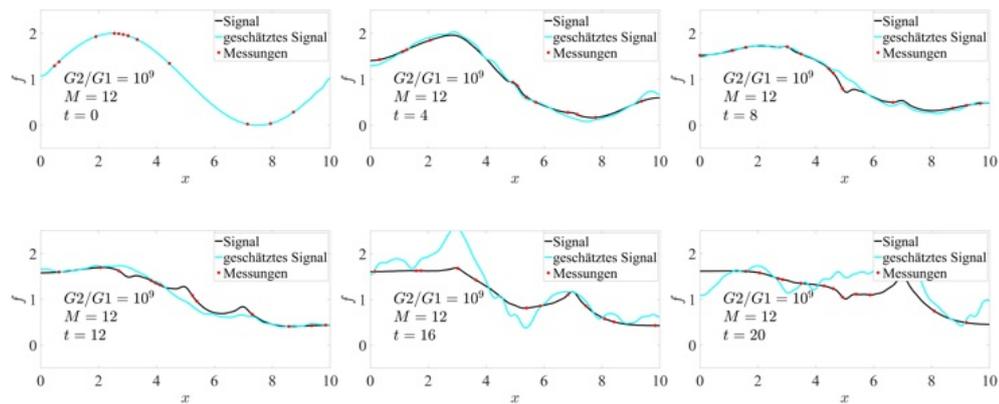


Abbildung 5.15: Rekonstruiertes Signal mit Kalman-Gleichung mit $G_2/G_1 = 10^9$.

Wir können davon ausgehen, dass die Gewichtungen G_1 und G_2 das Ergebnis stark beeinflussen. Um die Wirksamkeit der Gewichtungen zu analysieren, haben wir mehrere Simulationen gemacht, und danach die Fehler bewertet.

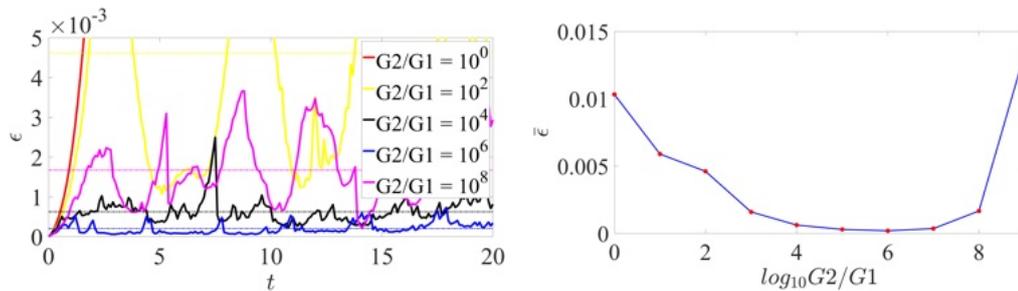


Abbildung 5.16: Fehler mit verschiedenen G_2/G_1 .

Es ist anschaulich, wenn G_2/G_1 zu groß oder zu klein ist, ist der Fehler sehr groß. Das beste Ergebnis kommt bei $G_2/G_1 \approx 10^6$ vor. Aber in der Praxis kennen wir jedoch das echte Signal nicht, sodass es schwierig ist, die Gewichtung G_1 und G_2 zu bestimmen. Deshalb brauchen wir ein anderes Verfahren, bei dem die Forderung auf der Gewichtung abgeschwächt wird.

5.6 Kalman-Filter mit Pseudo-Messungen

Das Kalman-Filter mit Pseudo-Messungen bedeutet, dass man das CS vor dem KF läuft, dann werden die geschätzte aus dem CS als die Messungen im Kalman-Filter einsetzen. In diesen Ansatz spielen die Gewichtungen G_1 und G_2 wenige Rolle.

5.6.1 Pseudo-Messungen

Im Gegensatz zur vorherigen Methode lassen wir bei dieser Methode zunächst das Compressive-Sensing unabhängig vom KF laufen, wie in der Abbildung 5.17 gezeigt.

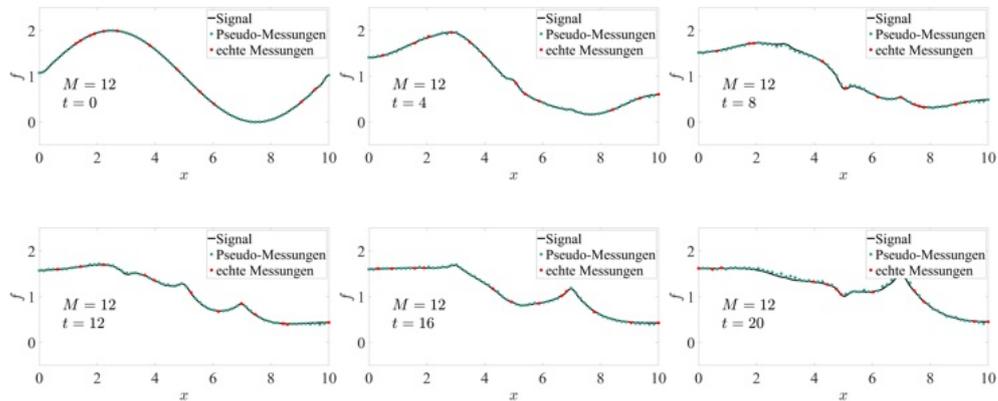


Abbildung 5.17: Pseudo-Messungen.

Diese rekonstruierte Punkte geben wir dann als die Pseudo-Messungen (PM) in das Kalman-Filter ein, und auf diese Weise können wir das Compressive-Sensing und das Kalman-Filter kombinieren.

5.6.2 Dynamische Gewichtung

Auf diese Weise können wir den Einfluss der Gewichtung reduzieren. Aber trotzdem können wir die Messungen noch verbessern. Da die Pseudo-Messungen keine reale Messungen sind, ist die Unsicherheit von Pseudo-Messungen σ_{PM} größer als die reale Messungen σ_M . Deshalb können wir den Pseudo-Messungen höhere Messunsicherheit geben. Es ist auch offenbar, wenn $\sigma_{PM} \rightarrow \infty$, ist das Kalman-Filter mit Pseudo-Messungen gleich wie das rsKF, d. h. nur die tatsächliche Messungen verwendet werden.

Wie wir aus der Abbildung 5.18 ersehen können, ist das Ergebnis nach der Einführung von dynamischer Gewichtung deutlich verbessert, vor allem an den Spitzen. Bemerkenswert ist, dass das Ergebnis an einigen Stellen nach der Einführung dynamischer Gewichtung sogar noch schlechter ist, d. h. es gilt nicht immer, je kleiner die Gewichtung der Pseudo-Messungen eingestellt wird, desto besser wird das Ergebnis. Wir werden später noch über die Messunsicherheit der PM diskutieren.

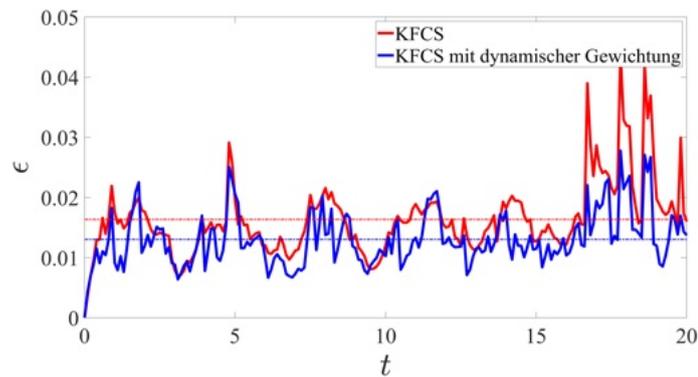


Abbildung 5.18: Vergleich ohne und mit dynamischer Gewichtung.

5.6.3 Fehler-Akkumulation

Da wir nur die Änderung der Koeffizienten $\Delta \underline{z}$ schätzen und die Messungen zufällig gewählt werden, führt dies dazu, dass der Fehler sich akkumuliert, wie in der Abbildung 5.19 dargestellt. Wir finden, dass dies daran liegt, dass wir nur die ℓ_1 -Norm-Optimierung vom Compressive-Sensing separat durchführen und die Pseudo-Messungen in das Kalman-Filter setzen. In diesem Prozess sind das Compressive-Sensing und das Kalman-Filter nicht vollständig fusioniert, d. h. die Ergebnisse vom CS wirken auf das KF, aber die Ergebnisse vom KF beeinflussen das CS nicht. Deshalb hoffen wir, dass das KF auch zu einer besserer Abschätzung vom CS beitragen wird. Um dies zu erreichen, können wir die Koeffizientenaktualisierung verwenden.

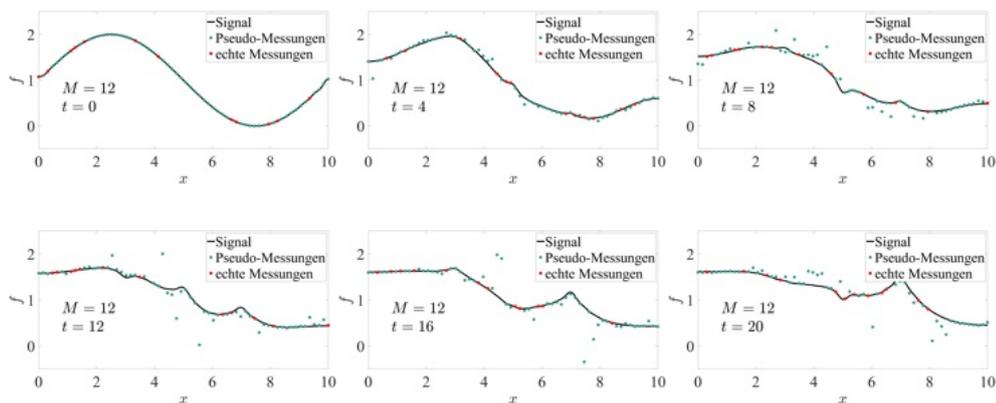


Abbildung 5.19: Pseudo-Messungen mit Wavelet-Basis.

5.7 Koeffizientenaktualisierung

Wie in der Abbildung 5.20 gezeigt, schätzen wir im vorherigen Prozess die Änderung der Koeffizienten $\Delta \underline{z}_k$ aus den zufälligen Messungen \tilde{y}_{k+1} mithilfe des Compressive-

Sensing, dann berechnen wir $z_{k+1} = z_k + \Delta z_k$. Mit z_{k+1} rekonstruieren wir die Pseudo-Messungen y_{k+1}^{PM} . Dann setzen wir die Pseudo-Messungen in das Kalman-Filter ein, um den aktuellen Zustandsgröße T_{k+1} zu schätzen, schließlich berechnen wir die Temperaturverteilung f (bei der FDM ist die Zustandsgröße direkt f).

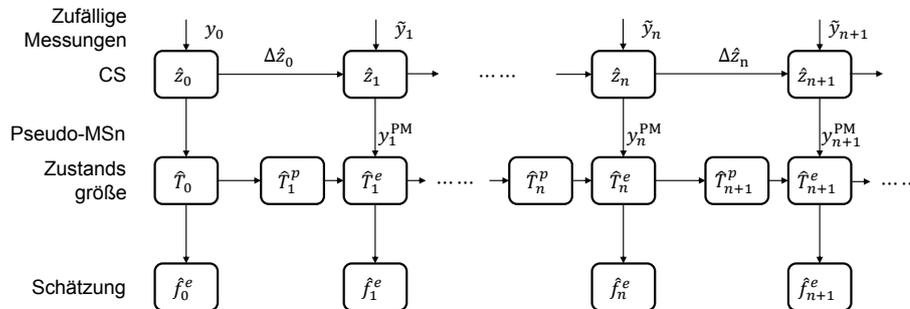


Abbildung 5.20: Verlauf des vorherigen KFCS.

Wie wir aus dem obenstehenden Diagramm ersehen können, ist der Grund für die Fehlerakkumulation darin zu sehen, dass es keine Rückkopplung im Prozess gibt, weshalb wir eine Rückkopplung einführen sollten, die es der Ausgangsgröße f ermöglicht, auf den Eingang (Compressive-Sensing) zu wirken.

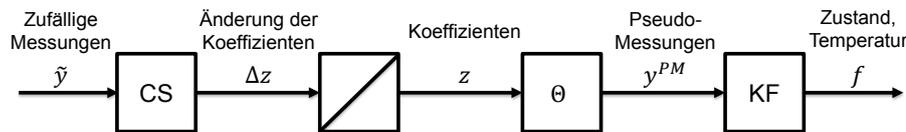


Abbildung 5.21: Verlauf des vorherigen KFCS.

Ausgehend von dieser Idee können wir den Prozess in die folgende Struktur 5.22 ändern

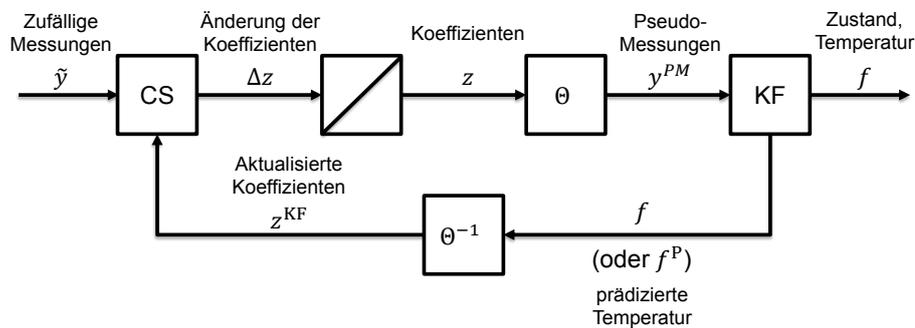


Abbildung 5.22: Strukturbild des KFCS mit Rückkopplung.

Dazu gibt es zwei Möglichkeiten: Koeffizientenaktualisierung aus der Schätzung und aus der Prädiktion.

5.7.1 Aktualisierung aus der Schätzung

Die erste Möglichkeit ist, wie in der Abbildung 5.23 gezeigt, dass wir die Temperaturverteilung \underline{f}_k nach der Schätzung der Zustandsgröße \underline{T}_k oder \underline{f}_k vom Kalman-Filter berechnen, und dann berechnen wir die Koeffizienten $\underline{z}_k^{\text{KF}}$ aus der Temperaturverteilung \underline{f}_k auf der Basis Θ . Dann verwenden wir $\underline{z}_k^{\text{KF}}$ als den Anfangswert, um $\Delta \underline{z}_k$ zu berechnen, wobei $\underline{z}_{k+1} = \underline{z}_k^{\text{KF}} + \Delta \underline{z}_k$.

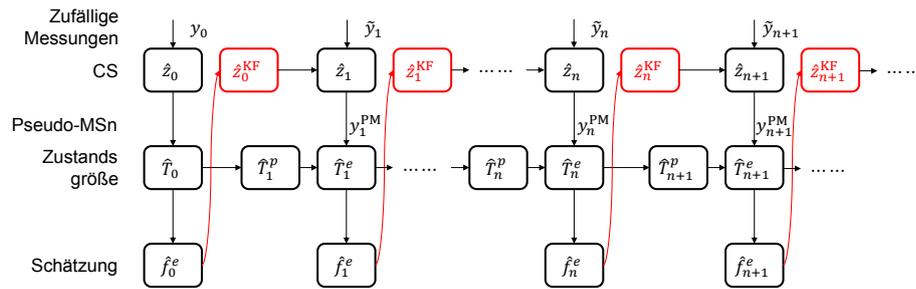


Abbildung 5.23: Verlauf für Koeffizientenaktualisierung aus der Schätzung.

Dieser Ansatz ist eher auf Fälle anwendbar, in denen die Anregung unbekannt ist. Für den Fall, dass die Anregung bekannt ist, können wir die Koeffizienten \underline{z}_k mit der Temperaturverteilung aus der Prädiktion aktualisieren, um bessere Ergebnisse zu erhalten.

5.7.2 Aktualisierung aus der Prädiktion

Die zweite Möglichkeit ist, wie in der Abbildung 5.24 gezeigt, dass wir die Temperaturverteilung \underline{f}_k aus der Prädiktion der Zustandsgröße $\underline{T}_k^{\text{P}}$ oder $\underline{f}_k^{\text{P}}$ vom Kalman-Filter berechnen, und dann berechnen wir die Koeffizienten $\underline{z}_k^{\text{P}}$ aus der prädizierten Temperaturverteilung auf der Basis Θ . Dann verwenden wir $\underline{z}_k^{\text{P}}$ als den Anfangswert, um $\Delta \underline{z}_k$ zu berechnen, wobei $\underline{z}_{k+1} = \underline{z}_k^{\text{P}} + \Delta \underline{z}_k$.

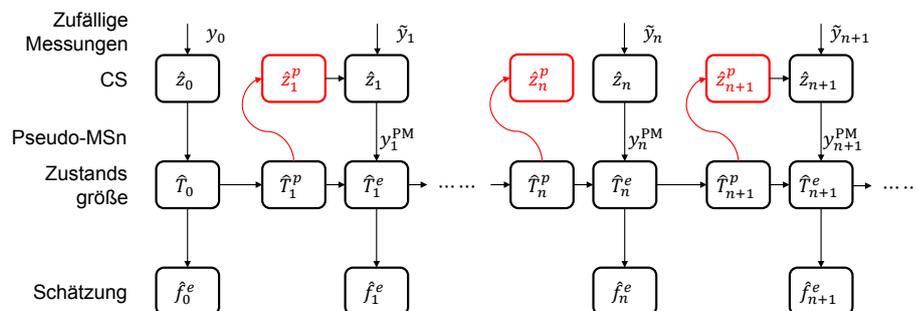


Abbildung 5.24: Verlauf für Koeffizientenaktualisierung aus der Prädiktion.

5.8 Iteratives Verfahren

Da dieses Verfahren nichtlinear ist, können wir auch ein iteratives Verfahren verwenden, um die Schätzung zu verbessern. Zu jedem Zeitpunkt k berechnen wir die Zustandsgrößen $(\underline{T}_k)^i$ und die Koeffizienten $(z_k)^i$ mit einer Beobachtung \tilde{y}_k und Prädiktion \hat{T}_k^P iterativ, bis sie konvergieren oder die maximale Anzahl von Iterationen erreicht ist. Die konvergierenden Koeffizienten $(z_k)^i$ werden als die z_k^{KF} zur Berechnung der Δz_{k+1} zum nächsten Zeitpunkt verwendet. Das Verfahren ist wie in der folgenden Grafik 5.25 gezeigt:

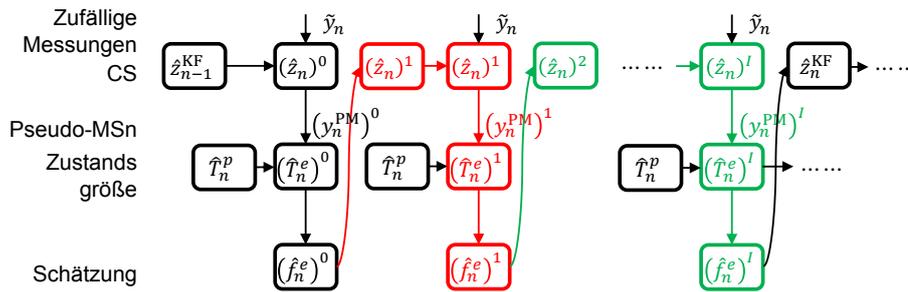


Abbildung 5.25: Verlauf für iteratives Verfahren.

5.9 Simulation

Wir rekonstruieren die Temperaturverteilung im vorherigen Beispiel mit verschiedenen Verfahren, nämlich dem Kalman-Filter mit 12, 40 und 64 äquidistanten Sensoren, das rsKF mit 12 Messungen mit 64 verfügbaren Sensoren, das KFCS (Kalman-Filter mit Pseudo-Messungen ohne Rückkoppelung) mit 12 Messungen mit 64 verfügbaren Sensoren, und das iterative KFCS mit 12 Messungen mit 64 verfügbaren Sensoren.

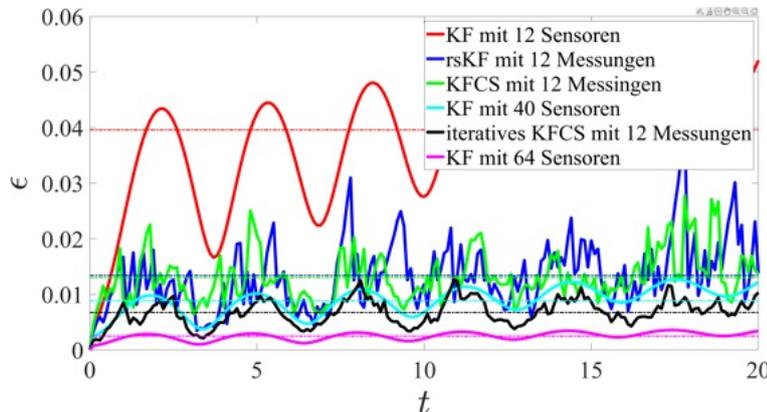


Abbildung 5.26: Vergleich aller Verfahren.

Wir können aus der Abbildung 5.26 sehen, dass das Ergebnis von rsKF mit 12 Messungen und KFCS mit 12 Messungen ähnlich sind. Der Grund ist, wie wir im Abschnitt 5.3.2 gesagt haben, wenn die dynamische Gewichtung der Pseudo-Messungen zu niedrig gewählt werden, sind die beiden Methoden gleichwertig. Aber nur, weil wir die Pseudo-Messungen eingeführt und die Struktur von CS-KF-CS aufgebaut haben, konnte das Verfahren im Vergleich zu rsKF noch weiter verbessert werden. Als Endergebnis können wir sehen, dass iteratives KFCS mit 12 Messungen bereits besseres Ergebnis liefert als das traditionelle Kalman-Filter mit 40 Sensoren. Das bedeutet, dass das neue Verfahren in der Arbeit sehr effektiv ist.

5.10 Zusammenfassung

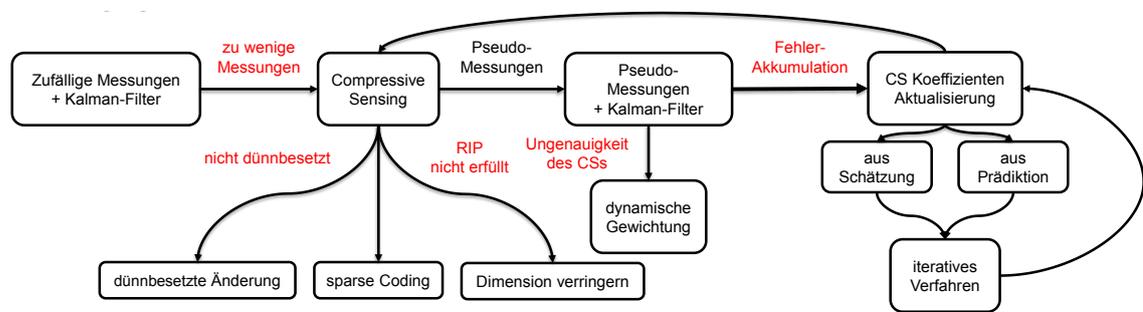


Abbildung 5.27: Vollständiger Gedankengang zu diesem Kapitel.

In diesem Kapitel untersuchten wir zunächst das Kalman-Filter mit zufälligen Messungen (rsKF). Und wir fanden den Nachteil von rsKF: Es ist möglich, dass es an einigen Stellen keine Messpunkte gibt. Deswegen wollen wir das Kalman-Filter und das Compressive-Sensing, also die zwei Vorwissen: das Systemmodell und die Dünnbesetztheit des Signals kombinieren. In diesem Fall haben wir zwei Ansätze versucht: die Kombination der Kalman-Gleichung mit der ℓ_p -Norm-Optimierung und die Kombination vom Compressive-Sensing und dem Kalman-Filter mittels Pseudo-Messungen. Tatsächlich müssen wir im ersten Ansatz die Gewichte der beiden Teile bestimmen, was in dem Fall, dass das echte Signal unbekannt ist, sehr schwierig ist. Deshalb verfolgen wir hauptsächlich den zweiten Ansatz. Beim zweiten Ansatz besteht jedoch eine hohe Anforderung an die Dünnbesetztheit des Signals, und das Ergebnis ist bei nicht-dünnbesetzten Signalen oft sehr schlecht, d. h. die Pseudo-Messungen haben wenigen Sinn. Dazu werden drei Verfahren vorgeschlagen, um die Anforderungen auf die Dünnbesetztheit des Signals abzuschwächen. Darüber hinaus schlagen wir zwei zusätzliche Methoden vor, um die Fusion der beiden Vorwissen zu verbessern und damit die Genauigkeit der Schätzung weiter zu erhöhen. Zunächst richten wir eine Rückkoppelung ein, d. h. wir aktualisieren die Koeffizienten \underline{z} aus

dem Ergebnis des Kalman-Filters (Schätzung oder Prädiktion) als Anfangswert vom Compressive-Sensing. Zweitens führen wir ein iteratives Verfahren ein, bei dem wir die Koeffizienten zu jedem Zeitpunkt iterativ berechnen, bis sie konvergieren. Durch diese Verfahren können wir mit 12 Messungen ein besseres Ergebnis im Vergleich zu traditionellem Kalman-Filter mit 40 Messungen erreichen.

Erweiterungen des Verfahrens

In den vorhergehenden Kapiteln haben wir eine vollständige Struktur entworfen, die es ermöglicht, das Kalman-Filter und das Compressive-Sensing vollständig zu fusionieren. Als Nächstes werden wir das Verfahren erweitern. Wir werden das Verfahren in rauschbehafteten Fällen und 2D-Fällen anwenden und schließlich werden wir die Fehler quantitativ bewerten.

6.1 KFCS für rauschbehaftete Signale

In der früheren Arbeit nehmen wir an, dass die Anregungen unbekannt sind, und daher stellen wir kein zusätzliches Systemrauschen hinzu. In diesem Abschnitt werden wir den Fall analysieren, in dem die Anregungen bekannt sind, während wir Systemrauschen hinzufügen. Und dann die von uns vorgeschlagene Verfahren zur Schätzung des Signals verwenden. Das ist die Temperaturverteilung mit Systemrauschen.

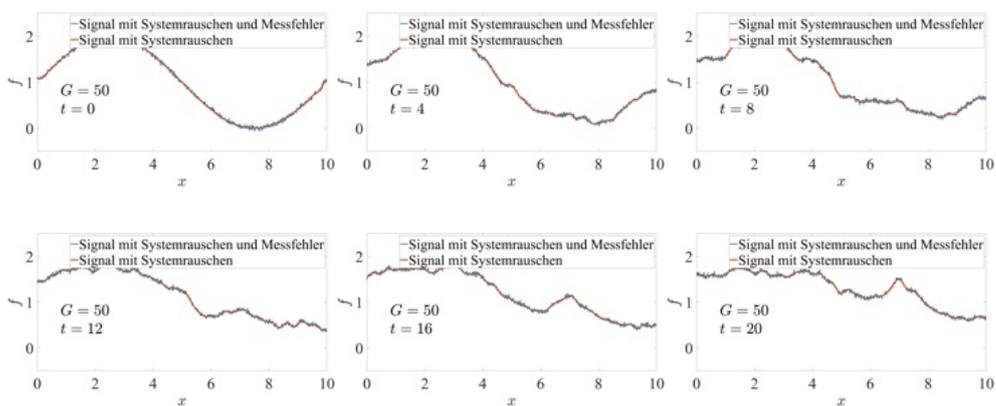


Abbildung 6.1: Temperaturverteilung mit Rauschen.

Hier ist das Ergebnis der Schätzung mit traditionellem Kalman-Filter mit 12 Messungen.

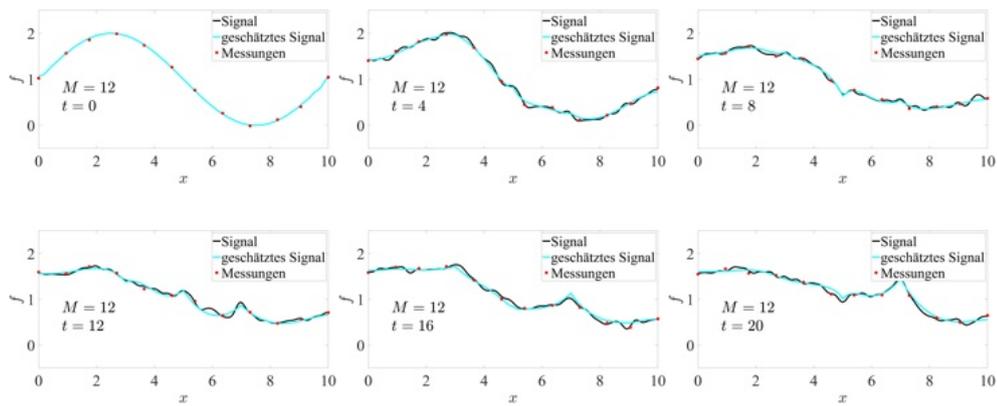


Abbildung 6.2: Kalman-Filter für rauschbehaftetes Signal mit 12 Sensoren.

Folgende Abbildung zeigt das Ergebnis aus dem Kalman-Filter mit 12 zufälligen Messungen.

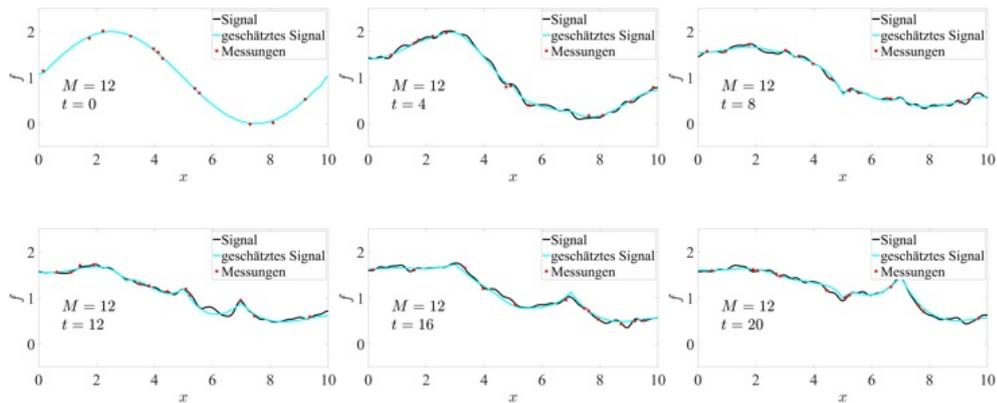


Abbildung 6.3: rsKF für rauschbehaftetes Signal mit 12 Messungen.

Folgende Abbildung zeigt das Ergebnis aus meinem Verfahren mit 12 zufälligen Messungen.

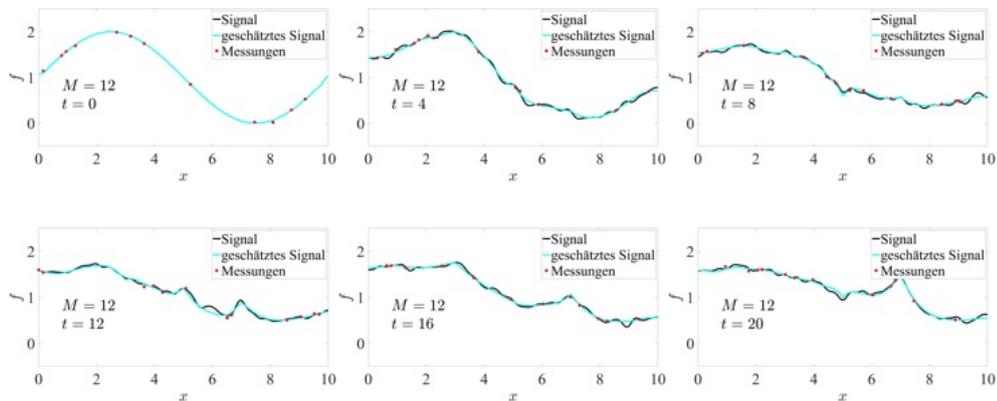


Abbildung 6.4: KFCS für rauschbehaftetes Signal mit 12 Messungen.

Es ist nicht anschaulich, durch die Abbildungen zu bestimmen, welches Verfahren besser ist. Deswegen vergleichen wir die Ergebnisse aus allen Verfahren quantitativ, wie in der Abbildung 6.5 dargestellt. Wir können sehen, dass unser Verfahren im Vergleich zum rauschfreien Fall relativ unwirksam ist. Aber im rauschbehafteten Fall gibt es noch eine Verbesserung um 25 Prozent im Vergleich zum traditionellen Kalman-Filter. Gleichzeitig können wir sehen, dass der Unterschied zwischen dem rsKF und dem Kalman-Filter in diesem Fall gering ist, weil die Anregung bekannt ist, d. h. selbst wenn es keine Messung an der Spitze gibt, können wir aus dem Systemmodell die Temperaturänderung hier noch erkennen.

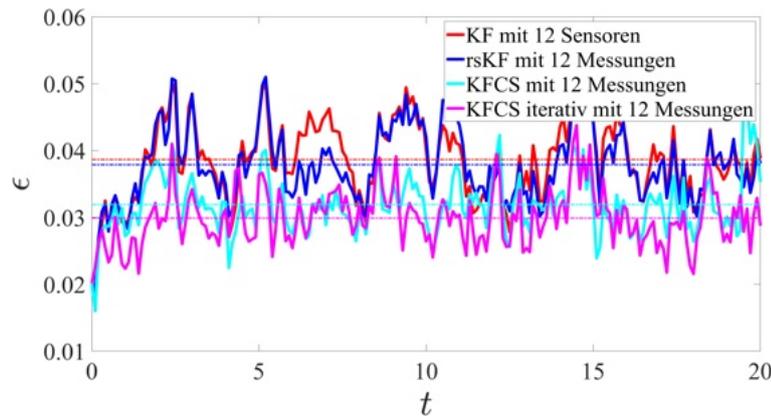


Abbildung 6.5: Vergleich für rauschbehaftetes Signal.

6.2 KFCS für 2D-Signale

In diesem Abschnitt werden wir das 2D-Signal mit dem KF und dem KFCS rekonstruieren.

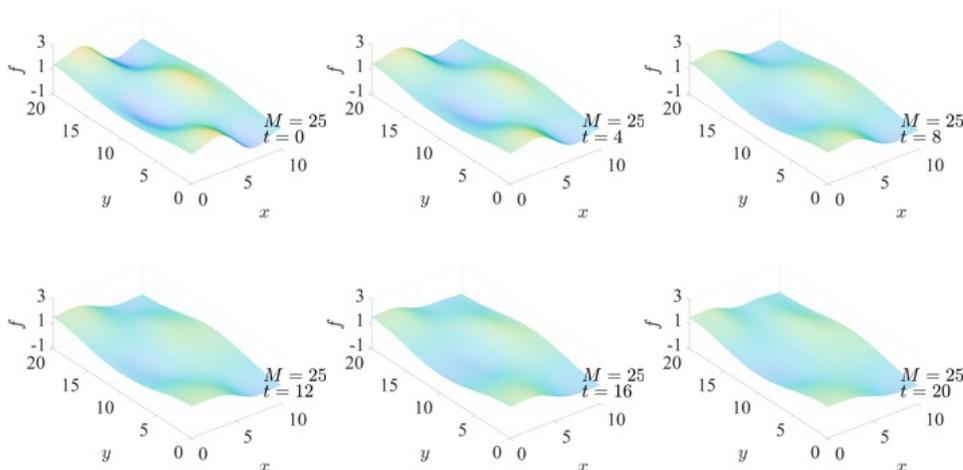


Abbildung 6.6: 2D-Kalman-Filter mit 25 Sensoren.

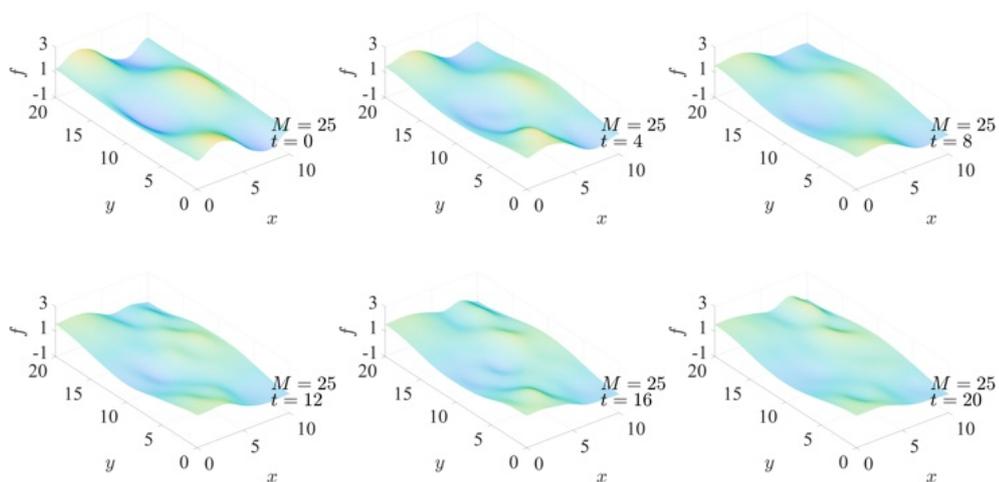


Abbildung 6.7: 2D-KFCS mit 25 Messungen.

Um die Ergebnisse anschaulicher zu vergleichen, haben wir die Abbildung 6.8 erstellt. Wir können aus der Abbildung feststellen, dass KFCS im 2D-Fall bessere Ergebnisse erzielt als das Kalman-Filter, und dass diese Verbesserung etwa 25 Prozent beträgt.

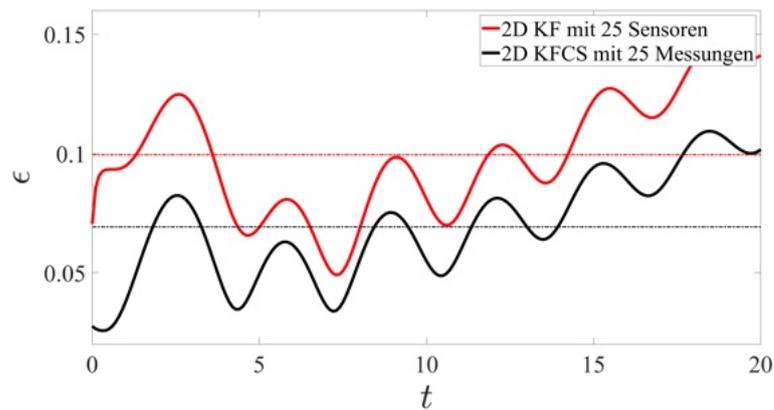


Abbildung 6.8: Vergleich für 2D-Signal.

6.3 Fehlerbewertung

Aus der vorherigen Simulation können wir erkennen, dass die Fehler in verschiedenen Fällen unterschiedlich sind und dass es unterschiedliche Grade gibt, wie gut oder schlecht das KFCS und das Kalman-Filter in verschiedenen Fällen sind. Deswegen werden 10 verschiedene Systemrauschen und 10 verschiedene Messunsicherheiten ausgewählt. Dann kombinieren wir sie und bekommen wir 100 verschiedene Kombinationen. Für jede Kombination werden 300 Simulationen durchgeführt, wobei das Signal mit dem Kalman-Filter bzw. dem KFCS rekonstruiert wird und die Fehler berechnet werden. Das Ergebnis wird in den folgenden Abbildungen gezeigt.

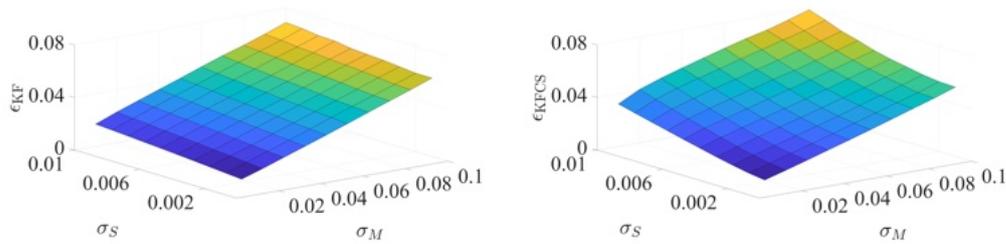


Abbildung 6.9: Fehler, links: Kalman-Filter, rechts: KFCS.

Um den Unterschied deutlicher zu sehen, haben wir ihre Unterschiede berechnet, also $\Delta\varepsilon = \varepsilon_{\text{KF}} - \varepsilon_{\text{KFCS}}$. Dann bekommen wir die folgende Abbildung

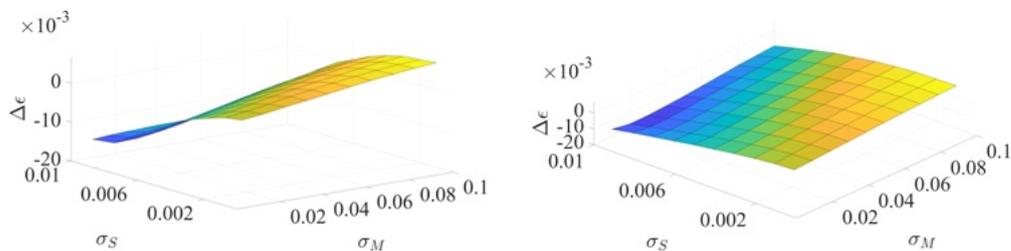


Abbildung 6.10: Differenzfehler von verschiedenen Aspekten.

Aus der Definition können wir wissen, dass das KFCS besser ist als das Kalman-Filter, wenn $\Delta\varepsilon$ positiv ist und umgekehrt. So können wir feststellen, dass das Kalman-Filter besser als KFCS ist, wenn das Systemrauschen relativ größer als das Messrauschen ist. Das KFCS ist besser als das Kalman-Filter, wenn das Messrauschen relativ größer als das Systemrauschen ist. Der Grund dafür ist, wenn das Systemrauschen groß ist, ist das Signal nicht mehr dünnbesetzt, d. h. das Vorwissen, dass das Signal dünnbesetzt ist, funktioniert nicht mehr. Und wenn das Systemrauschen sehr groß ist, kann das Compressive-Sensing sogar negative Auswirkung auf der Schätzung machen.

6.4 Zusammenfassung

In diesem Kapitel verwenden wir das Kalman-Filter und das KFCS im rauschbehafteten und 2D-Fall und wir analysieren die Fehler. Wir stellen fest, dass KFCS in diesen Fällen immer noch besser ist als das KF und dass es eine Verbesserung von etwa 25 Prozent gibt. Schließlich führten wir eine Fehlerbewertung von Signalen mit unterschiedlichem Rauschen durch und wir fanden heraus, dass das KFCS deutlich besser als das Kalman-Filter ist, wenn das Messrauschen relativ größer als das Systemrauschen ist. D. h. wenn wir das System genau modellieren können und

das Signal mit relativ schlechten Sensoren messen, können wir bessere Ergebnisse erhalten. Dies passt auch zu unserem Ziel, Energie zu sparen.

Zusammenfassung und Ausblick

7.1 Zusammenfassung

In dieser Arbeit haben wir die Wärmeleitung, das Kalman-Filter, das Compressive-Sensing, die Signalanalyse, die Informationsfusion und die Fehlerbewertung untersucht.

In dieser Arbeit wurde die Wirksamkeit des Verfahrens am Beispiel der Wärmeleitung überprüft. Deswegen haben wir zuerst die analytischen und numerischen Lösungen der Temperaturverteilung aus dem Diffusionsmodell hergeleitet und die verschiedenen Fälle simuliert.

Wir haben dann das Kalman-Filter hergeleitet und die Signalschätzung der Temperaturverteilung simuliert. Dann wurde das Compressive-Sensing vorgestellt und das Compressive-Sensing mit den 1D- und 2D-Signalen als Beispiele gezeigt.

In Bezug auf die Signalanalyse hat diese Arbeit die Eigenschaften von Signalen analysiert und drei Methoden vorgeschlagen, die es ermöglichen, das Compressive-Sensing auch auf nicht-dünnbesetzte Signale anzuwenden. Die Methoden sind nämlich dünnbesetzte Änderung, Dimensionreduzierung und Sparse-Coding-Representation.

Bei der Informationsfusion hat die Arbeit zwei Fusionsverfahren vorgeschlagen, Fusion durch die Kalman-Gleichung und durch die Pseudo-Messungen. In den ersten Ansatz ist die Gewichtung in der Praxis schwierig zu bestimmen, deswegen haben wir den zweiten Ansatz vorgeschlagen, also das Verfahren mit Pseudo-Messungen. In diesem Verfahren spielt die Gewichtung weniger Rolle. Darüber hinaus fusionieren wir das Kalman-Filter und das Compressive-Sensing weiter und wir setzen nicht nur das Ergebnis aus dem Compressive-Sensing in das Kalman-Filter ein, sondern wir führen das Ergebnis vom Kalman-Filter auch zu dem Compressive-Sensing zurück, wodurch die Struktur kompakter und vollständiger wird. Diese Arbeit hat dann ein iteratives Verfahren zur Überwindung der Nichtlinearitäten des Problems vorgeschlagen. Schließlich haben wir durch die Simulationen herausgefunden, dass das

neue Verfahren in idealen Fällen eine Verbesserung von etwa 70 Prozent gegenüber dem traditionellen Kalman-Filter bietet.

Schließlich hat diese Arbeit das Verfahren auf rauschbehaftete und 2D-Signale angewendet und festgestellt, dass das neue Verfahren in diesen Fällen eine Verbesserung von 25 Prozent bringt. Um den Zusammenhang zwischen den Rauschen und den Ergebnissen zu untersuchen, wurden in dieser Arbeit 100 verschiedene Kombinationen von Systemrauschen und Messunsicherheiten ausgewählt und 300 Simulationen für jede Kombination durchgeführt, um den durchschnittlichen Fehler zu berechnen. Durch unsere Analyse haben wir festgestellt, dass unser Verfahren deutlich besser ist als das Kalman-Filter, wenn das Systemrauschen relativ gering als die Messunsicherheit ist.

7.2 Ausblick

Die Arbeit ist aber noch nicht perfekt. In der Zukunft kann das Verfahren in folgenden Aspekten noch verbessert werden.

- Adaptive Pseudo-Messungen

In unserem Verfahren waren die Positionen der Pseudo-Messungen fest, z. B. wurden in dieser Arbeit die Pseudo-Messungen an den möglichen Sensorstellen ausgewählt. In zukünftigen Studien kann man versuchen, die Position der Pseudo-Messungen adaptiv zu wählen, z. B. zufällig gewählt und die Wahrscheinlichkeit der Pseudo-Messungen ist proportional zur Unsicherheit. Dadurch wird die Unsicherheit der Gesamtschätzung minimiert.

- Bestimmen der Fehlerverteilung des Compressive-Sensing

In dieser Arbeit wurde der Fehler vom Compressive-Sensing als Gauß'sche Verteilung behandelt. Tatsächlich ist die Fehlerverteilung vom Compressive-Sensing nicht eine Gauß'sche Verteilung, daher ist das Kalman-Filter nicht optimal. Obwohl wir versuchen, die optimale Lösung durch ein iteratives Verfahren zu finden, ist der Rechenaufwand sehr hoch. Daher kann man in der Zukunft die Fehlerverteilung, insbesondere die Kovarianzmatrix, vom Compressive-Sensing untersuchen, dann lässt sich KF besser einsetzen, oder ein besserer Filter entworfen wird, um die optimale Lösung direkt zu finden.

ANHANG

Anhang

A.1 Herleitung der Temperaturverteilung

A.1.1 Modalanalyse

Fangen wir zuerst mit dem 1D-Fall an.

Zunächst betrachten wir nur den homogenen Teil:

$$\frac{\partial f}{\partial t}(x, t) = k \frac{\partial^2 f}{\partial x^2}(x, t) \quad (\text{A.1})$$

Bei der Modalanalyse wird die Gleichung in einen ortsabhängigen Teil $X(x)$ und einen zeitabhängigen Teil $T(t)$ zerlegt, also

$$f(x, t) =: X(x)T(t) . \quad (\text{A.2})$$

Durch Einsetzen von (A.2) in die DGL (A.1) erhalten wir

$$\dot{T}(t)X(x) = kT(t)\ddot{X}(x) .$$

Wir separieren nun die Variablen und erhalten

$$\frac{1}{k} \frac{\dot{T}(t)}{T(t)} = \frac{\ddot{X}(x)}{X(x)} \stackrel{!}{=} C .$$

Es ergeben sich zwei gewöhnliche Differenzialgleichungen (DGLs):

- örtliche DGL

$$\ddot{X}(x) - CX(x) = 0 .$$

- zeitliche DGL

$$\dot{T}(t) - kCT(t) = 0 .$$

Lösen wir zuerst die örtliche DGL.

Für $C = 0$ gilt

$$\begin{aligned}\ddot{X}(x) &= 0 , \\ \dot{X}(x) &= B , \\ X(x) &= Bx + D .\end{aligned}$$

Setzen wir die Gleichung in 1. Randbedingung ein, gilt

$$\dot{X}(0) = B = 0 ,$$

d. h. $X(x)$ ist konstant.

Für $C > 0$ gilt

$$\begin{aligned}\ddot{X}(x) &= CX(x) , \\ X(x) &= B \exp \left\{ \sqrt{C}x \right\} + D \exp \left\{ -\sqrt{C}x \right\} , \\ \dot{X}(x) &= B\sqrt{C} \exp \left\{ \sqrt{C}x \right\} - D\sqrt{C} \exp \left\{ -\sqrt{C}x \right\} .\end{aligned}$$

Setzen wir die Gleichung in die erste Randbedingung ein, gilt

$$\dot{X}(0) = \sqrt{C}(B - D) = 0 ,$$

d. h. $B = D$. Dann gilt

$$X(x) = B \left(\exp \left\{ \sqrt{C}x \right\} + \exp \left\{ -\sqrt{C}x \right\} \right) .$$

Setzen wir die Gleichung in die zweite Randbedingung ein, gilt

$$\dot{X}(L) = B\sqrt{C} \underbrace{\left(\exp \left\{ \sqrt{C}L \right\} - \exp \left\{ -\sqrt{C}L \right\} \right)}_{>0} = 0 ,$$

d. h. $B = 0$, $X(x)$ ist konstant.

Für $C < 0$ gilt

$$\begin{aligned}X(x) &= B \sin \left(\underbrace{\sqrt{-C}x}_{=: \lambda} \right) + D \cos \left(\underbrace{\sqrt{-C}x}_{=: \lambda} \right) \\ &= B \sin (\lambda x) + D \cos (\lambda x) .\end{aligned}$$

Setzen wir die Gleichung in die erste Randbedingung ein, gilt

$$\dot{X}(0) = B\lambda = 0 ,$$

d. h. $B = 0$. Setzen wir die Gleichung in die zweite Randbedingung ein, gilt

$$\dot{X}(L) = -D\lambda \sin(\lambda L) = 0 .$$

Daraus folgt $D = 0$ oder

$$\lambda L = i\pi, \quad i = 0, 1, 2, \dots$$

Schließlich erhalten wir

$$X_i(x) = D \cos(\lambda_i x), \quad \lambda_i = i \frac{\pi}{L} . \quad (\text{A.3})$$

Nun formen wir die zeitliche DGL um.

$$\begin{aligned} \frac{1}{k} \frac{\dot{T}(t)}{T(t)} &= C = -\lambda^2 \\ \Rightarrow \dot{T}_i(t) &= -k\lambda_i^2 T_i(t) \end{aligned} \quad (\text{A.4})$$

Setzen wir (A.3) und (A.4) zusammen, erhalten wir

$$f(x, t) = \sum_{i=0}^{\infty} T_i(t) X_i(x) . \quad (\text{A.5})$$

Um (A.5) zu verallgemeinern, normalisieren wir nun die Funktionen $X(x)$:

$$\begin{aligned} f(x, t) &= \sum_{i=0}^{\infty} T_i(t) X_i(x) \\ &= \sum_{i=0}^{\infty} \underbrace{T_i(t)}_{=: T_i(t)} \underbrace{\frac{X_i(x)}{E_{X_i}}}_{=: \psi_i(x)} \\ &= \sum_{i=0}^{\infty} T_i(t) \psi_i(x) \end{aligned} \quad (\text{A.6})$$

, wobei E_{X_i} der Energie der Funktion X_i entspricht

$$E_{X_i} = \sqrt{\int_{x=0}^L |X_i(x)|^2 dx} .$$

Die normalisierten Funktionen $\psi_i(x)$ lauten:

$$\begin{aligned} \psi_0(x) &= \sqrt{\frac{1}{L}} , \\ \psi_i(x) &= \sqrt{\frac{2}{L}} \cos(\lambda_i x), \quad i = 1, 2, \dots . \end{aligned}$$

Sie erfüllen die Orthogonalitätsbedingung:

$$\int_0^L \psi_i(x)\psi_j(x)dx = \begin{cases} 1, & i = j , \\ 0, & i \neq j . \end{cases}$$

Deswegen bilden ψ_i eine Basis. Wir bezeichnen die ψ_i als Eigenfunktionen. Dementsprechend bezeichnen wir die T_i als Koeffizienten zu den ψ .

Jetzt leiten wir das Systemmodell aus der Modalanalyse her. Laut (A.4) und (A.6) und nach der zeitlichen und örtlichen Diskretisierung mit jeweils M und N Knoten sowie der Wahl von $i_{\max} = G$ (G -Grad-Approximation) erhalten wir:

$$\underbrace{\begin{bmatrix} T_1[k+1] \\ \vdots \\ T_G[k+1] \end{bmatrix}}_{=: \underline{T}_{k+1}} = \underbrace{\begin{bmatrix} 1 - \Delta tk \lambda_1^2 & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & 1 - \Delta tk \lambda_G^2 \end{bmatrix}}_{=: \mathbf{A}} \underbrace{\begin{bmatrix} T_1[k] \\ \vdots \\ T_G[k] \end{bmatrix}}_{=: \underline{T}_k},$$

$$\underbrace{\begin{bmatrix} f_1[k+1] \\ \vdots \\ f_N[k+1] \end{bmatrix}}_{=: \underline{f}_{k+1}} = \underbrace{\begin{bmatrix} \psi_1[1] & \cdots & \psi_G[1] \\ \vdots & \ddots & \vdots \\ \psi_1[N] & \cdots & \psi_G[N] \end{bmatrix}}_{=: \mathbf{\Psi}} \underbrace{\begin{bmatrix} T_1[k+1] \\ \vdots \\ T_G[k+1] \end{bmatrix}}_{=: \underline{T}_{k+1}}.$$

Also wir bekommen

$$\underline{T}_{k+1} = \mathbf{A} \cdot \underline{T}_k ,$$

$$\underline{f}_{k+1} = \mathbf{\Psi} \cdot \underline{T}_{k+1} .$$

Jetzt berechnen wir \underline{T}_0 . Da die ψ_i eine Basis bilden, können die Koeffizienten T_i durch das Innenprodukt (Projektion) von $f(x, t_0)$ und ψ_i berechnet werden, also

$$\begin{bmatrix} T_1[0] \\ \vdots \\ T_G[0] \end{bmatrix} = \begin{bmatrix} \psi_1[1] & \cdots & \psi_1[N] \\ \vdots & \ddots & \vdots \\ \psi_G[1] & \cdots & \psi_G[N] \end{bmatrix} \begin{bmatrix} f_1[0] \\ \vdots \\ f_N[0] \end{bmatrix} ,$$

d. h.

$$\underline{T}_0 = \mathbf{\Psi}^T \cdot \underline{f}_0 .$$

Dann leiten wir die numerische Lösung für den inhomogenen Fall (mit Anregung u) her. Das Diffusionsmodell mit Anregung lautet

$$\frac{\partial f}{\partial t}(x, t) = k \frac{\partial^2 f}{\partial x^2}(x, t) + u(x, t) .$$

Analog stellen wir die Anregung durch die Koeffizienten $U_i(t)$ und Eigenfunktionen ψ_i dar, also

$$u(x, t) = \sum_{i=0}^{\infty} U_i(t) \psi_i(x) .$$

Es gilt dann

$$\frac{\partial}{\partial t} \left(\sum_{i=0}^{\infty} T_i(t) \psi_i(x) \right) = k \frac{\partial^2}{\partial x^2} \left(\sum_{i=0}^{\infty} T_i(t) \psi_i(x) \right) + \sum_{i=0}^{\infty} U_i(t) \psi_i(x) ,$$

also

$$\sum_{i=0}^{\infty} \dot{T}_i(t) \psi_i(x) = k \sum_{i=0}^{\infty} T_i(t) \psi_i''(x) + \sum_{i=0}^{\infty} U_i(t) \psi_i(x) . \quad (\text{A.7})$$

Da

$$\begin{aligned} \psi_i(x) &= \begin{cases} \sqrt{\frac{1}{L}}, & i = 0 , \\ \sqrt{\frac{2}{L}} \cos(\lambda_i x), & i = 1, 2, \dots , \end{cases} \\ \psi_i'(x) &= -\lambda_i \sqrt{\frac{2}{L}} \sin(\lambda_i x) , \\ \psi_i''(x) &= -\lambda_i^2 \sqrt{\frac{2}{L}} \cos(\lambda_i x) = -\lambda_i^2 \psi_i(x) , \end{aligned} \quad (\text{A.8})$$

gilt sich aus (A.7) und (A.8)

$$\begin{aligned} \sum_{i=0}^{\infty} \dot{T}_i(t) \psi_i(x) &= \sum_{i=0}^{\infty} (-k) \lambda_i^2 T_i(t) \psi_i(x) + \sum_{i=0}^{\infty} U_i(t) \psi_i(x) , \\ \sum_{i=0}^{\infty} \dot{T}_i(t) \psi_i(x) &= \sum_{i=0}^{\infty} (-k \lambda_i^2 T_i(t) + U_i(t)) \psi_i(x) , \\ \dot{T}_i(t) &= -k \lambda_i^2 T_i(t) + U_i(t) . \end{aligned} \quad (\text{A.9})$$

Laut (A.9) erhalten wir das Systemmodell für den inhomogenen Fall, also

$$\begin{aligned} \underline{T}_{k+1} &= \mathbf{A} \cdot \underline{T}_k + \Delta t \underbrace{\begin{bmatrix} U_1[k] \\ \vdots \\ U_G[k] \end{bmatrix}}_{=: \underline{U}_k} \\ &= \mathbf{A} \cdot \underline{T}_k + \underline{U}_k , \\ \underline{f}_{k+1} &= \mathbf{\Psi} \cdot \underline{T}_{k+1} . \end{aligned}$$

Die Berechnung von \underline{U}_k ist analog wie die von T_0 , also

$$\underline{U}_k = \Delta t \begin{bmatrix} U_1[k] \\ \vdots \\ U_G[k] \end{bmatrix} = \Delta t \cdot \Psi^T \begin{bmatrix} u_1[k] \\ \vdots \\ u_N[k] \end{bmatrix} .$$

Nun Lösen wir die Temperaturverteilung in den 2D-Fall.

Das Diffusionsmodell für den 2D-Fall ist

$$\frac{\partial f}{\partial t}(x, y, t) = k \left(\frac{\partial^2 f}{\partial x^2}(x, y, t) + \frac{\partial^2 f}{\partial y^2}(x, y, t) \right) + u(x, y, t) . \quad (\text{A.10})$$

Separieren wir $f(x, y, t)$ in $f(x, y, t) = X(x)Y(y)T(t)$, erhalten wir

$$f(x, y, t) = \sum_{i=0}^{\infty} \sum_{j=0}^{\infty} T_{i,j}(t) \psi_i(x) \varphi_j(y)$$

und

$$X(x)Y(y)\dot{T}(t) = kT(t) \left(\ddot{X}(x)Y(y) + X(x)\ddot{Y}(y) \right) .$$

Dann ergibt sich

$$\frac{1}{k} \frac{\dot{T}(t)}{T(t)} = \frac{\ddot{X}(x)}{X(x)} + \frac{\ddot{Y}(y)}{Y(y)} = C . \quad (\text{A.11})$$

Um eine Lösung zu finden, setzen wir nun zunächst

$$\frac{\ddot{X}(x)}{X(x)} = K \frac{\ddot{Y}(y)}{Y(y)}, \quad K = \text{Konst.}$$

Dann ergibt sich

$$\frac{1}{k} \frac{\dot{T}(t)}{T(t)} = (1 + K) \frac{\ddot{X}(x)}{X(x)} = C ,$$

also

$$\underbrace{\frac{1}{k(1+K)}}_{=:k'} \frac{\dot{T}(t)}{T(t)} = \frac{\ddot{X}(x)}{X(x)} = \underbrace{\frac{C}{1+K}}_{=:C'} .$$

Analog wie im 1D-Fall erhalten wir schließlich zwei DGL jeweils in x - und y -Richtung.

$$\begin{aligned} X(x) &= D_x \cos(\lambda_i x), & \lambda_i &= i \frac{\pi}{L_x} , \\ Y(y) &= D_y \cos(\sigma_j y), & \sigma_j &= j \frac{\pi}{L_y} . \end{aligned}$$

In diesem Fall gilt K

$$K_{ij} = \frac{\ddot{X}_i(x)/X_i(x)}{\ddot{Y}_j(y)/Y_j(y)} = \frac{\lambda_i^2}{\sigma_j^2} .$$

Nun normalisieren wir die beiden örtlichen Funktionen:

- Eigenfunktionen in x -Richtung:

$$\psi_i(x) = \begin{cases} \sqrt{\frac{1}{L_x}}, & i = 0, \\ \sqrt{\frac{2}{L_x}} \cos(\lambda_i x), & i = 1, 2, \dots \end{cases}$$

- die Eigenfunktionen in y -Richtung:

$$\varphi_j(y) = \begin{cases} \sqrt{\frac{1}{L_y}}, & j = 0, \\ \sqrt{\frac{2}{L_y}} \cos(\lambda_j y), & j = 1, 2, \dots \end{cases}$$

Setzen wir die Eigenfunktionen in (A.11) ein, erhalten wir

$$\dot{T}_{i,j}(t) = -k (\lambda_i^2 + \sigma_j^2) T_{i,j}(t) .$$

Betrachten wir nun die Anregung. Stellen wir die Anregung $u(x, y, t)$ durch Eigenfunktionen dar:

$$u(x, y, t) = \sum_{i=0}^{\infty} \sum_{j=0}^{\infty} U_{i,j}(t) \psi_i(x) \varphi_j(y)$$

Wegen (A.10) gilt dann

$$\dot{T}_{i,j}(t) = - \underbrace{k (\lambda_i^2 + \sigma_j^2)}_{=:\kappa_{i,j}} T_{i,j} + U_{i,j}(t) .$$

Bevor wir das Systemmodell beschreiben, nummerieren wir zuerst die Knoten. D. h. wir führen eine Umformung

$$\begin{aligned} \mathbf{T} \in \mathbb{R}^{G_x \times G_y} &\mapsto \underline{T} \in \mathbb{R}^{G_x G_y \times 1}, \\ \mathbf{f} \in \mathbb{R}^{N_x \times N_y} &\mapsto \underline{f} \in \mathbb{R}^{N_x N_y \times 1}, \end{aligned}$$

durch

$$\begin{aligned} \underline{T}_{index} &= \mathbf{T}_{i,j}, \text{ mit } index := i + (j - 1)G_x, \\ \underline{f}_{index} &= \mathbf{f}_{i,j}, \text{ mit } index := i + (j - 1)N_x, \end{aligned}$$

wie gezeigt in der Abbildung A.1.

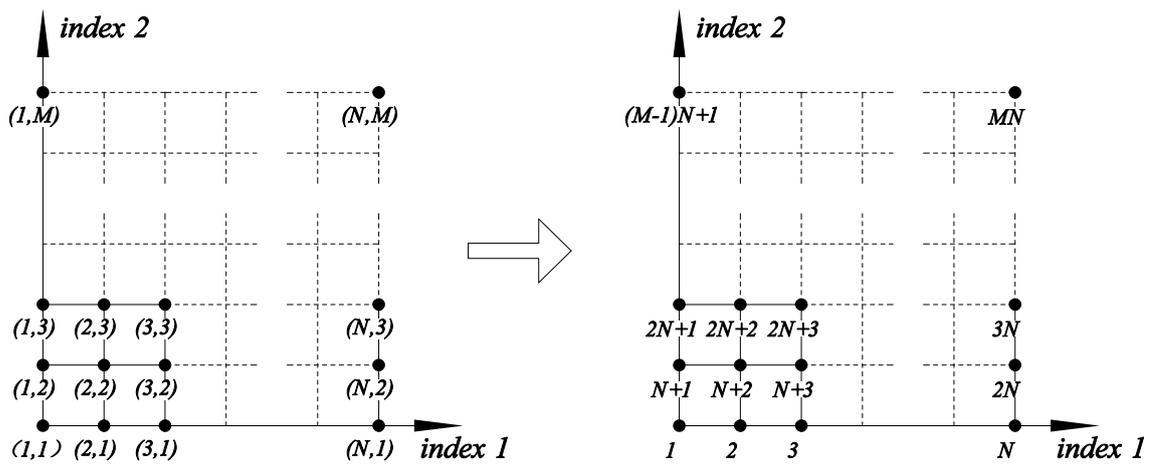


Abbildung A.1: Nummerierung

Schließlich erhalten wir nach der Diskretisierung das Systemmodell

$$\underline{T}_{k+1} = \mathbf{A} \cdot \underline{T}_k + \underline{U}_k ,$$

$$\underline{f}_{k+1} = \Psi \cdot \underline{T}_{k+1} ,$$

wobei

$$\begin{aligned} \underline{T}_k &= \left[T_1[K] \quad \cdots \quad T_{G_x G_y}[K] \right]^T \\ &= \left[T_{1,1}[K] \quad \cdots \quad T_{G_x,1}[K] \quad \cdots \quad T_{1,G_y}[K] \quad \cdots \quad T_{G_x,G_y}[K] \right]^T , \\ \mathbf{A} &= \begin{bmatrix} 1 - \kappa_{1,1} & & & & & \underline{0} \\ & \ddots & & & & \\ & & 1 - \kappa_{G_x,1} & & & \\ & & & \ddots & & \\ & & & & 1 - \kappa_{1,G_y} & \\ & & & & & \ddots \\ \underline{0} & & & & & & 1 - \kappa_{G_x,G_y} \end{bmatrix} , \\ \underline{U}_k &= \Delta t \cdot \left[U_1[K] \quad \cdots \quad U_{G_x G_y}[K] \right]^T \\ &= \Delta t \cdot \left[U_{1,1}[K] \quad \cdots \quad U_{G_x,1}[K] \quad \cdots \quad U_{1,G_y}[K] \quad \cdots \quad U_{G_x,G_y}[K] \right]^T , \end{aligned}$$

$$\begin{aligned}
\Psi &= \begin{bmatrix} \psi_1[1]\varphi_1[1] & \cdots & \psi_{G_x}[1]\varphi_1[1] & \cdots & \psi_1[1]\varphi_{G_y}[1] & \cdots & \psi_{G_x}[1]\varphi_{G_y}[1] \\ \vdots & \ddots & \vdots & & \vdots & \ddots & \vdots \\ \psi_1[N_x]\varphi_1[1] & \cdots & \psi_{G_x}[N_x]\varphi_1[1] & \cdots & \psi_1[N_x]\varphi_{G_y}[1] & \cdots & \psi_{G_x}[N_x]\varphi_{G_y}[1] \\ \vdots & & \vdots & \ddots & \vdots & & \vdots \\ \psi_1[1]\varphi_1[N_y] & \cdots & \psi_{G_x}[1]\varphi_1[N_y] & \cdots & \psi_1[1]\varphi_{G_y}[N_y] & \cdots & \psi_{G_x}[1]\varphi_{G_y}[N_y] \\ \vdots & \ddots & \vdots & & \vdots & \ddots & \vdots \\ \psi_1[N_x]\varphi_1[N_y] & \cdots & \psi_{G_x}[N_x]\varphi_1[N_y] & \cdots & \psi_1[N_x]\varphi_{G_y}[N_y] & \cdots & \psi_{G_x}[N_x]\varphi_{G_y}[N_y] \end{bmatrix} \\
&= \underbrace{\begin{bmatrix} \varphi_1[1] & \cdots & \varphi_{G_y}[1] \\ \vdots & \ddots & \vdots \\ \varphi_1[N_y] & \cdots & \varphi_{G_y}[N_y] \end{bmatrix}}_{\Psi_y} \otimes \underbrace{\begin{bmatrix} \psi_1[1] & \cdots & \psi_{G_x}[1] \\ \vdots & \ddots & \vdots \\ \psi_1[N_x] & \cdots & \psi_{G_x}[N_x] \end{bmatrix}}_{\Psi_x}.
\end{aligned}$$

Hierbei steht das Zeichen „ \otimes “ für das Kronecker-Produkt, mit dem wir die Messabbildung Ψ klarer und einfacher beschreiben können.

A.1.2 Finite-Differenzen-Verfahren

Beim Finite-Differenzen-Verfahren approximieren wir die partiellen Terme durch eine Taylor-Entwicklung, also

$$\begin{aligned}
\frac{\partial f}{\partial t}(x[n], t[k]) &\approx \frac{f(x[n], t[k]) - f(x[n], t[k-1])}{\Delta t}, \\
\frac{\partial^2 f}{\partial x^2}(x[n], t[k]) &\approx \frac{f(x[n-1], t[k]) - 2f(x[n], t[k]) + f(x[n+1], t[k])}{\Delta x^2}.
\end{aligned}$$

In Kurzschreibweise:

$$\frac{\partial f_n^k}{\partial t} \approx \frac{f_n^k - f_n^{k-1}}{\Delta t}, \quad (\text{A.12})$$

$$\frac{\partial^2 f_n^k}{\partial x^2} \approx \frac{f_{n-1}^k - 2f_n^k + f_{n+1}^k}{\Delta x^2}. \quad (\text{A.13})$$

Fügen (A.12) und (A.13) in das Diffusionsmodell ein, ergibt sich dann

$$\frac{f_n^k - f_n^{k-1}}{\Delta t} = k \frac{f_{n-1}^k - 2f_n^k + f_{n+1}^k}{\Delta x^2} + u_n^k.$$

Formen wir die Gleichung um, bekommen wir

$$-f_n^k + \underbrace{\frac{\Delta x^2}{\Delta x^2 + 2k\Delta t}}_{=:A} f_n^{k-1} + \underbrace{\frac{k\Delta t}{\Delta x^2 + 2k\Delta t}}_{=:B} f_{n-1}^k + \underbrace{\frac{k\Delta t}{\Delta x^2 + 2k\Delta t}}_{=:B} f_{n+1}^k = \underbrace{\frac{\Delta x^2 \Delta t}{\Delta x^2 + 2k\Delta t}}_{=:C} u_n^k.$$

Laut Taylorentwicklung

$$\begin{aligned}\frac{\partial f_{m,n}^k}{\partial t} &\approx \frac{f_{m,n}^k - f_{m,n}^{k-1}}{\Delta t}, \\ \frac{\partial^2 f_{m,n}^k}{\partial x^2} &\approx \frac{f_{m-1,n}^k - 2f_{m,n}^k + f_{m+1,n}^k}{\Delta x^2}, \\ \frac{\partial^2 f_{m,n}^k}{\partial y^2} &\approx \frac{f_{m,n-1}^k - 2f_{m,n}^k + f_{m,n+1}^k}{\Delta y^2},\end{aligned}$$

ergibt sich dann für alle Knoten außer den Knoten an den Rändern, also

$$-f_{m,n}^k + Af_{m,n}^{k-1} + Bf_{m,n-1}^k + Bf_{m,n+1}^k + Cf_{m-1,n}^k + Cf_{m+1,n}^k = Du_{m,n}^k.$$

Für die Randbedingungen gilt

$$\begin{aligned}-f_{2,n}^k + f_{1,n}^k &= 0, \\ -f_{M,n}^k + f_{M-1,n}^k &= 0, \\ -f_{m,1}^k + f_{m,2}^k &= 0, \\ -f_{m,N}^k + f_{m,N-1}^k &= 0.\end{aligned}$$

Nach der Nummerierung wie gezeigt in der Abbildung A.2

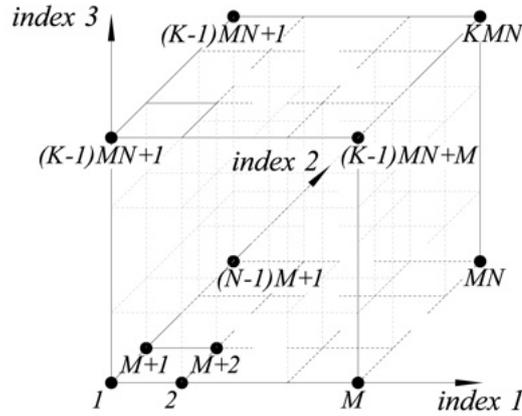


Abbildung A.2: Nummerierung

erhalten wir ein LGS in der Form von

$$\mathbf{H} \cdot \underline{f} = \underline{b}.$$

Dann können wir die Temperaturverteilung nach der Lösung dieses LGS.

A.2 Herleitung des Kalman-Filters

Es wird so angenommen, dass das Systemmodell

$$\begin{aligned}\underline{x}_{k+1} &= \mathbf{A} \cdot \underline{x}_k + \mathbf{B} \cdot u_k + \underline{w}_k \\ \underline{y}_{k+1} &= \mathbf{H}_{k+1} \cdot \underline{x}_{k+1} + \underline{v}_{k+1}\end{aligned}$$

bekannt ist. Die Systemrauschen \underline{w}_k und die Messrauschen \underline{v}_k sind Gaußsche Verteilungen mit

$$\begin{aligned}E\{\underline{w}_k\} &= 0, & \text{Cov}\{\underline{w}_i, \underline{w}_j\} &= \mathbf{C}_k^w \cdot \delta_{i,j} \\ E\{\underline{v}_k\} &= 0, & \text{Cov}\{\underline{v}_i, \underline{v}_j\} &= \mathbf{C}_k^v \cdot \delta_{i,j}\end{aligned}$$

1. Schritt: Prädiktion

Die Prädiktion ist Erwartungstreue, d. h.

$$\begin{aligned}\hat{\underline{x}}_{k+1}^p &= E\left\{\underline{x}_{k+1} | \underline{y}_k\right\} \\ &= E\left\{\mathbf{A}_k \cdot \underline{x}_k + \mathbf{B}_k \cdot \underline{u}_k + \underline{w}_k | \underline{y}_k\right\} \\ &= \mathbf{A}_k \cdot E\left\{\underline{x}_k | \underline{y}_k\right\} + \mathbf{B}_k \cdot E\left\{\underline{u}_k | \underline{y}_k\right\} + E\left\{\underline{w}_k | \underline{y}_k\right\} \\ &= \mathbf{A}_k \cdot E\left\{\underline{x}_k | \underline{y}_k\right\} + \mathbf{B}_k \cdot E\left\{\underline{u}_k | \underline{y}_k\right\} \\ &= \mathbf{A}_k \cdot \hat{\underline{x}}_k^e + \mathbf{B}_k \cdot \underline{u}_k\end{aligned}$$

Die Fehlerkovarianzmatrix nach der Prädiktion:

Da

$$\begin{aligned}\underline{x}_{k+1} &= \mathbf{A}_k \cdot \underline{x}_k + \mathbf{B}_k \cdot \underline{u}_k + \underline{w}_k \\ \hat{\underline{x}}_{k+1}^p &= \mathbf{A}_k \cdot \hat{\underline{x}}_k^e + \mathbf{B}_k \cdot \underline{u}_k\end{aligned}$$

gilt

$$\underline{x}_{k+1} - \hat{\underline{x}}_{k+1}^p = \mathbf{A}_k \cdot (\underline{x}_k - \hat{\underline{x}}_k^e) + \underline{w}_k$$

dann gilt

$$\begin{aligned}
\mathbf{C}_{k+1}^p &= E \left\{ (\underline{x}_{k+1} - \hat{\underline{x}}_{k+1}^p) (\underline{x}_{k+1} - \hat{\underline{x}}_{k+1}^p)^\top | \underline{y}_k \right\} \\
&= E \left\{ (\mathbf{A}_k \cdot (\underline{x}_k - \hat{\underline{x}}_k^e) + \underline{w}_k) (\mathbf{A}_k \cdot (\underline{x}_k - \hat{\underline{x}}_k^e) + \underline{w}_k)^\top | \underline{y}_k \right\} \\
&= E \left\{ (\mathbf{A}_k \cdot (\underline{x}_k - \hat{\underline{x}}_k^e) + \underline{w}_k) \left((\underline{x}_k - \hat{\underline{x}}_k^e)^\top \cdot \mathbf{A}_k^\top + \underline{w}_k^\top \right) | \underline{y}_k \right\} \\
&= E \left\{ \mathbf{A}_k \cdot (\underline{x}_k - \hat{\underline{x}}_k^e) \cdot (\underline{x}_k - \hat{\underline{x}}_k^e)^\top \cdot \mathbf{A}_k^\top + \underline{w}_k \underline{w}_k^\top | \underline{y}_k \right\} \\
&= E \left\{ \mathbf{A}_k \cdot (\underline{x}_k - \hat{\underline{x}}_k^e) \cdot (\underline{x}_k - \hat{\underline{x}}_k^e)^\top \cdot \mathbf{A}_k^\top | \underline{y}_k \right\} + E \left\{ \underline{w}_k \underline{w}_k^\top | \underline{y}_k \right\} \\
&= \mathbf{A}_k \cdot E \left\{ (\underline{x}_k - \hat{\underline{x}}_k^e) \cdot (\underline{x}_k - \hat{\underline{x}}_k^e)^\top | \underline{y}_k \right\} \cdot \mathbf{A}_k^\top + E \left\{ \underline{w}_k \underline{w}_k^\top | \underline{y}_k \right\} \\
&= \mathbf{A}_k \cdot \mathbf{C}_k^e \cdot \mathbf{A}_k^\top + \mathbf{C}_k^w
\end{aligned}$$

2. Schritt: Filterung

Erwartungsstreuung

$$\begin{aligned}
\hat{\underline{x}}_{k+1}^e &= \mathbf{K}_{k+1}^p \cdot \hat{\underline{x}}_{k+1}^p + \mathbf{K}_{k+1} \cdot \underline{y}_{k+1} \\
E \left\{ \hat{\underline{x}}_{k+1}^e \right\} &= E \left\{ \mathbf{K}_{k+1}^p \cdot \hat{\underline{x}}_{k+1}^p + \mathbf{K}_{k+1} \cdot \underline{y}_{k+1} \right\} \\
&= E \left\{ \mathbf{K}_{k+1}^p \cdot \hat{\underline{x}}_{k+1}^p + \mathbf{K}_{k+1} \cdot (\mathbf{H}_{k+1} \cdot \underline{x}_{k+1} + \underline{v}_{k+1}) \right\} \\
&= \mathbf{K}_{k+1}^p \cdot E \left\{ \hat{\underline{x}}_{k+1}^p \right\} + \mathbf{K}_{k+1} \cdot \mathbf{H}_{k+1} \cdot E \left\{ \underline{x}_{k+1} \right\} + \mathbf{K}_{k+1} \cdot E \left\{ \underline{v}_{k+1} \right\} \\
&= (\mathbf{K}_{k+1}^p + \mathbf{K}_{k+1} \cdot \mathbf{H}_{k+1}) \cdot E \left\{ \underline{x}_{k+1} \right\} \\
&\stackrel{!}{=} E \left\{ \underline{x}_{k+1} \right\}
\end{aligned}$$

d. h.

$$\mathbf{K}_{k+1}^p = \mathbf{I} - \mathbf{K}_{k+1} \cdot \mathbf{H}_{k+1}$$

also

$$\begin{aligned}
\hat{\underline{x}}_{k+1}^e &= \mathbf{K}_{k+1}^p \cdot \hat{\underline{x}}_{k+1}^p + \mathbf{K}_{k+1} \cdot \underline{y}_{k+1} \\
&= (\mathbf{I} - \mathbf{K}_{k+1} \cdot \mathbf{H}_{k+1}) \cdot \hat{\underline{x}}_{k+1}^p + \mathbf{K}_{k+1} \cdot \underline{y}_{k+1}
\end{aligned}$$

Die Fehlerkovarianzmatrix nach der Filterung:

Es gilt

$$\begin{aligned}
\underline{x}_{k+1} - \hat{\underline{x}}_{k+1}^e &= \underline{x}_{k+1} - (\mathbf{I} - \mathbf{K}_{k+1} \cdot \mathbf{H}_{k+1}) \cdot \hat{\underline{x}}_{k+1}^p - \mathbf{K}_{k+1} \cdot \underline{y}_{k+1} \\
&= \underline{x}_{k+1} - (\mathbf{I} - \mathbf{K}_{k+1} \cdot \mathbf{H}_{k+1}) \cdot \hat{\underline{x}}_{k+1}^p - \mathbf{K}_{k+1} \cdot (\mathbf{H}_{k+1} \cdot \underline{x}_{k+1} + \underline{v}_{k+1}) \\
&= \underline{x}_{k+1} - (\mathbf{I} - \mathbf{K}_{k+1} \cdot \mathbf{H}_{k+1}) \cdot \hat{\underline{x}}_{k+1}^p - \mathbf{K}_{k+1} \cdot \mathbf{H}_{k+1} \cdot \underline{x}_{k+1} - \mathbf{K}_{k+1} \cdot \underline{v}_{k+1} \\
&= (\mathbf{I} - \mathbf{K}_{k+1} \cdot \mathbf{H}_{k+1}) \cdot \underline{x}_{k+1} - (\mathbf{I} - \mathbf{K}_{k+1} \cdot \mathbf{H}_{k+1}) \cdot \hat{\underline{x}}_{k+1}^p - \mathbf{K}_{k+1} \cdot \underline{v}_{k+1} \\
&= (\mathbf{I} - \mathbf{K}_{k+1} \cdot \mathbf{H}_{k+1}) \cdot (\underline{x}_{k+1} - \hat{\underline{x}}_{k+1}^p) - \mathbf{K}_{k+1} \cdot \underline{v}_{k+1}
\end{aligned}$$

dann gilt

$$\begin{aligned}\mathbf{C}_{k+1}^e &= E \left\{ (\underline{x}_{k+1} - \hat{\underline{x}}_{k+1}^e) (\underline{x}_{k+1} - \hat{\underline{x}}_{k+1}^e)^T \right\} \\ &= (\mathbf{I} - \mathbf{K}_{k+1} \mathbf{H}_{k+1}) \mathbf{C}_{k+1}^p (\mathbf{I} - \mathbf{K}_{k+1} \mathbf{H}_{k+1})^T + \mathbf{K}_{k+1} \mathbf{C}_{k+1}^v \mathbf{K}_{k+1}^T\end{aligned}\quad (\text{A.14})$$

Bestimmung der Verstärkungsmatrix \mathbf{K}_{k+1} :

Es ist offensichtlich:

$$\begin{aligned}\text{Spur}(\mathbf{C}_{k+1}^e) &= \text{Spur} \left(E \left\{ (\underline{x}_{k+1} - \hat{\underline{x}}_{k+1}^e) (\underline{x}_{k+1} - \hat{\underline{x}}_{k+1}^e)^T \right\} \right) \\ &= E \left\{ \text{Spur} \left((\underline{x}_{k+1} - \hat{\underline{x}}_{k+1}^e) (\underline{x}_{k+1} - \hat{\underline{x}}_{k+1}^e)^T \right) \right\} \\ &= E \left\{ \text{Spur} \left((\underline{x}_{k+1} - \hat{\underline{x}}_{k+1}^e)^T (\underline{x}_{k+1} - \hat{\underline{x}}_{k+1}^e) \right) \right\} \\ &= E \left\{ \|\underline{x}_{k+1} - \hat{\underline{x}}_{k+1}^e\|_2^2 \right\}\end{aligned}$$

Deswegen können wir durch die Minimierung von $\text{Spur}(\mathbf{C}_{k+1}^e)$ die Verstärkungsmatrix \mathbf{K}_{k+1} bestimmen.

$$\begin{aligned}\frac{\partial}{\partial \mathbf{K}_{k+1}} \text{Spur}(\mathbf{C}_{k+1}^e) \\ = -2(\mathbf{I} - \mathbf{K}_{k+1} \cdot \mathbf{H}_{k+1}) \cdot \mathbf{C}_{k+1}^p \cdot \mathbf{H}_{k+1}^T + 2\mathbf{K}_{k+1} \cdot \mathbf{C}_{k+1}^v \stackrel{!}{=} \mathbf{0}\end{aligned}$$

Es gilt dann

$$\mathbf{K}_{k+1} = \mathbf{C}_{k+1}^p \cdot \mathbf{H}_{k+1}^T \cdot (\mathbf{H}_{k+1} \cdot \mathbf{C}_{k+1}^p \cdot \mathbf{H}_{k+1}^T + \mathbf{C}_{k+1}^v)^{-1}\quad (\text{A.15})$$

Setzen wir (A.15) in (A.14) ein, erhalten wir

$$\mathbf{C}_{k+1}^e = (\mathbf{I} - \mathbf{K}_{k+1} \cdot \mathbf{H}_{k+1}) \cdot \mathbf{C}_{k+1}^p$$

Fassen wir nun das Kalman-Filter zusammen:

- Prädiktion

$$\begin{aligned}\hat{\underline{x}}_{k+1}^p &= \mathbf{A}_k \cdot \hat{\underline{x}}_k^e + \mathbf{B}_k \cdot \underline{u}_k \\ \mathbf{C}_{k+1}^p &= \mathbf{A}_k \cdot \mathbf{C}_k^e \cdot \mathbf{A}_k^T + \mathbf{C}_k^w\end{aligned}$$

- Filterung

$$\begin{aligned}\hat{\underline{x}}_{k+1}^e &= (\mathbf{I} - \mathbf{K}_{k+1} \cdot \mathbf{H}_{k+1}) \cdot \hat{\underline{x}}_{k+1}^p + \mathbf{K}_{k+1} \cdot \underline{y}_{k+1} \\ \mathbf{C}_{k+1}^e &= (\mathbf{I} - \mathbf{K}_{k+1} \cdot \mathbf{H}_{k+1}) \cdot \mathbf{C}_{k+1}^p\end{aligned}$$

mit

$$\mathbf{K}_{k+1} = \mathbf{C}_{k+1}^p \cdot \mathbf{H}_{k+1}^T \cdot (\mathbf{H}_{k+1} \cdot \mathbf{C}_{k+1}^p \cdot \mathbf{H}_{k+1}^T + \mathbf{C}_{k+1}^v)^{-1}$$

Literaturverzeichnis

- [1] Laska, Jason N., et al. "Theory and implementation of an analog-to-information converter using random demodulation." 2007 IEEE International Symposium on Circuits and Systems. IEEE, 2007.
- [2] Herman, Matthew A., et al. "A higher-speed compressive sensing camera through multi-diode design." Compressive Sensing II. Vol. 8717. International Society for Optics and Photonics, 2013.
- [3] Wu, Dalei, Wei-Ping Zhu, and M. N. S. Swamy. "A compressive sensing method for noise reduction of speech and audio signals." 2011 IEEE 54th International Midwest Symposium on Circuits and Systems (MWSCAS). IEEE, 2011.
- [4] Takhar, Dharmpal, et al. "A new compressive imaging camera architecture using optical-domain compression." Computational Imaging IV. Vol. 6065. International Society for Optics and Photonics, 2006.
- [5] Bobin, Jérôme, Jean-Luc Starck, and Roland Ottensamer. "Compressed sensing in astronomy." IEEE Journal of Selected Topics in Signal Processing 2.5 (2008): 718-726.
- [6] Dong, Xiao, and Yunhua Zhang. "A MAP approach for 1-bit compressive sensing in synthetic aperture radar imaging." IEEE Geoscience and Remote Sensing Letters 12.6 (2015): 1237-1241.
- [7] Duarte, Marco F., et al. "Single-pixel imaging via compressive sampling." IEEE signal processing magazine 25.2 (2008): 83-91.
- [8] Laska, Jason N., et al. "Theory and implementation of an analog-to-information converter using random demodulation." 2007 IEEE International Symposium on Circuits and Systems. IEEE, 2007.
- [9] Zelinski, Adam C., et al. "Sparsity-enforced slice-selective MRI RF excitation pulse design." IEEE transactions on medical imaging 27.9 (2008): 1213-1229.
- [10] Dai, Wei, et al. "Compressive sensing DNA microarrays." EURASIP journal on bioinformatics and systems biology 2009.1 (2008): 162824.

- [11] Levin, Anat, et al. "Image and depth from a conventional camera with a coded aperture." *ACM transactions on graphics (TOG)* 26.3 (2007): 70-es.
- [12] Seite "Modalanalyse". In: Wikipedia, Die freie Enzyklopädie. Bearbeitungsstand: 17. Dezember 2019, 22:22 UTC. URL: <https://de.wikipedia.org/w/index.php?title=Modalanalyse&oldid=194999069>
- [13] Kalman, Rudolph Emil. "A new approach to linear filtering and prediction problems." (1960): 35-45.
- [14] Kiencke, Uwe, Michael Schwarz, and Thomas Weickert. *Signalverarbeitung: Zeit-Frequenz-Analyse und Schätzverfahren*. Oldenbourg Verlag, 2009.
- [15] Mazumder, Sandip. *Numerical methods for partial differential equations: finite difference and finite volume methods*. Academic Press, 2015.
- [16] Shannon, Claude E. "A mathematical theory of communication." *The Bell system technical journal* 27.3 (1948): 379-423.
- [17] Davenport, Mark A., et al. "Signal processing with compressive measurements." *IEEE Journal of Selected topics in Signal processing* 4.2 (2010): 445-460.
- [18] Candes, Emmanuel J., Justin K. Romberg, and Terence Tao. "Stable signal recovery from incomplete and inaccurate measurements." *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences* 59.8 (2006): 1207-1223.
- [19] Donoho, David L. "Compressed sensing." *IEEE Transactions on information theory* 52.4 (2006): 1289-1306.
- [20] Haupt, Jarvis, and Robert Nowak. "Compressive sampling for signal detection." *2007 IEEE International Conference on Acoustics, Speech and Signal Processing-ICASSP'07*. Vol. 3. IEEE, 2007.
- [21] Davenport, Mark A., Michael B. Wakin, and Richard G. Baraniuk. "Detection and estimation with compressive measurements." Dept. of ECE, Rice University, Tech. Rep (2006).
- [22] Mark A. Davenport, Petros T. Boufounos, Richard Baraniuk. *Compressive Domain Interference Cancellation*. SPARS'09 - Signal Processing with Adaptive Sparse Structured Representations, Inria Rennes - Bretagne Atlantique, Apr 2009, Saint Malo, France. [finria-00369498](https://hal.inria.fr/finria-00369498)
- [23] Candes, Emmanuel J., and Terence Tao. "Decoding by linear programming." *IEEE transactions on information theory* 51.12 (2005): 4203-4215.

- [24] Baraniuk, Richard, et al. "An introduction to compressive sensing." *Connexions e-textbook* (2011): 24-76.
- [25] Baraniuk, Richard G. "Compressive sensing [lecture notes]." *IEEE signal processing magazine* 24.4 (2007): 118-121.
- [26] Cohen, Albert, Wolfgang Dahmen, and Ronald DeVore. "Compressed sensing and best ℓ_1 -term approximation." *Journal of the American mathematical society* 22.1 (2009): 211-231.
- [27] Baraniuk, Richard, et al. "A simple proof of the restricted isometry property for random matrices." *Constructive Approximation* 28.3 (2008): 253-263.
- [28] Candès, Emmanuel J. "Compressive sampling." *Proceedings of the international congress of mathematicians*. Vol. 3. 2006.
- [29] Donoho, David L., and Michael Elad. "Optimally sparse representation in general (nonorthogonal) dictionaries via ℓ_1 minimization." *Proceedings of the National Academy of Sciences* 100.5 (2003): 2197-2202.
- [30] Donoho, David L. "De-noising by soft-thresholding." *IEEE transactions on information theory* 41.3 (1995): 613-627.
- [31] Tsaig, Yaakov, and David L. Donoho. "Extensions of compressed sensing." *Signal processing* 86.3 (2006): 549-571.
- [32] Candes, Emmanuel J., and Terence Tao. "Near-optimal signal recovery from random projections: Universal encoding strategies?." *IEEE transactions on information theory* 52.12 (2006): 5406-5425.
- [33] Olshausen, Bruno A., and David J. Field. "Sparse coding with an overcomplete basis set: A strategy employed by V1?." *Vision research* 37.23 (1997): 3311-3325.
- [34] Cai, Deng, Hujun Bao, and Xiaofei He. "Sparse concept coding for visual analysis." *CVPR 2011*. IEEE, 2011.
- [35] Mairal, Julien, et al. "Online dictionary learning for sparse coding." *Proceedings of the 26th annual international conference on machine learning*. 2009.
- [36] Bao, Chenglong, Jian-Feng Cai, and Hui Ji. "Fast sparsity-based orthogonal dictionary learning for image restoration." *Proceedings of the IEEE International Conference on Computer Vision*. 2013.
- [37] Candes, Emmanuel, and Justin Romberg. "Sparsity and incoherence in compressive sampling." *Inverse problems* 23.3 (2007): 969.
- [38] Coifman, Ronald, F. Geshwind, and Yves Meyer. "Noiselets." *Applied and Computational Harmonic Analysis* 10.1 (2001): 27-44.

- [39] Tropp, Joel A., and Anna C. Gilbert. "Signal recovery from random measurements via orthogonal matching pursuit." *IEEE Transactions on information theory* 53.12 (2007): 4655-4666.
- [40] Candès, Emmanuel J., Justin Romberg, and Terence Tao. "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information." *IEEE Transactions on information theory* 52.2 (2006): 489-509.
- [41] Needell, Deanna, and Roman Vershynin. "Uniform uncertainty principle and signal recovery via regularized orthogonal matching pursuit." *Foundations of computational mathematics* 9.3 (2009): 317-334.
- [42] Blumensath, Thomas, and Mike E. Davies. "Iterative hard thresholding for compressed sensing." *Applied and computational harmonic analysis* 27.3 (2009): 265-274.
- [43] Cohen, Albert, Wolfgang Dahmen, and Ronald DeVore. "Instance optimal decoding by thresholding in compressed sensing." *Proc. 8th Int. Int. Conf.*, Jun. 1620, 2010, El Escorial, Madrid, Spain. Vol. 505. 2008.
- [44] Needell, Deanna, and Joel A. Tropp. "CoSaMP: Iterative signal recovery from incomplete and inaccurate samples." *Applied and computational harmonic analysis* 26.3 (2009): 301-321.
- [45] Tropp, Joel A., et al. "Random filters for compressive sampling and reconstruction." *2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*. Vol. 3. IEEE, 2006.
- [46] Asif, Muhammad Salman. *Dynamic compressive sensing: Sparse recovery algorithms for streaming signals and video*. Diss. Georgia Institute of Technology, 2013.
- [47] Asif, M. Salman, and Justin Romberg. "Dynamic Updating for ℓ_1 Minimization." *IEEE Journal of selected topics in signal processing* 4.2 (2010): 421-434.
- [48] Vaswani, Namrata. "Kalman filtered compressed sensing." *2008 15th IEEE International Conference on Image Processing*. IEEE, 2008.
- [49] Vaswani, Namrata. "Analyzing least squares and Kalman filtered compressed sensing." *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2009.
- [50] Vaswani, Namrata. "LS-CS-residual (LS-CS): compressive sensing on least squares residual." *IEEE Transactions on Signal Processing* 58.8 (2010): 4108-4120.

- [51] Ji, Shihao, Ya Xue, and Lawrence Carin. "Bayesian compressive sensing." *IEEE Transactions on signal processing* 56.6 (2008): 2346-2356.
- [52] Baron, Dror, Michael B. Wakin, Marco F. Duarte, Shriram Sarvotham, and Richard G. Baraniuk. "Distributed compressed sensing." (2005).
- [53] Carmi, Avishy, Pini Gurfil, and Dimitri Kanevsky. "Methods for sparse signal recovery using Kalman filtering with embedded pseudo-measurement norms and quasi-norms." *IEEE Transactions on Signal Processing* 58.4 (2009): 2405-2409.
- [54] Karimi, Hazhar Sufi, and Balasubramaniam Natarajan. "Recursive Dynamic Compressive Sensing in Smart Distribution Systems." *2020 IEEE Power Energy Society Innovative Smart Grid Technologies Conference (ISGT)*. IEEE, 2020.
- [55] Vaswani, Namrata, and Jinchun Zhan. "Recursive recovery of sparse signal sequences from compressive measurements: A review." *IEEE Transactions on Signal Processing* 64.13 (2016): 3523-3549.
- [56] Karseras, Evripidis, Kin Leung, and Wei Dai. "Tracking dynamic sparse signals using hierarchical Bayesian Kalman filters." *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2013.
- [57] Uwe D. Hanebeck, Benjamin Noack, et al. "Skriptum zur Vorlesung Informationsverarbeitung in Sensornetzwerken."