

Universität Duisburg-Essen
Fakultät für Ingenieurwissenschaften
Lehrstuhl für Mechatronik

Längsdynamik eines Plug-in-Hybriden, physikalische Modelle und künstliche neuronale Netze im Vergleich

*Longitudinal Dynamics of a Plug-In Hybrid, Comparing
Physical Models and Artificial Neural Networks*

Masterarbeit
Maschinenbau / Mechatronik

Leonard Voss
Matr.-Nr.: 3016200

Erstprüfer Prof. Dr.-Ing. Dr. h.c. Dieter Schramm
Zweitprüfer Dr.-Ing. Frédéric Etienne Kracht
Betreuer Sebastian Reicherts, M.Sc.

30. Oktober 2020

**UNIVERSITÄT
DUISBURG
ESSEN**

Offen im Denken

Kurzfassung

In dieser Arbeit wird die Anwendung künstlicher neuronaler Netze in der Modellbildung der Fahrzeuglängsdynamik untersucht, mit der Idee ohne aufwendige Versuche oder Wissen über den spezifischen Antriebsstrang und dessen Parameter ein Modell auf Basis einfach zu gewinnender Daten zu erstellen. Als exemplarische Grundlage werden über den CAN-Bus abgerufene Daten fahrzeuginterner Sensoren eines Ford C-Max Energie Plug-In Hybrid verwendet. Als Vergleichsmodell wird ein physikalisches Modell basierend auf den Fahrwiderstandskräften aufgestellt. Als Eingangsgrößen der Modelle werden die Gas- und Bremspedalstellung sowie die aktuelle Geschwindigkeit gewählt. Die Beschleunigung des Fahrzeuges in Längsrichtung bildet den Ausgang, dessen Integration den zum Vergleich herangezogenen Geschwindigkeitsverlauf liefert. Es werden verschiedene künstliche neuronale Netzwerkstrukturen konstruiert und anhand der Daten trainiert.

Die Ergebnisse dieser Arbeit zeigen, dass bereits ein einfaches, dreischichtiges, vorwärtsgerichtetes, künstliches neuronales Netz in der Lage ist, die Fahrzeuglängsdynamik zu modellieren, wenn auch die Genauigkeit leicht hinter dem des physikalischen Referenzmodells zurückbleibt. Dabei ist dieser Ansatz vollständig ohne Wissen über das Fahrzeug umsetzbar. Wird jedoch die Struktur des Netzes dahingehend verändert, dass weitere fahrzeuginterne Größen während des Trainings als Zwischenausgänge verwendet werden, entsteht das Modell mit der höchsten Güte dieser Arbeit. Dieses Einbringen von Wissen in den Prozess des maschinellen Lernens liefert ein Modell mit geringeren Abweichungen im Vergleich zum physikalischen Modell. Ein rekurrentes Netz mit *Long short-term memory*, mit dem Vorteil zeitabhängige Vorgänge abbilden zu können, liefert ebenfalls ein funktionierendes Modell, dessen Abweichungen jedoch trotzdem die der anderen Modelle überschreiten.

Hohe Korrelationswerte der Ausgaben der Modelle untereinander, die größtenteils die Korrelation der Werte zum gemessenen Geschwindigkeitsverlauf überschreiten,

weisen auf eine gemeinsame Fehlerquelle in der Modellierung hin. Insbesondere die fehlenden Eingangsgrößen der Windgeschwindigkeit sowie der Steigung sind als Ursachen für die Abweichungen der Modelle wahrscheinlich, weshalb angenommen werden kann, dass ein Training der Modelle bis nahe an das Optimum der in den Daten vorhandenen Abhängigkeiten möglich ist. Es wird vorgeschlagen die Güte der Daten und dessen Umfang mit weiteren zusätzlichen Sensoren zu verbessern. Vor allem um den Nutzen des zusätzlichen Aufwandes bewerten zu können und so eine abschließende Bewertung der, im Rahmen dieser Arbeit erstellten, Modelle vornehmen zu können.

Inhaltsverzeichnis

Abbildungsverzeichnis	vii
Tabellenverzeichnis	ix
1 Einleitung	1
1.1 Motivation	1
1.2 Zielsetzung	2
1.3 Aufbau der Arbeit	2
2 Grundlagen	5
2.1 Fahrzeuglängsdynamik	5
2.2 Künstliche Neuronale Netze	7
2.2.1 Feed Forward Netze	10
2.2.2 Rekurrente Netze	11
2.2.3 Keras/Tensorflow	11
2.3 Ford C-Max	11
2.3.1 Datalogging	12
2.3.2 Antriebstopologie	12
3 Modellbildung	17
3.1 Physikalisch-datenbasiertes Modell	18
3.2 Genauere physikalische Modellierung	25
3.3 KNN Modellbildung	27
3.3.1 Datengrundlage und Training	27
3.3.2 Tiefes KNN	29
3.3.3 Rekurrentes Netz	30
3.3.4 Strukturiertes Netz	31

3.3.5	AutoML	33
3.3.6	Weitere Ansätze	35
4	Ergebnisse	37
4.1	Physikalisch-datenbasiertes Modell	39
4.2	Tiefes KNN	41
4.3	Rekurrentes Netz	43
4.4	Strukturiertes Netz	44
4.5	Vergleich und Bewertung der Ergebnisse	47
4.6	Datenqualität	48
4.7	Wie viele Daten sind genug?	49
5	Bewertung und Ausblick	55
5.1	Auswertung	55
5.2	Bewertung der Ansätze	61
5.3	Grenzen der KNN	63
5.4	Fazit und Ausblick	66
	Literaturverzeichnis	69

Abbildungsverzeichnis

2.1	Schematische Darstellung eines biologischen Neurons	7
2.2	Schematische Darstellung eines künstlichen Neurons	8
2.3	Schematische Darstellung eines tiefen KNN	9
2.4	Antriebstopologie	13
2.5	Fahrmodi	14
3.1	Grundstruktur der Modelle	18
3.2	Getriebeübersetzung des Elektromotors	19
3.3	Reifenradius	21
3.4	Getriebeübersetzung des Verbrennungsmotors	22
3.5	<i>Full Pedal Map</i>	23
3.6	Lade- und Entladelimit der Traktionsbatterie	25
3.7	Fahrmodiwechsel	26
3.8	Geschwindigkeitsverteilung	28
3.9	Tiefes KNN	30
3.10	LSTM	31
3.11	Strukturiertes KNN	33
3.12	<i>AutoML</i> generiertes KNN	35
4.1	Testfahrten	39
4.2	Physikalisch-datenbasiertes Modell Testfahrt 1	40
4.3	Physikalisch-datenbasiertes Modell Testfahrt 2	40
4.4	Physikalisch-datenbasiertes Modell Testfahrt 3	41
4.5	Tiefes KNN Testfahrt 1	41
4.6	Tiefes KNN Testfahrt 2	42
4.7	Tiefes KNN Testfahrt 3	42
4.8	Rekurrentes KNN Testfahrt 1	43

4.9	Rekurrentes KNN Testfahrt 2	44
4.10	Rekurrentes KNN Testfahrt 3	44
4.11	Strukturiertes KNN Testfahrt 1	45
4.12	Strukturiertes KNN Testfahrt 2	45
4.13	Strukturiertes KNN Testfahrt 3	46
4.14	Simuliertes verbrennungsmotorisches Antriebsmoment	46
4.15	Simuliertes elektrisches Antriebsmoment	47
4.16	Integration der Beschleunigungsdaten	49
4.17	Einfluss der Datenmenge	51
4.18	Kerndichteschätzung	52
4.19	Kullback-Leibler-Divergenz	53
5.1	Korrelationsmatrix Testfahrt 1	56
5.2	Korrelationsmatrix Testfahrt 2	57
5.3	Korrelationsmatrix Testfahrt 3	57
5.4	Hohe Korrelation der Modelle	59
5.5	Hohe Übereinstimmung	60
5.6	Plausibilitätsprüfungen tiefes KNN	64
5.7	Plausibilitätsprüfungen strukturiertes KNN	64
5.8	Plausibilitätsprüfungen rekurrentes KNN	65

Tabellenverzeichnis

2.1	Auswahl an gemessenen Variablen mit Verwendung in dieser Arbeit. .	12
3.1	Koeffizienten der Gaspedal-Karte.	24
3.2	Koeffizienten der Bremspedal-Karte.	24
4.1	Eckdaten der Testfahrten.	39
4.2	RMSE der simulierten Testfahrten und MSE des Testdatensatzes. . .	47
4.3	Einfluss der Datenmenge auf das Training.	50

Einleitung

1.1 Motivation

Die Entwicklung mächtiger Frameworks, wie TensorFlow, PyTorch, Scikit-Learn u.v.w., sowie Leistungssteigerungen moderner Computer vergrößern zunehmend den Gebrauch Künstlicher Neuronaler Netze (KNN). Im Rahmen dieser Entwicklung wird die Lösung immer weiterer Problemstellungen mit Hilfe KNN in Angriff genommen. Ursprünglich als mathematische Abstraktion biologischer Neuronen konzeptioniert, werden KNN als universeller Lösungsansatz erprobt. Insbesondere bei Problemstellungen bei denen eine vollständige mathematische oder physikalische Beschreibung nicht mehr in geeignetem Maße möglich ist erweisen sich KNN als hilfreich.

Die mathematische Beschreibung der vollständigen Dynamik von Fahrzeugen ist nicht in geeignetem Maße möglich. Als Beispiele der nicht trivial zu beschreibenden Effekte der Fahrzeugdynamik seien hier der Luftwiderstand, die Elastokinematik der Radaufhängungen und die Kraftübertragung der Reifen genannt. Präzise Modellierung kann in vielen Bereichen nur über aufwendige Simulationen erfolgen. Diese Komplexität der Modellierung wird häufig mit stark vereinfachten Modellen umgangen. Diese Modelle treffen verschiedenste Annahmen, Vereinfachungen und Linearisierungen und ermöglichen Aussagen in nur begrenzten und vorher definierten Bereichen.

Doch zusätzlich erfordern alle diese Modellierungen hohes Wissen über das zu untersuchende Fahrzeug, welches in aufwendigen Versuchen bspw. in aerodynamischen Untersuchungen im Windkanal ermittelt werden muss.

Der Ansatz der Modellierung der Fahrzeugdynamik mit Hilfe von KNN könnte

die benötigte Menge an spezifischen Informationen reduzieren. Große Mengen leicht zu messender Daten zum Training der KNN ersetzen benötigtes spezifisches Wissen und komplexe Simulationen. An die Stelle aufwendiger Versuche und der Verwendung von Prüfständen würden reale Fahrten zur Datengewinnung treten. Je nach angestrebter Komplexität des Modells könnten bereits serienmäßig verbaute Sensoren der Fahrstabilitäts- und Fahrassistenzsysteme die Datengrundlage des Trainings des KNN liefern.

Das zu modellierende Fahrzeug wird mit der Möglichkeit zur Speicherung der Daten der verbauten Sensoren ausgestattet. Somit sollte die Datenaufzeichnung und Modellierung bei fast jedem auch bereits älterem Fahrzeug möglich sein.

1.2 Zielsetzung

Im Rahmen dieser Masterarbeit wird nun der Einsatz von KNN in der Modellierung der Fahrzeuglängsdynamik untersucht. Insbesondere erfolgt eine Einschätzung gegenüber der klassischen physikalisch-mathematischen Modellbildung.

Die Untersuchungen erfolgen am Beispiel eines Ford C-Max Plug-In Hybriden. Als Versuchsfahrzeug ist es mit Datenlogger-Hardware erweitert, welche die über den CAN Bus gesendeten Sensorwerte abspeichert. Die Datenaufzeichnung kann somit ohne zusätzliche Sensorik und ohne großen Aufwand erfolgen. Die Messdaten der internen preisoptimierten Sensorik des Fahrzeuges werden somit einem neuen Zweck zugeführt. Damit stellt dies nicht den für diese Aufgabenstellung perfekten sensorischen Aufbau da und die Qualität der Daten ist nicht einwandfrei. Mehrere tausend Kilometer dieser Daten aus normalen Fahrten liegen vor. Fragestellungen zur ausreichenden Qualität sowie dem Umfang der Messdaten entscheiden über den Nutzen des KNN.

Im Rahmen dieser Masterarbeit sollen geeignete Modellierungsansätze, sowie Topologien, Kosten- und Optimierungsfunktionen für ein zielführendes Training KNN gefunden werden. Das Ziel ist es, die Umsetzbarkeit der Idee zu untersuchen, eine Einschätzung und Bewertung aller Ergebnisse und Umstände zu treffen und mit einem KNN als Fahrzeuglängsdynamikmodell zu enden.

1.3 Aufbau der Arbeit

Diese Masterarbeit wird in Kapitel 2 mit einer Darlegung der Grundlagen beginnen. Hierbei wird auf die Längsdynamik von Fahrzeugen und insbesondere auf die Fahrwiderstandskräfte eingegangen. Künstliche Neuronale Netze (KNN) werden in ihren Grundzügen erläutert und im nächsten Abschnitt des Kapitels veranschaulicht. Hierbei liegt der Schwerpunkt auf den später in dieser Arbeit verwendeten Strukturen. Im letzten Abschnitt des Kapitel 2 wird das dieser Arbeit als Grundlage dienende Hybrid-Fahrzeug behandelt. Insbesondere wird auf die Datenerfassung sowie den

Antriebsstrang eingegangen. Kapitel 3 enthält die verschiedenen Modellierungen der Längsdynamik. Das physikalisch-datenbasierte Modell macht den Anfang. Daraufhin folgen verschiedene Fahrzeuglängsdynamikmodelle basierend auf unterschiedlichen KNN. Im Kapitel 4 werden die Ergebnisse dargestellt und verglichen. Im letzten Kapitel erfolgt dann eine Auswertung der in dieser Arbeit erstellten Modelle und Ansätze. Zusätzlich dazu werden auch die Grenzen sowie ein Ausblick gegeben.

Grundlagen

In diesem Kapitel erfolgt eine Darstellung der benötigten Grundlagen bevor mit der Erläuterung der im Rahmen dieser Arbeit erfolgten Modellierung der Fahrzeuglängsdynamik begonnen werden kann.

2.1 Fahrzeuglängsdynamik

Im Rahmen dieser Arbeit liegt der Fokus auf einer einfachen Darstellung der Fahrzeuglängsdynamik. Auf eine vollständige Einführung der kompletten Fahrzeugdynamik mit Längs- und Querdynamik sowie Modellen für die Räder und Radaufhängungen bis hin zu Mehrkörpersimulationen kann verzichtet werden [16].

Eine Betrachtung der Längsdynamik losgelöst von der restlichen Dynamik des Fahrzeuges ist sinnvoll und ausreichend. So kann die Längsdynamik auf eine Dimension entlang der Längsrichtung des Fahrzeuges und vereinfacht angenommen auch der Bewegungsrichtung des Fahrzeuges reduziert werden. Somit wird auch das Fahrzeug lediglich auf eine Punktmasse mit einem Freiheitsgrad reduziert. Die üblichste Beschreibung erfolgt mit Hilfe eines Gleichgewichts aller in Längsrichtung des Fahrzeuges wirkenden Kräfte.

Die Summe aller Fahrwiderstandskräfte muss vom Antrieb aufgebracht werden. Folgend werden alle üblichen Komponenten des Fahrwiderstands betrachtet.

Der **Luftwiderstand** oder auch Strömungswiderstand bezeichnet die Kraft, die die Luft dem Fahrzeug entgegensetzt.

$$F_{\text{Luft}} = c_W \cdot A \cdot \rho_{\text{Luft}} \cdot \frac{(v_{\text{Fzg}} + v_{\text{Wind}})^2}{2} \quad (2.1)$$

mit

- F_{Luft} : Luftwiderstandskraft
- c_{W} : Strömungswiderstandskoeffizient
- A : Stirnfläche des Fahrzeuges
- ρ_{Luft} : Dichte der Luft
- v_{Fzg} : Geschwindigkeit des Fahrzeuges
- v_{Wind} : Windgeschwindigkeit entgegen der Fahrtrichtung

Der **Rollwiderstand** beschreibt die Kraft, die durch die Reibung der Reifen auf der Straße auftritt. Diese Kraft lässt sich mit Hilfe eines Rollwiderstandskoeffizienten und der Normalkraft wie folgt vereinfachen.

$$F_{\text{Roll}} = f_{\text{Roll}} \cdot (m_{\text{Fzg}} + m_{\text{Zu}}) \cdot g \cdot \cos(\alpha) \quad (2.2)$$

mit

- F_{Roll} : Rollwiderstandskraft
- f_{Roll} : Rollwiderstandskoeffizient
- m_{Fzg} : Fahrzeugmasse
- m_{Zu} : Zuladung des Fahrzeuges
- g : Erdbeschleunigung
- α : Steigungswinkel

Der **Steigungswiderstand** beschreibt die Komponente der Gewichtskraft des Fahrzeuges, die beim Befahren einer Steigung in Fahrzeuginnenrichtung wirkt.

$$F_{\text{Steig}} = (m_{\text{Fzg}} + m_{\text{Zu}}) \cdot g \cdot \sin(\alpha) \quad (2.3)$$

mit

- F_{Steig} : Steigungswiderstandskraft
- m_{Fzg} : Fahrzeugmasse
- m_{Zu} : Zuladung des Fahrzeuges
- g : Erdbeschleunigung
- α : Steigungswinkel

Der **Beschleunigungswiderstand** ist eine vereinfachte Zusammenfassung der Trägheitskräfte, bei dem zusätzlich zu den translatorischen Trägheitskräften rotatorische Anteile über einen Massenfaktor λ berücksichtigt werden.

$$F_{\text{Besch}} = \lambda \cdot (m_{\text{Fzg}} + m_{\text{Zu}}) \cdot a \quad (2.4)$$

mit

- F_{Besch} : Beschleunigungswiderstandskraft
- λ : Massenfaktor der rotatorischen Widerstände
- m_{Fzg} : Fahrzeugmasse
- m_{Zu} : Zuladung des Fahrzeuges
- a : Beschleunigung des Fahrzeuges

Die Antriebskraft wirkt der Summe aller Widerstandskräfte entgegen.

$$F_{\text{An}} = -F_{\text{Gesamt}} = -\sum F_i = -(F_{\text{Luft}} + F_{\text{Roll}} + F_{\text{Steig}} + F_{\text{Besch}}) \quad (2.5)$$

Diese Vereinfachungen der Längsdynamik auf ein Kräftegleichgewicht der Widerstandskräfte sind üblich. Bei Kombination mit dem Einspur- oder Zweispurmodell kann eine hinreichende Fahrzeugdynamik Beschreibung erfolgen.

2.2 Künstliche Neuronale Netze

Biologische Neuronen, auch Nervenzellen genannt, sind die Basis menschlicher und tierischer Nervensysteme. Die Aufgabe eines Neurons innerhalb eines Neuronalen Netzes ist die Signalleitung und -übertragung. Neuronen nehmen über die Dendriten Erregungen auf, welche gewichtet, abhängig von der Länge im Zellkörper, zusammengeführt werden (siehe Abbildung 2.1). Übersteigt die Erregung am Axonhügel das Schwellenpotential, gibt das Neuron über das Axon das Signal weiter. Die Verbindung zwischen zwei Neuronen wird als Synapse bezeichnet.

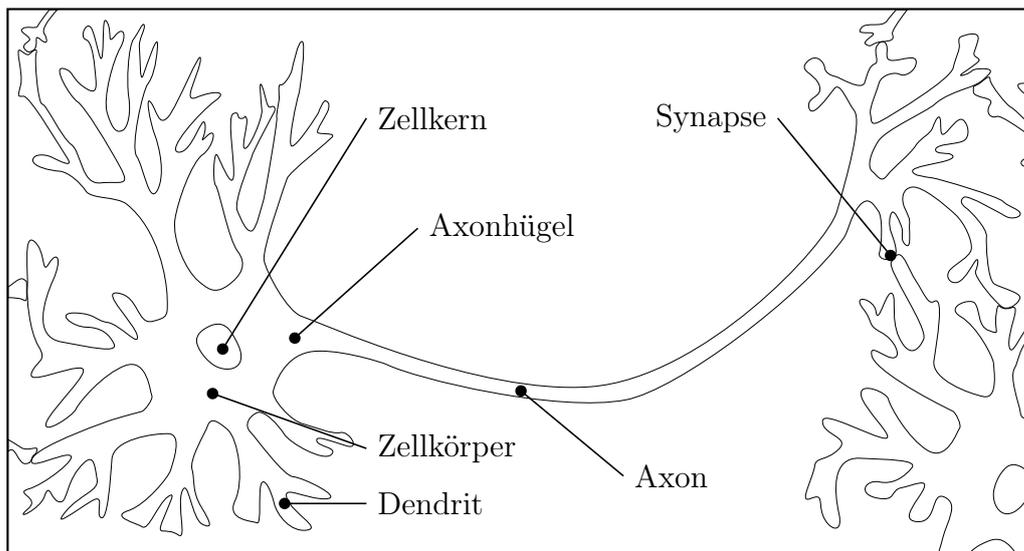


Abbildung 2.1: Schematische Darstellung eines biologischen Neurons (vgl. [4]).

Bei der Signalübertragung erfolgt somit eine Gewichtung abhängig vom Ausbau der Dendriten und der Synapsen. Unterschiedlich starke Verwendung der synaptischen Verbindungen führt zu Veränderungen in der Gewichtung. Häufigere Verwendung der Synapsen führen zu einem Ausbau, ohne Verwendung baut sich die Verbindung zurück. Diese Fähigkeit der Anpassung der Verbindungen zwischen den Neuronen und somit des gesamten Netzes ermöglicht die Fähigkeit zu Lernen.

Künstliche Neuronale Netze (KNN) sind Modelle der Neuroinformatik bestehend aus künstlichen Neuronen [13], welche die Funktionsweise biologischer Nervensysteme nachbilden (Abbildung 2.2). Hierbei erfolgt die Gewichtung, der zu übertragenden Signale, durch Multiplikation der Eingänge mit Gewichtungsfaktoren.

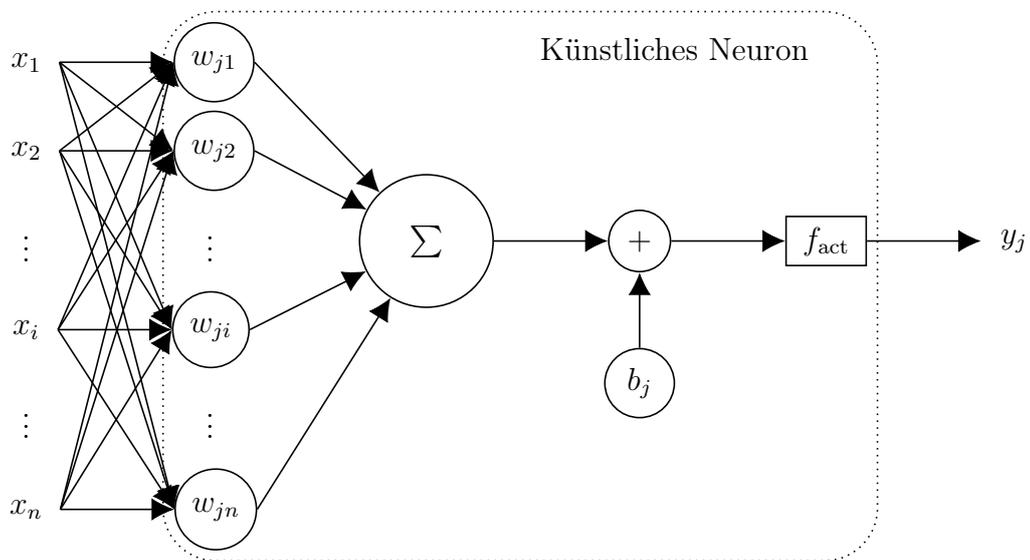


Abbildung 2.2: Schematische Darstellung eines künstlichen Neurons j mit n Eingängen.

Somit ergibt sich eine *propagation function*, die für die eingehenden Signale x_i aller n Neuronen den Ausgang y_j eines Neurons j ermittelt.

$$y_j(t) = f_{\text{act}} \left(\sum_{i=1}^n x_i(t)w_{ji} + b_j \right) \quad (2.6)$$

$y_j \in \mathbb{R}$: Ausgang des Neurons j

$x_i \in \mathbb{R}$: Eingang des Neurons i

$w_{ji} \in \mathbb{R}$: Gewicht für Eingang i des Neurons j

$b_j \in \mathbb{R}$: Bias des Neurons j

$f_{\text{act}} : \mathbb{R} \rightarrow \mathbb{R}$: Aktivierungsfunktion des Neurons j

Zur mathematischen Vereinfachung erfolgt eine Strukturierung des Netzes in Schichten. Somit werden die Eingangssignale zu Vektoren und die Gewichte zu Matrizen. Werden nun mehrere Schichten miteinander verbunden ergeben sich (tiefe) KNN.

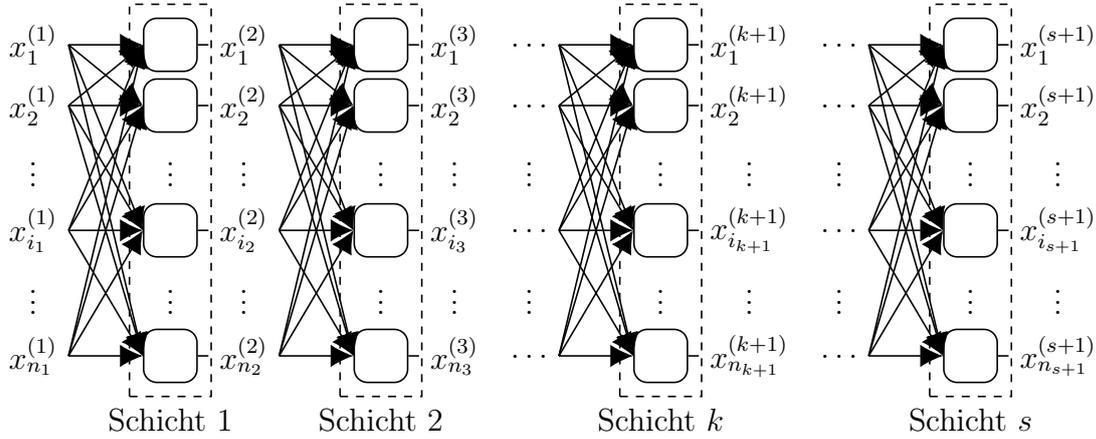


Abbildung 2.3: Schematische Darstellung eines tiefen KNN mit s Schichten.

Die vektorielle Zusammenfassung der Schichten liefert folgende mathematische Beschreibung einer Schicht k .

$$\mathbf{x}^{(k+1)} = \mathbf{f}_{\text{act}}^{(k)} \left(\mathbf{W}^{(k)} \mathbf{x}^{(k)} + \mathbf{b}^{(k)} \right) \quad (2.7)$$

$\mathbf{x}^{(k+1)} \in \mathbb{R}^{n_{k+1}}$: Ausgangsvektor

$\mathbf{x}^{(k)} \in \mathbb{R}^{n_k}$: Eingangsvektor

$\mathbf{W}^{(k)} \in \mathbb{R}^{n_{k+1} \times n_k}$: Gewichtsmatrix

$\mathbf{b}^{(k)} \in \mathbb{R}^{n_{k+1}}$: Biasvektor

$\mathbf{f}_{\text{act}}^{(k)} : \mathbb{R}^{n_{k+1}} \rightarrow \mathbb{R}^{n_{k+1}}$: Aktivierungsfunktion

Wobei $\mathbf{x}^{(1)}$ den Inputvektor der Eingangsschicht und $\mathbf{x}^{(k)}$ den Vektor des Einganges der Schicht $1 \leq k \leq s$ beschreibt. Der Ausgang des KNN mit s Schichten ist der Vektor $\mathbf{y} = \mathbf{x}^{(s+1)}$.

Dies bildet die mathematische Grundstruktur eines KNN. Damit dieses jedoch eine Funktion erfüllen kann, müssen auch geeignete Lernverfahren, auch Trainingsverfahren genannt, aus dem biologischen Vorbild abstrahiert werden. Innerhalb des Trainings eines KNN erfolgt die Auswahl eines geeigneten Satzes an Gewichtsmatrizen. Das gesamte KNN kann als hochdimensionale Funktion mit einer Vielzahl an Parametern $\omega = \left(\mathbf{W}^{(1)}, \dots, \mathbf{W}^{(s)}, \mathbf{b}^{(1)}, \dots, \mathbf{b}^{(s)} \right) \in \Omega$ verstanden werden.

$$F_{\text{KNN}}(\mathbf{x}) = \mathbf{y} \quad (2.8)$$

Beim Training eines KNN erfolgt die Wahl der optimalen Parameter $\omega^* = (\mathbf{W}^{(1)}, \dots, \mathbf{W}^{(s)}, \mathbf{b}^{(1)}, \dots, \mathbf{b}^{(s)})^*$ zur Minimierung einer Kostenfunktion (engl. *Cost-function*) $C(\omega) \rightarrow \mathbb{R}$. Liegen Ein- und Ausgangsdaten bspw. eines realen Systems vor, so kann das KNN anhand diesen Daten trainiert werden das reale System zu imitieren. Eine hierfür übliche Kostenfunktion ist die mittlere quadratische Abweichung (MSE aus dem engl. *mean squared error*). Sie gibt ein Maß der Abweichung des KNN über einen Satz Trainingsdaten an.

$$C(\omega) = \text{MSE} = \frac{1}{2} \sum_{i=1}^N (F_{\text{KNN}}(\mathbf{x}_i) - \mathbf{y}_i)^2 \quad (2.9)$$

Hierbei ist $F_{\text{KNN}}(\mathbf{x}_i)$ das Ergebnis des zu trainierenden KNN. Üblicherweise wird die Kostenfunktion mit Hilfe des Gradientenverfahrens (engl. *gradient descent*) oder einer Abwandlung dessen minimiert. Bei diesem Optimierungsverfahren wird am aktuellen Satz der Parameter ω begonnen und ein Schritt $\Delta\omega$ in die Richtung des negativen Gradienten der Kostenfunktion $\frac{\partial C}{\partial \omega}$, welcher die größte Änderungsrate darstellt, vorgenommen. Hierzu müssen für die Kostenfunktion die partiellen Ableitungen nach jedem Parameter $\omega = (\mathbf{W}^{(1)}, \dots, \mathbf{W}^{(s)}, \mathbf{b}^{(1)}, \dots, \mathbf{b}^{(s)})$ ermittelt werden. Da die Kostenfunktion eine Funktion des Ausgangs der letzten Schicht ist, müssen die Gradienten der vorherigen Schichten mit Hilfe einer Fehlerrückführung (engl. *Backpropagation*) ermittelt werden. Somit ergibt sich ein einzelner Gradient zu einer rückwärts ins KNN gerichteten Kette an partiellen Ableitungen.

Mit diesem Verfahren werden alle Parameter nach einem Durchlauf eines Trainingsdatenpaketes auch *Batch* genannt optimiert. Der Durchlauf aller Trainingsdatenpakete und die darauf folgende Optimierung der Parameter werden als Epoche bezeichnet. Nach einer gewissen Anzahl Epochen wird so ein Minimum der Kostenfunktion gefunden. Vorausgesetzt dass die Trainingsdaten alle benötigten Informationen enthalten, bildet das trainierte KNN nun ein Modell des realen Sachverhaltes der Trainingsdaten. Ein großer Vorteil hierbei ist, dass kein Wissen über den Sachverhalt sondern lediglich die Trainingsdaten vorliegen müssen.

2.2.1 Feed Forward Netze

Feed Forward Netze sind Netze, dessen Signale, wie oben dargestellt, nur vorwärts gerichtet sind. Es gibt eine Eingangsschicht von der aus die Signale das Netz bis hin zur Ausgangsschicht nur in eine Richtung durchlaufen. Weißt ein KNN weitere Schichten zwischen Ein- und Ausgangsschicht auf, so spricht man von einem

deep neural network bzw. vom *deep learning*. Diese Netzwerkarchitekturen sind vergleichsweise einfach, aber bereits in der Lage die meisten Aufgaben zu lösen. Jede Schicht kann also nur Verbindungen zur darauf folgenden Schicht besitzen.

Der Ansatz eines tiefen vorwärts gerichteten KNN wird in Kapitel 3.3.2 zur Modellierung der Fahrzeuglängsdynamik verwendet.

2.2.2 Rekurrente Netze

Rekurrente Netze weisen eine Rückführung in Form von Verbindungen von Neuronen zur selben oder zu vorherigen Schichten auf. Diese Rückführungen erlauben die Abbildung zeitlich dynamischer Vorgänge. Wenn bspw. der Ausgang eines Neurons wieder zum selben Neuron als Eingang zurückgeführt wird, besitzt dieses Neuron einen internen Zustand auch *internal memory*. Rekurrente Netze weisen eine größere Komplexität auf und sind aufwendiger zu optimieren, sind jedoch zur Darstellung zeitliche dynamischer Vorgänge notwendig.

Rekurrente Netze als Ansatz zur Abbildung der zeitlichen Vorgänge der Längsdynamik werden im Kapitel 3.3.3 verfolgt.

2.2.3 Keras/Tensorflow

Im Rahmen dieser Masterarbeit wird auf die neuroinformatische Implementierung des Keras Frameworks zurückgegriffen [3]. Als *deep learning* Bibliothek, welche TensorFlow [1] als Backend verwendet, ermöglicht es eine einfache Implementierung KNN und deren Lernverfahren in der Programmiersprache Python. Das ursprünglich von François Chollet als Teil des ONEIROS (Open-ended Neuro-Electronic Intelligent Robot Operating System) initiierte Projekt ist nun als Keras unter einer Open-Source-Lizenz veröffentlicht. Keras übernimmt die Implementierung der in der Neuroinformatik üblichen Schichtstrukturen, Aktivierungs-, Kostenfunktionen und Optimierungsalgorithmen. Dies reduziert die Programmierarbeit und ermöglicht so eine Anwendung durch Personengruppen außerhalb der Neuroinformatik.

2.3 Ford C-Max

Im Rahmen dieser Arbeit erfolgt die Untersuchung der Ansätze und insbesondere die Modellbildung anhand eines Ford C-Max Energi Plug-In Hybrid als Versuchsfahrzeug. Die Methoden des universellen Ansatzes können jedoch auf vergleichbare Anwendungen und somit andere Fahrzeuge übertragen werden. Ausgestattet mit der dritten Generation der Ford-Hybridtechnologie kann das Fahrzeug als Powersplit-Hybrid sowohl rein elektrisch als auch mit Unterstützung des Verbrennungsmotor fahren. Dieses Fahrzeug wurde im Rahmen anderer Forschungsarbeiten mit Datalogging Hardware ausgestattet. Anstelle von Datengewinnung in speziellen Manövern oder durch Fahrten auf Teststrecken werden die Daten im realen Verkehr gewonnen.

2.3.1 Datalogging

Moderne Fahrzeuge, wie auch der Ford C-Max, haben bereits eine Vielzahl an Sensoren verbaut. Diese Sensoren werden insbesondere für Fahrerassistenzsysteme benötigt. Insbesondere das ESP oder höhere Fahrdynamikstabilitäts-Assistenten basieren auf Sensorik zur fahrzeuginternen Modellierung der Fahrzeugdynamik und -stabilität. Somit ist es ausreichend, zur Datengewinnung lediglich auf bereits im Fahrzeug verbaute Sensoren zurück zugreifen. Ein Einplatinenrechner greift über das DLC-Interface auf den CAN-Bus des Fahrzeug zu und kann so die Sensordaten während der realen Testfahrten abspeichern [14].

Mit Hilfe dieses Aufbaus wurden zum Zeitpunkt der Modellierungen in dieser Arbeit bereit ca. 10000 km aufgenommen. Die Größe der Datenmenge legt eine fundierte Datengrundlage für das Training KNN nahe.

CAN Label	Variable	Beschreibung	Einheit
UDS_APP	p_{ACC}	Gaspedalstellung	%
UDS_BRK_PDL	p_{Brk}	Bremspedalstellung	mm
UDS_ENG_TQ	M_{VM}	Antriebsmoment des VM	Nm
UDS_LF_BRK_PRS	$p_{Brk,LF}$	Bremsdruck vorne links	kPa
UDS_LONG_ACCL	a_{long}	Längsbeschleunigung	$\frac{m}{s^2}$
UDS_MTQ_OUT	M_{EM}	Antriebsmoment des EM	Nm
UDS_M_SPEED	ω_{EM}	Drehzahl des EM	$\frac{rad}{s}$
UDS_RF_BRK_PRS	$p_{Brk,RF}$	Bremsdruck vorne rechts	kPa
UDS_VSS	v_{Fzg}	Fahrzeuggeschwindigkeit	$\frac{m}{s}$
UDS_WHEEL_SPD_LF	ω_{LF}	Raddrehzahl vorne links	$\frac{rad}{s}$
UDS_WHEEL_SPD_RF	ω_{RF}	Raddrehzahl vorne rechts	$\frac{rad}{s}$
UDS_BATT_CHAR	SOC	State of charge	%
UDS_CHARGELIM	$P_{max,in}$	Max. Batterieladeleistung	W
UDS_DISCHARGELIM	$P_{max,out}$	Max. Batterieentladeleistung	W

Tabelle 2.1: Auswahl an gemessenen Variablen mit Verwendung in dieser Arbeit.

2.3.2 Antriebstopologie

Der Antrieb des Ford C-Max Energi Plug-In Hybrid basiert auf der dritten Generation der Ford-Hybridtechnologie [15]. Die Hybride Antriebstechnologie erlaubt einen rein elektrischen Betrieb bis zu einer Geschwindigkeit von ca. $137 \frac{km}{h}$. Bei höheren Geschwindigkeiten oder bei niedrigem Ladestand der Antriebsbatterie erfolgt eine

Unterstützung durch den Verbrennungsmotor. Als Verbrennungsmotor wird ein 2.0l Atkinson 4 Zylinder Motor mit 105 kW verwendet. Der elektrische Antrieb erfolgt mit Hilfe eines Permanentmagnet-Synchronmotor mit einer Leistung von 88 kW. Die Lithium-Ionen Traktionsbatterie hat einen Energiegehalt von 7.6 kWh mit einer maximalen Leistung von 35 kW. Die Gesamtleistung des Antriebsstranges ist mit 140 kW angegeben [6].

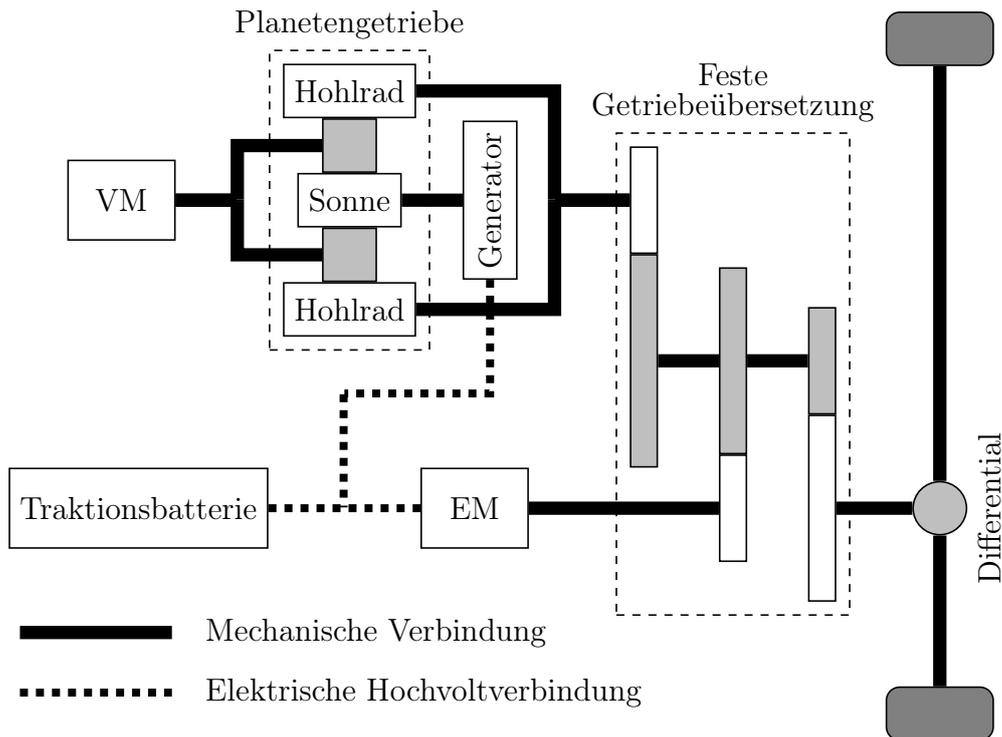


Abbildung 2.4: Powersplit-Architektur des Ford C-Max Energi Plug-In Hybriden (vgl. [15]).

Es handelt sich um einen leistungsverzweigten Hybrid, eine Powersplit-Architektur, d.h. sie setzt sich aus einem Verbrennungsmotor und zwei Elektromotoren zusammen. Der Verbrennungsmotor ist dabei über ein Planetengetriebe direkt mit einem der Elektromotoren verbunden. Dieser Elektromotor wird üblicherweise im generatorischen Betrieb genutzt und wird deshalb auch einfach als Generator bezeichnet. Das Zusammenspiel des Verbrennungsmotors mit dem Generator im Planetengetriebe erlaubt eine kontinuierliche und stufenlose Getriebeübersetzung und wird von Ford deshalb *electronically controlled continuously variable transmission (eCVT)* genannt. Das Planetengetriebe sowie der zweite Elektromotor sind dann mit einer festen Getriebeübersetzung mit dem Achsdifferential der Vorderachse verbunden. Die beiden Elektromotoren hängen zusammen an der Traktionsbatterie [6, 15].

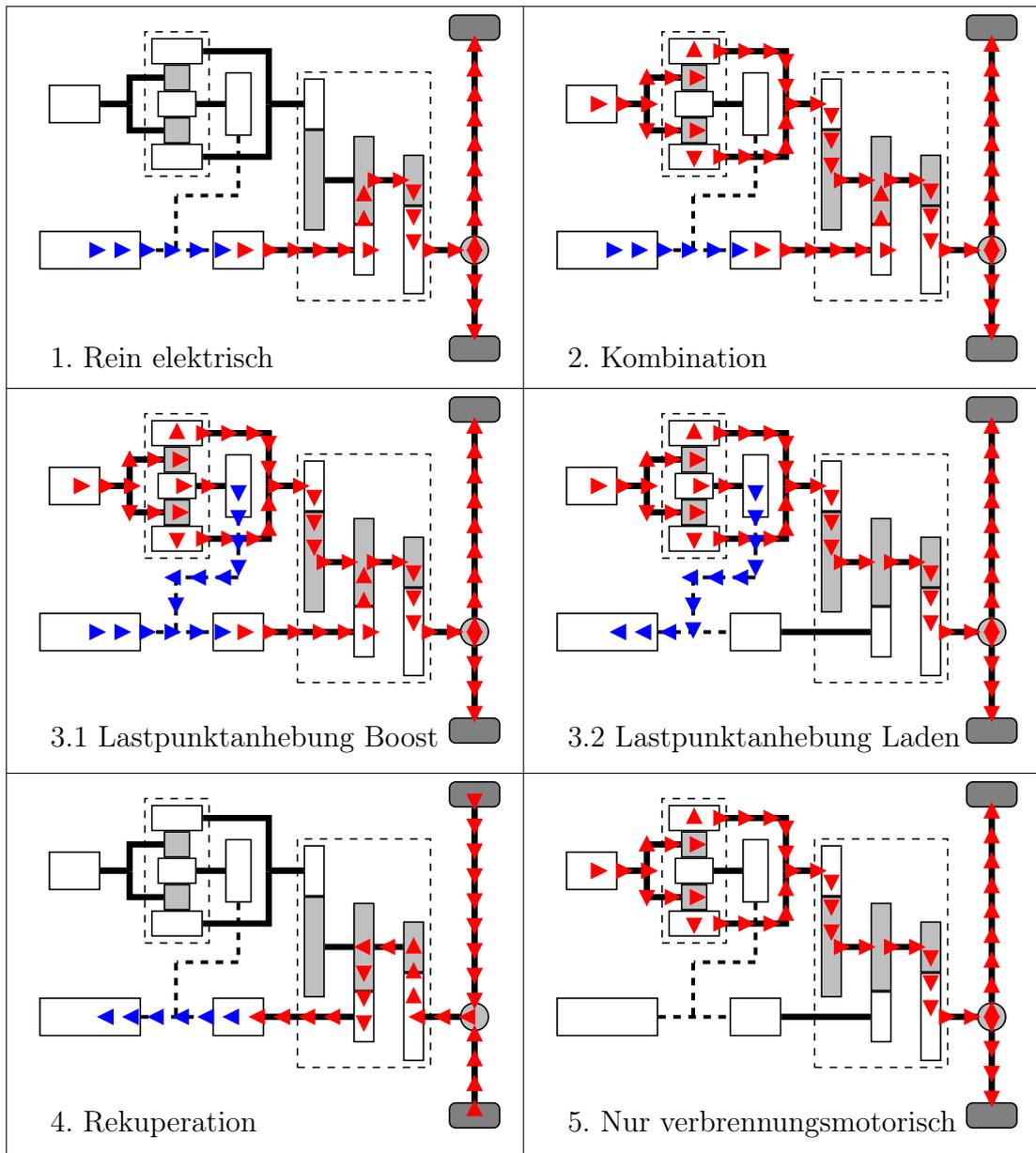


Abbildung 2.5: Fahrmodi der Powersplit-Architektur mit in Rot dargestelltem mechanischen Energiefluss und in Blau dargestelltem elektrischem Energiefluss.

Diese Architektur erlaubt mehrere verschiedene Fahrmodi.

1. Der rein elektrische Antrieb des Fahrzeuges, hierbei versorgt die Batterie den Elektromotor mit der benötigten Energie. Der Verbrennungsmotor kann hierbei ausgeschaltet sein.

2. Der Verbrennungsmotor sowie der Elektromotor liefern Antriebsmoment: Hierbei wird der Elektromotor nur von der Batterie versorgt.
3. Der Verbrennungsmotor liefert Antriebsmoment, der Generator entzieht aber einen Teil der Leistung, die dann zur Verfügung steht. Hierbei wird der Lastpunkt des Verbrennermotors in einen Bereich höherer Effizienz angehoben.
 - 3.1. Die zusätzliche elektrische Energie wird zum Antrieb des Elektromotors verwendet. Sowohl Verbrennungs- als auch Elektromotor liefern Antriebsmoment.
 - 3.2. Die zusätzliche elektrische Energie wird zum Laden der Traktionsbatterie verwendet.
4. Rekuperatives Bremsen. Hierbei befindet sich der Elektromotor im generatorischen Betrieb und nimmt wieder Energie auf und speist diese zurück in die Batterie.
5. Der rein verbrennungsmotorische Betrieb des Fahrzeuges bei niedrigem Ladezustand der Batterie.

Die Wahl der Fahrmodi erfolgt so, dass der Fahrerwunsch regulär erfüllt wird, aber alle Systeme des Fahrzeuges abhängig von allen Parametern den höchsten Wirkungsgrad aufweisen. Dazu wird der Freiheitsgrad der stufenlosen Übersetzung des Planetengetriebes genutzt, um die Betriebspunkte der drei verwendeten Antriebseinheiten anzupassen. Die Regelung des Antriebsstranges verschiebt die Betriebspunkte in den Wirkungsgradkennfeldern und optimiert so den Gesamtwirkungsgrad abhängig vom Fahrerwunsch und Ladezustand der Batterie. Die Regelstrategie optimiert ebenfalls die Rekuperation des Fahrzeuges. Bei schwacher Bremsung wird lediglich mit dem Elektromotor verzögert. Gibt der Fahrerwunsch eine stärkere Bremsung vor, erfolgt eine Kombination mit den mechanischen Bremsen. Somit ergibt sich eine hoch komplexe Regelungsstrategie.

KAPITEL 3

Modellbildung

In diesem Kapitel erfolgt nun die eigentliche Modellbildung als Kernthema dieser Arbeit. Den Anfang macht ein physikalisches Modell. Um diesen Ansatz universell zu halten, wird nur auf einfach verfügbare Fahrzeugparameter zurückgegriffen. Komplexe Motorkennfelder werden nicht verwendet sondern durch einfachen datenbasierten Modellierungen ersetzt. Im darauf folgenden Abschnitt wird die Modellbildung mit Hilfe KNN dargestellt. Zuerst wird die Datenaufbereitung erläutert, danach folgen verschiedene KNN Topologien, die im Rahmen dieser Arbeit zur Modellbildung der Fahrzeuglängsdynamik herangezogen werden.

Die Eingangsgrößen der zu erstellenden Modelle sind mit der Gaspedalstellung p_{ACC} und der Bremspedalstellung p_{Brk} alle für die Fahrzeuglängsdynamik relevanten Eingänge des Fahrers. Auf Grund der Geschwindigkeitsabhängigkeit der Längsdynamik bildet die Fahrzeuggeschwindigkeit v_{Fzg} den dritten Eingang. Die Fahrzeuggeschwindigkeit stellt ebenfalls eine leicht verständliche Möglichkeit für die Ausgangsgröße der Modelle da. Aufgrund der geringen Änderung der Ausgangsgröße bspw. während einer Autobahnfahrt ist diese Größe nicht für das Training von KNN geeignet. Bei Änderungen der Eingangsgrößen sind die Auswirkungen auf die Beschleunigung a_{long} wesentlich direkter. Somit wird die Beschleunigung als Ausgang gewählt. In einer Simulation eines vollständigen Modells liefert eine Integration der Beschleunigung z.B. mit dem expliziten Euler-Verfahren die neue Geschwindigkeit als Eingang des nächsten Zeitschrittes.

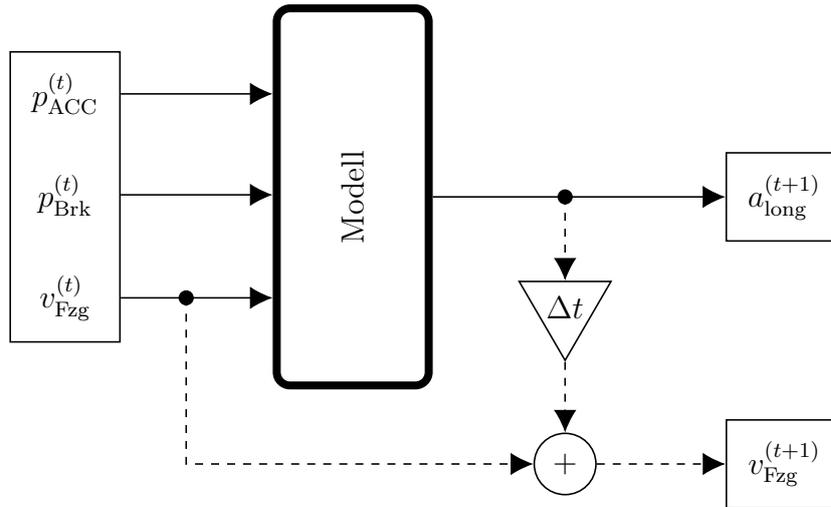


Abbildung 3.1: Schematische Darstellung der Grundstruktur der Modelle.

3.1 Physikalisch-datenbasiertes Modell

Bei der Konstruktion dieses Modells wird die bekannte Antriebstopologie als Grundlage verwendet. Dabei sind die meisten Parameter jedoch unbekannt und müssen aus den aufgenommenen Daten extrahiert werden. Sowohl das Antriebsmoment des Elektromotors M_{EM} als auch des Verbrennungsmotors am Ausgang des Planetengetriebes M_{VM} sind bekannt. Beide Momente sind über eine feste Getriebeübersetzung verbunden.

$$M_{Antrieb} = M_{EM} \cdot i_{EM} + M_{VM} \cdot i_{VM} \quad (3.1)$$

Die Übersetzungsverhältnisse i_{EM} und i_{VM} sind jedoch unbekannt und werden über die Fahrdaten rekonstruiert. Beginnend vom allgemeinen Ansatz für Getriebeübersetzungen wird zuerst die Übersetzung des Elektromotors berechnet.

$$i = \frac{M_A}{M_B} = \frac{n_B}{n_A} = \frac{\omega_B}{\omega_A} \quad (3.2)$$

Da sowohl die Winkelgeschwindigkeiten ω_{LF} und ω_{RF} beider Räder sowie des Elektromotors ω_{EM} unter den gemessenen Daten sind, lässt sich die Übersetzung einfach berechnen. Zur Vereinfachung werden eventuelle Übersetzungsverhältnisse des Achsdifferentials einfach der festen Getriebeübersetzung zugeschlagen. Dementsprechend kann für die Drehzahl vor dem Differential der Durchschnitt der beiden Raddrehzahlen angenommen werden.

$$\omega_{Diff} = \frac{\omega_{LF} + \omega_{RF}}{2} \quad (3.3)$$

Zwischen Elektromotor und Differential ergibt sich folgender Zusammenhang.

$$i_{\text{EM}} = \frac{\omega_{\text{EM}}}{\omega_{\text{Diff}}} = \frac{2 \cdot \omega_{\text{EM}}}{\omega_{\text{LF}} + \omega_{\text{RF}}} \quad (3.4)$$

Die aufgenommenen Fahrdaten sind jedoch mindestens als verrauscht zu bezeichnen, zum Teil sogar fehlerbehaftet. Unter Annahme von gleichverteilten Messfehlern kann der Durchschnitt als Getriebeübersetzung angenommen werden.

$$i_{\text{EM}} = \frac{1}{N} \sum_{n=0}^N \frac{\omega_{\text{EM},n}}{\omega_{\text{Diff},n}} = 10.3914 \quad (3.5)$$

N : Anzahl der Messwerte

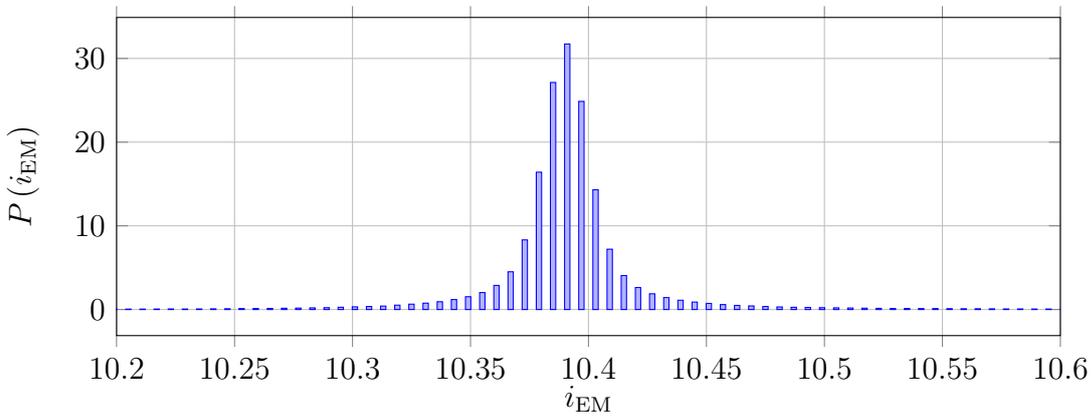


Abbildung 3.2: Verteilung der Getriebeübersetzung des Elektromotors.

Da die Hohlradzahl des Planetengetriebes, an dem das gemessene Antriebsmoment des Verbrennungsmotors anliegt unbekannt ist, kann die Berechnung der Übersetzung i_{VM} nicht analog erfolgen. Stattdessen wird die Übersetzung aus dem benötigten Antriebsmoment über die Fahrwiderstandskräfte ermittelt. Wie in Kapitel 2 beschrieben gilt für die Fahrwiderstandskräfte folgende nachfolgende Summe.

$$F_{\text{Gesamt}} = \sum F_i = F_{\text{Luft}} + F_{\text{Roll}} + F_{\text{Steig}} + F_{\text{Besch}} \quad (3.6)$$

Aus der einfachen und daher jedoch begrenzten Datenerfassung des Versuchsfahrzeuges ergeben sich Beschränkungen. Diese führen zu Vereinfachungen bei einzelnen Fahrwiderstandskräften. Für den Luftwiderstand kann die vorherrschende Windgeschwindigkeit am Fahrzeug nicht trivial gemessen werden. Somit entfällt die Windgeschwindigkeit aus der Berechnung. Der Betrag des hieraus resultierenden Fehlers kann jedoch insbesondere bei hohen Geschwindigkeit groß sein. Es kann jedoch

angenommen werden, dass die, über einen viele Monate langen Zeitraum aufgenommene, große Anzahl an Datensätzen eine normal Verteilung der Windrichtung und Geschwindigkeit nahelegt. Somit kann auch bei der aus den Daten berechneten Luftwiderstandskraft mit Vernachlässigung des Windes ein normalverteilter Fehler angenommen werden. Für die vereinfachte Widerstandskraft gilt folgender Zusammenhang.

$$F_{\text{Luft}} = c_W \cdot A \cdot \rho_{\text{Luft}} \cdot \frac{v_{\text{Fzg}}^2}{2} \quad (3.7)$$

Strömungswiderstandskoeffizient c_W sowie die Stirnfläche des Fahrzeuges lassen sich aus öffentlichen Datenbanken entnehmen [5]. Für die Dichte der Luft wird ebenfalls ein konstanter Wert für alle Fahrten angenommen.

$$\begin{aligned} c_W &= 0.32 \\ A &= 2.57\text{m}^2 \\ \rho_{\text{Luft}} &= 1.2041 \frac{\text{kg}}{\text{m}^3} \end{aligned}$$

Die getroffenen Vereinfachungen führen zur Annahme eines konstanten Wertes für den Rollwiderstandskoeffizienten f_{Roll} in der Rollwiderstandskraft. Eine spezifische Berechnung der Normalkraft ist ebenfalls nicht möglich. Da eine Erfassung des Steigungswinkel α nicht erfolgt, wird für das Modell eine Ebene angenommen. Der daraus resultierende Fehler ist gemäß der Kleinwinkelnäherung $\cos(\alpha) \approx 1 + O(\alpha^2)$ für $\alpha \ll 1$ als gering anzunehmen. Ebenfalls nicht erfasst ist die Zuladung und letztendlich die Fahrzeuggesamtmasse. Hier wird die angegebene Masse des Fahrzeugherstellers sowie eine konstante Zuladung angenommen. Die Vereinfachungen liefern eine konstante Rollwiderstandskraft.

$$F_{\text{Roll}} = f_{\text{Roll}} \cdot (m_{\text{Fzg}} + m_{\text{Zu}}) \cdot g \quad (3.8)$$

$$\begin{aligned} f_{\text{Roll}} &= 0.013 \\ m_{\text{Fzg}} &= 1654 \text{ kg} \\ m_{\text{Zu}} &= 100 \text{ kg} \end{aligned}$$

Auf Grund der fehlenden Messung des Steigungswinkels α entfällt die Berechnung einer Steigungswiderstandskraft vollständig. Die geringen Steigungen der Straßen der Region, in der die Daten aufgenommen werden lässt jedoch einen ebenfalls kleinen Fehler aus dieser Vernachlässigung vermuten. Außerdem kann von Normalverteilung ausgegangen werden.

$$F_{\text{Steig}} = 0 \quad (3.9)$$

Für den Beschleunigungswiderstand wird ein Massenfaktor der rotatorischen Widerstände λ angenommen.

$$F_{\text{Besch}} = \lambda \cdot (m_{\text{Fzg}} + m_{\text{Zu}}) \cdot a \quad (3.10)$$

$$\lambda = 1.1$$

Folgender Zusammenhang ergibt sich für das Widerstandsmoment an den Rädern.

$$M_{\text{Gesamt}} = F_{\text{Gesamt}} \cdot r_{\text{Rad}} = (F_{\text{Luft}} + F_{\text{Roll}} + F_{\text{Besch}}) \cdot r_{\text{Rad}} \quad (3.11)$$

Der Radius des Reifens r_{Rad} lässt sich aus den gemessenen Winkelgeschwindigkeiten $\omega_{\text{LF}}, \omega_{\text{RF}}$ der Räder sowie der Fahrzeuggeschwindigkeit v_{Fzg} bestimmen. Von den beiden gemessenen Raddrehzahlen wird der Durchschnitt verwendet.

$$r_{\text{Rad}} = \frac{v_{\text{Fzg}}}{\frac{\omega_{\text{LF}} + \omega_{\text{RF}}}{2}} \quad (3.12)$$

Für niedrige Geschwindigkeiten ist mit höheren Beschleunigungen und mehr Schlupf sowie stärkerer Verformung des Reifens aufgrund des Fahrzeuggewichtes zu rechnen. Deshalb werden zur Bestimmung des Reifenradius r_{Rad} nur die höheren Geschwindigkeiten betrachtet. Das Auftragen der aus den Messdaten berechneten Radradien über die Fahrzeuggeschwindigkeit bestätigt durch wesentlich geringere Verteilungen die Vorgehensweise. Somit lässt sich auch auf den Reifenradius eine Gaußsche Glockenkurve anpassen.

$$f(x|\mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \cdot e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad (3.13)$$

$$\mu = 0.3221 \text{ m} = r_{\text{Rad}}$$

$$\sigma = 7.007 \times 10^{-7}$$

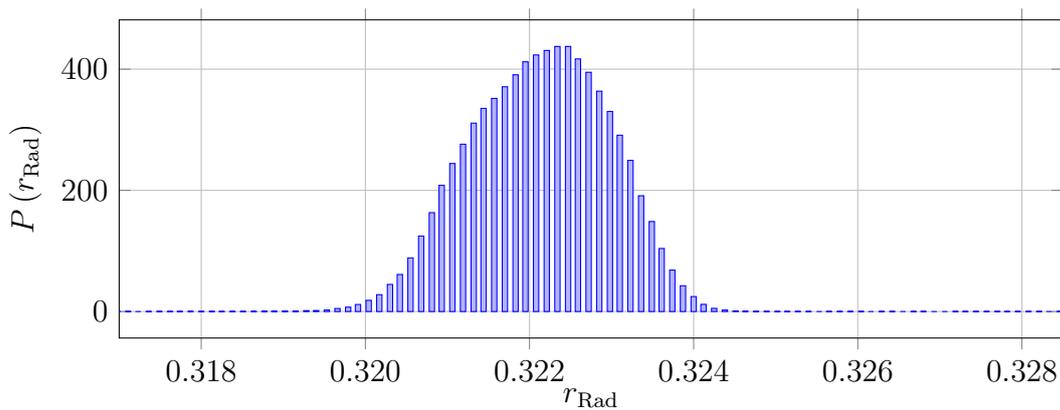


Abbildung 3.3: Normalverteilung des Reifenradius für hohe Geschwindigkeiten.

Nun lässt sich aus den gemessenen Daten für alle Fahrten das Widerstandsmoment an den Räder bestimmen. Da dieses, solange nicht mechanisch gebremst wird, dem Antriebsmoment entsprechen muss, lässt sich auch die Getriebeübersetzung des Verbrennungsmotors bestimmen.

$$M_{\text{Gesamt}} = (F_{\text{Luft}} + F_{\text{Roll}} + F_{\text{Besch}}) \cdot r_{\text{Rad}} = M_{\text{EM}} \cdot i_{\text{EM}} + M_{\text{VM}} \cdot i_{\text{VM}} - M_{\text{B,mech}} \quad (3.14)$$

bei $M_{\text{Brk}} = 0 \text{ Nm}$

Somit ergibt sich für die gesuchte Getriebeübersetzung folgender Zusammenhang.

$$i_{\text{VM}} = \frac{(F_{\text{Luft}} + F_{\text{Roll}} + F_{\text{Besch}}) \cdot r_{\text{Rad}} - M_{\text{EM}} \cdot i_{\text{EM}}}{M_{\text{VM}}} \quad (3.15)$$

Auch hier liefert die Normalverteilung eine Näherung der Getriebeübersetzung mit $\mu = 3.07649 = i_{\text{VM}}$ bei einer Standardabweichung von $\sigma = 0.775$.

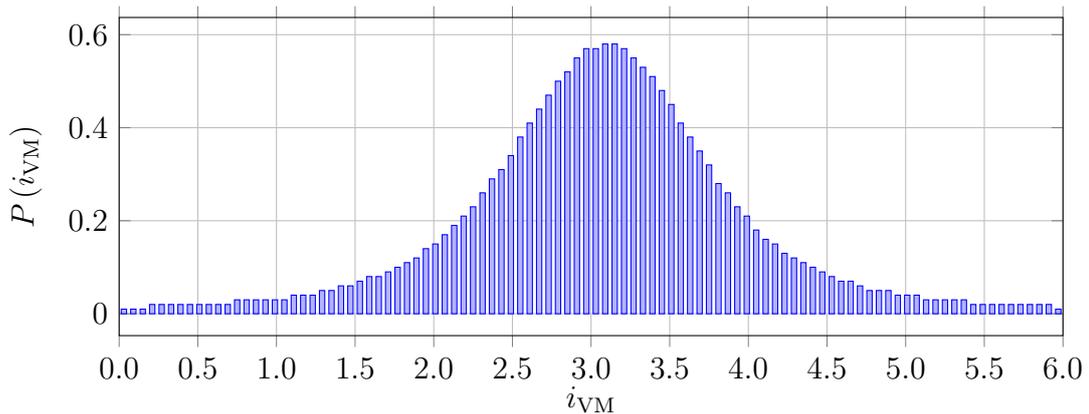


Abbildung 3.4: Normalverteilung der Getriebeübersetzung des Verbrennungsmotors.

Somit ist das Antriebsmoment des hybriden Antriebsstranges aus den gemessenen Daten bestimmt. Außerdem ermöglicht eine umgestellte Momentengleichung die Beschleunigung des Fahrzeuges zu berechnen. Hierbei sind die Antriebs- und Bremsmomente von den Pedalstellungen abhängig.

$$a = \frac{(M_{\text{EM}}(p_{\text{ACC}}, p_{\text{Brk}}) \cdot i_{\text{EM}} + M_{\text{VM}}(p_{\text{ACC}}) \cdot i_{\text{VM}} - M_{\text{Brk}}(p_{\text{Brk}})) \cdot r_{\text{Rad}}^{-1} - F_{\text{Luft}} - F_{\text{Roll}}}{\lambda \cdot (m_{\text{Fzg}} + m_{\text{Zu}})} \quad (3.16)$$

Um jedoch von den Pedalen als Eingängen auf die Antriebsmomente zu kommen, wird eine *Throttle map* bzw. Pedal-Karte benötigt. Diese stellt das Antriebsmoment

in Abhängigkeit der Pedalstellung und der Geschwindigkeit bzw. Drehzahl da. Ein solches Kennfeld könnte auf einem Rollenprüfstand bestimmt werden. Alternativ wird hier der datenbasierte Ansatz gewählt und eine vollständige Pedalkarte aus den Daten generiert. Da es keinen Unterschied im elektrischen und verbrennungsmotorischen Antriebsmoment bei gleicher Pedalstellung gibt, wird das Antriebsmoment beider Motoren an den Rädern zusammengefasst. Zusätzlich wird das Kennfeld um das Bremspedal in negativer Gaspedal Richtung erweitert. So ergeben sich zwei Flächen, die das komplette Antriebsverhalten inklusive der Hybrid- und Rekuperationsstrategie des Antriebsstranges abbilden.

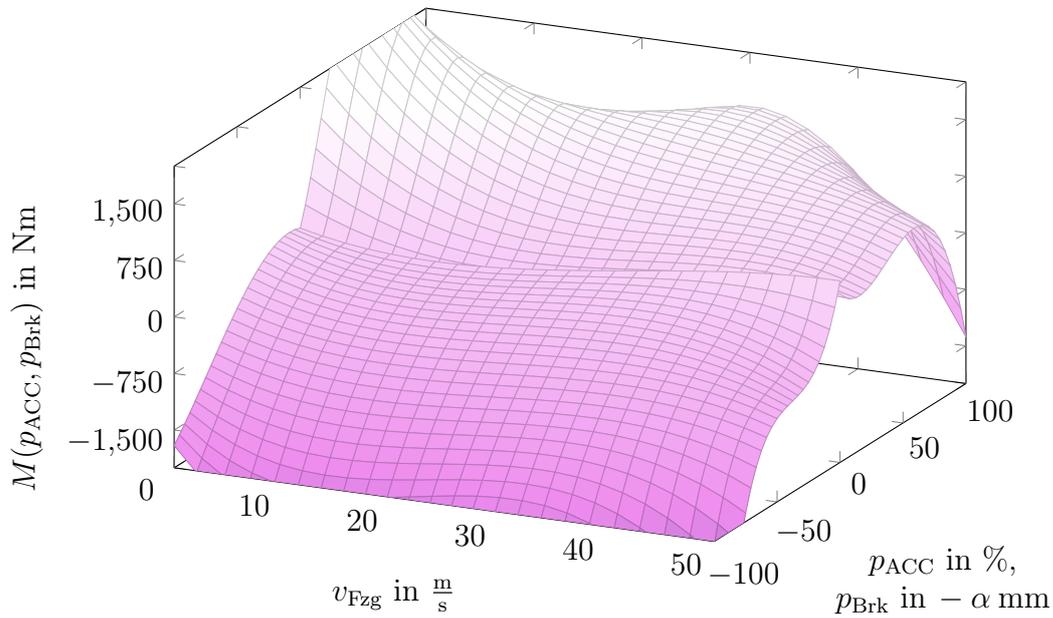


Abbildung 3.5: *Throttle map* des Antriebsmoment erweitert mit dem Bremspedal und -moment.

Sowohl die Fläche, die die Geschwindigkeit und die Gaspedalstellung auf das Antriebsmoment abbildet, als auch die Fläche des Bremsmomentes werden als Polynome 3. Grades approximiert. Es erfolgt somit eine Vereinfachung der einzelnen physikalischen Momente an den Rädern auf zwei approximierten Momente.

$$\left. \begin{array}{l} M_{EM}(p_{ACC}, p_{Brk}, v_{Fzg}) \in \mathbb{R} \\ M_{VM}(p_{ACC}, v_{Fzg}) \in \mathbb{R}^{\geq 0} \\ M_{Brk}(p_{Brk}, v_{Fzg}) \in \mathbb{R}^{\leq 0} \end{array} \right\} \implies \begin{array}{l} M_{An}(p_{ACC}, v_{Fzg}) \in \mathbb{R}^{\geq 0} \\ M'_{Brk}(p_{Brk}, v_{Fzg}) \in \mathbb{R}^{\leq 0} \end{array} \quad (3.17)$$

Die mathematische Beschreibung erfolgt mit Hilfe von Polynomen 3. Grades.

$$P(x, y) = \sum_{i=0}^3 \sum_{j=0}^3 a_{ij} \cdot x^i \cdot y^j \quad (3.18)$$

Für die Koeffizienten des angepassten Polynom dritten Grades der Gaspedal-Karte ergeben sich folgende Werte.

a_{ij}	x^0	x^1	x^2	x^3
y^0	1.202×10^2	-4.079×10^1	1.670×10^0	-1.616×10^{-2}
y^1	6.013×10^1	-6.038×10^0	2.416×10^{-1}	-3.118×10^{-3}
y^2	1.555×10^0	-1.502×10^{-2}	-2.631×10^{-3}	5.543×10^{-5}
y^3	-3.043×10^{-2}	1.529×10^{-3}	-1.522×10^{-5}	-1.471×10^{-7}

Tabelle 3.1: Koeffizienten der Gaspedal-Karte.

Analog dazu ergeben sich die Koeffizienten der Bremspedal-Karte.

b_{ij}	x^0	x^1	x^2	x^3
y^0	1.316×10^2	-3.493×10^1	1.750×10^0	-1.881×10^{-2}
y^1	-7.568×10^0	2.499×10^0	-1.376×10^{-1}	2.473×10^{-3}
y^2	-4.773×10^{-1}	7.708×10^{-2}	-4.421×10^{-3}	7.742×10^{-5}
y^3	-2.179×10^{-3}	6.583×10^{-4}	-3.844×10^{-5}	6.626×10^{-7}

Tabelle 3.2: Koeffizienten der Bremspedal-Karte.

Damit ist die Herleitung des physikalisch-datenbasierten Fahrzeuglängsdynamik Modells abgeschlossen.

$$a(p_{\text{ACC}}, p_{\text{Brk}}, v_{\text{Fzg}}) = \frac{(M_{\text{An}}(p_{\text{ACC}}, v_{\text{Fzg}}) + M_{\text{Brk}}(p_{\text{Brk}}, v_{\text{Fzg}})) \cdot r_{\text{Rad}}^{-1} - F_{\text{Luft}}(v_{\text{Fzg}}) - F_{\text{Roll}}}{\lambda \cdot (m_{\text{Fzg}} + m_{\text{Zu}})} \quad (3.19)$$

$$M_{\text{An}}(p_{\text{ACC}}, v_{\text{Fzg}}) = \sum_{i=0}^3 \sum_{j=0}^3 a_{ij} \cdot v_{\text{Fzg}}^i \cdot p_{\text{ACC}}^j$$

$$M_{\text{Brk}}(p_{\text{Brk}}, v_{\text{Fzg}}) = \sum_{i=0}^3 \sum_{j=0}^3 b_{ij} \cdot v_{\text{Fzg}}^i \cdot p_{\text{Brk}}^j$$

3.2 Genauere physikalische Modellierung

Es sind auch noch kompliziertere physikalische Modellansätze und insbesondere auch präzisere Parametrierung denkbar. Diese würden jedoch vom Ansatz, nur Daten aus regulären Fahrten als Datengrundlage zu verwenden, abweichen. Aufwendige Versuche könnten genauere Fahrzeugparameter liefern. Beispielsweise könnte im Windtunnel ein besserer Wert für den Luftwiderstandsbeiwert bestimmt werden, oder auf einem Rollenprüfstand eine präzise *throttle map* erstellt werden. Doch auch wenn alle Fahrzeugparameter eindeutig bestimmt sind, kommt die physikalische Modellbildung an ihre Grenzen, wenn es um die Abbildung der fahrzeuginternen Antriebssoftware geht. Das in dieser Arbeit herangezogene Hybridfahrzeug verwendet beispielsweise eine komplizierte Strategie zur Kombination des Elektro- und Verbrennungsmotors. Die Frage, ob das Fahrzeug nur elektrisch fährt oder auch noch der Verbrennungsmotor zum Antrieb benötigt wird, ist von vielen Parametern abhängig. So gibt es eine maximale Entlade- und Laderate der Traktionsbatterie (Abbildung 3.6), diese wiederum ist unter anderem vom State-of-charge abhängig, sowie vermutlich der Batterietemperatur. Bei höherem Ladestand wird bis zur maximalen Entladerate rein elektrisch gefahren. Ist der Ladezustand zu niedrig wird der Verbrennungsmotor nicht nur zum Fahren sondern auch noch über den Generator zum Laden der Batterie verwendet. Hinzu kommen noch dynamische Effekte wie eine verzögerte Bereitstellung des Antriebsmoments beim Hinzuschalten des Verbrennungsmotors. Muss der Verbrennungsmotor erst auf eine andere Drehzahl gebracht oder sogar erst gestartet werden um Antriebsmoment liefern zu können, kann es zu erheblichen Verzögerungen kommen. Ist der Verbrennungsmotor bereits gestartet, gibt es keine Verzögerung.

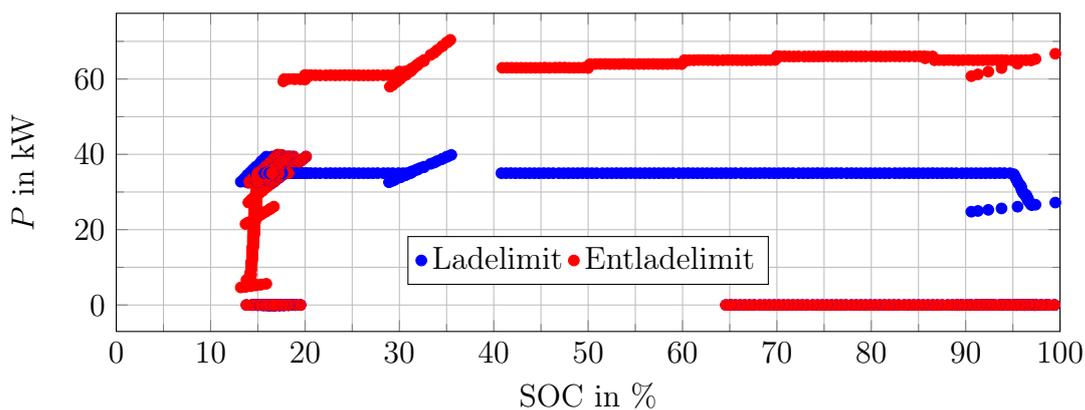


Abbildung 3.6: Lade- und Entladelimit der Traktionsbatterie.

In Abbildung 3.7 sei beispielhaft ein Ausschnitt der Messdaten gezeigt, der Aufschluss auf die vom Fahrzeug gewählte Hybridstrategie gibt. Dargestellt sind das

vom Fahrer gewünschte Antriebsmoment an den Rädern, basierend auf den Ergebnissen des physikalisch-datenbasierten Modells sowie das maximale elektrische Antriebs- und Bremsmoment basierend auf den Lade- und Entladelimits der Traktionsbatterie. Zusätzlich wird das tatsächliche gemessene hybride Antriebsmoment an den Rädern abgebildet. Bereiche in denen verbrennungsmotorisches Antriebsmoment anliegt sind blau eingefärbt und Bereiche in denen die mechanischen Bremsen zusätzliches Bremsmoment bereitstellen sind rot eingefärbt. Zu Beginn des dargestellten Abschnittes ist vom Fahrer mehr Antriebsmoment gefordert als rein elektrisch von der Batterie bereitstellbar. Aus diesem Grund muss zusätzlich der Verbrennungsmotor Antriebsmoment bereitstellen. Sinkt das benötigte Moment, wird der Verbrennungsmotor, wenn auch zeitlich verzögert, abgeschaltet. Vom Zeitpunkt $t = 218$ s an wird gebremst. Hierbei ist zu sehen, wie das Bremsmoment sich aus der für die Batterie maximal möglichen Rekuperation sowie einem mechanischem Anteil zusammensetzt. Um den Zeitpunkt $t = 226$ s ist der rein elektrische Antriebsmodus zu beobachten, hierbei liegt das gewünschte Antriebsmoment unter dem maximal von der Batterie bereitstellbaren Moment. Wenn danach mehr Antriebsmoment benötigt wird, wird zurück in den kombinierten Antriebsmodus geschaltet. Auch hier ist wieder eine kleine Verzögerung beobachtbar, bis der Verbrennungsmotor über das Planetengetriebe das Fahrzeug antreiben kann.

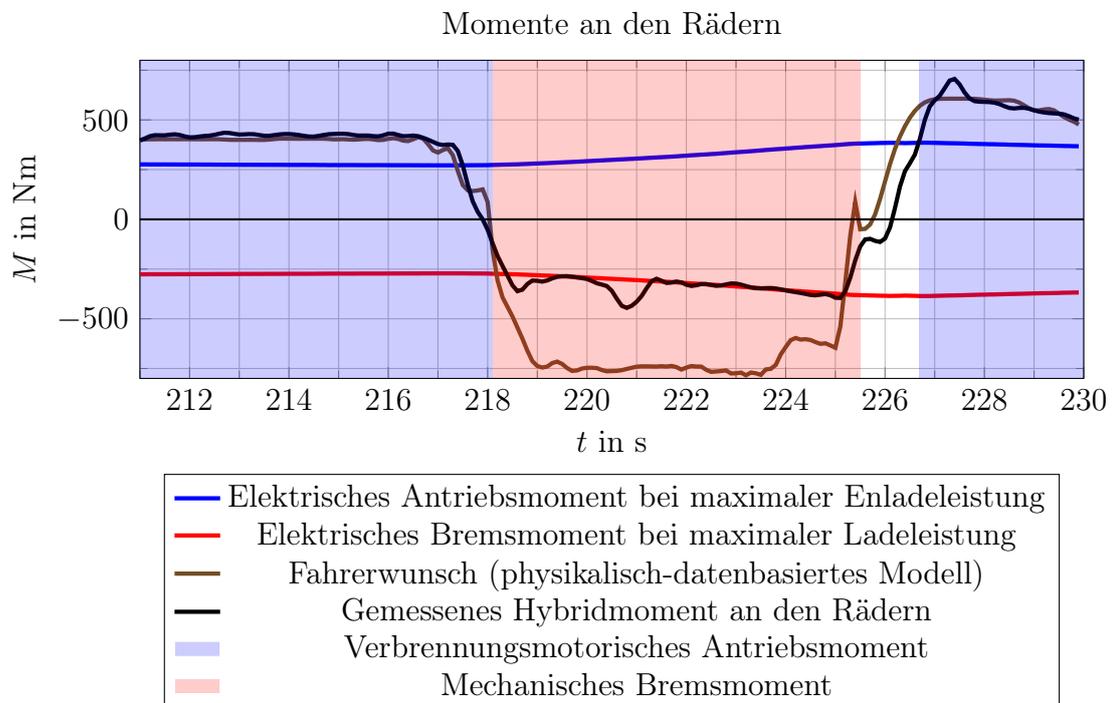


Abbildung 3.7: Fahrmodiwechsel in Testfahrt 2.

Somit sind für ein präziseres Modell die vollständigen Simulationen der drei Motoren nötig. Ebenso muss die Batterie mit ihrem Ladezustand und den elektrischen Strömen im Fahrzeug simuliert werden. Nebenverbraucher, Batterietemperatur, Batteriespannung, Dynamik der Batterie bei Entnahme hoher Ströme, Dynamik des Verbrennungsmotors, die Hybridstrategie und vieles mehr sind für ein präziseres physikalisches Modell nötig. Die begrenzte Datengrundlage ermöglicht diese volle Modellbildung und Simulation jedoch nicht.

Ebenso sind auch noch höhere physikalische Modelle denkbar. Mit Mehrkörpersimulation und Aerodynamiksimulation seien an dieser Stelle zwei Beispiele genannt. Die Anwendung höherer Modelle hätte jedoch wesentlich mehr Daten zum spezifischen Fahrzeug benötigt und würde somit nicht dem Ansatz dieser Arbeit gerecht werden. Außerdem benötigen solche Modelle wesentlich mehr Rechenleistung.

3.3 KNN Modellbildung

Aus den oben genannten Gründen wird auf kompliziertere physikalische Modellbildung verzichtet. Stattdessen werden KNN als Ansätze, die kein Detailwissen über das Fahrzeug benötigen, vorgeschlagen. Die Idee hinter diesem Ansatz ist das Erlernen komplizierterer Vorgänge im Fahrzeug, die ein einfaches physikalisches Modell nicht abbilden kann. Ohne Wissen über das Fahrzeug, das vom Hersteller und aus aufwendigen Fahrversuchen stammen kann, soll bereits ein komplizierteres und präziseres Modell mit Hilfe von maschinellen Lernverfahren erstellt werden.

Für die hier gezeigten Modellbildungen mit Hilfe von KNN wird kein spezifisches Wissen über das Fahrzeug verwendet. Es wird lediglich ein großer Datensatz an einfach zu messenden Größen in regulären Fahrten zum Training verwendet. Ob durch das Einbringen eines geringen Grades an Fachwissen über die Antriebstopologie in die Netzwerktopologie eine Verbesserung erzielen lässt, wird im Abschnitt 3.3.4 untersucht.

3.3.1 Datengrundlage und Training

Als Datengrundlage dieser Arbeit werden 430 aufgenommene Fahrten verwendet. Diese sind im Zeitraum vom 02.05.2018 bis 10.07.2020 in regulären Fahrten aufgenommen worden. Betrachtet man den Geschwindigkeitsverlauf aller aufgenommenen Fahrten, wird die Überrepräsentation der niedrigen Geschwindigkeiten deutlich (Abbildung 3.8). Insbesondere wird jedoch deutlich, dass die Geschwindigkeit von $v_{\text{Fzg}} = 0 \frac{\text{m}}{\text{s}}$ mit insgesamt ca. 700000 Messungen wesentlich häufiger vorkommt als alle anderen Geschwindigkeiten. Für das Training der KNN sind jedoch möglichst gleich verteilte Geschwindigkeiten vorteilhaft. Ein Ungleichgewicht in den Daten führt zu einem schlechteren Ergebnis in den weniger häufig trainierten Bereichen. Aus diesem Grund werden alle Fahrten, bei denen weniger als 1000 m zurückgelegt und bei denen nicht mindestens einmal eine Geschwindigkeit von mindestens $5 \frac{\text{m}}{\text{s}}$

erreicht wird aussortiert. Fehlmessungen und zu kurze Fahrten werden dadurch aussortiert. Die Aufzeichnung einer Fahrt beginnt automatisch mit dem Starten des Fahrzeuges und nimmt so lange Standzeiten auf. Ohne das Entfernen der Geschwindigkeiten von $v_{\text{Fzg}} = 0 \frac{\text{m}}{\text{s}}$ kann das Ergebnis eines Trainings ein KNN sein, dessen Ausgaben unabhängig von den Eingängen stets 0 ist. Die dauerhafte Ausgabe von 0 stellt auf Grund langer Standzeiten in den Daten ein lokales Minimum der Kostenfunktion da, das es für ein erfolgreiches Training zu vermeiden gilt. Übrig bleiben so noch 345 aufgezeichnete Fahrten mit einer zurückgelegten Strecke von insgesamt 9727 km. Um die Überbewertung von Standzeiten zu vermeiden, werden diese nach einer Dauer von 2 s entfernt.

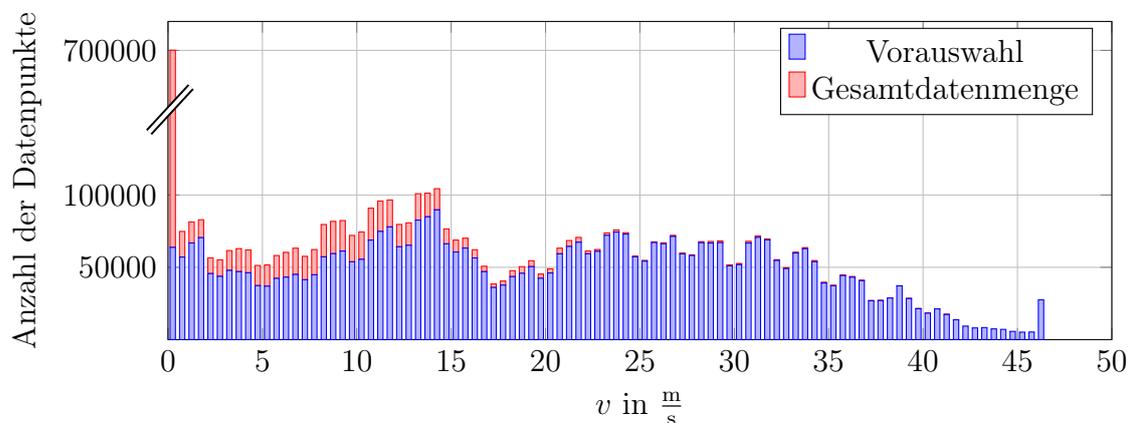


Abbildung 3.8: Geschwindigkeitsverteilung vor und nach der Vorauswahl.

Durch diese Vorauswahl ist die Geschwindigkeitsverteilung wesentlich gleichmäßiger und zum Training der KNN besser geeignet. Zuletzt werden alle Fahrten noch in gleich lange Stücke zerteilt, sodass die Datengrundlage aus 1525 Fahrten von jeweils 300 s Länge besteht. Übrig gebliebene Reststücke kürzer als 300 s werden weggelassen. Die Gesamtstrecke beläuft sich somit auf 9198 km. Alle aufgezeichneten Messgrößen wurden bereits vor dieser Masterarbeit in Matlab gefiltert und auf 10 Hz *sampled*. Aus dem Geschwindigkeitsverlauf wird ebenfalls die elektronisch begrenzte maximale Geschwindigkeit von $v_{\text{Fzg,max}} = 102 \text{ mph} = 46 \frac{\text{m}}{\text{s}}$ deutlich.

Zum Training der KNN wird die Datengrundlage in drei Datensätze geteilt. Zum einen werden 125 Fahrten als Testdatensatz zurückgelegt, um erst am Ende dieser Arbeit einen Vergleich ziehen zu können. Die 1400 Fahrten werden zu 90 % zum Training verwendet, während die verbliebenen 10 % den Validierungsdatsatz bilden.

Das Training der KNN erfolgt mit der Kostenfunktion des *mean squared error*.

$$\text{MSE} = \frac{1}{N} \sum_{n=1}^N (\hat{y} - y)^2 \quad (3.20)$$

MSE : Mittlere quadratische Abweichung

N : Anzahl der Messwerte

\hat{y} : Ausgang des KNN

y : Gewünschter Ausgang des KNN

Üblich ist die Optimierung der Kostenfunktion C mit Hilfe des Gradientenverfahrens (engl. *gradient descent*). Hierbei wird vom aktuellen Satz an Parametern ω ein Schritt $\Delta\omega$ in die Richtung des negativen Gradienten der Kostenfunktion $\frac{\partial C}{\partial \omega}$ vorgenommen. So nähert sich der Parametersatz einem Minimum der Kostenfunktion an. Ohne Konvexität der Kostenfunktion ist eine Konvergenz zum globalen Maximum nicht garantiert. Um diesem entgegen zu wirken, wird dem Optimierungsschritt ein stochastischer Anteil hinzugefügt. Dieses Optimierungsverfahren wird *stochastic gradient descent* genannt.

$$\Delta\omega = -\eta \frac{\partial C}{\partial \omega} + \xi \quad (3.21)$$

C : Kostenfunktion

ω : Menge aller trainierbaren Parameter des KNN

η : Lernrate

ξ : Stochastischer Anteil

Mit dem *Adam optimizer* wird hier als Optimierungsalgorithmus jedoch eine Verbesserung des *stochastic gradient descent* verwendet [11].

Der Name *Adam* kommt vom *adaptive moment estimation* und beschreibt den größten Vorteil des Verfahrens. Anstelle einer einzelnen Lernrate wird für jeden Parameter eine eigene Lernrate als exponentieller gleitender Durchschnitt des Gradienten und des quadratischen Gradienten gebildet. So ist das Verfahren in der Lage, schneller und besser die globalen Minima der Kostenfunktion zu finden.

3.3.2 Tiefes KNN

Als erster Ansatz wird in diesem Kapitel ein vorwärtsgerichtetes tiefes KNN vorgestellt. Es wird hierbei auf die *Dense Layer* Implementierung von Keras zurückgegriffen. Tiefe Neuronale Netze sind als universeller Ansatz für die meisten Probleme erfolgreich anwendbar. Die hier verwendete Topologie des KNN besteht aus drei Schichten mit jeweils 32 Neuronen sowie einer Ausgangsschicht mit einem Neuron. Dies stellt sich als ein sinnvoller Mittelweg zwischen Komplexität und Rechenaufwand da. Um dem *dying ReLU* Problem entgegen zu wirken, wird anstelle der üblicherweise verwendeten *rectified linear unit* der Tangens hyperbolicus in allen

Schichten als Aktivierungsfunktion verwendet. Das *dying ReLU* Problem beschreibt den Fall, dass alle Neuronen mit *ReLU* während des Trainings inaktiv werden und nur noch Nullen als Ausgang für jeden möglichen Eingangswert liefern.

Als Eingänge des KNN werden wie beim physikalischen Modell die Gaspedalstellung p_{ACC} , Bremspedalstellung p_{Brk} sowie die Fahrzeuggeschwindigkeit v_{Fzg} gewählt. Diese Werte werden manuell für alle Fahrten identisch genormt. Der Ausgang des KNN ist die Beschleunigung des Fahrzeuges a zum nächsten Zeitschritt ebenfalls mit Hilfe eines festen Faktors normiert.

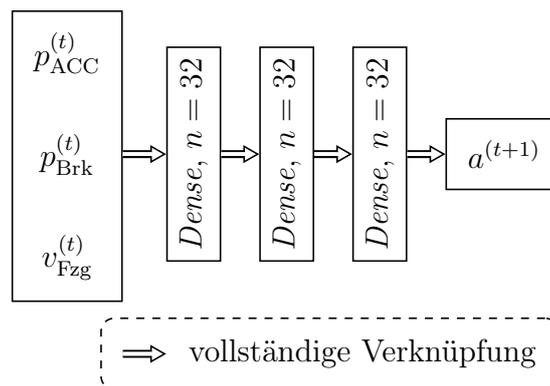


Abbildung 3.9: Schematische Darstellung des tiefen KNN.

Das Training erfolgte über insgesamt 100 Epochen und erreicht eine mittlere quadratische Fehlerabweichung von $\text{MSE} = 0.0018$.

3.3.3 Rekurrentes Netz

Als zweiter Ansatz zur Modellierung der Längsdynamik mit Hilfe von KNN wird mit dem *Long short-term memory (LSTM)* Layer eine rekurrente Netzwerkstruktur gewählt [8]. Eine rekurrente Netzwerkstruktur ermöglicht zusätzlich das Erlernen zeitlich abhängiger Vorgänge und ist somit dem klassischen vorwärtsgerichteten Netz überlegen. Im Rahmen der Aufgabenstellung dieser Masterarbeit soll ein rekurrentes KNN die dynamischen Abhängigkeiten der Längsdynamik erlernen. Während das tiefe KNN (3.3.2) lediglich Aussagen innerhalb eines Zeitschrittes treffen kann, trifft das hier verwendete Netz seine Aussagen basierend auf den letzten 5 Zeitschritten.

Als rekurrente Schichten kommen *LSTM-Layer* zum Einsatz. Klassische rekurrente Strukturen ermöglichen lediglich die Verwendung der meist unmittelbar vorangegangenen Zustände. Das *LSTM* hingegen weist eine komplexere rekurrente Struktur auf, die eine Form des Kurzzeitgedächtnisses ermöglicht, das länger anhält. Somit kann ein *LSTM* auf frühere Erfahrungen reagieren. Zusätzlich ist das *LSTM* in besonderem Maße für die zum Lernen notwendige Fehlerrückführung geeignet.

Als Längsdynamikmodell wird eine dreischichtige Konfiguration mit jeweils 32 Neuronen gewählt. Zur Vermeidung von Überanpassung erfolgt eine Regularisierung

mit 20% *Dropout*. Während die ersten beiden Schichten für jeden Zeitschritt ein Ergebnis weitergeben, liefert die dritte *LSTM* Schicht lediglich den abschließenden Wert als Ergebnis des ganzen Modells.

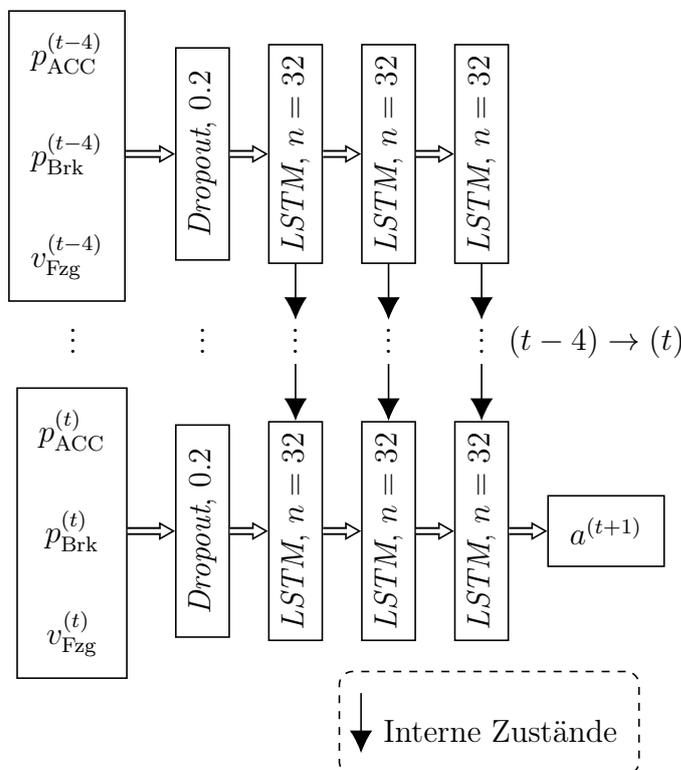


Abbildung 3.10: Die Struktur des rekurrenten KNN.

Zur Optimierung wird der *Adam optimizer* verwendet. Als Kostenfunktion wird die mittlere quadratische Abweichung gewählt. Nach 100 Epochen Training sinkt diese für den Validierungsdatenanteil auf $\text{MSE} = 0.0019$.

3.3.4 Strukturiertes Netz

In diesem Abschnitt soll untersucht werden, ob das Einbringen von Wissen über den Antriebsstrang in die Struktur des KNN einen Vorteil bringt. Der Ansatz des Tiefen KNN (3.2.2) wird nun erweitert. Ähnlich wie beim physikalisch-datenbasierten Modell sollen zuerst die Pedalstellungen p_{ACC} und p_{Brk} in Momente am Rad als Zwischengrößen umgewandelt werden. Danach erst erfolgt die Berechnung einer Längsbeschleunigung des Fahrzeuges aus den Momenten. Hierzu bekommt das KNN eine zweiteilige Form. Am Ausgang des ersten Teiles befinden sich die Zwischengrößen. Der zweite Teil bekommt diese dann zusätzlich zu den ursprünglichen Größen als Eingänge. In der letzten Ausgangsschicht befindet sich dann die gesuchte Längsbe-

schleunigung. Das Ziel dieser Strukturierung ist es, alle Komponenten des Gesamtmomentes als Zwischengrößen mit zu trainieren. Nach erfolgreichem Training kann das KNN jedoch identisch zum Tiefen KNN mit den gleichen Eingangsdaten betrieben werden. Es erfolgt jedoch intern zuerst eine Schätzung der Zwischengrößen. Das Gesamtmoment setzt sich aus den beiden Motormomenten M_{VM} und M_{EM} sowie dem Bremsmoment der mechanischen Bremsen M_{Brems} unter Berücksichtigung der Übersetzungsverhältnisse zusammen. Sowohl das elektromotorische als auch das verbrennungsmotorische Moment liegt als Messgröße vor und kann somit zum Training der Zwischengrößen verwendet werden. Lediglich das Bremsmoment der mechanischen Bremsen kann nicht gemessen werden. Die mechanischen Bremsen sind über hydraulische Kolben betriebene Scheibenbremsen. Für die Kraft F_{Zyl} des hydraulischen Zylinder in der Bremse gilt.

$$F_{Zyl} = p_{Brk} \cdot A \quad (3.22)$$

p_{Brk} : Bremsdruck

A : Querschnittsfläche des Kolben

Für den Anteil F_U der Umfangskraft, der per Reibung auf die Scheiben übertragen wurde gilt somit folgende Beziehung.

$$F_U = \mu \cdot F_{Zyl} \quad (3.23)$$

μ : Reibungskoeffizient

Mit dem Radius, an dem die Umfangskraft angreift, lässt sich nun das Bremsmoment bestimmen.

$$M_{Brems} = F_U \cdot r_{Brems} = A \cdot \mu \cdot r_{Brems} \cdot p_{Brk} \quad (3.24)$$

Das Bremsmoment einer Scheibenbremse kann damit als linear zum Bremsdruck angenommen werden. Unter den gemessenen Größen befinden sich zwei Bremsdrücke $p_{Brk,LF}$ und $p_{Brk,RF}$ die deshalb als Zwischengrößen des strukturierten KNN verwendet werden. Die Drehzahl des Elektromotors wird ebenfalls als Zwischengröße verwendet, sodass alle der gemessenen Größen mit Aussage über Antriebsstrang und Bremsen verwendet werden.

Für den vorderen Teil des KNN werden wieder drei Schichten mit jeweils 32 Neuronen verwendet. Die Zwischenausgangsschicht weist 5 Neuronen auf. Der hintere Teil des KNN setzt sich ebenfalls aus drei Schichten zusammen. Aufgrund der höheren Anzahl an Eingängen wird eine Neuronenanzahl von 64 gewählt. Zuletzt folgt eine Ausgangsschicht mit einem Neuron für die gewünschte Ausgangsgröße der Längsbeschleunigung.

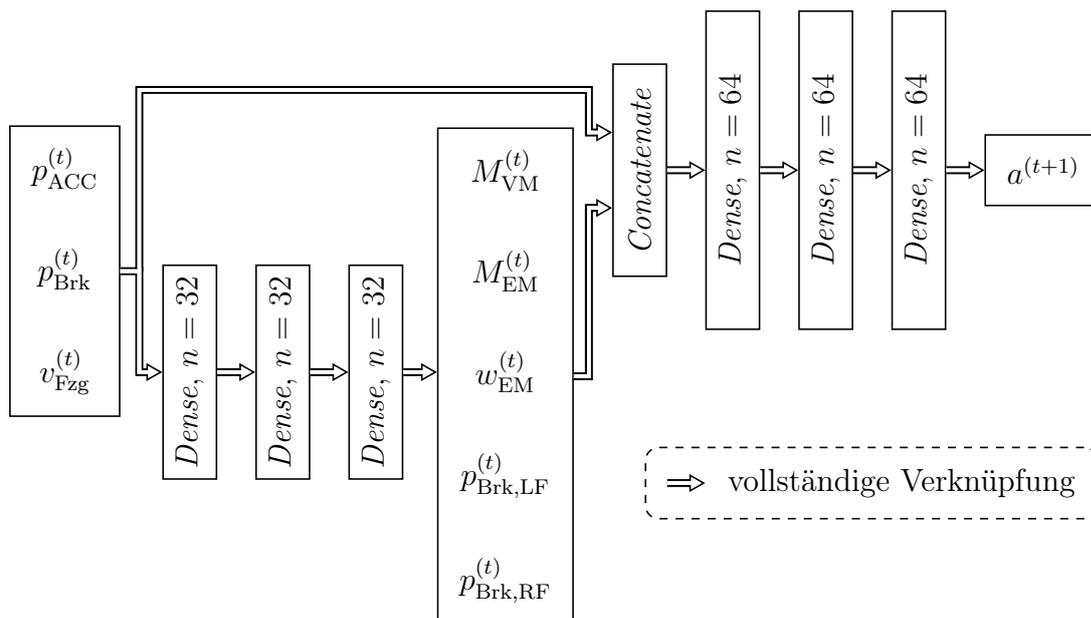


Abbildung 3.11: Schematische Darstellung des strukturierten KNN.

Als Optimierungsalgorithmus wird der *Adam optimizer* mit der mittleren quadratischen Abweichung als Kostenfunktion verwendet. Nach 100 Epochen Training erreicht das strukturierte KNN eine mittlere quadratische Fehlerabweichung der Validierungsdaten von $MSE = 0.0019$. Die Fehlerabweichung der Zwischenausgangsschicht liegt mit $MSE = 0.0064$ deutlich höher. Ein schlechteres Ergebnis für die Zwischenausgänge ist jedoch für die Anwendung des trainierten KNN nicht von Bedeutung. Im Anwendungsfall dieser Arbeit wird lediglich die Ausgangsgröße und nicht die Zwischengrößen verwendet.

3.3.5 AutoML

Die Automatisierung des maschinellen Lernens ist als *AutoML* bekannt. Mit dem Ziel der Vereinfachung des gesamten Prozesses von den Eingangsdaten bis hin zum fertigen Modell erfolgt eine vollständige Automatisierung. Insbesondere die automatisierte Suche nach der besten Netzwerkarchitektur sowie der Hyperparameteroptimierung liefern hochoptimierte Modelle, die häufig handgefertigte Modelle übertreffen. Den Anfang macht ein KNN, welches dann regulär trainiert wird. Anschließend wird eine Änderung am Netz vorgenommen und erneut trainiert. Ein Vergleich der Endergebnisse der Kostenfunktionen erlaubt dann eine Bewertung der durchgeführten Veränderung am Netz. Verändert werden u.a. Parameter wie Schichtanzahlen, Neuronenzahlen der einzelnen Schichten, Regularisierungen wie bspw. Dropout, Aktivierungsfunktionen bis hin zum Optimierungsalgorithmus selbst. Während eines *AutoML* Prozesses werden somit eine Vielzahl von KNN trainiert und deren Er-

gebnisse untereinander verglichen bis das Optimum gefunden wird, bzw. das KNN zufriedenstellende Ergebnisse liefert. Es handelt sich somit um ein noch höher dimensionales Optimierungsproblem als das Training eines einzelnen KNN und ist wesentlich rechenintensiver.

Konzerne wie Amazon und Google liefern Produkte für *AutoML* [7]. Aufgrund der hohen benötigten Rechenleistung handelt es sich um *Cloud Computing* Produkte, bei denen die *AutoML* Berechnungen auf spezieller Hardware in Rechenzentren durchgeführt wird. Die Preise für solche Produkte liegen außerhalb des für diese Arbeit sinnvollen Rahmens und werden deshalb nicht verwendet. Alternativ wird hier jedoch auf *AutoKeras* zurückgegriffen [10]. *AutoKeras* ist ein auf Keras basierendes *open-source AutoML* System entwickelt vom *DATA Lab* der Texas A&M Universität.

Dem *AutoKeras* Modell werden die gleichen Ein- und Ausgangsdaten wie dem tiefen KNN zur Verfügung gestellt. Nach dem zeitintensiven Training liefert *AutoKeras* ein KNN im Keras üblichen Datenformat zurück.

Nach ca. 67 Stunden liefert *AutoKeras* als optimales Netz für die gegebenen Daten ein tiefes KNN mit 2 Schichten. Mit den Neuronenanzahlen 64 für die erste Schicht und 128 für die Zweite. Als Aktivierungsfunktionen werden *ReLU* in allen Schichten verwendet. Der beste Optimierungsalgorithmus ist *Adam*. Zusätzlich erfolgt eine Regularisierung in Form eines *Dropout*.

Das *AutoKeras* Modell ist dem händisch entworfenen tiefen KNN sehr ähnlich. Anstelle von 3 Schichten mit je 32 Neuronen werden jedoch 2 Schichten mit höheren Neuronenanzahlen verwendet. Der *Dropout* ist eine Gegenmaßnahme, um *ReLU* als Aktivierungsfunktionen verwenden zu können. Nach 100 Epochen Training wird eine nahezu identische mittlere quadratische Fehlerabweichung von $MSE = 0.00176$ erreicht.

Da die Ergebnisse dieses KNN nicht weit genug von denen des händisch entworfenen tiefen KNN abweichen, wird in der Auswertung auf eine genauere Betrachtung der Ergebnisse des *AutoKeras* Modells verzichtet. Es konnte jedoch gezeigt werden, wie mit Hilfe von *AutoML* ohne viel Kenntnis über Neuroinformatik ein geeignetes KNN erstellt und trainiert werden kann. Des Weiteren kann das Ergebnis auf Grund der Ähnlichkeit als Bestätigung des händisch entworfenen tiefen KNN gewertet werden. Der im Rahmen dieser Arbeit durchgeführte Durchlauf war auf 100 Versuche begrenzt. Es wurden also 100 verschiedene Konfigurationen an KNN konstruiert, trainiert und miteinander verglichen. Ein besseres Ergebnis aus einer *AutoML* Optimierung mit deutlich höherer Berechnungszeit mit über 100 Versuchen ist möglich, die erwartete Verbesserung zum hier gefundenen Modell ist jedoch als minimal einzustufen. Das hier beschriebene beste KNN der *AutoML* Optimierung wurde bereits im Schritt 34 gefunden.

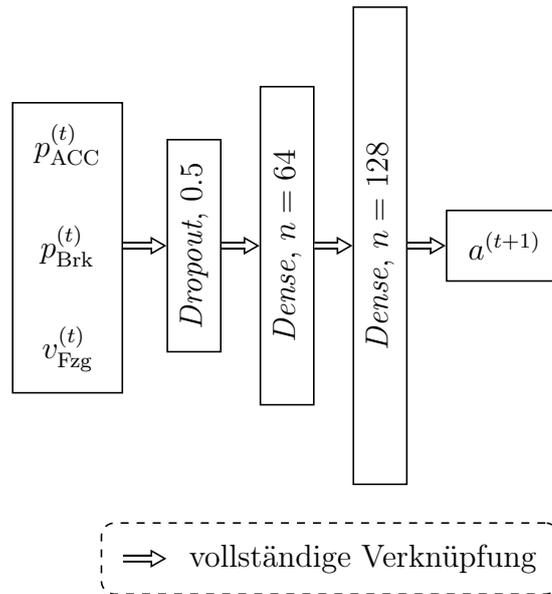


Abbildung 3.12: Die Struktur des *AutoML* generierten KNN.

3.3.6 Weitere Ansätze

Im Rahmen dieser Arbeit erfolgt eine Untersuchung vieler weiterer Ansätze. Insbesondere werden viele verschiedene Hyperparameterkombinationen getestet. In diesem Abschnitt sollen exemplarisch ein paar dieser Ansätze erläutert werden.

Ein Ansatz ist, analog zum Ansatz des strukturierten KNN, das Einbringen von physikalischem Wissen über den zu modellierenden Sachverhalt in die Struktur des KNN. Aus der physikalischen Längsdynamik Modellierung sind die Fahrwiderstandskräfte bekannt. Hierbei ist die Luftwiderstandskraft bei Vernachlässigung der Windgeschwindigkeit v_{Wind} abhängig vom Quadrat der Fahrzeuggeschwindigkeit v_{Fzg}^2 . Die Fahrzeuggeschwindigkeit ist einer der Eingänge des KNN. Somit muss das KNN während eines erfolgreichen Trainings die Fähigkeit zur Quadrierung dieses Eingangs erlangen. Die Idee ist somit, dem KNN bereits die quadrierte Geschwindigkeit als Eingang zur Verfügung zu stellen [9]. Entgegen der Erwartungen dieses Ansatzes ist das modifizierte KNN nicht in der Lage, bessere Ergebnisse als das tiefe KNN zu liefern. Stattdessen erreicht dieses KNN nach den gleichen 100 Epochen Training eine mittlere quadratische Fehlerabweichung von $\text{MSE} = 0.0019$. Weitere Modifizierungen der Hyperparameter dieses KNN bringen das Ergebnis der Kostenfunktion nicht unter das des tiefen KNN ohne v_{Fzg}^2 als Eingang. Somit weicht das Ergebnis dieses Ansatzes ebenfalls nur minimal von dem des tiefen KNN ab und wird im weiteren Verlauf dieser Arbeit nicht weiter betrachtet.

Ergebnisse

In diesem Kapitel werden reale Fahrten mit den in Kapitel 3 erstellten Modellen simuliert. Hierbei werden einmalig verschiedene Fahrten ausgewählt, welche nicht Teil der Trainingsdaten der KNN sind. Aus den Messdaten werden die realen Gas- und Bremspedalstellungen den Modellen als Eingänge zugeführt. Die Simulationen erfolgen ebenfalls zeitdiskret mit einer Frequenz von 10 Hz und weisen somit die gleiche Frequenz der Messdaten auf. Jeder Rechenschritt liefert eine neue Beschleunigung zurück, welche über das explizite Euler-Verfahren zu einer Geschwindigkeit integriert wird.

$$v^{(t+1)} = v^{(t)} + T \cdot a^{(t+1)} \quad (4.1)$$

$$T = 0.1 \text{ s}$$

Da die Modelle alle geschwindigkeitsabhängig sind, wird diese Geschwindigkeit dem nächsten Zeitschritt als Eingang zur Verfügung gestellt. Lediglich im ersten Zeitschritt wird die Startgeschwindigkeit $v^{(0)}$ aus den Messdaten gewählt.

Jedes Modell liefert damit einen Geschwindigkeitsverlauf für die simulierten Fahrten. Zur Auswertung werden diese mit dem realen gemessenen Geschwindigkeitsverlauf verglichen. Zur Bewertung wird die Wurzel aus der mittleren quadratischen Abweichung (*engl. RMSE*) herangezogen.

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{n=1}^N (\hat{y} - y)^2} \quad (4.2)$$

RMSE : Wurzel der mittleren quadratischen Abweichung

N : Anzahl der Messwerte

\hat{y} : Schätzung der Simulation

y : Gemessener Wert

Als weitere Metrik zum Vergleich der Modelle wird die *variance accounted for* berechnet (vgl. [9]). Erklärte Variation oder auch Bestimmtheitsmaß geben an, wie viel der Streuung der Daten durch das Modell erklärt werden können. Die *variance accounted for* gibt somit an, zu wie viel Prozent die Varianz der Daten mit dem Modell erklärt werden können.

$$\text{VAF} = 100 \left(1 - \frac{\text{var}(y - \hat{y})}{\text{var}(y)} \right) \quad (4.3)$$

VAF : *variance accounted for* Metrik

var : Varianz

\hat{y} : Schätzung der Simulation

y : Gemessener Wert

Die Fahrten zum exemplarischen Test, zur Bewertung und Validierung werden aus dem Anteil an Fahrten gewählt, der nicht zum Training der KNN verwendet wird. Es werden drei Fahrten mit möglichst unterschiedlichen Geschwindigkeitsprofilen gewählt. Testfahrt 1 hat eine geringe Durchschnittsgeschwindigkeit von $29.5 \frac{\text{km}}{\text{h}}$ bei einer maximalen Geschwindigkeit von $50.1 \frac{\text{km}}{\text{h}}$. Diese Fahrt enthält somit überwiegend Geschwindigkeiten üblich für innerstädtischen Verkehr und keinen Autobahnanteil. Für die Testfahrt 2 wird ein wesentlich schnelleres Geschwindigkeitsprofil gewählt. Repräsentativ für eine Autobahnfahrt mit einer Durchschnittsgeschwindigkeit von $126.4 \frac{\text{km}}{\text{h}}$. Die Geschwindigkeit für diese Testfahrt liegt zwischen minimal $64.0 \frac{\text{km}}{\text{h}}$ und maximal $151.1 \frac{\text{km}}{\text{h}}$. Für eine dritte Testfahrt wird ein Geschwindigkeitsprofil mit sowohl langsamen als auch schnellen Abschnitten gewählt. Die Höchstgeschwindigkeit beträgt $125.1 \frac{\text{km}}{\text{h}}$, die Durchschnittsgeschwindigkeit liegt bei $80.5 \frac{\text{km}}{\text{h}}$. Diese drei ausgewählten Fahrten sollen möglichst den vollständigen Fahrzeugdynamikbereich regulärer Autofahrten abdecken. Insbesondere sollen die gewählten Fahrten eine repräsentative Einschätzung der Modelle für innerstädtische, auf Autobahnen üblichen Geschwindigkeiten sowie die Kombination dieser liefern.

Testfahrt	1	2	3
v_{\min} in $\frac{\text{m}}{\text{s}}$	0.0	17.8	0.0
v_{\max} in $\frac{\text{m}}{\text{s}}$	13.9	42.0	34.7
v_{avg} in $\frac{\text{m}}{\text{s}}$	8.2	35.1	22.4
s_{ges} in m	2462	10854	6706

Tabelle 4.1: Eckdaten der Testfahrten.

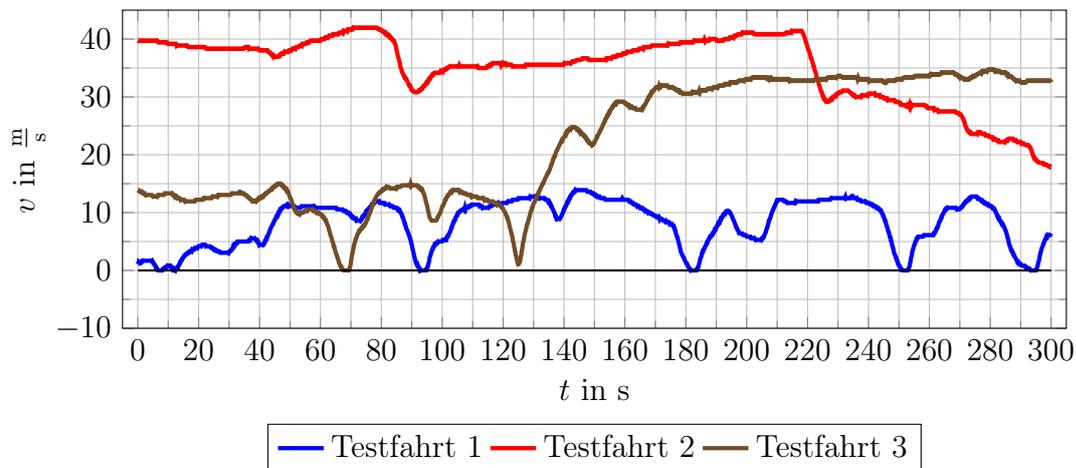


Abbildung 4.1: Geschwindigkeitsprofile der 3 ausgewählten Testfahrten.

Die Simulationen der Modelle liefern Geschwindigkeitsprofile, die in den folgenden Abschnitten vorgestellt werden. Nach der Vorstellung der Simulationsergebnisse der einzelnen Modelle folgt ein Vergleich sowie eine Bewertung. Anschließend wird auf die Qualität der Trainingsdaten und deren benötigte Menge eingegangen.

4.1 Physikalisch-datenbasiertes Modell

Die Simulation der ersten Testfahrt mit dem physikalisch-datenbasiertem Modell offenbart ein funktionierendes Längsdynamikmodell. Die Wurzel der mittleren Fehlerquadrate hat mit $\text{RMSE} = 1.060$ einen niedrigen Wert. Die *variance accounted for* ist mit dem hohen Wert von $\text{VAF} = 96.64\%$ ebenfalls gut. Die Abweichung ist für den überwiegenden Teil gering, insbesondere jedoch bei Abbremsvorgängen (bspw. bei $t = 90 \text{ s}$) bremst das Modell früher ab als das reale Fahrzeug. So kommt es zu einer maximalen Abweichung von $\Delta v (t = 89.5 \text{ s}) = 3.22 \frac{\text{m}}{\text{s}}$. Statische Geschwindigkeiten werden mit sehr geringen Abweichungen, meist innerhalb $-0.5 \frac{\text{m}}{\text{s}} \leq \Delta v \leq 0.5 \frac{\text{m}}{\text{s}}$, modelliert. Dynamische Vorgänge sind wie erwartet schlechter modelliert.

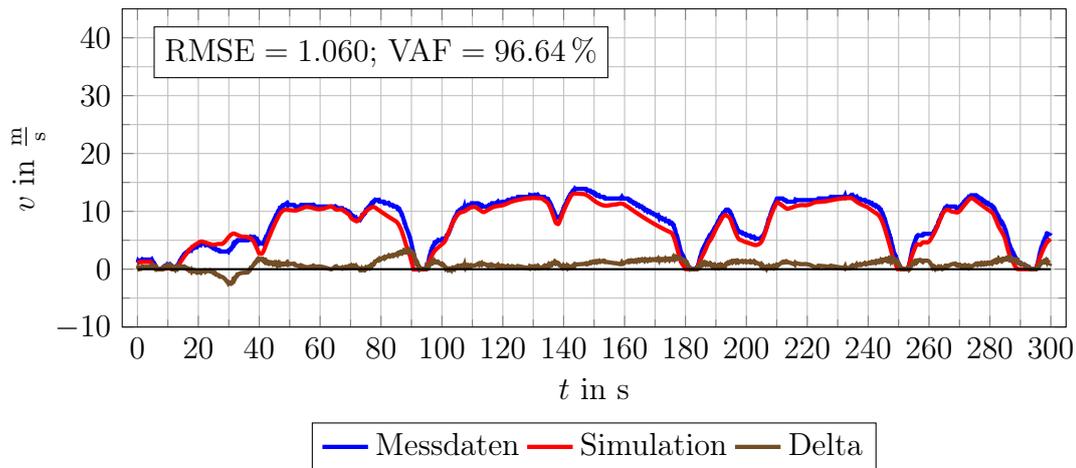


Abbildung 4.2: Testfahrt 1 des physikalisch-datenbasierten Modells.

Die zweite Testfahrt bei wesentlich höheren Geschwindigkeiten liefert einen leicht größeren Wert für die Wurzel der mittleren Fehlerquadrate von $\text{RMSE} = 1.363$. Die absolute Abweichung bei höheren Geschwindigkeiten ist größer als bei den niedrigen Geschwindigkeiten der Testfahrt 1.

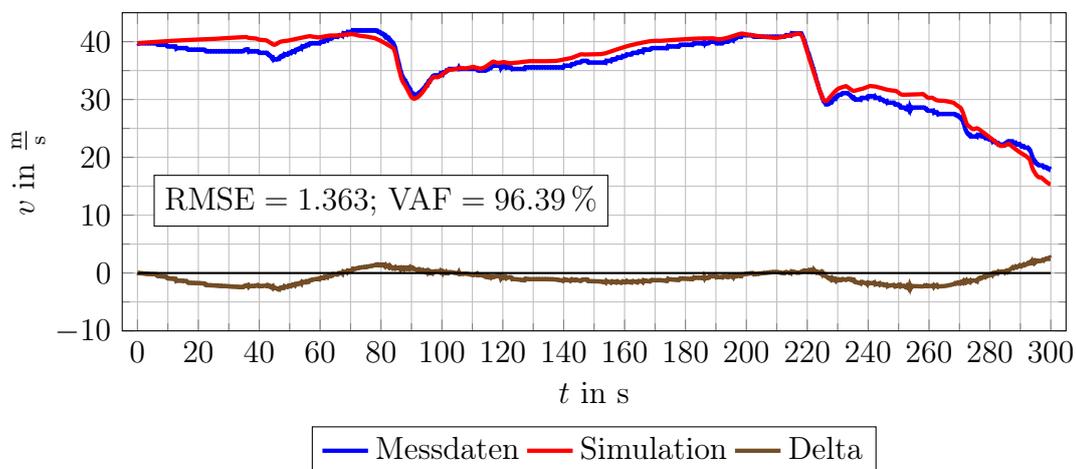


Abbildung 4.3: Testfahrt 2 des physikalisch-datenbasierten Modells.

Die dritte Testfahrt weist im vorderen Teil die zuvor festgestellte Güte mit sehr geringen Abweichungen auf. Im hinteren Teil gibt es jedoch eine statische Abweichung in der Größenordnung von $\Delta v = 5 \frac{\text{m}}{\text{s}}$. Aus diesem Grund liegt die Wurzel der mittleren quadratischen Abweichung bei einem hohen Wert von $\text{RMSE} = 3.054$. Die *variance accounted for* liegt jedoch mit $\text{VAR} = 94.48\%$ weiterhin in einem guten Bereich.

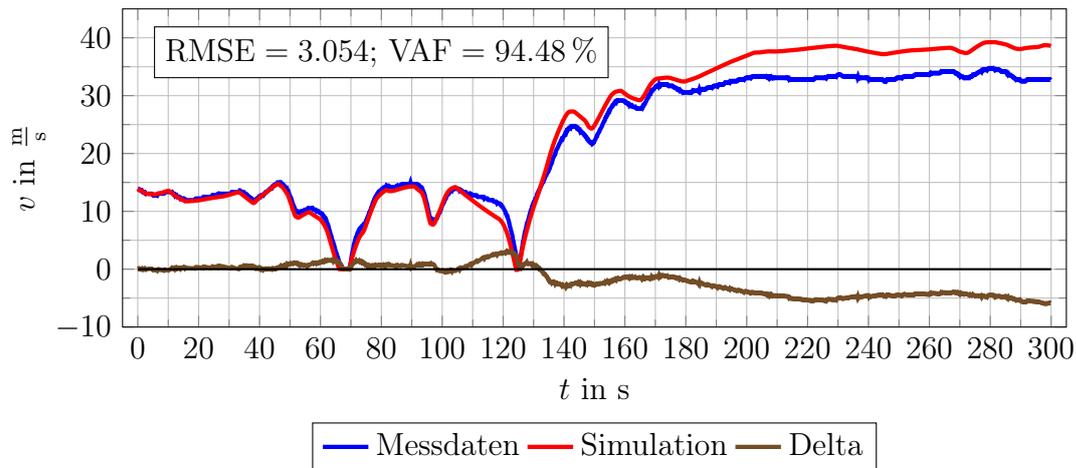


Abbildung 4.4: Testfahrt 3 des physikalisch-datenbasierten Modells.

4.2 Tiefes KNN

Nun sollen die Ergebnisse der Simulationen des tiefe KNN der drei Testfahrten dargestellt werden. Die Wurzel der mittleren Fehlerquadrate von $\text{RMSE} = 1.319$ bestätigt eine erfolgreiche Modellierung der Fahrzeuglängsdynamik. Der Verlauf der Abweichung zeigt Ähnlichkeiten zum physikalisch-datenbasierten Modell. Dynamische Vorgänge werden wie erwartet schlechter durch das Modell dargestellt, statische Geschwindigkeiten werden jedoch mit hoher Präzision simuliert. Das Modell weist zum gleichen Zeitpunkt, wie das physikalische Modell, die höchste Abweichung von $\Delta v(t = 89.5 \text{ s}) = 4.19 \frac{\text{m}}{\text{s}}$ auf.

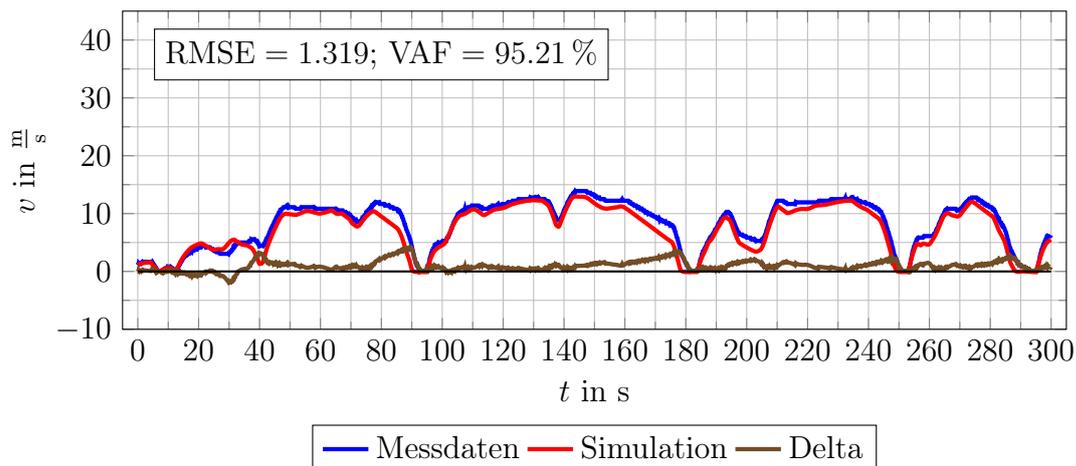


Abbildung 4.5: Testfahrt 1 des tiefen KNN.

Bei höheren Geschwindigkeiten zeigt auch dieses Modell höhere Abweichungen. Die Simulation der zweiten Testfahrt mit dem tiefen KNN weist nahezu konstant eine Abweichung in der Größenordnung von 10 % auf. Bei einem $\text{RMSE} = 2.411$ sowie $\text{VAF} = 97.49\%$ lässt das Ergebnis dennoch auf eine erfolgreiche Modellierung schließen.

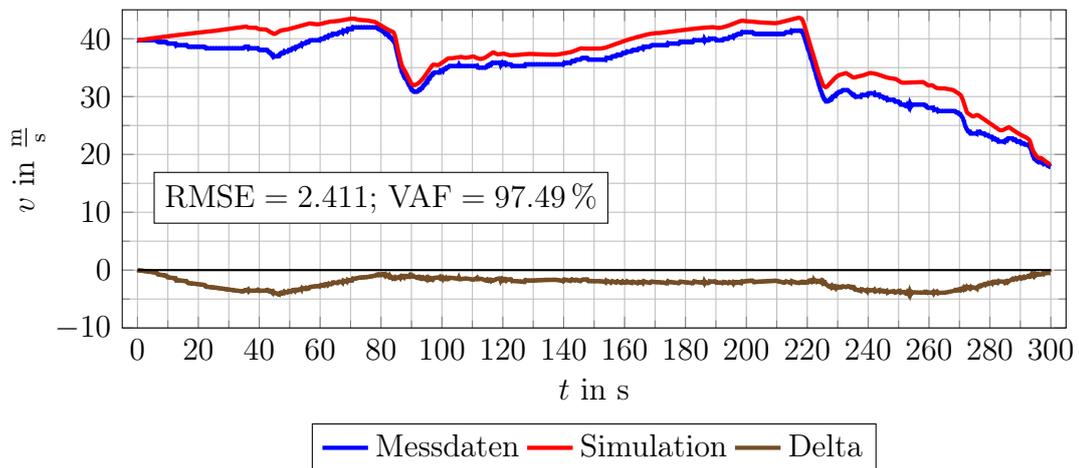


Abbildung 4.6: Testfahrt 2 des tiefen KNN.

Die Simulation der dritten Testfahrt bestätigt die Ergebnisse. Im vorderen Teil der niedrigen Geschwindigkeiten ist die Modellierung nahezu perfekt. Im hinteren Teil weist das Modell jedoch bei den höheren Geschwindigkeiten eine größere konstante Abweichung auf. Die Wurzel der mittleren quadratischen Fehler liegt somit bei $\text{RMSE} = 2.825$.

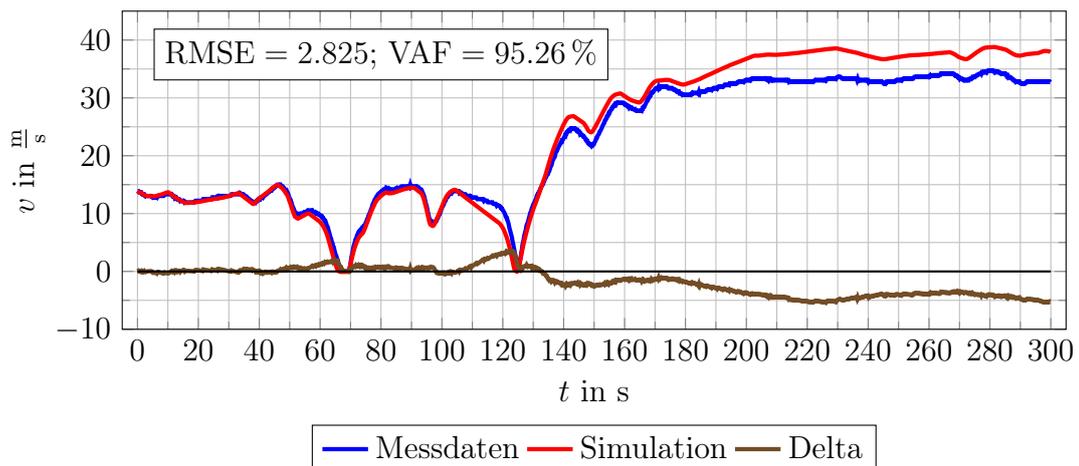


Abbildung 4.7: Testfahrt 3 des tiefen KNN.

4.3 Rekurrentes Netz

Die Simulation des rekurrenten Netzes zeigt, dass das Training erfolgreich war. Der Wert der simulierten Geschwindigkeit liegt nahe an der gemessenen Geschwindigkeit. Ebenfalls scheint das rekurrente KNN die dynamischen Zusammenhänge erlernt zu haben. Zur Simulation werden die Eingangswerte aus den ersten fünf Zeitschritten, für den ersten Berechnungsschritt, aus den Messdaten genommen. Bei allen weiteren Berechnungsschritten werden die Werte der Geschwindigkeit jeweils um einen Zeitschritt nach hinten verschoben sowie der letzte Zeitschritt mit der aus dem Ausgang berechneten Geschwindigkeit besetzt. Die Eingangswerte des Gas- und Bremspedals werden wie beim klassischen vorwärtsgerichteten KNN nur aus den Messdaten genommen.

Die Simulation der ersten Testfahrt zeigt einen Wert von $\text{RMSE} = 1.354$ für die Wurzel der mittleren Fehlerquadrate. Die Abweichung erreicht jedoch einen maximalen Wert von $\Delta v(t = 86.1 \text{ s}) = 4.27 \frac{\text{m}}{\text{s}}$. Die *variance accounted for* erreicht einen Wert von $\text{VAF} = 94.15 \%$.

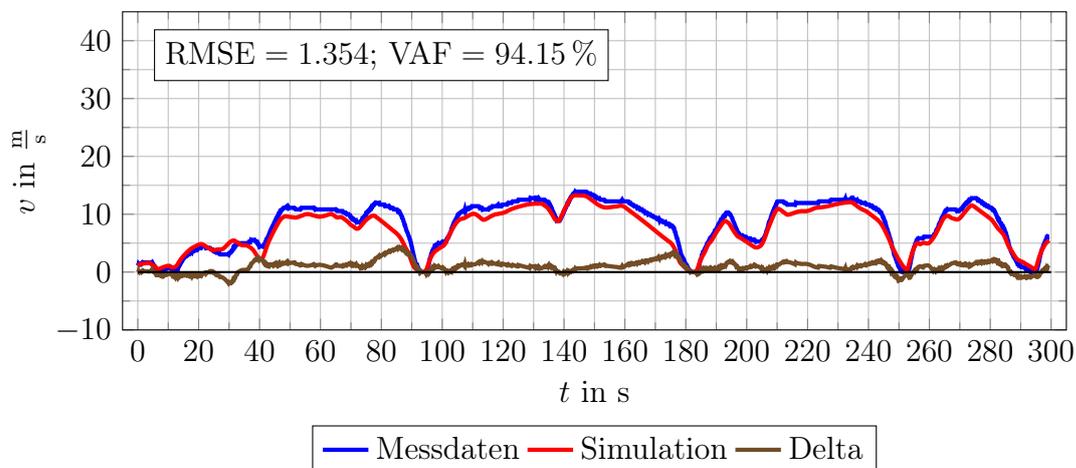


Abbildung 4.8: Testfahrt 1 des rekurrenten KNN.

Die Wurzel der mittleren quadratischen Abweichung der zweiten simulierten Testfahrt liegt bei $\text{RMSE} = 1.461$ bei einer *variance accounted for* von $\text{VAF} = 93.80 \%$.

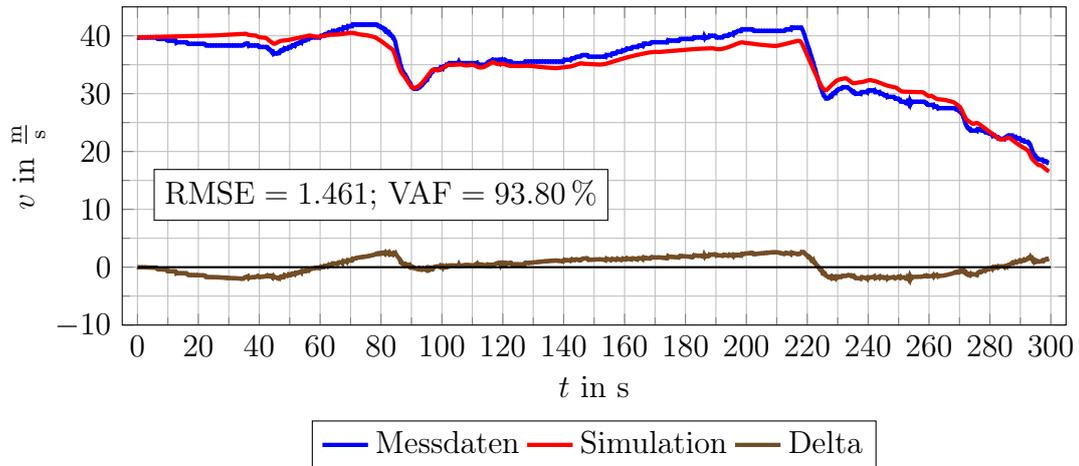


Abbildung 4.9: Testfahrt 2 des rekurrenten KNN.

Bei der Simulation der dritten Testfahrt, mit einer Mischung aus langsamer und schneller Fahrt, erreicht die Wurzel der mittleren Fehlerquadrate mit $\text{RMSE} = 0.964$ einen sehr niedrigen Wert. Die *variance accounted for* liegt mit einem Wert von $\text{VAF} = 99.21\%$ ebenfalls nahe an ihrem Maximum.

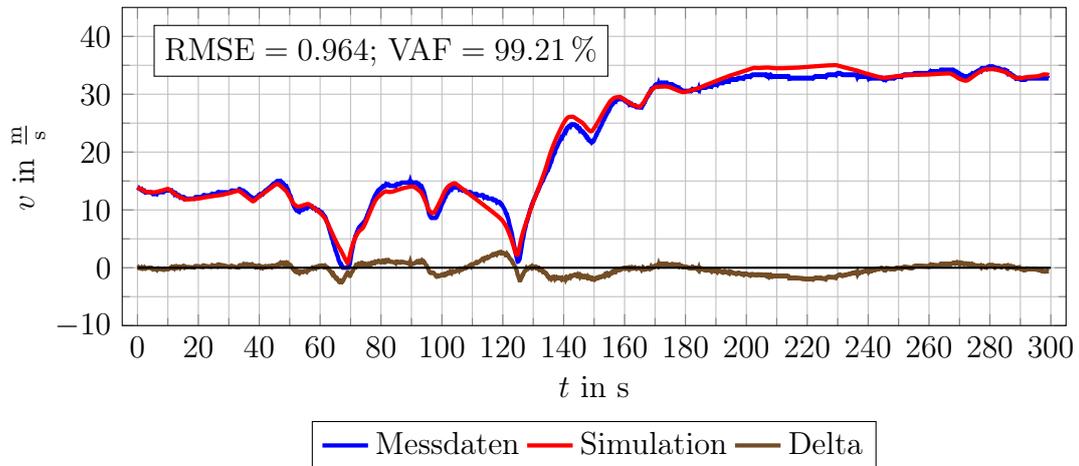


Abbildung 4.10: Testfahrt 3 des rekurrenten KNN.

4.4 Strukturiertes Netz

Das Training des strukturierten KNN ist erfolgreich. Die Wurzel der mittleren quadratischen Fehler liegt bei $\text{RMSE} = 1.542$. Die *variance accounted for* liegt bei $\text{VAF} = 93.80\%$. Mit der Ausnahme der Bremsvorgänge ist die Abweichung überwiegend gering. Der größte Fehler tritt wieder zum gleichen Zeitpunkt wie bei den anderen Modellen auf und beträgt $\Delta v(t = 89.5 \text{ s}) = 4.73 \frac{\text{m}}{\text{s}}$.

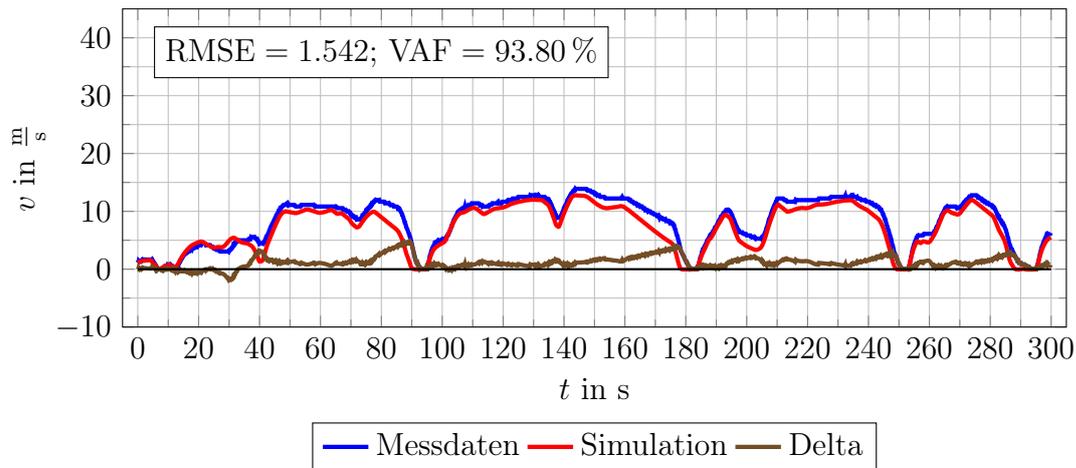


Abbildung 4.11: Testfahrt 1 des strukturierten KNN.

Die zweite Testfahrt offenbart sehr geringe Abweichungen, die sich in einer geringen Wurzel der mittleren Fehlerquadrate von $RMSE = 1.212$ niederschlagen. Die *variance accounted for* Metrik liegt mit $VAF = 96.59\%$ ebenfalls auf einem hohen Wert.

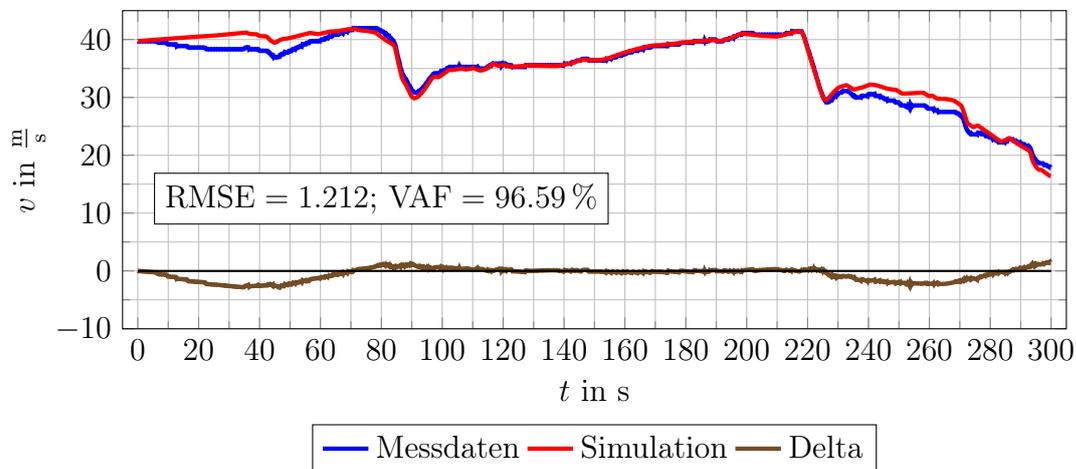


Abbildung 4.12: Testfahrt 2 des strukturierten KNN.

Die dritte Testfahrt zeigt wiederum einen geringen Fehler für die niedrigen Geschwindigkeiten und einen größeren Fehler für die höheren Geschwindigkeiten. Die Wurzel der mittleren Fehlerquadrate liegt bei $RMSE = 2.221$.

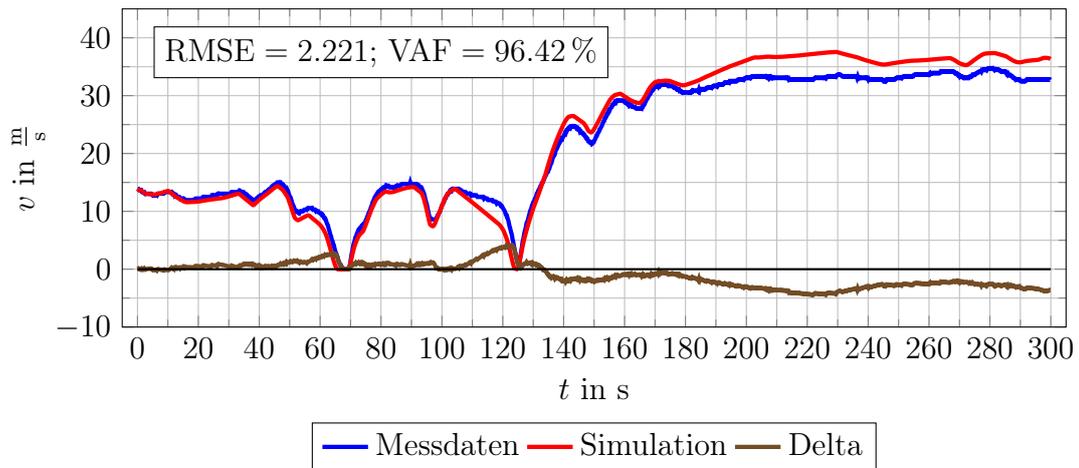


Abbildung 4.13: Testfahrt 3 des strukturierten KNN.

Bei Betrachtung der Verläufe der Zwischenausgänge wird deutlich, dass diese nicht die gleichen Güte des Trainingserfolges aufweisen. Werden verbrennungsmotorisches und elektrisches Antriebsmoment betrachtet, kann eine Aussage darüber getroffen werden, ob das strukturierte KNN die Hybridantriebsstrategie des Fahrzeuges erlernt hat. Der Verlauf des verbrennungsmotorischen Antriebsmoment zeigt, dass das KNN zwar die genauen Werte nicht erlernt hat, die Hybridstrategie mit ein- und ausgeschaltetem Verbrennungsmotor erscheint jedoch vom KNN in Grundzügen erlernt zu sein.

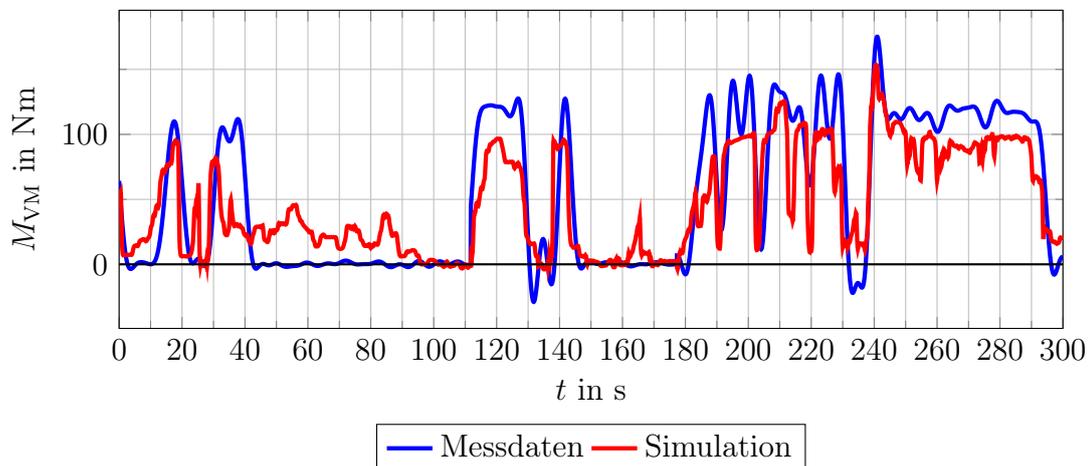


Abbildung 4.14: Verbrennungsmotorisches Antriebsmoment als Zwischenausgang des strukturierten KNN.

Diese Verläufe offenbaren jedoch auch, dass das strukturierte KNN zum Teil den falschen Fahrmodus wählt. Im Bereich von $250 \text{ s} \leq t \leq 290 \text{ s}$ simuliert das Modell

einen rein verbrennungsmotorischen Antrieb. Die Messdaten zeigen jedoch ein höheres Antriebsmoment für den Verbrennungsmotor und ein negatives Antriebsmoment für den Elektromotor. Somit weisen die Messdaten der realen Fahrt in dieser Zeitspanne eine Lastpunktanhebung zum Laden der Traktionsbatterie auf. Um die Fahrmodi vollständig abbilden zu können, müsste zusätzlich der Ladestand der Batterie simuliert werden. So ist es möglich, die Zwischenausgangsschicht um den Batterie-Strom I_{Bat} und Spannung U_{Bat} zu erweitern. Über externe Berechnungen, analog zur Integration der Beschleunigung zur Geschwindigkeit, ließe sich der Ladestand der Batterie als zusätzlicher Eingang des KNN realisieren. Unter Vernachlässigung der Nebenverbraucher ist so eine Simulation der Batterie möglich.

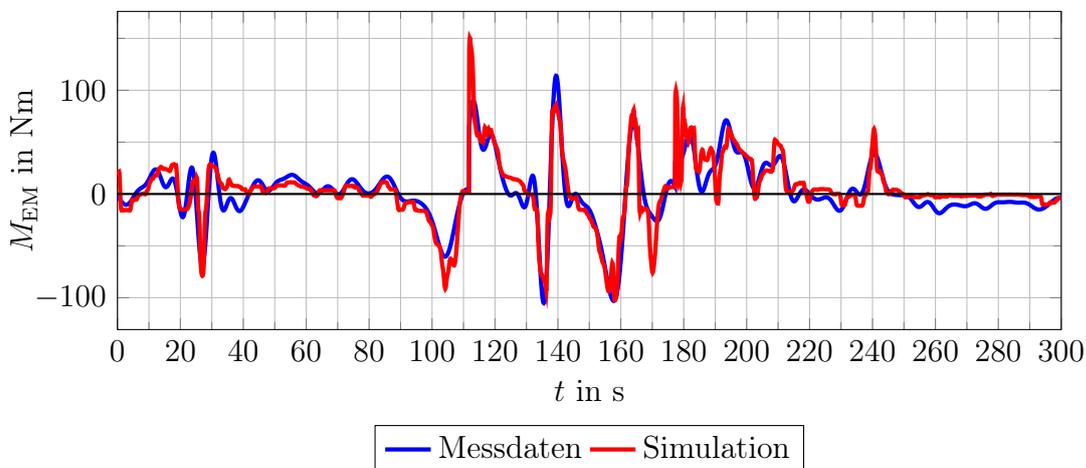


Abbildung 4.15: Elektrisches Antriebsmoment als Zwischenausgang des strukturierten KNN.

4.5 Vergleich und Bewertung der Ergebnisse

Nach der Simulation der drei Testfahrten mit den verschiedenen Modellen lassen sich deren Wurzeln der mittleren quadratischen Fehler vergleichen.

RMSE	Testfahrt 1	Testfahrt 2	Testfahrt 3	MSE_{Test}
phys.-datb. Modell	1.060	1.363	3.054	0.0019157
tiefes KNN	1.319	2.411	2.825	0.0015341
rekurrentes KNN	1.354	1.461	0.964	0.0022388
strukturiertes KNN	1.542	1.212	2.221	0.0015335

Tabelle 4.2: RMSE der simulierten Testfahrten und MSE des Testdatensatzes.

In der ersten Testfahrt mit niedrigen Geschwindigkeiten weist das physikalisch-datenbasierte Modell den niedrigsten Fehler auf. In der zweiten Testfahrt ist das strukturierte KNN das beste hier verglichene Modell. Das physikalisch-datenbasierte Modell sowie das tiefe KNN scheinen in niedrigeren Geschwindigkeiten genauere Ergebnisse zu liefern. Das strukturierte Modell weist jedoch in der zweiten Testfahrt mit den höchsten Geschwindigkeiten die geringste Abweichung auf. In der dritten Testfahrt erzielt jedoch das rekurrente KNN mit seinen *LSTM* Schichten mit Abstand das beste Ergebnis.

Wird die Kostenfunktion der KNN nach dem Training auf den Testdatensatz, bestehend aus den aktuellsten 125 der aufgenommenen Fahrten, angewandt. Ergibt sich für das Tiefe KNN ein Wert von $MSE_{\text{Test}} = 0.0015341$. Wird dies mit dem Ergebnis des *AutoML* generiertem KNN von $MSE_{\text{Test}} = 0.0015684$ verglichen, zeigt sich, dass das händisch angelegt KNN zwar wie in Kapitel 3.3.5 gezeigt ein minimal besseren Wert für den Validierungsdatensatz zeigt, aber im Testdatensatz schlechter abschneidet. Insgesamt sind die Ergebnisse dieser beiden KNN nahezu identisch, weshalb auf eine detaillierte Darstellung der Ergebnisse des *AutoML* generierten KNN verzichtet werden kann. Für das strukturierte KNN ergibt sich ebenfalls ein ähnlicher Wert für das mittlere Fehlerquadrat des Testdatensatz von $MSE_{\text{Test}} = 0.0015335$. Das rekurrente KNN liefert hingegen mit einem Wert von $MSE_{\text{Test}} = 0.0022388$ das schlechteste Ergebnis. Somit ist das strukturierte KNN, wenn auch nur mit minimal besserer Güte, als das beste der KNN Modelle anzusehen. Zum Vergleich der KNN mit dem physikalisch-datenbasierten Modell wird auch auf die Ausgaben dieses Modells die Kostenfunktion angewandt. Das mittlere Fehlerquadrat für den Testdatensatz ergibt sich zu $MSE_{\text{Test}} = 0.0019157$. Somit liefert das tiefe KNN geringere Abweichungen als das physikalisch-datenbasierte Modell und das strukturierte KNN liefert für den Testdatensatz die geringsten Abweichungen.

Im Vergleich der Modelle mit Hilfe der mittleren Fehlerquadrate angewandt auf einen nicht zum Training verwendeten Testdatensatz liegt somit das strukturierte KNN unter den hier verglichenen Modellen vorne. Eine ausführliche Auswertung, Bewertung und Erläuterung der Grenzen der Modelle erfolgt im Kapitel 5.

4.6 Datenqualität

Die größte Grenze der Güte der KNN ist der Datensatz an denen diese trainiert werden. Die in dieser Arbeit verwendeten Daten sind aus den fahrzeuginternen Sensoren gewonnen sind somit nur von geringer Qualität. Eine mögliche Begrenzung der Modelle liegt schon in der Messfrequenz von 10 Hz. Dies macht es unmöglich, höherfrequente Vorgänge überhaupt zu messen, geschweige denn in den Modellen abzubilden. Außerdem steigt mit der Periodendauer der Fehler bei der Integration der Geschwindigkeiten. Als Ausgangsgröße der Modelle wurde die Längsbeschleunigung a_{long} verwendet. Die aus dem Beschleunigungssensor des Ford C-Max gewonnenen

Daten sind jedoch stets verrauscht und fehlerbehaftet.

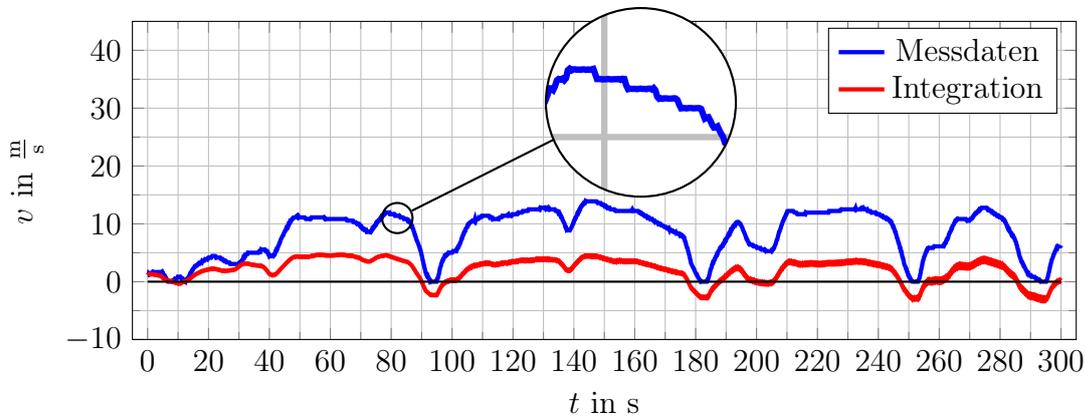


Abbildung 4.16: Integration der Beschleunigungsdaten der Testfahrt 1.

Betrachtet man den Geschwindigkeitsverlauf, der aus der Integration der Beschleunigungsdaten generiert werden kann, wird der Fehler der Sensordaten deutlich. Selbst in Situationen, in denen das Fahrzeug still steht, ergibt die Integration des verrauschten Signals eine Bewegung. Die KNN werden somit anhand dieses fehlerhaften Signals trainiert. Der Erfolg des Training spricht damit erstens für eine erfolgreiche Abstraktion der gewünschten physikalischen Abhängigkeiten, zweitens werden bessere Daten mit großer Wahrscheinlichkeit die Ergebnisse der KNN verbessern.

Der Versuch bessere Beschleunigungsdaten aus dem aufgezeichneten Geschwindigkeitsverlauf zu generieren, scheitert an dessen Wertdiskretheit. Eine Glättung des Geschwindigkeitsverlaufes ist zwar möglich, bringt jedoch neue Fehler mit sich.

Das wertdiskrete Geschwindigkeitssignal sorgt ebenfalls für schlechtere Werte der im Kapitel 4 angewendeten Wurzel der mittleren Fehlerquadrate. Somit schneiden die Modelle in diesen Testfahrten, verglichen zu einem realen wertkontinuierlichen Verlauf, schlechter ab. Betrachtet man bspw. den Fehler des strukturierten KNN in der zweiten Testfahrt wie in Abbildung 5.5, erkennt man, dass ein Teil der Abweichung mit den wertdiskreten Geschwindigkeitswerten zu erklären ist.

4.7 Wie viele Daten sind genug?

In diesem Abschnitt soll eine Einschätzung erfolgen, mit welcher Datenmenge ein erfolgreiches Training der KNN möglich ist. Die benötigte Datenmenge hat einen großen Einfluss auf die Realisierbarkeit der Ansätze des maschinellen Lernens zur Fahrzeuglängsdynamik Modellierung. Der generelle Zusammenhang zeigt, dass mehr Daten ein besseres Modell ermöglichen. Ab einer gewissen Menge bringen zusätzliche Daten jedoch nur noch geringen bis gar keinen Mehrwert im Training des KNN.

Somit existiert eine Menge an Trainingsdaten, die ein optimales Verhältnis zwischen Aufwand und Nutzen darstellt.

Zur Untersuchung des Einflusses der Datenmenge wird das tiefe KNN mit unterschiedlichen Anteilen der gesamten Datenmenge trainiert. Ein Vergleich der Ergebnisse der Kostenfunktion angewandt auf den gleichen Testdatensatz erlaubt eine Einschätzung der Modelle.

Anteil	Dauer	Strecke	MSE_{Test}
100 %	116.67 h	8391 km	0.001534
90 %	105.00 h	7703 km	0.001549
80 %	93.33 h	6794 km	0.001548
70 %	81.67 h	6097 km	0.001619
60 %	70.00 h	5070 km	0.001538
50 %	58.33 h	4388 km	0.001524
40 %	46.67 h	3565 km	0.001569
30 %	35.00 h	2674 km	0.001563
20 %	23.33 h	1744 km	0.001774
10 %	11.67 h	712 km	0.002561
5 %	5.83 h	374 km	0.004663
1 %	1.17 h	48 km	0.009931

Tabelle 4.3: Einfluss der Datenmenge auf das Training.

Nahezu identische Ergebnisse für Trainingsdatenanteile von 100 % bis 30 % bestätigen sofort, dass die im Rahmen dieser Arbeit verwendete Menge der Daten für das Training der KNN keinen limitierenden Faktor spielt. Es wird damit jedoch auch deutlich, dass mit bereits geringerem Aufwand eine geeignete Datenmenge aufgenommen hätte werden können. Statt der, für das Training der Modelle dieser Arbeit verwendeten, 8391 km hätten Fahrten mit einer Gesamtlänge in der Größenordnung von 2000 km bei 3000 km bereits ähnliche Ergebnisse liefern können.

Eine eindeutige Bewertung der unterschiedlichen Datenmengen erlaubt dieses einfache Testkriterium jedoch nicht. Das Training der KNN, als statistischer Prozess und insbesondere durch die Initialisierung der Gewichte mit Zufallsvariablen, liefert selbst für die gleichen Trainingsdaten stets andere Ergebnisse. Somit unterliegen die Werte der mittleren Fehlerquadrate einer gewissen Streuung. Der Abfall des Fehlers bei über 10 % der Daten ist jedoch trotzdem ein eindeutiges Indiz auf die Größenordnung der benötigten Datenmenge.

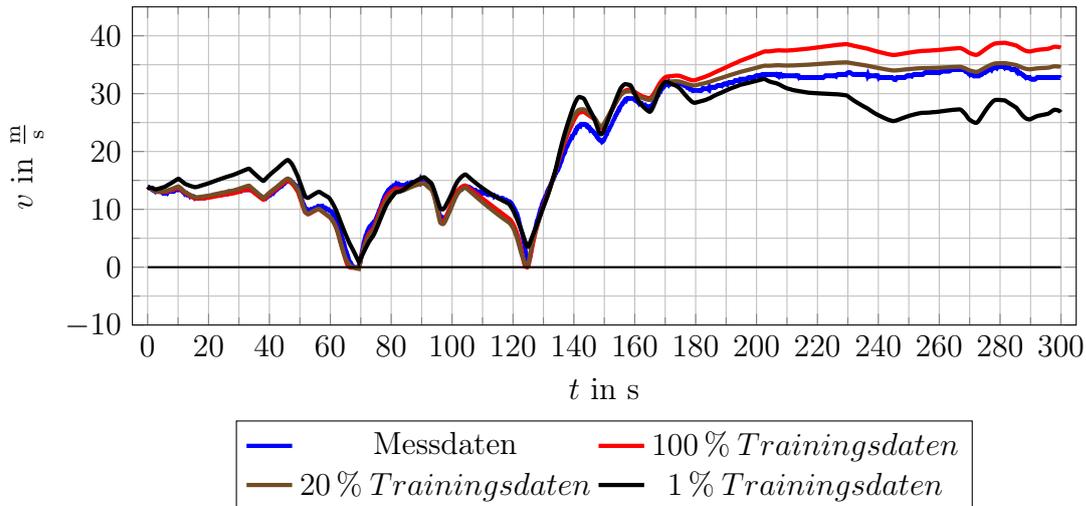


Abbildung 4.17: Einfluss der Datenmenge auf Testfahrt 3 des tiefen KNN.

Als alternative Methode zur Bestimmung einer ausreichenden Datenmenge soll ein statistisches Verfahren angewandt werden. Die korrekte Datenmenge wird durch den aus statistischer Sicht konvergenten Punkt der Dichtefunktion der gesammelten Daten abgeschätzt [18]. Die Dichtefunktion der Daten $\mathbf{x} = \{x_i\}_{i=1}^n$ sei $\hat{f}(x; n)$ für verschiedene Datenmengen n . In dieser Untersuchung wird die Anzahl der Messpunkte n jeweils aus der Menge der Fahrten mit je $t = 300$ s bestimmt. Die Änderung der Dichtefunktion wird mit dem Hinzufügen einer weiteren Fahrt mit m Messpunkten immer geringer.

$$\hat{f}(x; n) \sim \hat{f}(x; n + m), \text{ mit } n \rightarrow \infty, m \in \mathbb{R}^+ \quad (4.4)$$

Wenn das Hinzufügen weiterer Daten keine Änderung der Dichtefunktion liefert, kann die Datenmenge als ausreichend angesehen werden. Zur Schätzung der Dichtefunktion der Daten wird die Kerndichteschätzung angewandt [2].

$$\hat{f}(x; n) = \frac{1}{nh} \sum_{i=1}^n \kappa\left(\frac{x - x_i}{h}\right) \quad (4.5)$$

Hierbei ist h die Bandbreite. Als Kern κ wird der Gaußkern gewählt.

$$\kappa(t) = \frac{1}{\sqrt{2\pi}} e^{-\frac{t^2}{2}} \quad (4.6)$$

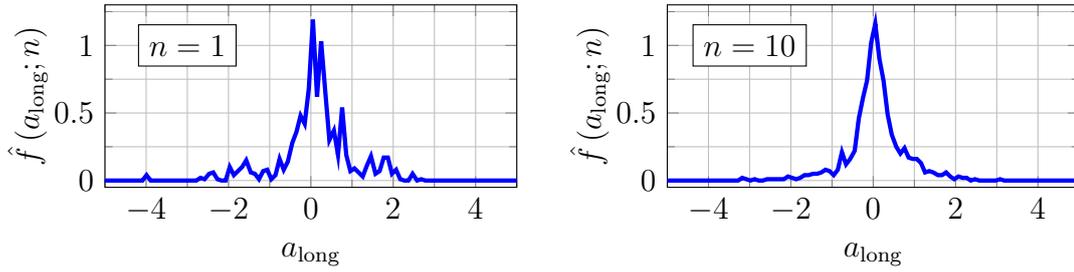


Abbildung 4.18: Kerndichteschätzung für unterschiedliche Datenmengen.

Um zwei Dichtefunktionen zu vergleichen, wird die Kullback-Leibler-Divergenz herangezogen [17] [12].

$$KL\left(\hat{f}(x; n+m) \parallel \hat{f}(x; n)\right) = \int_{-\infty}^{\infty} \hat{f}(x; n+m) \cdot \log \frac{\hat{f}(x; n+m)}{\hat{f}(x; n)} dx \quad (4.7)$$

Hierbei spricht ein großer Wert der Kullback-Leibler-Divergenz der beiden Dichtefunktionen $KL\left(\hat{f}(x; n+m) \parallel \hat{f}(x; n)\right)$ dafür, dass zwischen den beiden unterschiedlichen Dichtefunktionen $\hat{f}(x; n+m)$ und $\hat{f}(x; n)$ unterschieden werden kann. Somit enthalten die zusätzlichen m Messpunkte weitere nicht in den n Messpunkten enthaltene Daten und liefern einen Mehrwert zum erfolgreichen Training der KNN. Nähert sich die Kullback-Leibler-Divergenz jedoch 0 sind sich die beiden Dichtefunktionen sehr ähnlich und die m Messpunkte beinhalten keine neuen Informationen. Die Abbruchbedingung des statistischen Verfahrens wird also so formuliert, dass wenn $KL\left(\hat{f}(x; n+m) \parallel \hat{f}(x; n)\right)$ nur noch minimale Änderungen für zusätzliche Daten erfährt, die gewünschte Datenmenge erreicht worden ist.

$$\left| KL\left(\hat{f}(x; n+m) \parallel \hat{f}(x; n)\right) - KL\left(\hat{f}(x; n+2m) \parallel \hat{f}(x; n+m)\right) \right| \leq \epsilon, \epsilon \in \mathbb{R}^+ \quad (4.8)$$

Angewandt auf die drei Eingangs- und die Ausgangsgrößen der Längsdynamikmodelle $p_{ACC}, p_{Brk}, v_{Fzg}$ sowie a_{long} und mit der Abbruchbedingung von $\epsilon = 10^{-5}$ ergeben sich folgende Verläufe der Kullback-Leibler-Divergenzen.

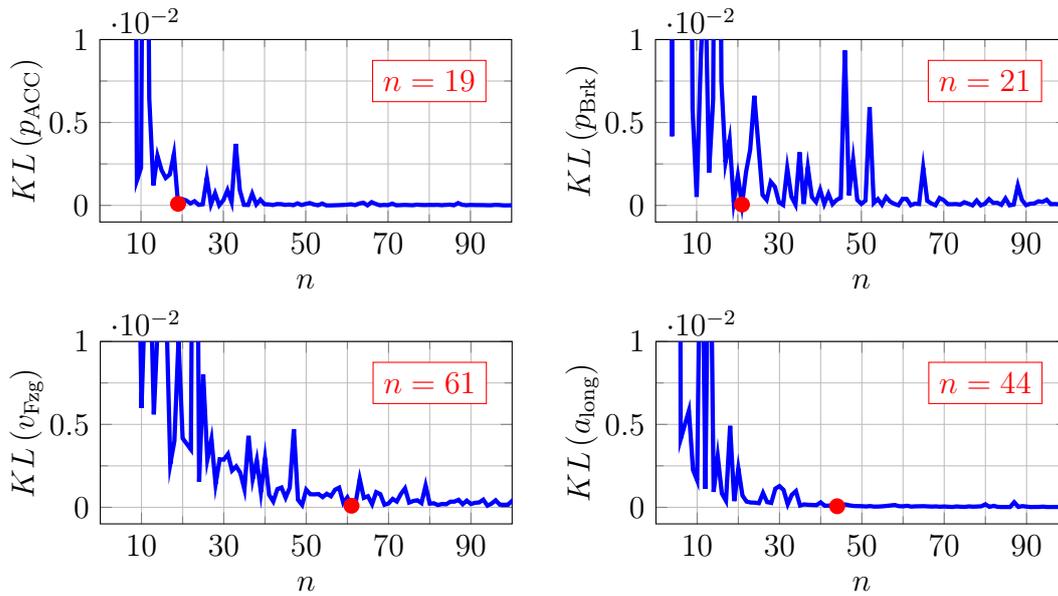


Abbildung 4.19: Kullback-Leibler-Divergenz mit der Abbruchstelle markiert als roter Punkt.

Die hier gewählte Abbruchbedingung dieses Verfahrens bestätigt somit ebenfalls eine ausreichende Menge an Daten zum Training der KNN. Außerdem wird mit $n = 61$ eine sehr geringe Anzahl an Fahrten mit einer Gesamtdauer von $t = 5.08$ h abgeschätzt. Das Verfahren berücksichtigt jedoch nicht das Zusammenspiel der einzelnen Größen untereinander und die Anwendung auf KNN ist auch nicht die ursprüngliche Verwendung. Das Verfahren kann also nur als bedingt geeignet zur Auswahl der benötigten Trainingsdatenmenge eines KNN bezeichnet werden. Das Ergebnis dieses Verfahrens ist insbesondere von der Wahl eines Wertes für die Abbruchbedingung ϵ abhängig. Somit bestimmt die Wahl eines Wertes für ϵ das Ergebnis dieses Kriteriums. Die Kernaussage dieses Verfahrens ist es, eine Aussage darüber zu treffen, ob ein neuer Datensatz weit genug vom bisher aufgenommenen Datensatz abweicht, um einen Mehrwert zu bringen. Auch wenn sich die Daten häufig wiederholen und damit die Kullback-Leibler-Divergenz unterhalb des Abbruchkriteriums liegt, kann ein weiterer Datensatz mit deutlich unterschiedlicher Verteilung folgen. Dies wird bspw. im Verlauf der Kullback-Leibler-Divergenz des Bremspedales p_{Brk} deutlich. Auch wenn der grobe Verlauf der Divergenz fallend ist, folgen nach dem Erreichen des Abbruchkriteriums noch hohe Divergenzwerte. Diese hohen Divergenzwerte weisen auf Datensätze mit einem Mehrwert für das Training der KNN durch bisher unbekannte Werte hin. Trotzdem kann das Verfahren einen quantifizierbaren statistischen Indiz über die benötigte Datenmenge liefern. Mit der Wahl eines geeigneten ϵ liegen die Ergebnisse nahe an denen des zuvor beschriebenen Trainings mit verschiedenen geringeren Datenanteilen.

Wichtiger als die reine Menge an Daten ist eine Abdeckung aller möglichen Fahrsituationen. Eine große Menge an alltäglichen Fahrten deckt alle üblichen Situationen ab. Alternativ sollte es möglich sein, mit noch geringeren Datenmengen KNN erfolgreich zu trainieren. Eine absichtliche Anregung aller möglichen längsdynamisch relevanten Situationen könnte einen noch wesentlich kleineren Datensatz ermöglichen, der bereits zum Training der KNN verwendet werden kann. Eine solche Fahrt müsste verschiedene starke Beschleunigungs- und Bremsvorgänge im gesamten Geschwindigkeitsspektrum des Fahrzeuges enthalten. Möglicherweise sind solche „künstlichen“ Daten bereits weit unterhalb 1000 km zum Training der KNN geeignet. Ein großer Datensatz alltäglicher Fahrten hat jedoch den Vorteil, dass die Anwendung der Modelle in den gleichen alltäglichen Fahrsituationen erfolgreich trainiert wird. Bei einem „künstlichen“ Datensatz besteht die Möglichkeit, dass in der Anwendung der Modelle nicht bedachte Situationen auftreten, bei denen der Erfolg zur Fähigkeit der Abstraktion nicht garantiert ist.

Die Trainingsdaten sind jedoch fehlerbehaftet. Diese gleichverteilten Fehler und Rauschen werden mit großen Mengen an Trainingsdaten ausgeglichen. Eine höhere Qualität der Daten mit geringerem Rauschen und Fehlern könnte somit die benötigte Menge an Daten weiter reduzieren.

Bewertung und Ausblick

In diesem Kapitel erfolgt eine Auswertung der verschiedenen Modelle. Insbesondere wird ein Vergleich der Korrelationen der Geschwindigkeitsprofile in den verschiedenen Testfahrten angestellt, um gemeinsame Fehler der unterschiedlichen Modelle aufzuzeigen. Anschließend erfolgt eine Bewertung der verschiedenen Modellierungsansätze sowie ein Ausblick.

5.1 Auswertung

An den Ergebnissen aller Modelle ist auffällig, dass diese alle ähnliche Ergebnisse und auch ähnliche Abweichungen zur gemessenen Geschwindigkeit aufweisen. Insbesondere zwischen dem physikalisch-datenbasierten Modell, dem tiefen KNN und dem strukturierten KNN gibt es große Gemeinsamkeiten zwischen den Ergebnissen. Um dies weiter zu verfolgen, erfolgt eine Untersuchung der Korrelation.

Zum Vergleich der verschiedenen Modelle mit den Messdaten und auch untereinander wird der Korrelationskoeffizient $r_{x,y}$ herangezogen. Sind zwei Vektoren \mathbf{x} und \mathbf{y} mit $i = 1, \dots, n$ Elementen und den arithmetischen Mitteln $\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i$ und $\bar{y} = \frac{1}{n} \sum_{i=1}^n y_i$ gegeben, ergibt sich für den Korrelationskoeffizienten folgende Beziehung.

$$r_{x,y} = \frac{C_{x,y}}{\sqrt{C_{x,x} \cdot C_{y,y}}} \quad (5.1)$$

Mit den Kovarianzen

$$C_{x,y} = \sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y}) \quad (5.2)$$

$$C_{x,x} = \sum_{i=1}^n (x_i - \bar{x})^2 \quad (5.3)$$

$$C_{y,y} = \sum_{i=1}^n (y_i - \bar{y})^2 \quad (5.4)$$

Der Korrelationskoeffizient ist eine dimensionslose Größe mit Werten zwischen -1 und 1 und ist ein Maß für den Grad des linearen Zusammenhanges der zwei Vektoren \mathbf{x} und \mathbf{y} . Ist der Zusammenhang zwischen den beiden Vektoren vollkommen positiv linear, liegt der Wert bei 1 . Bei negativer vollständiger linearer Abhängigkeit weist $r_{x,y}$ einen Wert von -1 auf. Ist jedoch keine (lineare) Abhängigkeit zwischen den beiden Vektoren vorhanden, liegt der Wert des Korrelationskoeffizienten bei 0 .

Es lassen sich nun für einzelne simulierte Fahrten sämtliche Korrelationskoeffizienten der verschiedenen Modelle und der Messdaten bestimmen und in einer Korrelationsmatrix darstellen. Der Wert des Korrelationskoeffizienten eines Vektors mit sich selbst beträgt immer 1 . Im Sachverhalt dieser Arbeit weisen hohe Werte auf hohe Korrelation der Simulationsergebnisse zweier Modelle oder eines Modells mit den Messdaten hin.

Werden nun die Korrelationskoeffizienten für die drei Testfahrten berechnet, ergeben sich folgende Korrelationsmatrizen.

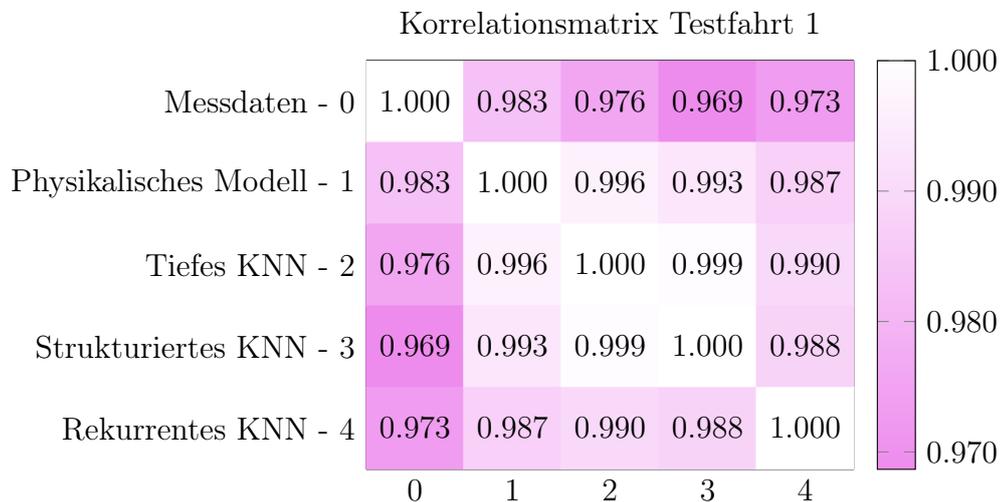


Abbildung 5.1: Korrelationsmatrix der ersten Testfahrt.

Hierbei sind für alle Koeffizienten hohe Korrelationen festzustellen. Was eine erfolgreiche Modellbildung in allen hier untersuchten Fällen bestätigt. Es wird jedoch

für alle Fahrten eine höhere Korrelation der Modelle untereinander als zu den Messdaten ermittelt. Insbesondere das tiefe KNN und das strukturierte KNN weisen einen hohen Korrelationskoeffizienten von 0.999 auf. Das strukturierte KNN kann, wenn die Zwischenausgänge im Anwendungsfall keinen Einfluss auf das Netz haben, als noch tieferes KNN angesehen werden. Hierbei ist die hohe Korrelation auf Grund der Ähnlichkeit der zugrunde liegenden Modellierung zu erwarten. Überraschend ist jedoch die hohe Korrelation des physikalisch-datenbasierten Modells zu den beiden vorwärtsgerichteten neuronalen Modellen von 0.993 und 0.996. Sogar zum rekurrenten KNN ist die Korrelation mit 0.987 größer als zu den Messdaten mit 0.983.

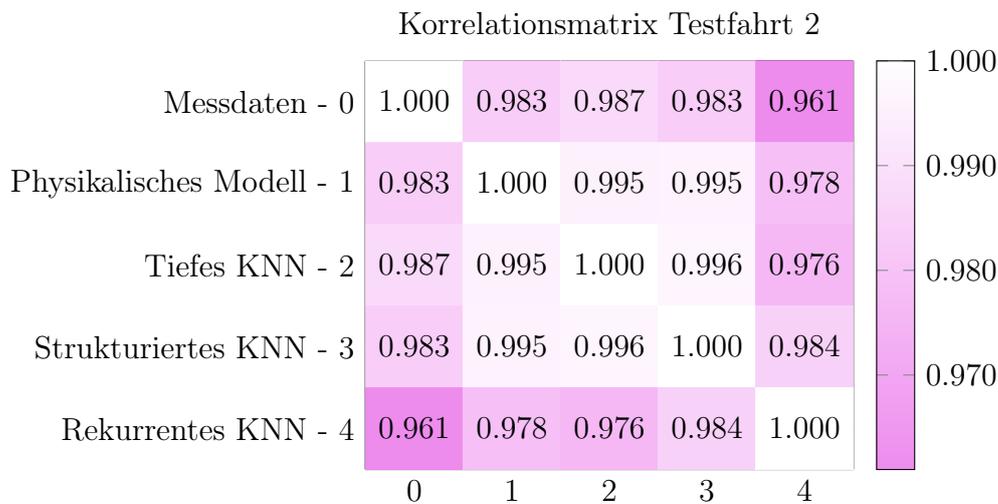


Abbildung 5.2: Korrelationsmatrix der zweiten Testfahrt.

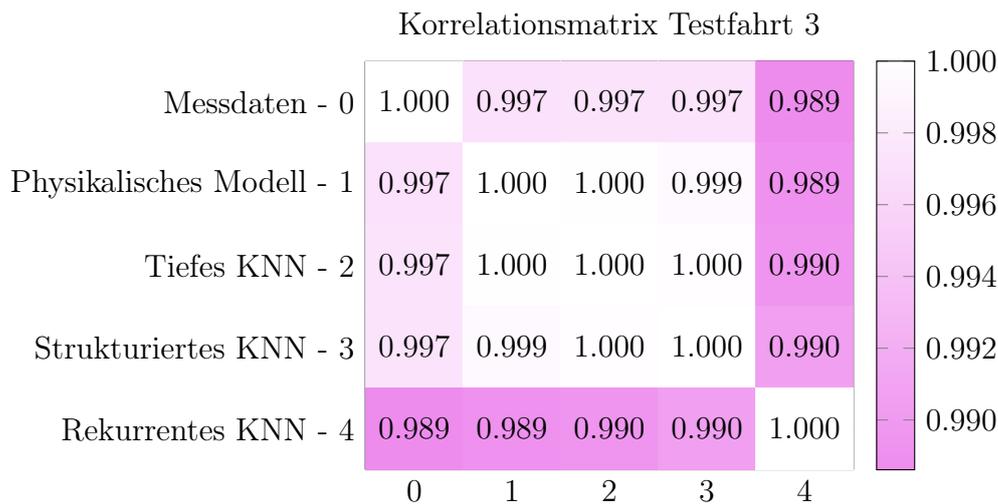


Abbildung 5.3: Korrelationsmatrix der dritten Testfahrt.

In den anderen beiden Korrelationsmatrizen bestätigt sich dieses Verhältnis. Insbesondere für das physikalisch-datenbasierte Modell und die beiden vorwärts gerichteten Modelle gibt es hohe Korrelationen. Die Modelle weisen stets höhere Korrelationen untereinander auf, obwohl die Strukturen der Modelle zum Teil vollständig unterschiedlich sind. In der dritten Testfahrt besteht sogar zwischen dem tiefen KNN und dem physikalisch-datenbasierten Modell eine größere Korrelation als zwischen den beiden neuronalen Modellen. Nur das rekurrente KNN weist zum Teil niedrigere Korrelationen zu den anderen Modellen auf. Dabei ist die Korrelation zum Messsignal jedoch stets niedriger zu den anderen Modellen.

Diese Korrelationen weisen damit auf einen gemeinsamen Fehler in allen der Modelle hin. Da diese aus mathematischer Sicht vollkommen verschieden sind, wird der Fehler in den Teilen der Modelle liegen, die alle gemein haben. Dies sind insbesondere die Datengrundlage sowie die zu Beginn getroffenen Vereinfachungen. Allen Modellen fehlt die Erfassung einiger relevanter Eingangsgrößen, Parameter oder auch Störgrößen. Vernachlässigt werden so bspw. die Veränderungen der gesamten Fahrzeugmasse durch unterschiedliche Anzahlen an Passagieren und unterschiedliche Beladungen. Ebenfalls vernachlässigt werden die unterschiedlichen Zustände der Reifen oder sogar der Einsatz unterschiedlicher Reifen. Ebenfalls nicht gemessen und somit nicht berücksichtigt sind die Fahrbahneigenschaften. All diese Vereinfachungen haben einen Einfluss auf die mit den Modellen abstrahierten Fahrwiderstandskräften. So ändert sich u.a. der Rollwiderstand mit dem Radius und Zustand des Reifens sowie beim Befahren unterschiedlicher Fahrbahnoberflächen. Alle Modelle verwenden statistische Verfahren und haben somit eine Form einer Mittlung dieser Parameter und Störgrößen vorgenommen. Bei einer Abweichung der realen Gegebenheiten von dieser Norm treten dementsprechend Fehler auf.

Insbesondere jedoch führt die Reduzierung der Eingangsdaten auf die leicht zu messenden Größen zur Vernachlässigung der Fahrwiderstandskräfte, die von anderen nicht gemessenen Größen abhängen. Hierbei sind insbesondere der Steigungswiderstand sowie die vom Wind induzierte Luftwiderstandskraft zu nennen. Die Effekte dieser Kräfte lassen sich nicht aus den Eingangsgrößen ermitteln und fehlen dementsprechend in allen im Rahmen dieser Arbeit erstellten Modellen. Die Korrelationen des rekurrenten KNN fallen hier jedoch niedriger aus, da dieses Modell zusätzlich zum ebenfalls gleichen Fehler noch dynamisches Verhalten abbilden kann. Dieses ergänzende Verhalten erklärt die zusätzlichen Unterschiede zu den anderen drei Modellen.

Aus diesem Grund lässt sich auch keine absolute Bewertung der Modelle untereinander anhand der Abweichungen zum Messsignal treffen. Ein Modell mit nahezu keinen Abweichungen ist unter den gegebenen Voraussetzungen nicht möglich und auch nicht als perfektes Modell anzusehen. Die oben verwendeten Metriken wie MSE oder RMSE, die auf den Abweichungen der Simulation zu den Messwerten basieren, können somit nicht mehr wesentlich bessere Werte als in den erstellten Modellen erreichen. Eine Optimierung der Modelle anhand dieser Größen ist unter Annah-

me eines gleichverteilten Fehlers dennoch sinnvoll. Der Steigungswiderstand kann als gleichverteilt angenommen werden, da die Höhe des Fahrzeugs über NN am Anfang und am Ende aller Testfahrten nahezu gleich ist. Somit erzeugt eine Vernachlässigung dieses Widerstandes ohne Berücksichtigung zusätzlicher Effekte einen gleichverteilten Fehler. Wenn Wind- und Fahrrichtungen sowie Windgeschwindigkeiten ebenfalls gleichverteilt sind, kann auch dieser Fehler annähernd gleichverteilt sein. Eine große Datenmenge gestattet diese Annahmen. Somit sind die abweichungsbasierten Metriken zur Optimierung der Modelle geeignet, perfekte Werte können jedoch nicht erreicht werden.

Der Umstand, dass die Korrelationen der Modelle untereinander größer sind als zum Messsignal, legt jedoch nahe, dass die Modelle sich dem Optimum angenähert haben. Wenn der Fehler aller Modelle gleich ist, und alle den gleichen Vereinfachungen in der Modellierung unterliegen, sollten sich die Modelle einem hypothetischen Verlauf des realen Fahrzeuges mit den gleichen Vereinfachungen angenähert haben.

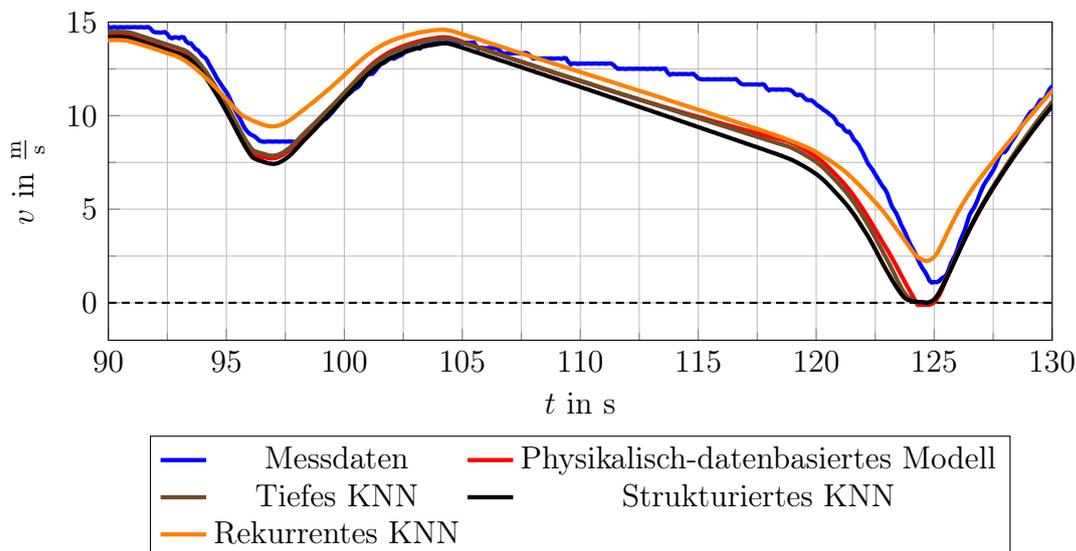


Abbildung 5.4: Abschnitt hoher Korrelation der Modelle in Testfahrt 3.

Als Beispiel dieser Annäherung der Modelle aneinander und an einen hypothetischen Verlauf des realen Fahrzeuges kann ein Abschnitt der dritten Testfahrt betrachtet werden. In der Spanne von $105\text{ s} \leq t \leq 125\text{ s}$ ergeben die Simulationen aller vier Modelle einen nahezu identischen Abfall der Geschwindigkeit (vgl. Abbildung 5.4). Das reale Messsignal legt jedoch einen langsameren Abfall der Geschwindigkeit vor. Die hohe Korrelation der vier Modelle bei gleichzeitiger Abweichung vom realen Signal kann somit wahrscheinlich mit einer Abweichung von den Modellierungsvereinfachungen erklärt werden. In diesem Fall hat möglicherweise Rückenwind und/oder eine abschüssige Fahrbahn dazu geführt, dass das Fahrzeug langsamer

abbremst. Möglicherweise fehlt den Modellen jedoch auch die dynamische Komponente dieser längsdynamischen Situation. Bei diesem Abbremsvorgang spielt bspw. die Trägheit der Fahrzeugmasse in der langsameren Bremsung eine Rolle. Drei der Modelle können diese jedoch nicht korrekt abbilden, da sie ihre Berechnungen nur auf einen Zeitschritt stützen. Eine Erklärung der Abweichungen in der hier diskutierten Zeitspanne mit der fehlenden Fähigkeit zur Abbildung dynamischer Vorgänge ist auf Grund der Ergebnisse des rekurrenten KNN jedoch unwahrscheinlich. Auch das zur Abbildung dynamischen Verhaltens fähige Modell weist an dieser Stelle eine ähnliche Abweichung auf.

Der Geschwindigkeitsverlauf sollte dementsprechend selbst bei perfekter Modellierung der Längsdynamik nur unter bestimmten Bedingungen die gleichen Werte wie die gemessene Geschwindigkeit aufweisen. Nur solange die Bedingungen, wie bspw. die Windgeschwindigkeit und die Steigung, den Vereinfachungen der Modellierung entsprechen, kann die Simulation mit den Messdaten übereinstimmen. Die Simulation der Testfahrt 2 des strukturierten KNN zeigt möglicherweise ein solches Szenario. Innerhalb der Zeitspanne von $80\text{ s} \leq t \leq 230\text{ s}$ ist die Abweichung stets kleiner als $\Delta v \leq 1.22 \frac{\text{m}}{\text{s}}$. Über einen Zeitraum von $t = 114.8\text{ s}$ ist der Fehler sogar kleiner als $\Delta v \leq 0.5 \frac{\text{m}}{\text{s}}$. Dies legt die Vermutung nahe, dass für diese Zeitspanne die wahren Gegebenheiten den Vereinfachungen der Modellierung entsprechen.

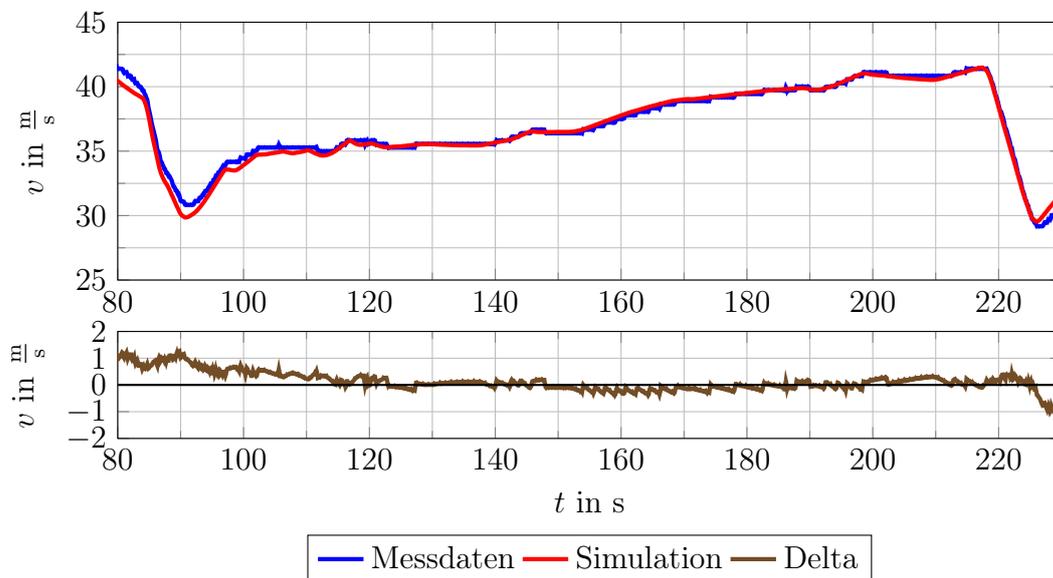


Abbildung 5.5: Abschnitt hoher Übereinstimmung in der zweiten Testfahrt des strukturierten KNN.

5.2 Bewertung der Ansätze

Klassische physikalische Längsdynamikmodelle liefern trotz großer Vereinfachungen gute Ergebnisse. Zur Modellbildung ist Wissen über das Fahrzeug nötig. Viele der benötigten Parameter sind jedoch allgemein unbekannt und müssen wenn benötigt über aufwendige Versuche ermittelt werden. Als Alternative dazu werden in dieser Arbeit weitere Ansätze vorgeschlagen und Modelle erstellt. Die Anforderungen der Ansätze reichen von rein datenbasiert ohne Wissen über das Fahrzeug bis hin zu einem klassischen physikalischen Modell, dessen fehlende Parameter über einfache Anpassungen an gemessene Daten gewonnen werden.

Das physikalisch-datenbasierte Modell zeigt die Grenzen der klassischen physikalischen Modellbildung auf. Ein Kompromiss zwischen der komplexen Realität und einfacher physikalischer Approximierung muss gewählt werden. Mit steigender Komplexität sinken die Verbesserung der Modellierung jedoch stark. Am Beispiel des in dieser Arbeit modellierten Ford C-Max Energi Plug-In Hybrid scheitert eine präzisere physikalische Modellierung insbesondere am fehlenden Wissen über die Hybridstrategie.

Die Modellierung der Längsdynamik über KNN wird als Alternative untersucht. Im Rahmen dieser Arbeit bestätigen die Ergebnisse der Modelle, dass einfache KNN eine den physikalischen Modellen ähnliche Güte liefern. Hierbei werden lediglich Daten aus fahrzeuginternen Sensoren des Serienfahrzeuges verwendet. Somit ist kein Wissen über das Fahrzeug und die Physik von Fahrzeugdynamik nötig. Bereits einfache tiefe KNN können mit dieser leicht zu gewinnenden Datenbasis erfolgreich trainiert werden. Die Güte scheint leicht hinter der des physikalisch-datenbasierten Modells zurückzubleiben, wobei wie in der Auswertung im Kapitel 5.1 erläutert eine abschließende Bewertung der Modelle nicht möglich ist. Das Fehlen wichtiger Eingangsgrößen wie der Windgeschwindigkeit führen zwangsweise zu Vereinfachungen der Modelle. Zusätzlich sind die Messdaten von geringer Güte. Somit kann bloß geschlossen werden, dass einfache tiefe KNN bereits bis nahe an das Maximum der in den Daten vorhandenen Zusammenhänge trainiert werden können. Der Ansatz über *AutoML* als Tool für automatisiertes maschinelles Lernen ermöglicht dies sogar bereits ohne viel Erfahrung im Bereich der Neuroinformatik.

Wird jedoch bei der Konstruktion des KNN zusätzlich Wissen über das Fahrzeug und beim Training zusätzliche Daten weiterer fahrzeuginterner Größen verwendet, kann ein Modell mit höherer Genauigkeit erstellt werden. Im Rahmen dieser Arbeit hat das strukturierte KNN bessere Ergebnisse in den simulierten Testfahrten geliefert als das physikalisch-datenbasierte Modell. Das zusätzlich benötigte Wissen über das Fahrzeug ist minimal, es werden lediglich mehr Daten benötigt. Solange diese ohne zusätzlichen Aufwand ebenfalls über den CAN-Bus abgerufen werden können, ist diese Vorgehensweise dem einfachen tiefen KNN vorzuziehen. Anstatt das Wissen über das Fahrzeug in Gleichungen zu formulieren und in Parametern festzuhalten, lässt sich dieses in die Struktur des KNN einbringen und sollte so ein besseres Mo-

dell liefern. Auch das in dieser Arbeit erstellt strukturierte KNN könnte um weitere relevante Zwischengrößen zu einem umfassenderen KNN erweitert werden. Als ein Beispiel wurde bereits der Ladestand mit seinem Einfluss auf die Hybridstrategie des Fahrzeuges genannt. Doch auch die Güte dieses Modells wird dadurch begrenzt, dass nicht alle relevanten Eingangsgrößen vorliegen.

Das rekurrente KNN ist ebenfalls zur Modellierung der Längsdynamik geeignet. Eine Bewertung anhand des Testdatensatzes offenbart jedoch die niedrigste Modellgüte unter den KNN. Die Fähigkeit zur Abbildung dynamischer Vorgänge ist für die Modellierung wünschenswert und führt so z.B. zu besseren Ergebnissen bei starken Bremsungen. Möglicherweise lässt sich die Güte des Modells durch weitere Hyperparameteroptimierung oder durch Einbringung von Wissen analog zum strukturierten KNN noch weiter verbessern. Letztendlich sind aber auch die Ergebnisse des rekurrenten KNN durch die Modellierungsvereinfachungen durch die fehlenden Eingangsdaten begrenzt.

Somit sind KNN als relevante Alternativen zur klassischen Modellbildung in der Fahrzeugdynamik bestätigt worden. Der Erfolg dieser trainierten Modelle hängt jedoch stark von den Trainingsdaten ab. Vom technischen Standpunkt ist das Aufnehmen dieser Daten ohne Probleme möglich, insbesondere da auf die fahrzeuginternen Sensoren moderner Fahrzeuge zurück gegriffen werden kann. In der praktischen Anwendung ist jedoch das Aufnehmen dieser Daten insbesondere in einer Vielzahl von Fahrten mit einem großen Aufwand verbunden.

Die Ergebnisse dieser Arbeit können jedoch zeigen, dass die hier verwendeten Daten aus regulären Alltagsfahrten zum Training der KNN ausreichen. Es müssen keine Fahrten mit speziellen Manövern unternommen werden um die Trainingsdaten zu generieren. Stattdessen können diese nebenbei und ohne zusätzlichen Aufwand generiert werden.

Somit ist der Ansatz KNN zur Modellierung von Längsdynamik als universeller Ansatz verwendbar. Von einem neuen zu modellierendem Fahrzeug müsste lediglich ein ausreichender Datensatz generiert werden. Die Struktur des tiefen KNN müsste nicht einmal angepasst werden und das Training des KNN würde ein funktionierendes Modell liefern. Eine mögliche Verwendung des vortrainierten KNN senkt dabei zusätzlich die benötigte Datenmenge. Physikalische Modelle müssten bspw. für andere Antriebstopologien angepasst werden, bei KNN wären diese Unterschiede irrelevant. Theoretisch sind die Modelle für jegliche Arten von Fahrzeugen anwendbar. Eine Anwendung vom Motorroller über PKW jeder Art bis hin zu LKW und Bussen ist denkbar. Somit ist der Ansatz der KNN wesentlich universeller einsetzbar als herkömmliche physikalische Modelle.

5.3 Grenzen der KNN

Da die internen Zusammenhänge der KNN nur schwer bis gar nicht nachzuvollziehen sind, ist eine Diskussion der Grenzen der KNN jedoch nötig. Das KNN ist nach erfolgreichem Training in der Lage, von gelernten Situationen beginnend die tatsächliche Dynamik des Fahrzeuges zu abstrahieren. Somit ist es in der Lage, auch unbekannte Eingänge entsprechend der gewünschten Modellierung erfolgreich zu verarbeiten. Trotzdem ist es niemals möglich die Funktion des KNN zu garantieren oder zu beweisen. Möglicherweise führen unbekannte Eingänge zu unvorhergesehenem Verhalten. Bei einem einfachen gleichungsgestützten physikalischen Modell ist das Verhalten des Modells eindeutig beschrieben. So wäre es bspw. möglich einen Wertebereich zu bestimmen. Die Werte der Ausgänge eines KNN hingegen können möglicherweise sogar unbeschränkt wachsen wenn unbekannte Eingänge vorliegen.

Als ein rudimentärer Test im Rahmen dieser Arbeit werden die KNN mit künstlichen Eingängen simuliert und deren Ergebnisse auf Plausibilität überprüft. Insgesamt werden 5 verschiedene Pedalstellungen als Eingänge für alle drei KNN geprüft. Auf Grund der Normierung der Eingangsgrößen kann auch der ursprünglich in mm gemessene Wert des Bremspedales als prozentual angesehen werden. Ein erster Test erfolgt mit $p_{ACC} = 0\%$ sowie $p_{Brk} = 0\%$ mit einem erwarteten Ausgang von $v_{Fzg} = 0 \frac{m}{s}$. Die zweite Plausibilitätsprüfung erfolgt mit $p_{ACC} = 0\%$ und $p_{Brk} = 100\%$, d.h. mit angezogener Bremse und ohne Druck auf das Gaspedal, auch hier wird eine konstante Geschwindigkeit von $v_{Fzg} = 0 \frac{m}{s}$ erwartet. Als dritte Prüfung wird ein Gaspedalwert von $p_{ACC} = 10\%$ rampenförmig innerhalb von 10 Sekunden aufgebaut und gehalten. Das Fahrzeug soll sich einem Zustand statischer Geschwindigkeit annähern. In der vierten Plausibilitätsprüfung mit $p_{ACC} = 100\%$ soll ein erfolgreich trainiertes Modelle sich der elektronisch abgeriegelten Maximalgeschwindigkeit des Ford C-Max Plug-In Hybrid von $v_{Fzg,max} = 102 \text{ mph} = 46 \frac{m}{s}$ annähern. Mit dem abschließenden fünften Test werden die Grenzen der KNN mit unbekanntem Eingängen von $p_{ACC} = p_{Brk} = 100\%$ ausgetestet.

Das tiefe KNN zeigt sowohl in der ersten als auch in der zweiten Plausibilitätsprüfung eine Annäherung an die konstante Geschwindigkeit von $v_{Fzg} = 1.98 \frac{m}{s}$. Damit weicht es von der geforderten Geschwindigkeit von $v_{Fzg} = 0 \frac{m}{s}$ ab, zeigt jedoch ein konstantes Verhalten. Die Abweichung lässt sich zum Teil auf die Fehler in den Trainingsdaten zurückführen. Auf die Fehler in den Beschleunigungsdaten wird in einem späteren Absatz genauer eingegangen. Die dritte und vierte Plausibilitätsprüfung zeigt das gewünschte Verhalten. Bei einer Gaspedalstellung $p_{ACC} = 10\%$ nähert sich das tiefe KNN einer Geschwindigkeit von $v_{Fzg} = 12.67 \frac{m}{s}$ an. Die Fahrt mit vollem Gaspedalausschlag liefert eine konstante Höchstgeschwindigkeit von $v_{Fzg,max} = 107.1 \text{ mph} = 47.9 \frac{m}{s}$, die damit nahe an der gemessenen und im Datenblatt angegebenen elektronisch abgeregelten Höchstgeschwindigkeit liegt. Im letzten Test mit gleichzeitig betätigtem Gas- und Bremspedal liefert das tiefe KNN mit einer konstanten Geschwindigkeit von $v_{Fzg} = -0.89 \frac{m}{s}$ ein plausibles Ergebnis, ebenfalls

mit einer geringen Abweichung.

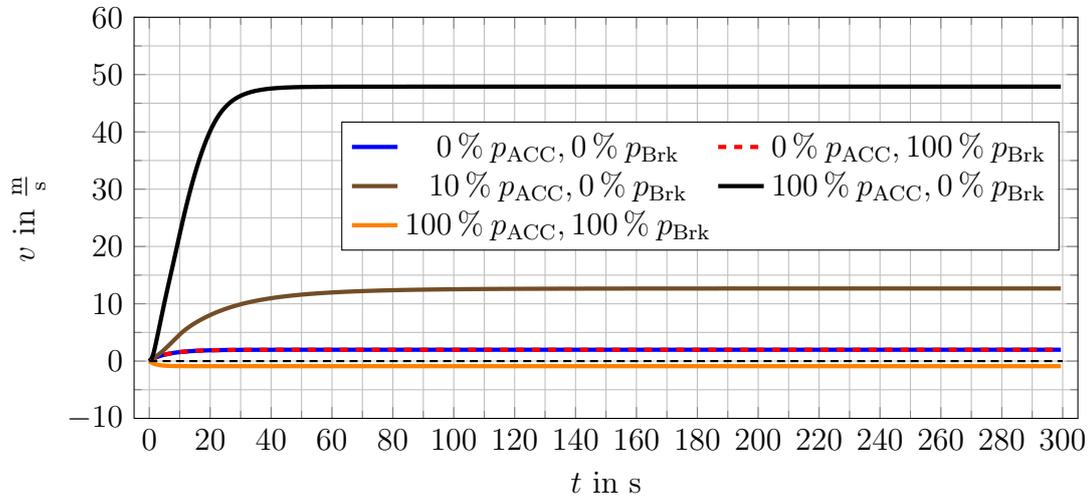


Abbildung 5.6: Simulierte Plausibilitätsprüfungen des tiefen KNN.

Für die ersten beiden Tests liefert das strukturierte KNN mit einer asymptotischen Annäherung an konstante Geschwindigkeiten von $v_{Fzg} = 1.78 \frac{m}{s}$ eine konstante Abweichung. Die beiden Plausibilitätsprüfungen mit konstanten Gaspedalstellungen liefern zwei konstante Geschwindigkeiten von $v_{Fzg} (p_{ACC} = 10\%) = 12.13 \frac{m}{s}$ und $v_{Fzg} (p_{ACC} = 100\%) = 102.6 \text{ mph} = 45.85 \frac{m}{s}$. Das strukturierte KNN liefert im Test mit gleichzeitig betätigtem Gas- und Bremspedal mit einer konstanten Geschwindigkeit von $v_{Fzg} = 0.15 \frac{m}{s}$ ein nahezu perfektes Ergebnis.

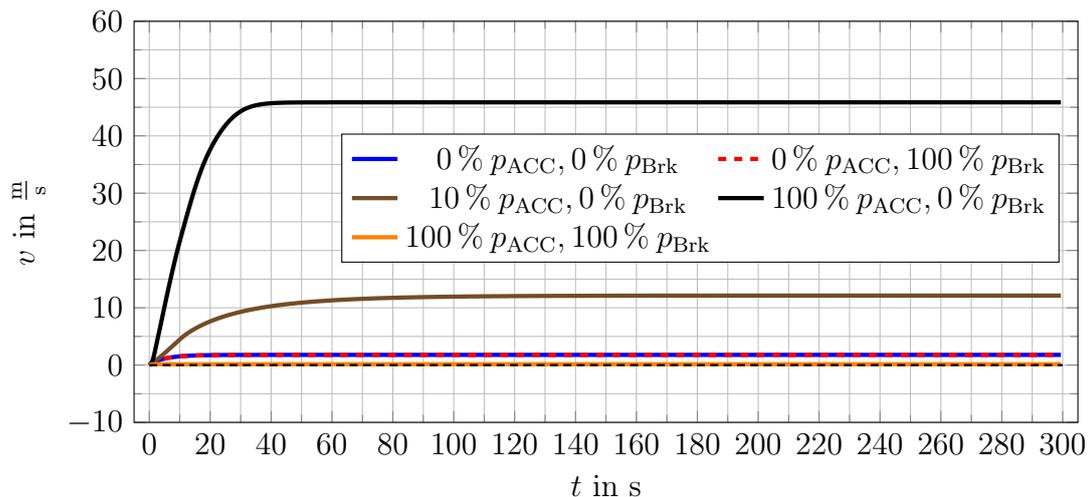


Abbildung 5.7: Simulierte Plausibilitätsprüfungen des strukturierten KNN.

Die Simulationen mit dem rekurrenten KNN liefern im Gegensatz zu den anderen KNN für die ersten beiden Überprüfungen keinen identischen konstanten Wert der sich innerhalb weniger Sekunden einstellt. Ohne Bremspedal liefert die erste Plausibilitätsprüfung eine konstante Geschwindigkeit von $v_{\text{Fzg}} = 1.75 \frac{\text{m}}{\text{s}}$. Bei Betätigung des Bremspedals verbessert sich der Wert deutlich auf einen Wert von $v_{\text{Fzg}} = 0.08 \frac{\text{m}}{\text{s}}$. Der dritte und vierte Test führt wieder analog zu den anderen KNN zu zwei konstanten Werten von $v_{\text{Fzg}}(p_{\text{ACC}} = 10\%) = 12.77 \frac{\text{m}}{\text{s}}$ und $v_{\text{Fzg}}(p_{\text{ACC}} = 100\%) = 129.8 \text{ mph} = 58.03 \frac{\text{m}}{\text{s}}$ für die Höchstgeschwindigkeit des Fahrzeuges. Der fünfte Test mit den dem KNN unbekanntem Eingangsdaten von $p_{\text{ACC}} = p_{\text{Brk}} = 100\%$ liefert mit der konstanten Geschwindigkeit von $v_{\text{Fzg}} = 5.18 \frac{\text{m}}{\text{s}}$ ein überraschend schlechtes Ergebnis.

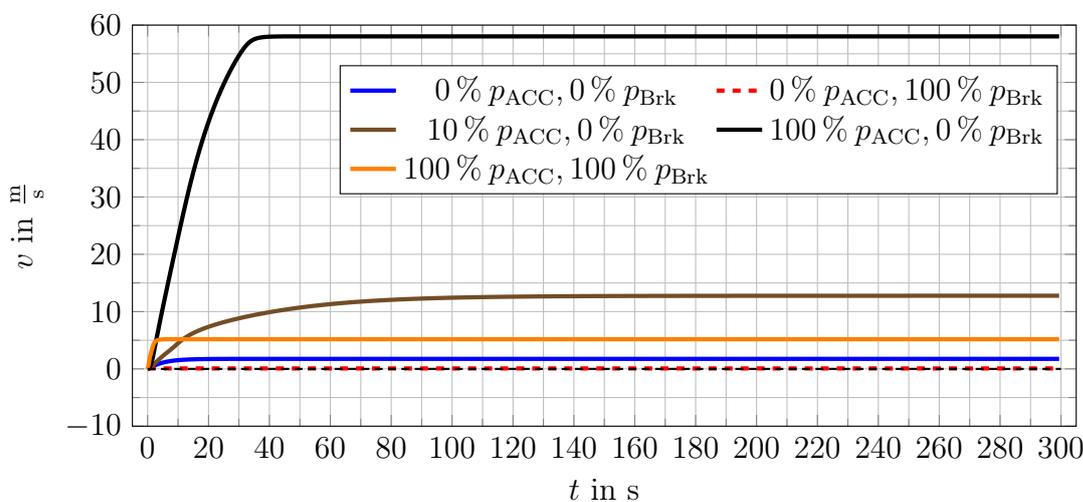


Abbildung 5.8: Simulierte Plausibilitätsprüfungen des rekurrenten KNN.

Während die in Kapitel 4 vorgestellten Ergebnisse eine erfolgreiche Modellbildung darlegen, werden aus diesen Überprüfungen Grenzen der KNN deutlich. Alle KNN haben die Annäherung an ein physikalisches Kräftegleichgewicht der Antriebskraft mit den Widerstandskräften erlernt. Die beiden vorwärts gerichteten KNN liefern jedoch konstante Abweichungen vom Stillstand des Fahrzeuges. Das LSTM scheint die geringste Abweichung vom Stillstand bei Betätigung der Bremse aufzuzeigen. Dafür hat das LSTM nicht die künstliche Begrenzung der Höchstgeschwindigkeit des Fahrzeuges auf $v_{\text{Fzg,max}} = 107.1 \text{ mph} = 47.9 \frac{\text{m}}{\text{s}}$ erlernt. Die anderen beiden KNN haben diese jedoch erlernt, auch wenn hier wieder konstante Abweichungen deutlich werden. Auf den Test mit den im Trainingsdatensatz nicht vorhandenen Fall, der gleichzeitigen vollständigen Betätigung der beiden Pedale, liefern beide vorwärtsgerichtete KNN plausible Ergebnisse. Hier liefert das LSTM jedoch falsche Werte.

Da die extremen Pedalwerte von 0% und 100% jedoch nicht häufig in den Trainingsdaten auftauchen, liefern die KNN für Werte dazwischen auch deutlich bes-

sere Ergebnisse. So liefert das tiefe KNN für die Pedalwerte von $p_{ACC} = 0\%$ und $p_{ACC} = 50\%$ eine konstante Geschwindigkeit von $v_{Fzg} = -0.06 \frac{m}{s}$.

Die Anwendung eines KNN ist somit immer mit einer gewissen Unsicherheit verbunden. Nur nach ausgiebigen Tests auch außerhalb der in den Trainingsdaten vorhandenen Eingangsgrößen ist eine Anwendung zu empfehlen. Insbesondere in sicherheitsrelevanten Anwendungsgebieten wie Fahrerassistenzsystemen ist von einem Einsatz dieser Modelle abzusehen. Mindestens müssen jedoch zusätzliche Sicherheitssysteme wie beispielsweise feste Wertebereiche die Ein- und Ausgänge der KNN bewerten und begrenzen. Sinnvolle Anwendungsgebiete sind stattdessen in Bereichen ohne große Sicherheitsrelevanz wie bspw. in Fahrsimulatoren zu suchen.

5.4 Fazit und Ausblick

Zusammenfassend lässt sich sagen, dass KNN geeignete Ansätze zur Modellierung von Fahrzeuglängsdynamik darstellen. Mit den in dieser Arbeit verwendeten Daten konnten mehrere geeignete Fahrzeuglängsdynamikmodelle für den Ford C-Max Energi Plug-in Hybrid erstellt werden. Die Güte dieser trainierten Modelle kann mit klassischer physikalischer Modellbildung mithalten oder sie unter Umständen sogar übertreffen. Generell sind die Modelle der KNN jedoch durch den Trainingsdatensatz begrenzt. Dabei ermöglichen große Datenmengen ein erfolgreiches Training bereits anhand verrauschter und fehlerbehafteter Messdaten. Dies ermöglicht die Verwendung der fahrzeuginternen Sensoren, dessen Werte über den CAN-Bus ohne großen Aufwand und Kosten aufgenommen werden können. Insbesondere jedoch werden die Ergebnisse der KNN durch das Fehlen längsdynamisch relevanter Eingangsgrößen wie bspw. der Windgeschwindigkeit und der Steigung begrenzt.

Mehrere Verbesserungspotentiale werden im Rahmen der Auswertung aufgezeigt. Eine Simulation des Batterieladestandes kann eine bessere Simulation der Fahrmodi des hybriden Antriebsstranges ermöglichen. Eine Strukturierung des rekurrenten KNN mit zusätzlichen fahrzeuginternen Größen in einer Zwischenausgangsschicht gilt es zu untersuchen. Insbesondere jedoch sollte untersucht werden, ob zusätzliche Größen aufgenommen werden können. Eine Möglichkeit zur Aufnahme der relativen Luftgeschwindigkeit, als Summe der Fahrzeug- und Windgeschwindigkeit, bietet eine Staudrucksonde. Die Steigungswerte der befahrenen Straßen könnten möglicherweise über ein aufgezeichnetes GPS-Signal und der Hilfe von 3D-Kartenmaterial erzeugt werden. Alternativ kann die Verwendung eines Gyroskopes diese Werte liefern. Ein externer Beschleunigungssensor oder eine Erhöhung der Messfrequenz kann zu einer Verbesserung der Datenqualität führen. Die Ergebnisse dieser Arbeit legen nahe, dass insbesondere die zusätzlichen Eingangsdaten zu einer verbesserten Modellbildung führen.

All diese Verbesserungen würden den Aufwand der Modellbildung und insbesondere der Datenaufnahme erhöhen. Aus diesem Grund soll der Mehrwert der einzelnen

Verbesserungen bestimmt und anhand des Verhältnisses zwischen Aufwand und Nutzen bewertet werden. Eine präzisere und umfassendere Datenaufnahme ist jedoch zum Zwecke der abschließenden Bewertung der einfacheren Modelle dieser Arbeit zu empfehlen. Möglicherweise liefert die Erhöhung des Aufwandes durch zusätzliche Messgrößen nicht die Verbesserungen in den gewünschten Größenordnungen. Trotzdem könnten diese Verbesserungen durch eine weitere Reduzierung der benötigten Datenmenge einen gewünschten Mehrwert bringen. Die vollständige Modellierung der Längsdynamik eines Fahrzeuges innerhalb von nur einer Stunde Fahrtzeit sollte mit diesen Verbesserungen im Rahmen des Möglichen liegen. Eine weitere zu untersuchende Verbesserungsmöglichkeit wäre die Kombination oder Fusion der Ausgaben verschiedener Modelle.

Die Verbesserungen durch den erhöhten Aufwand könnten den Sinn der gesamten Einfachheit des universellen datenbasierten Ansatzes jedoch unterminieren. Zumindest verschiebt sich die Rahmenbedingung dieses Ansatzes von einem geeigneten Ansatz mit geringen Aufwand hin zu einer komplexeren Lösung mit hohem Aufwand, dessen Vorteil bspw. in der geringen benötigten Zeit zur Modellierung liegen könnte. Je nach Anwendungsfall sind die Modelle, die im Rahmen dieser Arbeit auf Basis KNN mit verringertem Aufwand gegenüber dem physikalischen Modell erstellt wurden, bereits sehr gut geeignet.

Literaturverzeichnis

- [1] Martín Abadi u. a. “Tensorflow: Large-Scale Machine Learning on Heterogeneous Distributed Systems”. In: *arXiv preprint arXiv:1603.04467* (2016).
- [2] Christopher M Bishop. *Pattern Recognition and Machine Learning*. springer, 2006.
- [3] François Chollet u. a. *Keras*. <https://github.com/fchollet/keras>. 2015.
- [4] Wikipedia contributors. *Neuron — Wikipedia, The Free Encyclopedia*. 12. Okt. 2020. URL: <https://en.wikipedia.org/wiki/Neuron>.
- [5] Rüdiger Cordes und Ulven. *cw-Werte*. 12. Okt. 2020. URL: <http://rc.opelgt.org/indexcw.php>.
- [6] Ford. *2014 Ford C-Max Energi Technical Specifications*. 26. Okt. 2020. URL: https://media.ford.com/content/dam/fordmedia/North%5C%20America/US/2015_Specs/2014_CMAXEnergi_Specs.pdf.
- [7] Google. *Cloud AutoML*. 20. Okt. 2020. URL: <https://cloud.google.com/automl>.
- [8] Sepp Hochreiter und Jürgen Schmidhuber. “Long Short-Term Memory”. In: *Neural computation* 9.8 (1997), S. 1735–1780.
- [9] Sebastian S James, Sean R Anderson und Mauro Da Lio. “Longitudinal Vehicle Dynamics: A Comparison of Physical and Data-Driven Models Under Large-Scale Real-World Driving Conditions”. In: *IEEE Access* 8 (2020), S. 73714–73729.
- [10] Haifeng Jin, Qingquan Song und Xia Hu. “Auto-Keras: An Efficient Neural Architecture Search System”. In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 2019, S. 1946–1956.
- [11] Diederik P Kingma und Jimmy Ba. “Adam: A Method for Stochastic Optimization”. In: *arXiv preprint arXiv:1412.6980* (2014).

- [12] Kullback und Leibler. “On Information and Sufficiency”. In: *Ann. Math. Statist* 22 (1951), S. 76–86.
- [13] Warren S McCulloch und Walter Pitts. “A Logical Calculus of the Ideas Immanent in Nervous Activity”. In: *The bulletin of mathematical biophysics* 5.4 (1943), S. 115–133.
- [14] Sebastian Reicherts und Dieter Schramm. “Parameter Identification of Vehicle Dynamics Models using CAN Communication on Real-life Driving Data”. In: *AmE 2020-Automotive meets Electronics; 11th GMM-Symposium*. VDE. 2020, S. 1–6.
- [15] Andreas Schamel u. a. “Ford C-Max Mit Plug-In-Hybridtechnik”. In: *MTZ-Motortechnische Zeitschrift* 74.3 (2013), S. 186–193.
- [16] Dieter Schramm, Manfred Hiller und Roberto Bardini. *Modellbildung und Simulation der Dynamik von Kraftfahrzeugen*. Berlin Heidelberg New York: Springer-Verlag, 2013. ISBN: 978-3-642-33888-5.
- [17] Kullback und Solomon. *Information Theory and Statistics*. Courier Corporation, 1997.
- [18] Wenshuo Wang, Chang Liu und Ding Zhao. “How Much Data Are Enough? A Statistical Approach With Case Study on Longitudinal Driving Behavior”. In: *IEEE Transactions on Intelligent Vehicles* 2.2 (2017), S. 85–98.

DuEPublico

Duisburg-Essen Publications online

UNIVERSITÄT
DUISBURG
ESSEN

Offen im Denken

ub | universitäts
bibliothek

Dieser Text wird über DuEPublico, dem Dokumenten- und Publikationsserver der Universität Duisburg-Essen, zur Verfügung gestellt. Die hier veröffentlichte Version der E-Publikation kann von einer eventuell ebenfalls veröffentlichten Verlagsversion abweichen.

DOI: 10.17185/duepublico/74034

URN: urn:nbn:de:hbz:464-20210216-160429-7

Alle Rechte vorbehalten.