3.2 Fahrdynamikmodelle für autonomes Fahren und die aktive Fahrzeugsicherheit

Fahrdynamikmodelle spielen für das automatisierte Fahren und die integrale Fahrzeugsicherheit eine zentrale Rolle, weil sie dabei helfen, den aktuellen Zustand des Fahrzeugs zu analysieren und das künftige Verhalten des Fahrzeugs vorherzusagen bzw. zu planen. Aus diesem Grund sind Fahrdynamikmodelle ein Kernelement von Reglern und Prädiktionsalgorithmen, z. B. für die Situationsinterpretation und die Trajektorienplanung. Dabei muss nicht nur die Bewegung des eigenen Fahrzeugs, in diesem Buch **EGO** genannt, sondern auch aller anderen Verkehrsteilnehmer prädiziert werden. Hierbei ist es wichtig, geeignete Modelle zu nutzen. Für dynamische Objekte, deren Zustand mittels vorausschauender Sensoren geschätzt wird, ist es häufig aufgrund der fehlenden Eingangsgrößen für komplexe Fahrdynamikmodelle sinnvoll, einfache Bewegungsmodelle zu nutzen. Hingegen ist es für das EGO-Fahrzeug, gerade in kritischen Verkehrssituationen, in denen Manöver mit hoher Dynamik vorkommen können, sinnvoll, komplexere Fahrdynamikmodelle zu verwenden. Die Wahl eines adäquaten Modells ist von den verfügbaren Rechenressourcen und der umzusetzenden Funktionalität abhängig. Für eine Funktionalität, bei der nur geringe Querbeschleunigungen zu erwarten sind, wie z. B. dem automatisierten Fahren im Stau, ist es nicht erforderlich, komplexe Querdynamikmodelle einzusetzen. Dieses Unterkapitel soll auch dabei helfen, eine geeignete Wahl des Fahrdynamikmodells für Algorithmen zu treffen.

Ergänzende Inhalte zum Themengebiet der Fahrdynamikmodelle und zu Grundlagen der Kraftfahrzeugtechnik können beispielsweise in [Ril07, Sch07, Jaz12, HEG13, SHB13, MW15, Hak18] gefunden werden.

3.2.1 Relativbewegung

Zur Beschreibung der Bewegung von Fahrzeugen sind zwei Koordinatensysteme sehr wichtig: das **Fahrzeugkoordinatensystem** und ein **Inertialsystem**. Das Fahrzeugkoordinatensystem ist ein fahrzeugfestes, kartesisches Koordinatensystem, bezüglich dessen das Fahrzeug ruht. Ein Inertialsystem ist ein Bezugssystem, in dem es keine Trägheitskräfte gibt, d. h. in dem jeder kräftefreie Körper relativ zu diesem Bezugssystem in Ruhe verharrt oder sich unbeschleunigt und geradlinig bewegt. Die Trägheitskräfte, die durch die Rotation der Erde verursacht werden, sind für die Betrachtungen der Fahrdynamik vernachlässigbar, und damit wird im Folgenden ein mit der Erde fest verbundenes System als Inertialsystem gewählt.

In Abb. 3.10 wird der Ursprung des Fahrzeugkoordinatensystems im **Schwerpunkt** S des Fahrzeugs gewählt und die Achsen dieses Koordinatensystems mit den Kleinbuchstaben x, y und z bezeichnet. Diese Lage und die in Abb. 3.10 gezeigte Ausrichtung der Achsen des Fahrzeugkoordinatensystems entsprechen der DIN 70000. Das Inertialsystem hat seinen Ursprung auf der Erde, und seine Achsen wurden mit den Großbuchstaben X, Y und Z bezeichnet. Der **Radstand** des Fahrzeugs ist ℓ, der Abstand vom Schwerpunkt zur Vorderachse wird mit ℓ_v, der Abstand vom Schwerpunkt zur Hinterachse mit ℓ_h bezeichnet. Die **Höhe des Schwerpunkts** S über der Erde ist h_S und die **Spurweite** b.

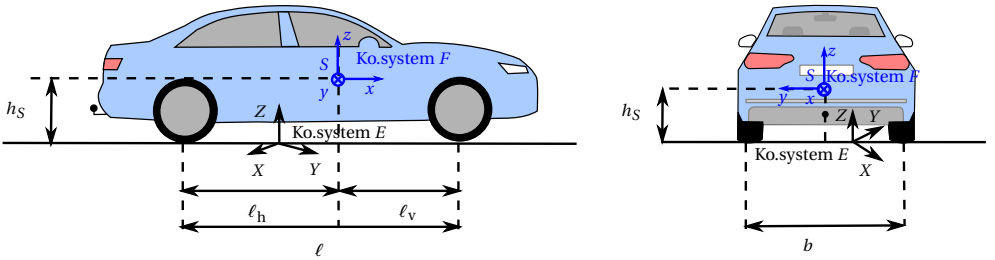

Abbildung 3.10 Fahrzeugkoordinatensystem F mit Achsen x, y, z und Ursprung im Schwerpunkt S sowie Inertialsystem E mit Achsen X, Y, Z und Ursprung auf der Erde.

Vektoren im Fahrzeug- und Erdkoordinatensystem und deren Notation

Im Folgenden wird, wie in Abb. 3.10 dargestellt, mit F das Fahrzeugkoordinatensystem und mit E das Erdkoordinatensystem bezeichnet. Ein Vektor $\boldsymbol{r} \in \mathbb{R}^3$, an dessen Spitze der Punkt P liegt und der den Ursprung des Erdkoordinatensystems E mit P verbindet, kann mit den **Einheitsvektoren** $\boldsymbol{e}_x = {}^F[1, 0, 0]^\mathrm{T}$, $\boldsymbol{e}_y = {}^F[0, 1, 0]^\mathrm{T}$ und $\boldsymbol{e}_z = {}^F[0, 0, 1]^\mathrm{T}$ des Fahrzeugkoordinatensystems F dargestellt werden. Dieser Vektor wird im Folgenden mit

$$\underset{E}{{}^F}\boldsymbol{r} = r_x \boldsymbol{e}_x + r_y \boldsymbol{e}_y + r_z \boldsymbol{e}_z \tag{3.42}$$

bezeichnet. Stellt man den Vektor mit den Einheitsvektoren des Erdkoordinatensystems E dar, also mit $\boldsymbol{e}_X = {}^E[1\ 0\ 0]^\mathrm{T}$, $\boldsymbol{e}_Y = {}^E[0, 1, 0]^\mathrm{T}$ und $\boldsymbol{e}_Z = {}^E[0, 0, 1]^\mathrm{T}$, so wird im Folgenden dieser Vektor mit

$$\underset{E}{{}^E}\boldsymbol{r} = r_X \boldsymbol{e}_X + r_Y \boldsymbol{e}_Y + r_Z \boldsymbol{e}_Z \tag{3.43}$$

bezeichnet, wobei in der Regel $r_X \neq r_x$, $r_Y \neq r_y$ und $r_Z \neq r_z$.

Der linke untere Index, hier E, bezeichnet das Koordinatensystem, von dessen Ursprung aus zu P der Ortsvektor definiert ist. Der linke obere Index bezeichnet das Koordinatensystem in dem der Ortsvektor dargestellt wird, d. h. mit dessen Einheitsvektoren er ausgedrückt wird. Der Vektor $\underset{E}{{}^F}\boldsymbol{r}$ ist also der Ortsvektor gegenüber dem Ursprung des Erdkoordinatensystems E und dargestellt in F. Analog dazu repräsentieren zum Beispiel der Geschwindigkeitsvektor $\underset{E}{{}^F}\boldsymbol{v}$ die Geschwindigkeit gegenüber dem Ursprung des Erdkoordinatensystems E und dargestellt in F, oder der Beschleunigungsvektor $\underset{E}{{}^F}\boldsymbol{a}$ die Beschleunigung gegenüber dem Ursprung des Erdkoordinatensystems E und dargestellt in F. Falls die beiden Koordinatensysteme links unten und links oben gleich sind, so kann zur Vereinfachung auch nur das obere linke angegeben werden, z. B. ${}^E\boldsymbol{r} = \underset{E}{{}^E}\boldsymbol{r}$.

Es ist zu beachten, dass es sich bei $\underset{E}{{}^F}\boldsymbol{r}$ und ${}^E\boldsymbol{r}$ um den gleichen Vektor $\boldsymbol{r}$ handelt, nämlich vom Ursprung von E nach P, dieser aber in unterschiedlichen Koordinatensystemen ausgedrückt wird. Um den Vektor aus einer Darstellung in die andere Darstellung zu transformieren, bedarf es einer **Rotationsmatrix**. Möchte man den Vektor $\underset{E}{{}^F}\boldsymbol{r}$ im Erdkoordinatensystem darstellen, so benötigt man die Rotationsmatrix ${}^E\boldsymbol{R}_F \in \mathbb{R}^{3\times3}$

$${}^E\boldsymbol{r} = {}^E\boldsymbol{R}_F \underset{E}{{}^F}\boldsymbol{r}, \tag{3.44}$$

und möchte man den Vektor ${}^E\boldsymbol{r}$ im Fahrzeugkoordinatensystem darstellen, so benötigt man die Rotationsmatrix ${}^F\boldsymbol{R}_E$

$$\underset{E}{{}^F}\boldsymbol{r} = {}^F\boldsymbol{R}_E {}^E\boldsymbol{r}, \tag{3.45}$$

wobei gilt

$$ {}^{E}\boldsymbol{R}_{F} = {}^{F}\boldsymbol{R}_{E}^{\mathrm{T}}. \tag{3.46} $$

Gl. (3.46) ergibt sich aufgrund der Tatsache, dass Rotationsmatrizen orthonormale Matrizen sind, d. h. die Spalten sind orthogonal zueinander und haben die Euklidische Norm 1. Es ist zu beachten, dass die erste Spalte von ${}^{E}\boldsymbol{R}_{F}$ die Darstellung von $\boldsymbol{e}_x$ im Inertialsystem E ist, weil sich diese bei der Multiplikation ${}^{E}\boldsymbol{R}_{F}\boldsymbol{e}_x$ ergibt. Analog dazu ist die zweite Spalte in ${}^{E}\boldsymbol{R}_{F}$ die Darstellung von $\boldsymbol{e}_y$ in E und die dritte Spalte in ${}^{E}\boldsymbol{R}_{F}$ die Darstellung von $\boldsymbol{e}_z$ in E.

Weil das Fahrzeugkoordinatensystem F ein bewegtes Koordinatensystem ist, sind seine Einheitsvektoren $\boldsymbol{e}_x$, $\boldsymbol{e}_y$ und $\boldsymbol{e}_z$ zeitabhängige Vektoren, und dies muss man bei allen Größen, die im Fahrzeugkoordinatensystem F repräsentiert werden, berücksichtigen. Diese Zeitabhängigkeit der Einheitsvektoren von F ist der Grund für die Trägheitskräfte, die sich in F ergeben. Um dies zu verstehen, werden im Folgenden Rotationen im Dreidimensionalen und die zeitliche Ableitung von Rotationsmatrizen betrachtet.

Euler-Winkel und Rotationsmatrizen

Die Rotation im dreidimensionalen Raum ist komplexer als im zweidimensionalen Raum und kann durch eine orthonormale Matrix $\boldsymbol{R} \in \mathbb{R}^{3\times 3}$ beschrieben werden. Entsprechend dem **Rotations-Theorem von Euler** kann jede Rotation eines Starrkörpers im dreidimensionalen Raum durch eine Abfolge dreier Drehungen um spezielle Achsen dargestellt werden. Möchte man das Koordinatensystem (X, Y, Z) in das Koordinatensystem (x, y, z) drehen, so entstehen nach jeder Drehung neue Koordinatensysteme. Dies ist beispielhaft in Abb. 3.11 visualisiert. Die erste Drehachse ist eine im Inertialsystem feste Achse, die beiden anderen sind vorher mitgedrehte Achsen. Die drei Drehwinkel werden **Euler-Winkel** genannt. Bei den Drehungen ist es nicht erlaubt, dass zwei aufeinanderfolgende Drehungen um die gleiche Achse stattfinden. Damit ergeben sich insgesamt 12 mögliche Folgen von Drehungen um die Achsen, und zwar $Xy'x''$, $Xz'x''$, $Yx'y''$, $Yz'y''$, $Zx'z''$, $Zy'z''$, $Xy'z''$, $Xz'y''$, $Yz'x''$, $Yx'z''$, $Zx'y''$ und $Zy'x''$. Es gibt in den unterschiedlichen Technologiefeldern Konventionen über die Drehfolge, die verwendet wird. In der Fahrzeugtechnik verwendet man die Drehfolge $Zy'x''$, um einen Vektor aus dem Inertialsystem E in das Fahrzeugkoordinatensystem F zu transformieren. Bezeichnet man den ersten Drehwinkel um die Z-Achse mit Ψ, den zweiten um die y'-Achse mit Θ und den dritten um die x''-Achse mit Φ, so ergibt sich für die Rotationsmatrix von E nach F mit den Abkürzungen $\mathrm{c}(\cdot) = \cos(\cdot)$ und $\mathrm{s}(\cdot) = \sin(\cdot)$

$$
\begin{aligned}
{}^{F}\boldsymbol{R}_{E} &= \boldsymbol{R}_{x}^{\mathrm{T}}(\Phi)\boldsymbol{R}_{y}^{\mathrm{T}}(\Theta)\boldsymbol{R}_{z}^{\mathrm{T}}(\Psi) \\
&= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \mathrm{c}(\Phi) & \mathrm{s}(\Phi) \\ 0 & -\mathrm{s}(\Phi) & \mathrm{c}(\Phi) \end{bmatrix} \begin{bmatrix} \mathrm{c}(\Theta) & 0 & -\mathrm{s}(\Theta) \\ 0 & 1 & 0 \\ \mathrm{s}(\Theta) & 0 & \mathrm{c}(\Theta) \end{bmatrix} \begin{bmatrix} \mathrm{c}(\Psi) & \mathrm{s}(\Psi) & 0 \\ -\mathrm{s}(\Psi) & \mathrm{c}(\Psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \\
&= \begin{bmatrix} \mathrm{c}(\Theta)\mathrm{c}(\Psi) & \mathrm{c}(\Theta)\mathrm{s}(\Psi) & -\mathrm{s}(\Theta) \\ \mathrm{s}(\Phi)\mathrm{s}(\Theta)\mathrm{c}(\Psi) - \mathrm{c}(\Phi)\mathrm{s}(\Psi) & \mathrm{s}(\Phi)\mathrm{s}(\Theta)\mathrm{s}(\Psi) + \mathrm{c}(\Phi)\mathrm{c}(\Psi) & \mathrm{s}(\Phi)\mathrm{c}(\Theta) \\ \mathrm{c}(\Phi)\mathrm{s}(\Theta)\mathrm{c}(\Psi) + \mathrm{s}(\Phi)\mathrm{s}(\Psi) & \mathrm{c}(\Phi)\mathrm{s}(\Theta)\mathrm{s}(\Psi) - \mathrm{s}(\Phi)\mathrm{c}(\Psi) & \mathrm{c}(\Phi)\mathrm{c}(\Theta) \end{bmatrix}.
\end{aligned} \tag{3.47}
$$

Um Gl. (3.47) besser zu verstehen, wird Abb. 3.11 verwendet. Wie daraus zu entnehmen ist, gelten die folgenden Beziehungen für die Komponenten des Vektors $\boldsymbol{r}$ in den entsprechenden Koordinatensystemen

$$\begin{bmatrix} r_{x'} \\ r_{y'} \\ r_{z'} \end{bmatrix} = \underbrace{\begin{bmatrix} \mathrm{c}(\Psi) & \mathrm{s}(\Psi) & 0 \\ -\mathrm{s}(\Psi) & \mathrm{c}(\Psi) & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{\boldsymbol{R}_z^{\mathrm{T}}(\Psi)} \begin{bmatrix} r_X \\ r_Y \\ r_Z \end{bmatrix} \text{ bzw. } \begin{bmatrix} r_X \\ r_Y \\ r_Z \end{bmatrix} = \underbrace{\begin{bmatrix} \mathrm{c}(\Psi) & -\mathrm{s}(\Psi) & 0 \\ \mathrm{s}(\Psi) & \mathrm{c}(\Psi) & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{\boldsymbol{R}_z(\Psi)} \begin{bmatrix} r_{x'} \\ r_{y'} \\ r_{z'} \end{bmatrix}, \tag{3.48}$$

$$\begin{bmatrix} r_{x''} \\ r_{y''} \\ r_{z''} \end{bmatrix} = \underbrace{\begin{bmatrix} \mathrm{c}(\Theta) & 0 & -\mathrm{s}(\Theta) \\ 0 & 1 & 0 \\ \mathrm{s}(\Theta) & 0 & \mathrm{c}(\Theta) \end{bmatrix}}_{\boldsymbol{R}_y^{\mathrm{T}}(\Theta)} \begin{bmatrix} r_{x'} \\ r_{y'} \\ r_{z'} \end{bmatrix} \text{ bzw. } \begin{bmatrix} r_{x'} \\ r_{y'} \\ r_{z'} \end{bmatrix} = \underbrace{\begin{bmatrix} \mathrm{c}(\Theta) & 0 & \mathrm{s}(\Theta) \\ 0 & 1 & 0 \\ -\mathrm{s}(\Theta) & 0 & \mathrm{c}(\Theta) \end{bmatrix}}_{\boldsymbol{R}_y(\Theta)} \begin{bmatrix} r_{x''} \\ r_{y''} \\ r_{z''} \end{bmatrix}, \tag{3.49}$$

$$\begin{bmatrix} r_x \\ r_y \\ r_z \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & \mathrm{c}(\Phi) & \mathrm{s}(\Phi) \\ 0 & -\mathrm{s}(\Phi) & \mathrm{c}(\Phi) \end{bmatrix}}_{\boldsymbol{R}_x^{\mathrm{T}}(\Phi)} \begin{bmatrix} r_{x''} \\ r_{y''} \\ r_{z''} \end{bmatrix} \text{ bzw. } \begin{bmatrix} r_{x''} \\ r_{y''} \\ r_{z''} \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & \mathrm{c}(\Phi) & -\mathrm{s}(\Phi) \\ 0 & \mathrm{s}(\Phi) & \mathrm{c}(\Phi) \end{bmatrix}}_{\boldsymbol{R}_x(\Phi)} \begin{bmatrix} r_x \\ r_y \\ r_z \end{bmatrix}. \tag{3.50}$$

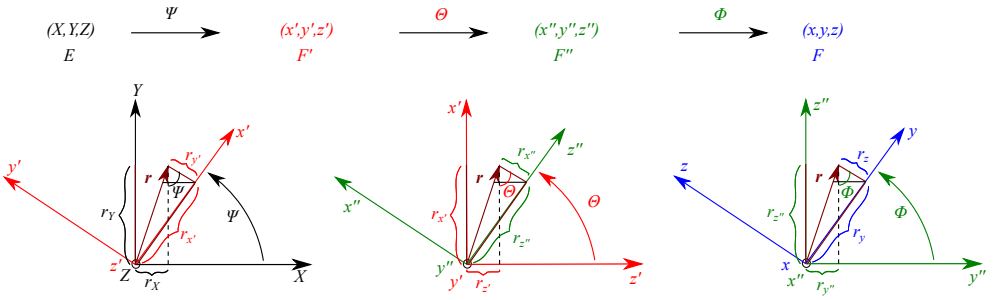

Abbildung 3.11 Rotationen mit den Winkeln Ψ, Θ und Φ um die Achsen Z, y' und x''.

Möchte man also einen Vektor aus dem Koordinatensystem E mit den Achsen X, Y, Z in das Koordinatensystem F mit den Achsen x, y, z durch Rotationen mit den Euler-Winkeln Φ, Θ und Ψ realisieren, so erfolgt zunächst die Rotation mit Ψ um die Z-Achse, dann die Rotation mit Θ um die y'-Achse und letztlich die Rotation mit Φ um die x''-Achse. Abb. 3.12 visualisiert diese Rotationen und in Gleichungen lässt sich schreiben

$$\underbrace{\begin{bmatrix} r_x \\ r_y \\ r_z \end{bmatrix}}_{{}^F\boldsymbol{r}} = \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & \mathrm{c}(\Phi) & \mathrm{s}(\Phi) \\ 0 & -\mathrm{s}(\Phi) & \mathrm{c}(\Phi) \end{bmatrix}}_{\boldsymbol{R}_x^{\mathrm{T}}(\Phi)} \underbrace{\begin{bmatrix} r_{x''} \\ r_{y''} \\ r_{z''} \end{bmatrix}}_{\parallel}$$

$$\underbrace{\begin{bmatrix} \mathrm{c}(\Theta) & 0 & -\mathrm{s}(\Theta) \\ 0 & 1 & 0 \\ \mathrm{s}(\Theta) & 0 & \mathrm{c}(\Theta) \end{bmatrix}}_{\boldsymbol{R}_y^{\mathrm{T}}(\Theta)} \underbrace{\begin{bmatrix} r_{x'} \\ r_{y'} \\ r_{z'} \end{bmatrix}}_{\parallel}$$

$$\underbrace{\begin{bmatrix} \mathrm{c}(\Psi) & \mathrm{s}(\Psi) & 0 \\ -\mathrm{s}(\Psi) & \mathrm{c}(\Psi) & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{\boldsymbol{R}_z^{\mathrm{T}}(\Psi)} \underbrace{\begin{bmatrix} r_X \\ r_Y \\ r_Z \end{bmatrix}}_{{}^E\boldsymbol{r}}$$

$$= \boldsymbol{R}_x^{\mathrm{T}}(\Phi)\boldsymbol{R}_y^{\mathrm{T}}(\Theta)\boldsymbol{R}_z^{\mathrm{T}}(\Psi)\,{}^E\boldsymbol{r} = {}^F\boldsymbol{R}_E\,{}^E\boldsymbol{r}. \tag{3.51}$$

Gl. (3.51) fasst Gl. (3.45) und Gl. (3.47) zusammen. Jede Orientierung des Fahrzeugs kann damit gegenüber der Lage des Erdkoordinatensystems E mit den Euler-Winkeln Ψ, Θ und Φ mit Hilfe der Matrix ${}^F\boldsymbol{R}_E$ aus Gl. (3.47) ausgedrückt werden.

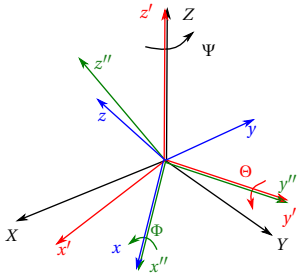

Abbildung 3.12 Rotation mit den Euler-Winkeln Φ, Θ und Ψ von E nach F.

Weil in einem fahrenden Fahrzeug die Winkel Ψ, Θ und Φ im Allgemeinen zeitabhängig sind, ist auch die Rotationsmatrix ${}^F\boldsymbol{R}_E$ zeitabhängig, und dies muss bei der zeitlichen Ableitung eines Vektors ${}^F_E\boldsymbol{r}$ berücksichtigt werden. Man erhält für die Ableitung des Vektors ${}^F_E\boldsymbol{r}$ nach der Zeit

$$\frac{\mathrm{d}}{\mathrm{d}t}\left({}^F_E\boldsymbol{r}\right) = \frac{\mathrm{d}}{\mathrm{d}t}\left({}^F\boldsymbol{R}_E\,{}^E\boldsymbol{r}\right) = \frac{\mathrm{d}}{\mathrm{d}t}\left({}^F\boldsymbol{R}_E\right){}^E\boldsymbol{r} + {}^F\boldsymbol{R}_E\frac{\mathrm{d}}{\mathrm{d}t}\left({}^E\boldsymbol{r}\right). \tag{3.52}$$

Um diesen Ausdruck zu vereinfachen, wird die zeitliche Ableitung der Rotationsmatrix ${}^F\boldsymbol{R}_E$ aus Gl. (3.47) benötigt. Die folgenden Überlegungen sind notwendig, um die zeitliche Ableitung von ${}^F\boldsymbol{R}_E$ zu berechnen.

Zeitliche Ableitung einer Rotationsmatrix $\boldsymbol{R}$

Jede Rotation eines Starrkörpers kann alternativ zur Darstellung mit den Euler-Winkeln auch durch die Rotation um eine Achse und einen entsprechenden Winkel beschrieben werden. Bei der Rotation eines Starrkörpers bewegt sich jeder Punkt des Starrkörpers auf einem Kreis. Dabei durchläuft die Senkrechte von jedem Punkt auf die Rotationsachse den gleichen Winkel θ. Bezeichnet man mit $\boldsymbol{n}$ den Einheitsvektor in Richtung der Rotationsachse, dann ist die Winkelgeschwindigkeit

$$\boldsymbol{\omega} = \frac{\mathrm{d}\theta}{\mathrm{d}t}\boldsymbol{n}, \quad \text{mit} \quad \boldsymbol{\omega} = [\omega_1, \omega_2, \omega_3]^{\mathrm{T}} \quad \text{und} \quad \boldsymbol{n} = [n_1, n_2, n_3]^{\mathrm{T}}. \tag{3.53}$$

Eine Rotationsmatrix $\boldsymbol{R}$, die durch drei Euler-Winkel beschrieben wird, lässt sich also auch durch einen Winkel θ und den entsprechenden Normalenvektor $\boldsymbol{n}$ darstellen. Abb. 3.13 visualisiert die Rotation von $\boldsymbol{r}$ nach $\boldsymbol{r}'$. Der Vektor $\boldsymbol{r}$ wird zunächst in seine Komponenten parallel und senkrecht zur Rotationsachse $\boldsymbol{n}$ zerlegt

$$\boldsymbol{r} = \boldsymbol{n}(\boldsymbol{n}^{\mathrm{T}}\boldsymbol{r}) - \boldsymbol{n} \times (\boldsymbol{n} \times \boldsymbol{r}). \tag{3.54}$$

Von der Rotation ist nur der zweite Term, also die Senkrechte zur Rotationsachse $\boldsymbol{n}$, betroffen und man erhält

$$\boldsymbol{r}' = \boldsymbol{n}(\boldsymbol{n}^{\mathrm{T}}\boldsymbol{r}) + \sin(\theta)(\boldsymbol{n} \times \boldsymbol{r}) - \cos(\theta)\boldsymbol{n} \times (\boldsymbol{n} \times \boldsymbol{r}). \tag{3.55}$$

4 Statistische Filterung

Insbesondere im Bereich des automatisierten Fahrens und in der aktiven Fahrzeugsicherheit spielen die **statistischen Filter** eine zentrale Rolle, weil sie die Grundlage für viele Signalverarbeitungsschritte bilden wie z. B. für das Schätzen von Zustandsgrößen des EGO-Fahrzeugs, für das Tracking von Objekten im Umfeld des EGO-Fahrzeugs oder für die Sensordatenfusion. Vorausschauende Sensoren sind beispielsweise in der Lage, die relative Position zu Objekten zu messen, aber in den Funktionsalgorithmen sind auch die Geschwindigkeit und die Beschleunigung der Verkehrsteilnehmer von zentraler Bedeutung, und diese werden mit Hilfe von statistischen Filtern bestimmt. In diesem Kapitel werden die Grundlagen der statistischen Filter vorgestellt und das Kalman-Filter als eines der wichtigsten statistischen Filter hergeleitet. Anschließend werden Anwendungen dieser Grundlagen für das Tracking von Verkehrsteilnehmern und für die Sensordatenfusion vorgestellt.

Lernziele in Kapitel 4

Der Lernende …

- versteht die unterschiedlichen Optimalitätskriterien für statistische Filter;
- versteht die Herleitung des Kalman-Filters;
- kann entscheiden, wann das Kalman-Filter genutzt werden sollte, und ist in der Lage, es in Anwendungen zu implementieren;
- kann einfache Trackingaufgaben modellieren und mit dem Kalman-Filter umsetzen;
- kennt Sensordatenfusionsmethoden wie die Fusion mittels Extended Kalman-Filter oder „Track-To-Track“ und kann diese implementieren.

4.1 Optimale statistische Filter

Dieses Unterkapitel führt statistische Filter ein und beschäftigt sich mit der Frage nach der Optimalität solcher Filter. Ausführliche Beschreibungen zu Methoden der statistischen Filterung und der Zustandsschätzung können beispielsweise in [BSL01, Sim06, Kay17] gefunden werden.

Ein statistisches Filter schätzt ein Nutzsignal oder einen Zustand aus verrauschten Messsignalen unter Verwendung des Wissens über die Statistik der Signale. Die ersten Arbeiten zu statistischen Filtern stammen von **Wiener** und **Kolmogorov** aus den 1940-er Jahren und behandeln Systeme, bei denen sich die statistischen Eigenschaften nicht mit der Zeit ändern, d. h. mit stationären Prozessen. Für stationäre Prozesse ist es leicht möglich, die statistischen Eigenschaften der Signale mit deren Frequenzeigenschaften in Verbindung zu setzen und somit diese Filter mit der klassischen Filtertheorie der elektrischen Netzwerke zu vereinbaren.

In den 1960-er Jahren entwickelte sich, getrieben durch Anwendungen, in denen die Signale nicht mehr als stationär modelliert werden können, das Gebiet der statistischen Filter für instationäre Prozesse, insbesondere die **Kalman-Filtertheorie**. Während bei der Frequenzraumdarstellung, wie in Gl. (2.287) dargestellt, nur ein Zusammenhang zwischen dem Ausgang und dem Eingang durch die Übertragungsfunktion $\boldsymbol{H}(s)$ hergestellt wurde, wird bei den statistischen Filtern explizit zwischen der Dynamik des Systems und dem Prozess seiner Messung unterschieden. Diese statistischen Filter gehen damit von der Zustandsbeschreibung des Systems aus und somit von einer Zeit- und nicht von einer Frequenzdarstellung.

Abb. 4.1 zeigt ein zeitdiskretes Modell für ein System, bei dem ein statistisches Filter zur Schät-

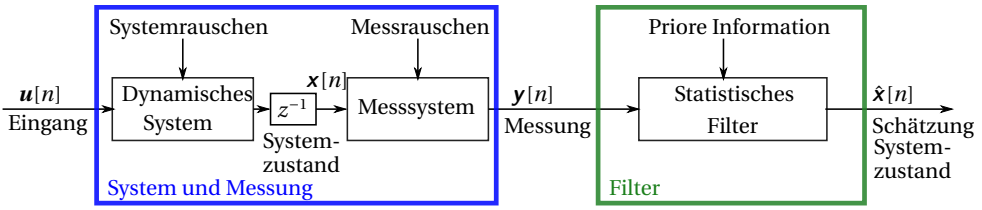

Abbildung 4.1 Modell für statistische Filter.

zung des Zustandes $\boldsymbol{x}[n]$ verwendet wird. Um das statistische Filter zu entwerfen, benötigt man:

1. Ein Modell für das dynamische System, unter Berücksichtigung des Systemrauschens $\boldsymbol{\eta}_{\mathrm{S}}[n]$

$$\boldsymbol{x}[n+1] = \boldsymbol{f}_n\left(\boldsymbol{x}[n], \boldsymbol{u}[n], \boldsymbol{\eta}_{\mathrm{S}}[n]\right); \tag{4.1}$$

2. Ein Modell für das Messsystem, unter Berücksichtigung des Messrauschens $\boldsymbol{\eta}_{\mathrm{M}}[n]$

$$\boldsymbol{y}[n] = \boldsymbol{h}_n\left(\boldsymbol{x}[n], \boldsymbol{u}[n], \boldsymbol{\eta}_{\mathrm{M}}[n]\right); \tag{4.2}$$

3. Die statistische Beschreibung der Rauschprozesse $\boldsymbol{\eta}_{\mathrm{S}}(m,\omega)$ und $\boldsymbol{\eta}_{\mathrm{M}}(m,\omega)$;
4. Priore Informationen:
   - Anfangszustand des Systems, d. h. $\boldsymbol{x}[0]$;
   - Eingang $\boldsymbol{u}[n]$.

Dieses benötigte Wissen ist auch in Abb. 4.2 dargestellt.

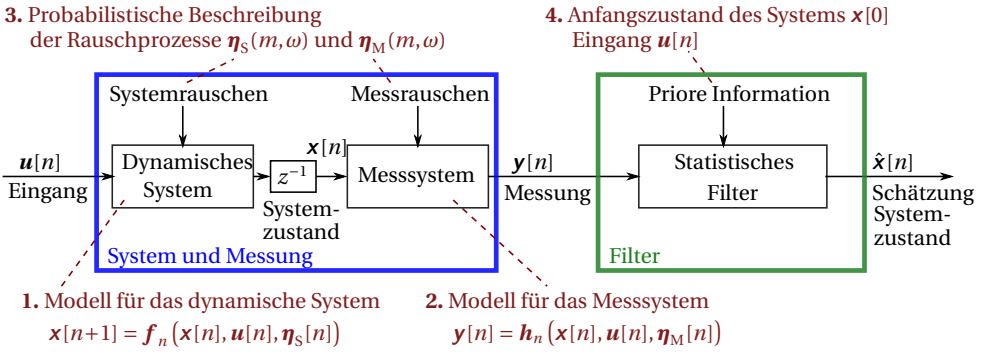

Abbildung 4.2 Benötigtes Wissen für ein statistisches Filter.

Beispielsweise ergeben sich für die System- und Messgleichungen (4.1) und (4.2) bei einem zeitdiskreten linearen System Gl. (2.197) und Gl. (2.198)

$$\boldsymbol{x}[n+1] = \boldsymbol{A}[n]\boldsymbol{x}[n] + \boldsymbol{B}[n]\boldsymbol{u}[n] + \boldsymbol{G}[n]\boldsymbol{\eta}_{\mathrm{S}}[n], \quad n \geq 0 \tag{4.3}$$

$$\boldsymbol{y}[n] = \boldsymbol{C}[n]\boldsymbol{x}[n] + \boldsymbol{D}[n]\boldsymbol{u}[n] + \boldsymbol{\eta}_{\mathrm{M}}[n], \tag{4.4}$$

wobei die Dimensionen der Vektoren und Matrizen in Unterkapitel 2.4.2 eingeführt wurden.

Die Aufgabe beim Entwurf des statistischen Filters für den Zeitpunkt, der durch den Index n gekennzeichnet ist, besteht darin, einen optimalen Schätzwert $\hat{\boldsymbol{x}}[n]$ von $\boldsymbol{x}[n]$ zu finden unter Verwendung von $\boldsymbol{f}_n$, $\boldsymbol{h}_n$, der statistischen Beschreibung der Rauschprozesse, des Anfangszustands $\boldsymbol{x}[0]$, sowie der Reihen von bisherigen Eingängen und Messwerten

$$\{\boldsymbol{u}[0], \boldsymbol{u}[1], \ldots, \boldsymbol{u}[n-1]\} \quad \text{bzw.} \quad \{\boldsymbol{y}[1], \boldsymbol{y}[2], \ldots, \boldsymbol{y}[n]\}. \tag{4.5}$$

Im Folgenden wird mit $\boldsymbol{u}_{0:n-1}$ der $(nM \times 1)$-Vektor mit allen Eingängen bis einschließlich zum Index $n-1$ und mit $\boldsymbol{y}_{1:n}$ der $(nK \times 1)$-Vektor mit allen Messungen bis einschließlich zum Index n bezeichnet

$$\boldsymbol{u}_{0:n-1} = [\boldsymbol{u}^{\mathrm{T}}[0], \boldsymbol{u}^{\mathrm{T}}[1], \ldots, \boldsymbol{u}^{\mathrm{T}}[n-1]]^{\mathrm{T}} \quad \text{bzw.} \quad \boldsymbol{y}_{1:n} = [\boldsymbol{y}^{\mathrm{T}}[1], \boldsymbol{y}^{\mathrm{T}}[2], \ldots, \boldsymbol{y}^{\mathrm{T}}[n]]^{\mathrm{T}}. \tag{4.6}$$

Man kann das statistische Filter beruhend auf unterschiedlichen Optimalitätskriterien entwerfen. Drei davon werden im Folgenden vorgestellt.

I. Optimales lineares statistisches Filter

Das **optimale lineare statistische Filter** ergibt sich aus der Minimierung des mittleren quadratischen Fehlers, wobei als Ansatz für das Filter gewählt wird

$$\hat{\boldsymbol{x}}_{\mathrm{lin}}[n] = \boldsymbol{W}\boldsymbol{y}_{1:n} + \boldsymbol{t}. \tag{4.7}$$

Die Optimierungsaufgabe lautet

$$\{\boldsymbol{W}_{\mathrm{lin}}, \boldsymbol{t}_{\mathrm{lin}}\} = \underset{\boldsymbol{W},\boldsymbol{t}}{\operatorname{argmin}} \left\{ \mathrm{E}_{\boldsymbol{x}[n],\boldsymbol{y}_{1:n}} \left\{ \left\| \boldsymbol{x}[n] - (\boldsymbol{W}\boldsymbol{y}_{1:n} + \boldsymbol{t}) \right\|_2^2 \right\} \right\}. \tag{4.8}$$

Die zu optimierende Kostenfunktion lässt sich schreiben als

$$\mathscr{L} = \mathrm{E}_{\boldsymbol{x}[n],\boldsymbol{y}_{1:n}} \left\{ \left\| \boldsymbol{x}[n] - (\boldsymbol{W}\boldsymbol{y}_{1:n} + \boldsymbol{t}) \right\|_2^2 \right\} \tag{4.9}$$

$$= \mathrm{E}_{\boldsymbol{x}[n],\boldsymbol{y}_{1:n}} \left\{ \mathrm{tr} \left\{ \left(\boldsymbol{x}[n] - (\boldsymbol{W}\boldsymbol{y}_{1:n} + \boldsymbol{t}) \right) \left(\boldsymbol{x}[n] - (\boldsymbol{W}\boldsymbol{y}_{1:n} + \boldsymbol{t}) \right)^{\mathrm{T}} \right\} \right\}. \tag{4.10}$$

Man erhält das Minimum dieser Kostenfunktion für diejenigen Parameter $\boldsymbol{W}$ und $\boldsymbol{t}$, für die die Ableitung von $\mathscr{L}$ null ergibt. Die Ableitung von $\mathscr{L}$ nach $\boldsymbol{t}$ lautet

$$\frac{\partial \mathscr{L}}{\partial \boldsymbol{t}} = \mathrm{E}_{\boldsymbol{x}[n],\boldsymbol{y}_{1:n}} \left\{ -\left(\boldsymbol{x}[n] - \boldsymbol{W}\boldsymbol{y}_{1:n} - \boldsymbol{t} \right) - \left(\boldsymbol{x}[n] - \boldsymbol{W}\boldsymbol{y}_{1:n} - \boldsymbol{t} \right) \right\} \tag{4.11}$$

$$= -2\,\mathrm{E}_{\boldsymbol{x}[n],\boldsymbol{y}_{1:n}} \left\{ \boldsymbol{x}[n] - \boldsymbol{W}\boldsymbol{y}_{1:n} - \boldsymbol{t} \right\} = -2\boldsymbol{\mu}_{\boldsymbol{x}[n]} + 2\boldsymbol{W}\boldsymbol{\mu}_{\boldsymbol{y}_{1:n}} + 2\boldsymbol{t} \overset{!}{=} \boldsymbol{0}_{N\times 1} \tag{4.12}$$

und damit gilt

$$\boldsymbol{t} = \boldsymbol{\mu}_{\boldsymbol{x}[n]} - \boldsymbol{W}\boldsymbol{\mu}_{\boldsymbol{y}_{1:n}}. \tag{4.13}$$

Die Ableitung von $\mathscr{L}$ nach $\boldsymbol{W}$ lautet unter Verwendung von Gl. (4.13)

$$\frac{\partial \mathscr{L}}{\partial \boldsymbol{W}} = \mathrm{E}_{\boldsymbol{x}[n],\boldsymbol{y}_{1:n}}\left\{-\left(\boldsymbol{x}[n]-\boldsymbol{W}\boldsymbol{y}_{1:n}-\boldsymbol{t}\right)\boldsymbol{y}_{1:n}^{\mathrm{T}}-\left(\boldsymbol{x}[n]-\boldsymbol{W}\boldsymbol{y}_{1:n}-\boldsymbol{t}\right)\boldsymbol{y}_{1:n}^{\mathrm{T}}\right\} \tag{4.14}$$

$$= -2\,\mathrm{E}_{\boldsymbol{x}[n],\boldsymbol{y}_{1:n}}\left\{\left(\boldsymbol{x}[n]-\boldsymbol{W}\boldsymbol{y}_{1:n}-\boldsymbol{t}\right)\boldsymbol{y}_{1:n}^{\mathrm{T}}\right\} = -2\boldsymbol{R}_{\boldsymbol{x}[n],\boldsymbol{y}_{1:n}}+2\boldsymbol{W}\boldsymbol{R}_{\boldsymbol{y}_{1:n},\boldsymbol{y}_{1:n}}+2\boldsymbol{t}\boldsymbol{\mu}_{\boldsymbol{y}_{1:n}}^{\mathrm{T}} \tag{4.15}$$

$$= -2\boldsymbol{R}_{\boldsymbol{x}[n],\boldsymbol{y}_{1:n}}+2\boldsymbol{W}\boldsymbol{R}_{\boldsymbol{y}_{1:n},\boldsymbol{y}_{1:n}}+2\left(\boldsymbol{\mu}_{\boldsymbol{x}[n]}-\boldsymbol{W}\boldsymbol{\mu}_{\boldsymbol{y}_{1:n}}\right)\boldsymbol{\mu}_{\boldsymbol{y}_{1:n}}^{\mathrm{T}} \tag{4.16}$$

$$= -2\underbrace{\left(\boldsymbol{R}_{\boldsymbol{x}[n],\boldsymbol{y}_{1:n}}-\boldsymbol{\mu}_{\boldsymbol{x}[n]}\boldsymbol{\mu}_{\boldsymbol{y}_{1:n}}^{\mathrm{T}}\right)}_{\boldsymbol{C}_{\boldsymbol{x}[n],\boldsymbol{y}_{1:n}}}+2\boldsymbol{W}\underbrace{\left(\boldsymbol{R}_{\boldsymbol{y}_{1:n},\boldsymbol{y}_{1:n}}-\boldsymbol{\mu}_{\boldsymbol{y}_{1:n}}\boldsymbol{\mu}_{\boldsymbol{y}_{1:n}}^{\mathrm{T}}\right)}_{\boldsymbol{C}_{\boldsymbol{y}_{1:n},\boldsymbol{y}_{1:n}}} \tag{4.17}$$

$$= -2\boldsymbol{C}_{\boldsymbol{x}[n],\boldsymbol{y}_{1:n}}+2\boldsymbol{W}\boldsymbol{C}_{\boldsymbol{y}_{1:n},\boldsymbol{y}_{1:n}} \stackrel{!}{=} \boldsymbol{0}_{N\times Kn} \tag{4.18}$$

und damit gilt für die Lösung der Optimierungsaufgabe in Gl. (4.8) unter Verwendung von Gl. (4.13)

$$\boldsymbol{W}_{\mathrm{lin}} = \boldsymbol{C}_{\boldsymbol{x}[n],\boldsymbol{y}_{1:n}}\boldsymbol{C}_{\boldsymbol{y}_{1:n},\boldsymbol{y}_{1:n}}^{-1}, \quad \boldsymbol{t}_{\mathrm{lin}} = \boldsymbol{\mu}_{\boldsymbol{x}[n]}-\boldsymbol{C}_{\boldsymbol{x}[n],\boldsymbol{y}_{1:n}}\boldsymbol{C}_{\boldsymbol{y}_{1:n},\boldsymbol{y}_{1:n}}^{-1}\boldsymbol{\mu}_{\boldsymbol{y}_{1:n}}. \tag{4.19}$$

Man sieht, dass für ein optimales lineares Filter nur statistische Momente erster und zweiter Ordnung (Erwartungswerte, Kovarianzen und Kreuzkovarianzen) notwendig sind. Ein Nachteil dieses Ansatzes ist, dass die Dimension der Matrix $\boldsymbol{W}_{\mathrm{lin}}$ mit jeder neuen Messung steigt.

II. „Maximum-a-posteriori“ statistisches Filter

Bei dem **Maximum-a-posteriori** (**MAP**) statistischen Filter wird die Tatsache ausgenutzt, dass die Rückschlusswahrscheinlichkeitsdichte (*A-posteriori*-WDF)

$$\mathrm{p}\Big(\boldsymbol{x}[n]\Big|\underbrace{\left[\boldsymbol{y}^{\mathrm{T}}[1],\boldsymbol{y}^{\mathrm{T}}[2],\ldots,\boldsymbol{y}^{\mathrm{T}}[n]\right]^{\mathrm{T}}}_{\boldsymbol{y}_{1:n}},\underbrace{\left[\boldsymbol{u}^{\mathrm{T}}[0],\boldsymbol{u}^{\mathrm{T}}[1],\ldots,\boldsymbol{u}^{\mathrm{T}}[n-1]\right]^{\mathrm{T}}}_{\boldsymbol{u}_{0:n-1}}\Big) \tag{4.20}$$

die gesamte Information beinhaltet, die zum Zeitpunkt n über den Systemzustand $\boldsymbol{x}[n]$ verfügbar ist. Der Ansatz beim MAP-Filter ist es, denjenigen Ausgang $\hat{\boldsymbol{x}}[n]$ zu liefern, für den die Rückschlusswahrscheinlichkeitsdichte am höchsten ist. Damit ergibt sich der Ausgang des MAP-Filters aus der Lösung der Optimierungsaufgabe

$$\hat{\boldsymbol{x}}_{\mathrm{MAP}}[n] = \underset{\boldsymbol{x}[n]}{\mathrm{argmax}}\left\{\mathrm{p}\big(\boldsymbol{x}[n]|\boldsymbol{y}_{1:n},\boldsymbol{u}_{0:n-1}\big)\right\}. \tag{4.21}$$

Das optimale MAP-Filter bestimmt zu jedem Zeitstempel n aus der *A-posteriori*-WDF einen optimalen Schätzwert entsprechend Gl. (4.21). In praktischen Anwendungen besteht die Schwierigkeit darin, die *A-posteriori*-WDF bzw. die statistischen Momente, die sie beschreiben, zu ermitteln. Wenn die Bestimmung der *A-posteriori*-WDF bzw. der statistischen Momente, die sie beschreiben, rekursiv durchgeführt wird, indem sie von Zeitstempel zu Zeitstempel geschätzt werden, so bezeichnet man diese Vorgehensweise als **Rekursive Bayesianische Filterung**, und man kann sie wie folgt darstellen:

$$\begin{aligned}
&\mathrm{p}(\boldsymbol{x}[0])\\
&\mathrm{p}\big(\boldsymbol{x}[1]|\boldsymbol{y}_{1:1},\boldsymbol{u}_{0:0}\big) \quad\longrightarrow\quad \hat{\boldsymbol{x}}_{\mathrm{MAP}}[1]\\
&\mathrm{p}\big(\boldsymbol{x}[2]|\boldsymbol{y}_{1:2},\boldsymbol{u}_{0:1}\big) \quad\longrightarrow\quad \hat{\boldsymbol{x}}_{\mathrm{MAP}}[2]\\
&\quad\vdots\\
&\mathrm{p}\big(\boldsymbol{x}[n]|\boldsymbol{y}_{1:n},\boldsymbol{u}_{0:n-1}\big) \quad\longrightarrow\quad \hat{\boldsymbol{x}}_{\mathrm{MAP}}[n].
\end{aligned} \tag{4.22}$$

Wie später gezeigt wird, ist das Kalman-Filter ein Spezialfall des MAP-Filters für den Fall, dass die *A-posteriori*-WDF Gaußsch ist.

III. „Minimum Mean Squared Error“ statistisches Filter

Der Ansatz beim **Minimum Mean Squared Error** (**MMSE**) statistischen Filter ist es, denjenigen Wert $\hat{\boldsymbol{x}}[n]$ zu berechnen, für den im Mittel der quadratische Fehler zwischen $\boldsymbol{x}[n]$ und dem Ausgang $\hat{\boldsymbol{x}}[n]$ minimal ist. Damit ergibt sich der Ausgang des MMSE-Filters aus der Lösung der Optimierungsaufgabe

$$\hat{\boldsymbol{x}}_{\mathrm{MMSE}}[n] = \underset{\hat{\boldsymbol{x}}[n]}{\operatorname{argmin}} \left\{ \mathrm{E}_{\boldsymbol{x}[n]|\boldsymbol{y}_{1:n},\boldsymbol{u}_{0:n-1}} \left\{ \|\boldsymbol{x}[n] - \hat{\boldsymbol{x}}[n]\|_2^2 \right\} \right\}. \tag{4.23}$$

Um die Lösung zu finden, werden zur Vereinfachung der Notation in diesem Abschnitt folgende Abkürzungen eingeführt

$$\boldsymbol{z} := \boldsymbol{y}_{1:n}, \boldsymbol{u}_{0:n-1}, \quad \boldsymbol{x} := \boldsymbol{x}[n] \quad \text{und} \quad \hat{\boldsymbol{x}} := \hat{\boldsymbol{x}}[n]. \tag{4.24}$$

Weiterhin ist für die weitere Herleitung zu beachten, dass $\hat{\boldsymbol{x}}$ nur eine Funktion der Messwerte $\boldsymbol{y}_{1:n}$ und der Eingänge $\boldsymbol{u}_{0:n-1}$ ist und damit bei gegebenem $\boldsymbol{z}$ für den bedingten Erwartungswert $\mathrm{E}_{\boldsymbol{x}|\boldsymbol{z}}\{\hat{\boldsymbol{x}}\}$ gilt

$$\mathrm{E}_{\boldsymbol{x}|\boldsymbol{z}}\{\hat{\boldsymbol{x}}\} = \hat{\boldsymbol{x}}. \tag{4.25}$$

Die Kostenfunktion, die zu minimieren ist, lautet

$$\mathscr{L} = \mathrm{E}_{\boldsymbol{x}|\boldsymbol{z}}\left\{(\boldsymbol{x}-\hat{\boldsymbol{x}})^{\mathrm{T}}(\boldsymbol{x}-\hat{\boldsymbol{x}})\right\} = \mathrm{E}_{\boldsymbol{x}|\boldsymbol{z}}\left\{\boldsymbol{x}^{\mathrm{T}}\boldsymbol{x} - \boldsymbol{x}^{\mathrm{T}}\hat{\boldsymbol{x}} - \hat{\boldsymbol{x}}^{\mathrm{T}}\boldsymbol{x} + \hat{\boldsymbol{x}}^{\mathrm{T}}\hat{\boldsymbol{x}}\right\} \tag{4.26}$$

$$= \mathrm{E}_{\boldsymbol{x}|\boldsymbol{z}}\left\{\boldsymbol{x}^{\mathrm{T}}\boldsymbol{x}\right\} - \mathrm{E}_{\boldsymbol{x}|\boldsymbol{z}}\left\{\boldsymbol{x}^{\mathrm{T}}\right\}\hat{\boldsymbol{x}} - \hat{\boldsymbol{x}}^{\mathrm{T}}\,\mathrm{E}_{\boldsymbol{x}|\boldsymbol{z}}\{\boldsymbol{x}\} + \hat{\boldsymbol{x}}^{\mathrm{T}}\hat{\boldsymbol{x}}. \tag{4.27}$$

Addiert und subtrahiert man zu diesem Ausdruck den Term $\mathrm{E}_{\boldsymbol{x}|\boldsymbol{z}}\left\{\boldsymbol{x}^{\mathrm{T}}|\boldsymbol{z}\right\}\mathrm{E}_{\boldsymbol{x}|\boldsymbol{z}}\{\boldsymbol{x}|\boldsymbol{z}\}$, so erhält man

$$\mathscr{L} = \mathrm{E}_{\boldsymbol{x}|\boldsymbol{z}}\left\{\boldsymbol{x}^{\mathrm{T}}\boldsymbol{x}\right\} - \mathrm{E}_{\boldsymbol{x}|\boldsymbol{z}}\left\{\boldsymbol{x}^{\mathrm{T}}\right\}\mathrm{E}_{\boldsymbol{x}|\boldsymbol{z}}\{\boldsymbol{x}\} + \left(\hat{\boldsymbol{x}} - \mathrm{E}_{\boldsymbol{x}|\boldsymbol{z}}\{\boldsymbol{x}\}\right)^{\mathrm{T}}\left(\hat{\boldsymbol{x}} - \mathrm{E}_{\boldsymbol{x}|\boldsymbol{z}}\{\boldsymbol{x}\}\right). \tag{4.28}$$

Die ersten zwei Terme sind nicht abhängig von $\hat{\boldsymbol{x}}$, und die Kostenfunktion $\mathscr{L}$ wird damit durch die Wahl von $\hat{\boldsymbol{x}}$ minimal, falls gilt

$$\hat{\boldsymbol{x}} - \mathrm{E}_{\boldsymbol{x}|\boldsymbol{z}}\{\boldsymbol{x}\} = \boldsymbol{0}_{N\times 1}. \tag{4.29}$$

Das heißt, dass der Ausgang des MMSE-Filters der **bedingte Erwartungswertschätzer** ist

$$\hat{\boldsymbol{x}}_{\mathrm{MMSE}}[n] = \mathrm{E}_{\boldsymbol{x}[n]|\boldsymbol{y}_{1:n},\boldsymbol{u}_{0:n-1}}\{\boldsymbol{x}[n]\}. \tag{4.30}$$

Dieses Ergebnis ist unabhängig von der Art der WDF $\mathrm{p}(\boldsymbol{x}[n]|\boldsymbol{y}_{1:n}, \boldsymbol{u}_{0:n-1})$.

Ist die *A-posteriori*-WDF Gaußsch, so ist der bedingte Erwartungswert gemäß Gl. (2.150) linear (vgl. Gl. (4.19)). Es gilt für die Gaußsche *A-posteriori*-WDF

$$\hat{\boldsymbol{x}}_{\mathrm{MMSE}}[n] = \boldsymbol{\mu}_{\boldsymbol{x}[n]} + \boldsymbol{C}_{\boldsymbol{x}[n],\boldsymbol{y}_{1:n}}\boldsymbol{C}_{\boldsymbol{y}_{1:n},\boldsymbol{y}_{1:n}}^{-1}\left(\boldsymbol{y}_{1:n} - \boldsymbol{\mu}_{\boldsymbol{y}_{1:n}}\right). \tag{4.31}$$

Da bei einer Gaußwahrscheinlichkeitsdichte das Maximum am Erwartungswert zu finden ist, ist das MMSE-Filter für Gaußsche-Rückschlusswahrscheinlichkeitsdichten gleichzeitig auch das MAP-Filter. Es gilt also für die Gaußsche *A-posteriori*-WDF

$$\hat{\boldsymbol{x}}_{\mathrm{MMSE}}[n] = \hat{\boldsymbol{x}}_{\mathrm{MAP}}[n] = \hat{\boldsymbol{x}}_{\mathrm{lin}}[n]. \tag{4.32}$$

Es ist zu beachten, dass Gl. (4.30) allgemein gilt und Gl. (4.31) und Gl. (4.32) für den Spezialfall einer Gaußschen *A-posteriori*-WDF gültig sind.

Übung 4.1

Bayes Zustandsschätzer

Das Konzept der Bayes-Zustandsschätzung bzw. Bayes-Filterung beruht darauf, dass vorhandenes Wissen, das *A-priori*-Wissen, für die Bestimmung der gesuchten Parameter bzw. Zustände eingebracht wird. Dafür verwendet man einen probabilistischen Rahmen, in dem das Messmodell durch die WDF $\mathrm{p}\left(\boldsymbol{y}[n]|\boldsymbol{x}_{0:n}, \boldsymbol{u}_{0:n}\right)$ und das Systemmodell durch die WDF $\mathrm{p}\left(\boldsymbol{x}[n]|\boldsymbol{x}_{0:n-1}, \boldsymbol{u}_{0:n-1}\right)$ beschrieben werden. Ziel der Bayes-Zustandsschätzung ist es, die *A-posteriori*-WDF $\mathrm{p}\left(\boldsymbol{x}[n]|\boldsymbol{y}_{1:n}, \boldsymbol{u}_{0:n-1}\right)$ zu bestimmen.

Um die *A-posteriori*-WDF effizient zu berechnen, ist eine rekursive Formulierung gewünscht. Dazu nutzt man den Satz von Bayes:

$$\begin{aligned}
\mathrm{p}\left(\boldsymbol{x}[n]|\boldsymbol{y}_{1:n}, \boldsymbol{u}_{0:n-1}\right) &= \frac{\mathrm{p}\left(\boldsymbol{y}_{1:n}|\boldsymbol{x}[n], \boldsymbol{u}_{0:n-1}\right)\mathrm{p}\left(\boldsymbol{x}[n]|\boldsymbol{u}_{0:n-1}\right)}{\mathrm{p}\left(\boldsymbol{y}_{1:n}|\boldsymbol{u}_{0:n-1}\right)} \\
&= \frac{\mathrm{p}\left(\boldsymbol{y}[n], \boldsymbol{y}_{1:n-1}|\boldsymbol{x}[n], \boldsymbol{u}_{0:n-1}\right)\mathrm{p}\left(\boldsymbol{x}[n]|\boldsymbol{u}_{0:n-1}\right)}{\mathrm{p}\left(\boldsymbol{y}[n], \boldsymbol{y}_{1:n-1}|\boldsymbol{u}_{0:n-1}\right)} \\
&= \frac{\mathrm{p}\left(\boldsymbol{y}[n]|\boldsymbol{y}_{1:n-1}, \boldsymbol{x}[n], \boldsymbol{u}_{0:n-1}\right)\mathrm{p}\left(\boldsymbol{y}_{1:n-1}|\boldsymbol{x}[n], \boldsymbol{u}_{0:n-1}\right)\mathrm{p}\left(\boldsymbol{x}[n]|\boldsymbol{u}_{0:n-1}\right)}{\mathrm{p}\left(\boldsymbol{y}[n]|\boldsymbol{y}_{1:n-1}, \boldsymbol{u}_{0:n-1}\right)\mathrm{p}\left(\boldsymbol{y}_{1:n-1}|\boldsymbol{u}_{0:n-1}\right)} \\
&= \frac{\mathrm{p}\left(\boldsymbol{y}[n]|\boldsymbol{y}_{1:n-1}, \boldsymbol{x}[n], \boldsymbol{u}_{0:n-1}\right)\mathrm{p}\left(\boldsymbol{x}[n]|\boldsymbol{y}_{1:n-1}, \boldsymbol{u}_{0:n-1}\right)\mathrm{p}\left(\boldsymbol{y}_{1:n-1}|\boldsymbol{u}_{0:n-1}\right)\mathrm{p}\left(\boldsymbol{x}[n]|\boldsymbol{u}_{0:n-1}\right)}{\mathrm{p}\left(\boldsymbol{y}[n]|\boldsymbol{y}_{1:n-1}, \boldsymbol{u}_{0:n-1}\right)\mathrm{p}\left(\boldsymbol{y}_{1:n-1}|\boldsymbol{u}_{0:n-1}\right)\mathrm{p}\left(\boldsymbol{x}[n]|\boldsymbol{u}_{0:n-1}\right)} \\
&= \frac{\mathrm{p}\left(\boldsymbol{y}[n]|\boldsymbol{y}_{1:n-1}, \boldsymbol{x}[n], \boldsymbol{u}_{0:n-1}\right)\mathrm{p}\left(\boldsymbol{x}[n]|\boldsymbol{y}_{1:n-1}, \boldsymbol{u}_{0:n-1}\right)}{\mathrm{p}\left(\boldsymbol{y}[n]|\boldsymbol{y}_{1:n-1}, \boldsymbol{u}_{0:n-1}\right)}.
\end{aligned} \tag{4.33}$$

Es kann aufgrund des Messmodells aus Gl. (4.2) angenommen werden, dass der aktuelle Messwert $\boldsymbol{y}[n]$ nur von dem aktuellen Zustand $\boldsymbol{x}[n]$, dem aktuellen Eingang $\boldsymbol{u}[n]$ und dem Messrauschen $\boldsymbol{\eta}_{\mathrm{M}}[n]$ abhängt, die Historie der Messwerte $\boldsymbol{y}_{1:n-1}$ also bereits in dem Zustand $\boldsymbol{x}[n]$ enthalten ist, so dass gilt

$$\mathrm{p}\left(\boldsymbol{y}[n]|\boldsymbol{y}_{1:n-1}, \boldsymbol{x}[n], \boldsymbol{u}_{0:n-1}\right) = \mathrm{p}\left(\boldsymbol{y}[n]|\boldsymbol{x}[n], \boldsymbol{u}[n]\right).$$

Die *A-posteriori*-WDF aus Gl. (4.33) lässt sich damit schreiben als

$$\begin{aligned}
\mathrm{p}\left(\boldsymbol{x}[n]|\boldsymbol{y}_{1:n}, \boldsymbol{u}_{0:n-1}\right) &= \frac{\mathrm{p}\left(\boldsymbol{y}[n]|\boldsymbol{x}[n], \boldsymbol{u}[n]\right)\mathrm{p}\left(\boldsymbol{x}[n]|\boldsymbol{y}_{1:n-1}, \boldsymbol{u}_{0:n-1}\right)}{\mathrm{p}\left(\boldsymbol{y}[n]|\boldsymbol{y}_{1:n-1}, \boldsymbol{u}_{0:n-1}\right)} \\
&= \frac{\mathrm{p}\left(\boldsymbol{y}[n]|\boldsymbol{x}[n], \boldsymbol{u}[n]\right)\mathrm{p}\left(\boldsymbol{x}[n]|\boldsymbol{y}_{1:n-1}, \boldsymbol{u}_{0:n-1}\right)}{\int_{\mathbb{X}}\mathrm{p}\left(\boldsymbol{y}[n], \boldsymbol{x}[n]|\boldsymbol{y}_{1:n-1}, \boldsymbol{u}_{0:n-1}\right)\mathrm{d}\boldsymbol{x}[n]} \\
&= \frac{\mathrm{p}\left(\boldsymbol{y}[n]|\boldsymbol{x}[n], \boldsymbol{u}[n]\right)\mathrm{p}\left(\boldsymbol{x}[n]|\boldsymbol{y}_{1:n-1}, \boldsymbol{u}_{0:n-1}\right)}{\int_{\mathbb{X}}\mathrm{p}\left(\boldsymbol{y}[n]|\boldsymbol{x}[n], \boldsymbol{y}_{1:n-1}, \boldsymbol{u}_{0:n-1}\right)\mathrm{p}\left(\boldsymbol{x}[n]|\boldsymbol{y}_{1:n-1}, \boldsymbol{u}_{0:n-1}\right)\mathrm{d}\boldsymbol{x}[n]} \\
&= \frac{\mathrm{p}\left(\boldsymbol{y}[n]|\boldsymbol{x}[n], \boldsymbol{u}[n]\right)\mathrm{p}\left(\boldsymbol{x}[n]|\boldsymbol{y}_{1:n-1}, \boldsymbol{u}_{0:n-1}\right)}{\int_{\mathbb{X}}\mathrm{p}\left(\boldsymbol{y}[n]|\boldsymbol{x}[n], \boldsymbol{u}[n]\right)\mathrm{p}\left(\boldsymbol{x}[n]|\boldsymbol{y}_{1:n-1}, \boldsymbol{u}_{0:n-1}\right)\mathrm{d}\boldsymbol{x}[n]}.
\end{aligned} \tag{4.34}$$

Die Komponenten dieser Gleichung können wie folgt interpretiert werden:

- Die *A-priori*-WDF $\mathrm{p}\left(\boldsymbol{x}[n]|\boldsymbol{y}_{1:n-1}, \boldsymbol{u}_{0:n-1}\right)$ fasst das gesamte Wissen der Vergangenheit über den aktuellen Zustand $\boldsymbol{x}[n]$ zusammen.

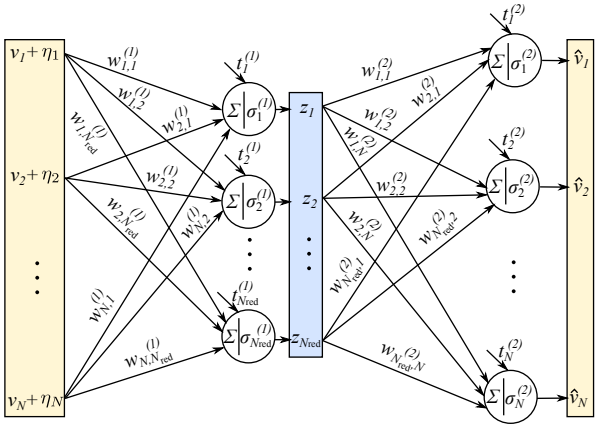

Abbildung 5.72 Denoising Autoencoder mit einer verdeckten Schicht.

Convolutional Autoencoder

Convolutional Autoencoder sind den in Unterkapitel 5.2.4 vorgestellten CNN sehr ähnlich und werden in der Regel für Bilddaten genutzt. Sie bestehen aus einer Folge von Faltungs-, Detektions- und Pooling-Schichten im Encoder und aus einer Folge von dazugehörigen Operationen im Decoder. Dabei muss berücksichtigt werden, dass durch die Faltung und durch das Pooling im Encoder ein Downsampling stattgefunden hat. Dieses muss im Decoder rückgängig gemacht werden, damit der Ausgang die gleiche Dimension hat wie der Eingang. Dies wird in der Regel durch eine Operation umgesetzt, die als **Upsampling** bekannt ist. Dabei wird zunächst jeder Pixelwert des Eingangs einer Decoderschicht mit einem Kernel, dessen Gewichte im Training angepasst werden, elementweise multipliziert. Das Ergebnis wird dann so zusammengesetzt, dass sich die gleiche Dimension ergibt wie beim Eingang in die dazugehörige Encoderschicht. Hierbei werden bei der Zusammensetzung Überlappungen additiv umgesetzt. Eine alternative Erklärung der Upsampling-Operation beruht darauf, dass die Faltung eine lineare Operation ist und sich als Matrix-Vektor-Multiplikation darstellen lässt. Während im Encoder die Faltungsmatrix das vektorisierte Eingangsbild auf ein Bild kleinerer Dimension abbildet, wird beim Upsampling die transponierte dieser Matrix genutzt.

So wie ein CNN als ein Spezialfall eines Multilayer Perceptrons betrachtet werden kann, ist dies auch bei den Autoencodern der Fall. Das „Parameter Sharing“ und die „Sparse Interaction“ führen dazu, dass deutlich weniger Parameter notwendig sind als bei einem vollvernetzten Autoencoder, was wiederum einen regularisierenden Effekt hat.

Probabilistische Sicht auf Autoencoder

Ein interessanter Zugang zu Autoencodern, deren Ziel es ist, am Ausgang den Eingang möglichst gut zu rekonstruieren, ergibt sich aus informationstheoretischen Betrachtungen und wurde in [VLL+10] vorgestellt. Der Ansatz dabei liegt darin zu fordern, dass die latente Darstellung $\boldsymbol{z}$ des Eingangs möglichst viel Information aus dem Eingang $\boldsymbol{v}$ behält. Dies kann mit Hilfe der in Unterkapitel 2.3.3 eingeführten Transinformation ausgedrückt werden, indem man fordert, dass die Transinformation $I(\boldsymbol{v};\boldsymbol{z})$ maximiert wird.

Dabei wird angenommen, dass es eine nicht notwendigerweise deterministische Abbildung $\boldsymbol{z} = f(\boldsymbol{v};\boldsymbol{\phi})$ von $\boldsymbol{v}$ nach $\boldsymbol{z}$ gibt, deren Parameter $\boldsymbol{\phi}$ gelernt werden sollen. Weil die WDF $\mathrm{p}(\boldsymbol{v})$

des Eingangs $\boldsymbol{v}$ nicht von $\boldsymbol{\phi}$ abhängt, spielt $H(\boldsymbol{v})$ keine Rolle bei der Maximierung der Transinformation. Mit Hilfe von Gl. (2.137) und Gl. (2.138) lässt sich die Maximierung der Transinformation schreiben als

$$\begin{aligned}\underset{\boldsymbol{\phi}}{\operatorname{argmax}}\{I(\boldsymbol{v};\boldsymbol{z})\} &= \underset{\boldsymbol{\phi}}{\operatorname{argmax}}\{H(\boldsymbol{v}) - H(\boldsymbol{v}|\boldsymbol{z})\} = \underset{\boldsymbol{\phi}}{\operatorname{argmax}}\{-H(\boldsymbol{v}|\boldsymbol{z})\} \\ &= \underset{\boldsymbol{\phi}}{\operatorname{argmax}}\left\{\mathrm{E}_{\mathrm{p}(\boldsymbol{v},\boldsymbol{z})}\left\{\log_2(\mathrm{p}(\boldsymbol{v}|\boldsymbol{z}))\right\}\right\},\end{aligned} \tag{5.348}$$

wobei $\boldsymbol{z} = f(\boldsymbol{v};\boldsymbol{\phi})$ und die Notation $\mathrm{E}_{\mathrm{p}(\boldsymbol{v},\boldsymbol{z})}\left\{\log_2(\mathrm{p}(\boldsymbol{v}|\boldsymbol{z}))\right\}$ verwendet wird für

$$\mathrm{E}_{\mathrm{p}(\boldsymbol{v},\boldsymbol{z})}\left\{\log_2(\mathrm{p}(\boldsymbol{v}|\boldsymbol{z}))\right\} = \int\limits_{\mathbb{R}^N}\int\limits_{\mathbb{R}^{N_{\mathrm{red}}}} \log_2(\mathrm{p}(\boldsymbol{v}|\boldsymbol{z}))\mathrm{p}(\boldsymbol{v},\boldsymbol{z})\mathrm{d}\boldsymbol{z}\mathrm{d}\boldsymbol{v}. \tag{5.349}$$

Weil $\mathrm{p}(\boldsymbol{v}|\boldsymbol{z})$ nicht bekannt ist, betrachtet man eine andere WDF $\mathrm{q}(\boldsymbol{v}|\boldsymbol{z})$ und erhält unter Berücksichtigung der Tatsache, dass die Kullback-Leibler-Divergenz immer größer oder gleich null ist,

$$\begin{aligned}\mathrm{KL}(\mathrm{p}(\boldsymbol{v}|\boldsymbol{z})\|\mathrm{q}(\boldsymbol{v}|\boldsymbol{z})) &= \int\limits_{\mathbb{R}^N} \mathrm{p}(\boldsymbol{v}|\boldsymbol{z})\ln\left(\frac{\mathrm{p}(\boldsymbol{v}|\boldsymbol{z})}{\mathrm{q}(\boldsymbol{v}|\boldsymbol{z})}\right)\mathrm{d}\boldsymbol{v} \\ &= \int\limits_{\mathbb{R}^N} \mathrm{p}(\boldsymbol{v}|\boldsymbol{z})\ln\big(\mathrm{p}(\boldsymbol{v}|\boldsymbol{z})\big) - \mathrm{p}(\boldsymbol{v}|\boldsymbol{z})\ln\big(\mathrm{q}(\boldsymbol{v}|\boldsymbol{z})\big)\,\mathrm{d}\boldsymbol{v} \geq 0\end{aligned} \tag{5.350}$$

nach Multiplikation mit $\mathrm{p}(\boldsymbol{z})$ und zusätzliche Integration über $\mathbb{R}^{N_{\mathrm{red}}}$ die Ungleichung

$$\mathrm{E}_{\mathrm{p}(\boldsymbol{v},\boldsymbol{z})}\left\{\log_2(\mathrm{p}(\boldsymbol{v}|\boldsymbol{z}))\right\} \geq \mathrm{E}_{\mathrm{p}(\boldsymbol{v},\boldsymbol{z})}\left\{\log_2(\mathrm{q}(\boldsymbol{v}|\boldsymbol{z}))\right\}. \tag{5.351}$$

Wenn die WDF $\mathrm{q}(\boldsymbol{v}|\boldsymbol{z})$ von den Parametern $\boldsymbol{\theta}$ abhängt, d. h. als $\mathrm{q}(\boldsymbol{v}|\boldsymbol{z};\boldsymbol{\theta})$ geschrieben werden muss, so ergibt sich aus Gl. (5.348) und Gl. (5.351), dass die Maximierung von $\mathrm{E}_{\mathrm{p}(\boldsymbol{v},\boldsymbol{z})}\left\{\log_2(\mathrm{q}(\boldsymbol{v}|\boldsymbol{z};\boldsymbol{\theta}))\right\}$, mit $\boldsymbol{z} = f(\boldsymbol{v};\boldsymbol{\phi})$, bezüglich $\boldsymbol{\phi}$ und $\boldsymbol{\theta}$ der Maximierung einer unteren Grenze der Transinformation $I(\boldsymbol{v};\boldsymbol{z})$ entspricht. Nimmt man nun an, dass die Abbildung $\boldsymbol{z} = f(\boldsymbol{v};\boldsymbol{\phi})$ deterministisch ist, so dass $\mathrm{p}(\boldsymbol{v},\boldsymbol{z}) = \mathrm{p}(\boldsymbol{v})$ und $\boldsymbol{z}$ als $\boldsymbol{z} = f_{\boldsymbol{\phi}}(\boldsymbol{v})$ geschrieben werden kann, so wird die untere Grenze der Transinformation zwischen $\boldsymbol{v}$ und $\boldsymbol{z}$ maximiert durch die Parameterwahl

$$\left\{\hat{\boldsymbol{\phi}},\hat{\boldsymbol{\theta}}\right\} = \underset{\boldsymbol{\phi},\boldsymbol{\theta}}{\operatorname{argmax}}\left\{\mathrm{E}_{\mathrm{p}(\boldsymbol{v})}\left\{\log_2(\mathrm{q}(\boldsymbol{v}|\boldsymbol{z};\boldsymbol{\phi},\boldsymbol{\theta}))\right\}\right\} = \underset{\boldsymbol{\phi},\boldsymbol{\theta}}{\operatorname{argmax}}\left\{\mathrm{E}_{\mathrm{p}(\boldsymbol{v})}\left\{\log_2(\mathrm{q}(\boldsymbol{v}|f_{\boldsymbol{\phi}}(\boldsymbol{v});\boldsymbol{\phi},\boldsymbol{\theta}))\right\}\right\}. \tag{5.352}$$

Betrachtet man nun Autoencoder, so kann der Ausgang $\hat{\boldsymbol{v}}$ als eine Kenngröße der WDF $\mathrm{q}(\boldsymbol{v}|\hat{\boldsymbol{v}})$ verstanden werden, für die $\mathrm{q}(\boldsymbol{v}|\hat{\boldsymbol{v}})$ hohe Werte annimmt. Dadurch hat $\boldsymbol{v}$ eine hohe WDF, wenn $\hat{\boldsymbol{v}}$ bekannt ist. In der Regel ist diese Kenngröße der WDF $\mathrm{q}(\boldsymbol{v}|\hat{\boldsymbol{v}})$ der Erwartungswert. Mit der deterministischen eineindeutigen Abbildung $\boldsymbol{z} = f_{\boldsymbol{\phi}}(\boldsymbol{v})$ und der deterministischen eineindeutigen Abbildung $\hat{\boldsymbol{v}} = g_{\boldsymbol{\theta}}(\boldsymbol{z})$ entspricht die WDF $\mathrm{q}(\boldsymbol{v}|\boldsymbol{z};\boldsymbol{\phi},\boldsymbol{\theta})$ aus Gl. (5.352) der WDF $\mathrm{q}(\boldsymbol{v}|\hat{\boldsymbol{v}} = g_{\boldsymbol{\theta}}(\boldsymbol{z});\boldsymbol{\phi},\boldsymbol{\theta})$

$$\mathrm{q}\left(\boldsymbol{v}|\boldsymbol{z};\boldsymbol{\phi},\boldsymbol{\theta}\right) = \mathrm{q}\left(\boldsymbol{v}|\hat{\boldsymbol{v}} = g_{\boldsymbol{\theta}}(\boldsymbol{z});\boldsymbol{\phi},\boldsymbol{\theta}\right). \tag{5.353}$$

Beim Training eines Autoencoders sollen die Parameter derart gefunden werden, dass $\boldsymbol{v}$ eine hohe WDF hat, wenn $\hat{\boldsymbol{v}}$ bekannt ist, d. h. derart, dass $\mathrm{q}(\boldsymbol{v}|\hat{\boldsymbol{v}})$ maximiert bzw. $-\ln\big(\mathrm{q}(\boldsymbol{v}|\hat{\boldsymbol{v}})\big)$ minimiert wird. Dies entspricht der Minimierung der Verlustfunktion

$$\mathscr{L}(\boldsymbol{v},\hat{\boldsymbol{v}}) = -\ln\big(\mathrm{q}(\boldsymbol{v}|\hat{\boldsymbol{v}})\big). \tag{5.354}$$

einer Linkskurve zur Unfallvermeidung einen Spurwechsel plant, ohne dabei die Fahrbahn zu verlassen.

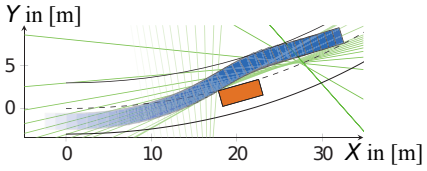

Abbildung 5.80 Bestimmung von Vermeidungsbeschleunigungen [HUBK15].

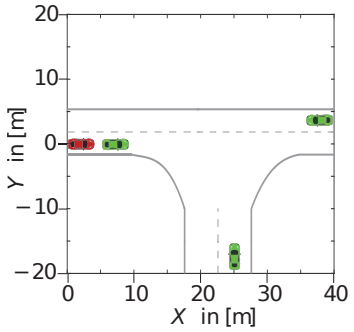

Abbildung 5.81 Verkehrsszenario zum Zeitpunkt t_0 für die Berechnung von prädizierten Belegungskarten [NBS17].

Weil die Lösung der MinMax-Optimierungsaufgabe einen Solver benötigt und deswegen nicht für den Einsatz auf einem Automotive-Mikrocontroller geeignet ist, wird in [HUBK15] auch ein Verfahren vorgestellt, das diese Art der Kritikalitätsschätzung mittels Vermeidungsbeschleunigungen viel recheneffizienter macht. Dieser Ansatz beruht auf der Berechnung der Kritikalität mit Hilfe des Random Forest-Klassifikators aus Unterkapitel 5.5. Dafür werden auf leistungsfähigen Rechnern viele kritische Szenarien generiert und die Vermeidungsbeschleunigungen jeweils mittels MinMax-Optimierung berechnet. Beruhend auf den Vermeidungsbeschleunigungen wird die Kritikalität in 4 Klassen eingeteilt und als Label für das Training des Random Forest-Klassifikators genutzt. Es wird in [HUBK15] gezeigt, dass der Random Forest-Klassifikator in der Lage ist, die Kritikalität von Verkehrsszenarien zu lernen.

Die Motivation, maschinelle Lernalgorithmen in dieser Anwendung zu nutzen, liegt also hauptsächlich in einer recheneffizienten Methode zur Klassifizierung der Kritikalität eines Verkehrsszenarios und kann der Kategorie II. „Rechenressourcen“ aus Abb. 5.77 zugeordnet werden. Das Verfahren kann aber auch als paralleler Pfad zu einem modellbasierten Ansatz zur Kritikalitätsberechnung genutzt werden, so dass es auch der Kategorie III. „Diversität“ aus Abb. 5.77 zugeordnet werden kann.

Prädizierte Belegungskarten

Eine hilfreiche Darstellung eines Verkehrsszenarios für die Schätzung der Kritikalität, aber auch für die Planung sicherer Trajektorien des EGO-Fahrzeugs sind die **prädizierten Belegungskarten** bzw. **Predicted Occupancy Grids**. Diese beruhen darauf, dass die Ebene, in der sich das EGO-Fahrzeug bewegt, in Gitterelemente unterteilt wird und für jedes der Gitterelemente für einen gewünschten Prädiktionszeitpunkt die Belegungswahrscheinlichkeit geschätzt wird. Diese Ortsdiskretisierung hat auch den Vorteil, dass ein Predicted Occupancy Grid als Bild mit einer festen Anzahl an „Pixeln“ interpretiert werden kann und damit als Eingang in weitere maschinelle Lernalgorithmen für die Kritikalitätsschätzung oder die Trajektorienplanung dienen kann. In [NB16], [NBS17] und [NBS18] wird ein Ansatz vorgestellt, in dem mittels maschineller Lernverfahren die Predicted Occupancy Grids recheneffizient berechnet werden können. Abb. 5.81 zeigt zur Verdeutlichung des Ansatzes ein Verkehrsszenario,

für das die Predicted Occupancy Grids bestimmt werden sollen. Für den Prädiktionszeitpunkt von 2,0 s ist das sich ergebende Predicted Occupancy Grid in Abb. 5.82 visualisiert. Je heller ein Gitterelement ist, umso wahrscheinlicher ist es, dass es belegt sein wird. Abb. 5.82 links

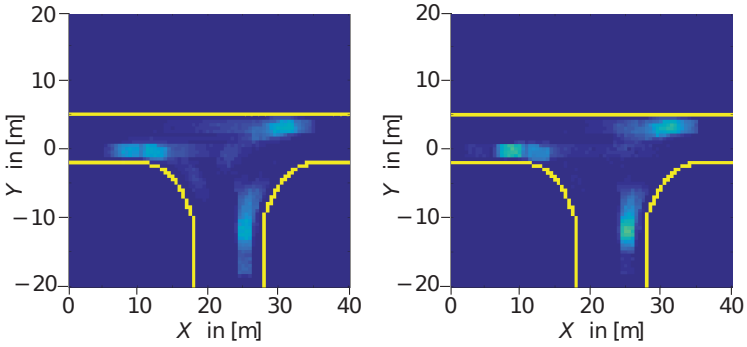

Abbildung 5.82 Prädizierte Belegungskarten für das Verkehrsszenario aus Abb. 5.81: modellbasiert (links) und mittels maschinellem Lernen (rechts) für die Prädiktionszeit von 2,0 s [NBS17].

zeigt das resultierende Predicted Occupancy Grid, das mit einem modellbasierten Ansatz berechnet wurde, und Abb. 5.82 rechts das Predicted Occupancy Grid des maschinellen Lernalgorithmus. Der modellbasierte Ansatz berechnet mit Hilfe der in Unterkapitel 3.2 vorgestellten Bewegungsmodelle und Differenzengleichungen für jedes Objekt mehrere Hypothesen für Trajektorien numerisch, was zu einem sehr hohen Rechenressourcenbedarf führt. Als Eingang für den maschinellen Lernalgorithmus dient die Darstellung eines Szenarios zum aktuellen Zeitpunkt t_0 als „Augmented Occupancy Grid“, d. h. für jedes Gitterelement gibt es jeweils einen Vektor in dem gespeichert wird, zu welchem Objekttyp das jeweilige Gitterelement gehört, welche Geschwindigkeit das dazugehörige Objekt hat, etc. Die Details dazu sind in [NB16] zu finden. Das Ziel des maschinellen Lernalgorithmus ist es, aus einem Augmented Occupancy Grid ein Predicted Occupancy Grid zu generieren. Zwei Architekturen, um dieses Ziel zu erreichen, sind in Abb. 5.83 zu sehen. In der oberen Architektur wird ein Stacked Denoising Autoencoder und ein Satz von Random Forest-Algorithmen (RF) für Regression verwendet. Stacked Denoising Autoencoder wurden in Unterkapitel 5.6.3 und der Random Forest-Algorithmus in Unterkapitel 5.5 eingeführt. Der Autoencoder wird für diese Anwendung mit den Augmented Occupancy Grids antrainiert, und anschließend wird der Decoder verworfen und nur der Encoder genutzt. Der Nachteil dieser Methode ist die noch sehr hohe Anzahl an benötigten Random Forest-Algorithmen, auch wenn diese sehr gut parallelisiert werden können, weil pro „Pixel“ eines Predicted Occupancy Grids jeweils ein Random Forest notwendig ist. Die untere Architektur zeigt einen Weg, um die Anzahl der Random Forest-Algorithmen stark zu reduzieren, indem zwei Stacked Denoising Autoencoder genutzt werden. Der erste Autoencoder wird mit den Augmented Occupancy Grids antrainiert und der zweite mit den Predicted Occupancy Grids. Beim ersten wird nur der Encoder behalten und beim zweiten nur der Decoder. Zwischen den beiden Komponenten ist ein Satz von Random Forest (RF)-Regressionsalgorithmen, die von der niederdimensionalen Darstellung der Augmented Occupancy Grids auf die niederdimensionale Darstellung der prädizierten Belegungskarten abbildet. Diese Architektur ermöglicht eine insgesamt recheneffiziente Berechnung von Predicted Occupancy Grids. Ein weiterer Vorteil des Verfahrens liegt darin, dass auch Interaktionen und Reaktionen der Verkehrsteilnehmer untereinander bei der Berechnung der Predicted Occupancy Grids ohne zusätzlichen Aufwand berücksichtigt werden können. Falls Interaktionen und Reaktionen zwi-

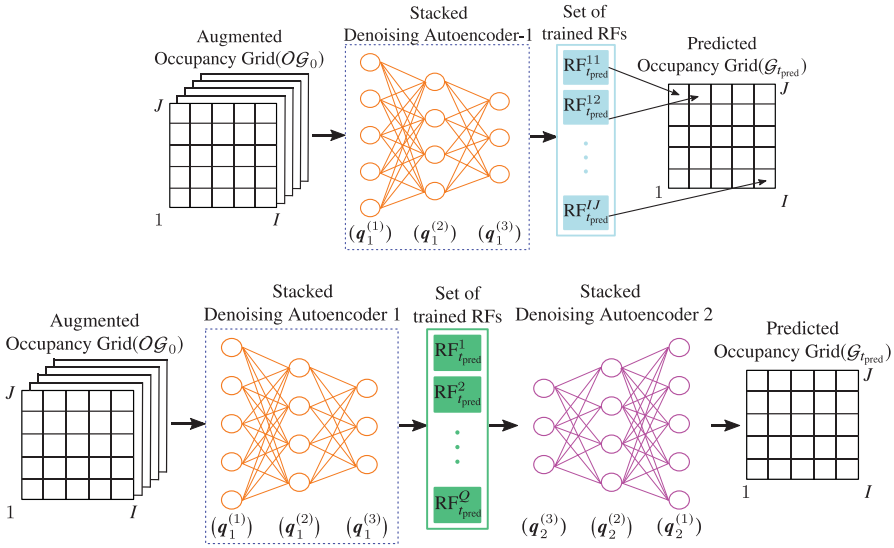

Abbildung 5.83 Maschinelle Lernverfahren für die Prädiktion von Belegungskarten [NBS18].

schen den Verkehrsteilnehmern in den Trainingsdaten enthalten sind, werden diese auch gelernt.

Die Motivation, maschinelle Lernalgorithmen in dieser Anwendung zu nutzen, liegt also in einer recheneffizienten Methode zur Berechnung von prädizierten Belegungskarten und kann der Kategorie II. „Rechenressourcen“ aus Abb. 5.77 zugeordnet werden.

5.7.2 Prädiktion der Crashschwere

Zusätzlich zur Prädiktion, ob eine Kollision stattfinden wird, ist eine zweite Größe für die Fahrzeugsicherheit von zentraler Bedeutung, nämlich die Crashschwere, also das Schadensausmaß aus Gl. (1.1). Der Begriff der Crashschwere wurde in Unterkapitel 3.1.1 bereits erläutert. In [MNB+16] und [MLB+18] wird ein Verfahren vorgestellt, das auch Anteile des maschinellen Lernens enthält, um die **prädiktive Crashschwereschätzung** auch auf Automotive-Mikrocontrollern möglich zu machen. Das Blockschaltbild aus Abb. 5.84 visualisiert die Methode. Einstiegspunkt für den Algorithmus ist die Unvermeidbarkeit einer Kollision. Um die

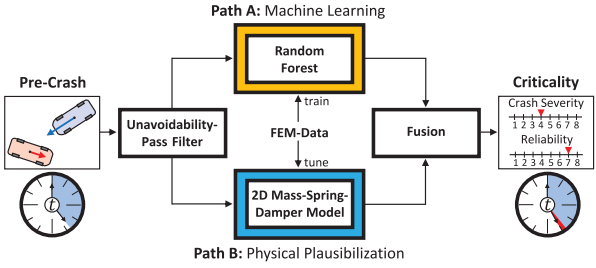

Abbildung 5.84 Architektur der prädiktiven Crashschwereschätzung [MLB+18].

Unvermeidbarkeit zweier Fahrzeuge zu ermitteln, wird ausgehend von deren aktueller Pre-

sifikationsaufgabe in vielen Fällen äußerst schwierig, und der maschinelle Lernalgorithmus kann deswegen für die finale Entscheidungen einen wertvollen Beitrag leisten.

Die Motivation, maschinelle Lernalgorithmen in dieser Anwendung zu nutzen, liegt also in einer alternativen Methode zur Auslösung von Rückhaltesystemen und kann der Kategorie I.„Modellierung nicht möglich bzw. zu komplex“ und der Kategorie III. „Diversität“ aus Abb. 5.77 zugeordnet werden.

5.7.5 Clusterung von Verkehrsszenarien

Ein zentraler Aspekt beim automatisierten Fahren ist die Absicherung, d. h. Fragen wie „Erbringt das System tatsächlich den beabsichtigten Nutzen?“ oder „Zeigt das System im Praxiseinsatz unerwartete Reaktionen?“ müssen beantwortet werden. Eine wesentliche Komponente dabei sind die sogenannten Felderprobungen, bei denen die Fahrzeuge im realen Straßenverkehr getestet werden. Um die Erprobungen gezielter zu steuern, wurde in [KWB18] ein Verfahren vorgestellt, das auf dem in Unterkapitel 5.6.2 vorgestellten Random Forest für unüberwachtes Lernen basiert und die automatisierte Identifikation von Verkehrsszenarien im Mittelpunkt hat. Es handelt sich um ein unüberwachtes maschinelles Lernverfahren, so dass keine händischen Kategorisierungen der Szenarien notwendig sind.

Die Identifikation eines Verkehrsszenarios stellt einen Grundpfeiler für weitergehende Analysen und Bewertungen dar. In dem Verfahren aus [KWB18] werden Szenarien durch einen Satz an Merkmalen beschrieben, der neben den EGO-Fahrzeugdaten das Umfeld und dessen Änderungen in zeitlicher Abfolge erfasst. Abb. 5.92 visualisiert das vorgeschlagene Verfahren zur Clusterung der Szenarien. Der Datensatz wird vom Random Forest für unüberwachtes Lernen

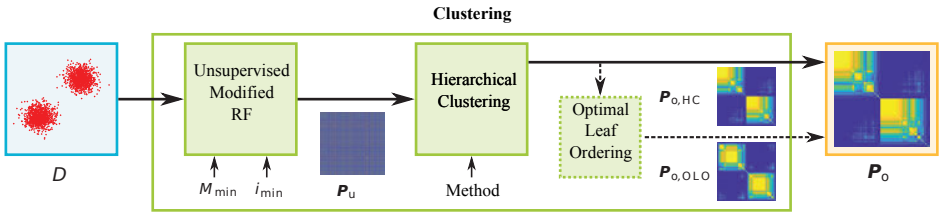

Abbildung 5.92 In [KWB18] genutzte Architektur zum Clustern von Verkehrsszenarien.

genutzt, um ein datenadaptives Ähnlichkeitsmaß zu bestimmen. Dieses Ähnlichkeitsmaß wird anschließend von hierarchischen Cluster-Verfahren, wie die in Unterkapitel 5.6.1 vorgestellten Methoden, genutzt, um Gruppen von ähnlichen Verkehrsszenarien zu bilden. In der resultierenden Proximity Matrix $\boldsymbol{P}_o$ spiegelt jedes Element den Vergleich zweier Verkehrsszenarien wider. Die Elemente der Matrix werden so angeordnet, dass sich ähnliche Punkte als Cluster visualisieren lassen. Liegt diese Strukturierung vor, lassen sich im nächsten Schritt Testfälle ableiten, indem man Repräsentanten je Cluster auswählt. Systeme des automatisierten Fahrens werden vorzugsweise auf repräsentativen Verkehrsszenarien geprüft. In einem weiteren Schritt ist es anschließend möglich, beruhend auf einem angelernten Klassifikationsalgorithmus, während einer Fahrt Szenarien einem der Cluster zuzuweisen. Damit erhält man eine Übersicht über die aufgezeichneten, bzw. auch über neue Verkehrsszenarien, und das ermöglicht quantitative Aussagen über den bereits geleisteten Beitrag für die Absicherung der Systeme des automatisierten Fahrens. Die identifizierten Testfälle lassen sich zusätzlich auch für die Weiterentwicklung dieser Systeme nutzen.

In [KWB18] wird das Cluster-Verfahren detailliert beschrieben. Es bietet die Möglichkeit, nach einer ersten Kategorisierung der Szenarien die Vorgehensweise für ein Cluster zu wiederholen und wie mit einer Lupe weitere Kategorien innerhalb dieses Clusters zu finden. So ergibt sich beispielsweise aus einem Datensatz, bestehend aus zunächst 22519 Verkehrsszenarien, die in Abb. 5.93 dargestellte Proximity Matrix, die nur urbane Verkehrsszenarien beinhaltet. In

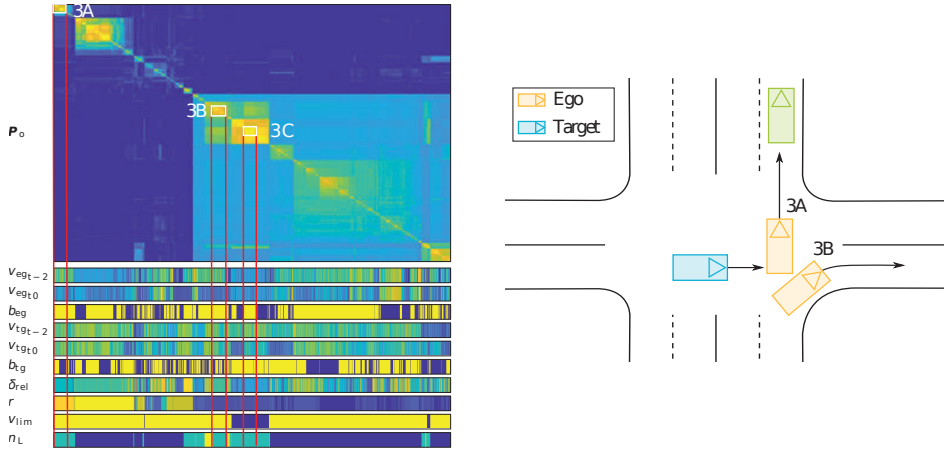

Abbildung 5.93 Proximity Matrix für urbane Verkehrsszenarien und genutzte Merkmale (links); Visualisierung von Kreuzungsszenarien, die zu zwei Clustern führen (rechts) [KWB18].

dieser Proximity Matrix lassen sich Cluster finden, deren Verkehrsszenarien bereits schon viele Gemeinsamkeiten haben. So enthalten die Boxen $3A$ und $3B$ nur Kreuzungsszenarien und die Box $3C$ nur Kreiselszenarien. In den Szenarien der Boxen $3A$ fährt das EGO-Fahrzeug weiter geradeaus, und in den Szenarien der Boxen $3B$ biegt es nach rechts ab. Es ist zu beachten, dass es sich um ein rein unüberwachtes maschinelles Lernverfahren handelt und keine vorherige Kategorisierung stattgefunden hat, was auf eine sehr gute Eignung des Verfahrens für die vorliegende Aufgabe hindeutet. In [KWM+19] wird das Verfahren durch die sogenannte „Path Proximity“ weiterentwickelt und um nachgelagerte Klassifikatoren erweitert, so dass während einer Fahrt Szenarien einem der Cluster zugewiesen bzw. als neue Szenarien identifiziert werden können.

Die Motivation, maschinelle Lernalgorithmen in dieser Anwendung zu nutzen, liegt also in einer automatisierten Methode, um Gruppen von ähnlichen Verkehrsszenarien zu finden, und kann der Kategorie I. „Modellierung nicht möglich bzw. zu komplex“ aus Abb. 5.77 zugeordnet werden.

5.7.6 Generierung von Szenarien mittels Variational Autoencodern

Die in Unterkapitel 5.6.4 eingeführten Variational Autoencoder (VAE) können zur Generierung von Verkehrsszenarien genutzt werden und sind damit für unterschiedliche Anwendungen im Bereich des sicheren automatisierten Fahrens nutzbar.

Übung 5.23

Sie wollen mit der Methode der automatischen Differentiation im Rückwärtsmodus die Ableitungen $\frac{\partial y}{\partial x_1}$ und $\frac{\partial y}{\partial x_2}$ von

$$y = f(x_1, x_2) = \mathrm{e}^{\sin(x_1)+\cos(x_1 x_2)} + \frac{x_2}{\ln(x_1 x_2)}.$$

berechnen.

a) Erstellen Sie dafür einen vorwärtsgerichteten azyklischen Graphen und den dazugehörigen Backpropagation-Graphen.

b) Welche Werte ergeben sich für die Ableitungen $\frac{\partial y}{\partial x_1}$ und $\frac{\partial y}{\partial x_2}$, wenn $x_1 = 1$ und $x_2 = 2$ sind? Berechnen Sie diese Ableitungen mit der automatischen Differentiation im Rückwärtsmodus.

c) Verifizieren Sie das Ergebnis durch symbolische Differentiation.

Übung 5.24

Erweitern Sie den vorwärtsgerichteten azyklischen Graphen aus Abb. 5.27 um eine weitere Verarbeitungseinheit g_7, die aus dem Ausgang $\hat{\boldsymbol{y}}$ die Verlustfunktion $a_7 = z = \mathscr{L}\left(\hat{\boldsymbol{y}}, \boldsymbol{y}\right)$ berechnet. Dabei sollen in der ersten und zweiten verdeckten Schicht alle Aktivierungsfunktionen gleich sein, d. h. $\boldsymbol{\sigma}^{(1)} = \boldsymbol{\sigma}$, $\boldsymbol{\sigma}^{(2)} = \boldsymbol{\sigma}$, und in der nichtlinearen Verarbeitungseinheit des Ausgangs sei $\boldsymbol{\sigma}^{(3)} = \boldsymbol{h}$. Skizzieren Sie den dazugehörigen Backpropagation-Graphen.

Übung 5.25

Zur Lösung einer Klassifikationsaufgabe mit 4 Klassen und einem 2-dimensionalen Eingangsvektor, wie z. B. in Abb. 5.7 und Abb. 5.8 dargestellt ist, soll ein künstliches neuronales Netz mit zwei verdeckten Schichten und einem softmax-Ausgang genutzt werden. Als Verlustfunktion soll die Kreuzentropie verwendet werden. Dabei sollen sowohl die erste als auch die zweite verdeckte Schicht jeweils drei Neuronen haben, die alle als Aktivierungsfunktion die logistische Sigmoidfunktion nutzen. Abb. 5.100 präsentiert zwei Darstellungen dieses künstlichen neuronalen Netzes.

a) Erstellen Sie den zur zweiten Darstellung gehörigen Backpropagation-Graphen.

b) Schreiben Sie mit Hilfe des Backpropagation-Graphen aus der Teilaufgabe a) den zweiten Schritt des Backpropagation-Verfahrens, so wie dies in Gl. (5.169) bis Gl. (5.178) gemacht wurde.

c) Schreiben Sie ein Matlab Skript, das den ersten und zweiten Schritt des Backpropagation-Verfahrens für das Multilayer Perceptron aus Abb. 5.100 umsetzt.

d) Verwenden Sie den Datensatz aus Übung 5.17 und das Matlab Skript aus der Teilaufgabe c), um die Klassifikationsaufgabe aus Übung 5.17 zu lösen. Visualisieren Sie das Ergebnis, so wie in Abb. 5.8.

Übung 5.26

a) Betrachten Sie den Teil c) der Übung 5.8 und zeigen Sie, dass es sich bei der dort gefundenen Lösung um ein RBF-Netzwerk handelt.

b) Verwenden Sie statt der Gaußschen Basisfunktionen die inversen quadratischen radialen Basisfunktionen aus Tab. 5.3 und lösen Sie damit Teil c) der Übung 5.8. Nutzen Sie dabei $\alpha = 4{,}0$. Die Zentren $\boldsymbol{c}_i$ seien Vektoren $\boldsymbol{\mu}_i$ aus Übung 5.8.

Mit $x_1 = 1$ und $x_2 = 2$ erhält man $\frac{\partial y}{\partial x_1} = -6.12$ und $\frac{\partial y}{\partial x_2} = -2.03$. Der Unterschied zur Teilaufgabe b) in der zweiten Nachkommastelle von $\frac{\partial y}{\partial x_1}$ ergibt sich aufgrund der Rundungsfehler, die durch die Beschränkung auf zwei Nachkommastellen in der Teilaufgabe b) entstehen.

Lösung zu Übung 5.24

Die beiden Graphen sind in Abb. 5.135 zu sehen.

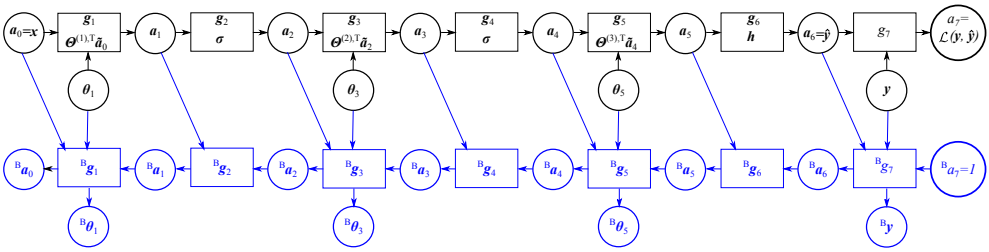

Abbildung 5.135 Vorwärtsgerichteter azyklischer Graph und Backpropagation-Graph (blau) zu Übung 5.24.

Lösung zu Übung 5.25

a) Es handelt sich um die Graphen aus Übung 5.24, die in Abb. 5.135 dargestellt sind.

b) Mit

$$z = \mathscr{L}(\boldsymbol{y}, \hat{\boldsymbol{y}}) = -\sum_{\ell=1}^{4} y[\ell] \ln\left(\hat{y}[\ell]\right) \quad \text{wobei} \quad \hat{\boldsymbol{y}} = \begin{bmatrix} \hat{y}[1] \\ \hat{y}[2] \\ \hat{y}[3] \\ \hat{y}[4] \end{bmatrix} = \begin{bmatrix} a_6[1] \\ a_6[2] \\ a_6[3] \\ a_6[4] \end{bmatrix} = \begin{bmatrix} \hat{\mathrm{p}}(c_1|\boldsymbol{x}) \\ \hat{\mathrm{p}}(c_2|\boldsymbol{x}) \\ \hat{\mathrm{p}}(c_3|\boldsymbol{x}) \\ \hat{\mathrm{p}}(c_4|\boldsymbol{x}) \end{bmatrix}$$

aus Gl. (5.86) lautet der zweite Schritt des Backpropagation-Verfahrens:

$${}^{\mathrm{B}}a_7^{\mathrm{T}} = \frac{\partial z}{\mathrm{vec}\{a_7\}^{\mathrm{T}}} = 1;$$

$${}^{\mathrm{B}}\boldsymbol{a}_6^{\mathrm{T}} = \frac{\partial z}{\partial \mathrm{vec}\{\boldsymbol{a}_6\}^{\mathrm{T}}} = 0 + \underbrace{\frac{\partial z}{\partial \mathrm{vec}\{a_7\}^{\mathrm{T}}}}_{\text{bereits bekannt}} \cdot \frac{\partial \mathrm{vec}\{a_7\}}{\partial \mathrm{vec}\{\boldsymbol{a}_6\}^{\mathrm{T}}}, \quad \text{mit} \quad a_7 = g_7(\boldsymbol{a}_6) = -\sum_{\ell=1}^{4} y[\ell] \ln(a_6[\ell])$$

und daraus ${}^{\mathrm{B}}\boldsymbol{a}_6^{\mathrm{T}} = \left[-y[1]\frac{1}{a_6[1]}, -y[2]\frac{1}{a_6[2]}, -y[3]\frac{1}{a_6[3]}, -y[4]\frac{1}{a_6[4]} \right];$

$${}^{\mathrm{B}}\boldsymbol{a}_5^{\mathrm{T}} = \frac{\partial z}{\partial \mathrm{vec}\{\boldsymbol{a}_5\}^{\mathrm{T}}} = 0 + \underbrace{\frac{\partial z}{\partial \mathrm{vec}\{\boldsymbol{a}_6\}^{\mathrm{T}}}}_{\text{bereits bekannt}} \cdot \frac{\partial \mathrm{vec}\{\boldsymbol{a}_6\}}{\partial \mathrm{vec}\{\boldsymbol{a}_5\}^{\mathrm{T}}},$$

$$\text{mit} \quad \boldsymbol{a}_6 = \boldsymbol{g}_6(\boldsymbol{a}_5) = \left[\frac{\mathrm{e}^{a_5[1]}}{\sum_{k=1}^{4} \mathrm{e}^{a_5[\ell]}}, \frac{\mathrm{e}^{a_5[2]}}{\sum_{k=1}^{4} \mathrm{e}^{a_5[\ell]}}, \frac{\mathrm{e}^{a_5[3]}}{\sum_{k=1}^{4} \mathrm{e}^{a_5[\ell]}}, \frac{\mathrm{e}^{a_5[4]}}{\sum_{k=1}^{4} \mathrm{e}^{a_5[\ell]}} \right]^{\mathrm{T}} = \begin{bmatrix} \hat{p}(c_1|\boldsymbol{x}) \\ \hat{p}(c_2|\boldsymbol{x}) \\ \hat{p}(c_3|\boldsymbol{x}) \\ \hat{p}(c_4|\boldsymbol{x}) \end{bmatrix}$$

und daraus ergibt sich mit Gl. (5.80) bis Gl. (5.82)

$${}^{\mathrm{B}}\boldsymbol{a}_5^{\mathrm{T}} = {}^{\mathrm{B}}\boldsymbol{a}_6^{\mathrm{T}} \begin{bmatrix} \hat{p}(c_1|\boldsymbol{x})(1-\hat{p}(c_1|\boldsymbol{x})) & -\hat{p}(c_1|\boldsymbol{x})\hat{p}(c_2|\boldsymbol{x}) & -\hat{p}(c_1|\boldsymbol{x})\hat{p}(c_3|\boldsymbol{x}) & -\hat{p}(c_1|\boldsymbol{x})\hat{p}(c_4|\boldsymbol{x}) \\ -\hat{p}(c_2|\boldsymbol{x})\hat{p}(c_1|\boldsymbol{x}) & \hat{p}(c_2|\boldsymbol{x})(1-\hat{p}(c_2|\boldsymbol{x})) & -\hat{p}(c_2|\boldsymbol{x})\hat{p}(c_3|\boldsymbol{x}) & -\hat{p}(c_2|\boldsymbol{x})\hat{p}(c_4|\boldsymbol{x}) \\ -\hat{p}(c_3|\boldsymbol{x})\hat{p}(c_1|\boldsymbol{x}) & -\hat{p}(c_3|\boldsymbol{x})\hat{p}(c_2|\boldsymbol{x}) & \hat{p}(c_3|\boldsymbol{x})(1-\hat{p}(c_3|\boldsymbol{x})) & -\hat{p}(c_3|\boldsymbol{x})\hat{p}(c_4|\boldsymbol{x}) \\ -\hat{p}(c_4|\boldsymbol{x})\hat{p}(c_1|\boldsymbol{x}) & -\hat{p}(c_4|\boldsymbol{x})\hat{p}(c_2|\boldsymbol{x}) & -\hat{p}(c_4|\boldsymbol{x})\hat{p}(c_3|\boldsymbol{x}) & \hat{p}(c_4|\boldsymbol{x})(1-\hat{p}(c_4|\boldsymbol{x})) \end{bmatrix};$$

$${}^{\mathrm{B}}\boldsymbol{\theta}_5^{\mathrm{T}} = \frac{\partial z}{\partial \mathrm{vec}\{\boldsymbol{\theta}_5\}^{\mathrm{T}}} = 0 + \underbrace{\frac{\partial z}{\partial \mathrm{vec}\{\boldsymbol{a}_5\}^{\mathrm{T}}}}_{\text{bereits bekannt}} \cdot \frac{\partial \mathrm{vec}\{\boldsymbol{a}_5\}}{\partial \mathrm{vec}\{\boldsymbol{\theta}_5\}^{\mathrm{T}}}, \quad \text{mit } \boldsymbol{a}_5 = \boldsymbol{g}_5(\boldsymbol{a}_4, \boldsymbol{\theta}_5) = \boldsymbol{\Theta}^{(3),\mathrm{T}} \tilde{\boldsymbol{a}}_4; \ \mathrm{vec}\{\boldsymbol{\theta}_5\} = \mathrm{vec}\left\{\boldsymbol{\Theta}^{(3)}\right\}$$

Index