



Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Bachelorarbeit

Frederic Hoops

Leistungsmaximierung eines Windparks mit Hilfe eines digitalen
Zwillings sowie Methoden des maschinellen Lernens

*Fakultät Technik und Informatik
Department Maschinenbau und Produktion*

*Faculty of Engineering and Computer Science
Department of Mechanical Engineering and
Production Management*

Frederic Hoops

**Leistungsmaximierung eines Windparks mit Hilfe
eines digitalen Zwillings sowie Methoden des
maschinellen Lernens**

Bachelorarbeit eingereicht im Rahmen der Bachelorprüfung

im Studiengang Bachelor of Science Maschinenbau - Entwicklung und Konstruktion
am Department Maschinenbau und Produktion
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Erstprüfer: Prof. Peter Dalhoff

Zweitprüferin: Prof. Dr. Sarah Hallerberg

Eingereicht am: 29.09.2022

Zusammenfassung

Frederic Hoops

Thema der Bachelorthesis

Leistungsmaximierung eines Windparks mit Hilfe eines digitalen Zwillings sowie Methoden des maschinellen Lernens

Stichworte

Windparkregelung, Leistungsmaximierung, digitaler Zwilling, maschinelles Lernen

Kurzzusammenfassung

In der vorliegenden Bachelorthesis wird untersucht, inwieweit eine Leistungsmaximierung eines Windparks anhand eines digitalen Zwillings und Methoden des maschinellen Lernens, speziell künstlichen neuronalen Netzen, stattfinden kann. Ein digitaler Zwilling des Windparks, der die Leistung von diesem in Abhängigkeit von realen Umgebungsparametern berechnen kann, wird in Python entwickelt und anhand realer Windpark-Daten validiert. Für Daten aus dem Jahr 2019 wichen die berechneten Einzelleistungen im Mittel um 24,05 % von den realen ab. Für die Gesamtwindparkleistung beträgt die mittlere Abweichung 17,27 %. Es folgt die Entwicklung eines künstlichen neuronalen Netzes, in welchem der digitale Zwilling als sog. Kostenfunktion integriert wird sowie dessen Validierung anhand von Test-Datensätzen. Es konnte bewiesen werden, dass das entwickelte Netz lernfähig ist. Für vereinzelnde Datensätze konnte bereits eine maximale Leistungserhöhung zwischen 4,63 % und 11,78 % (für verschiedene Netz-Parameter) erzielt werden, im Mittel allerdings noch nicht.

Frederic Hoops

Title of the paper

Maximizing the performance of a wind farm using a digital twin and machine learning methods

Keywords

wind farm control, performance maximization, digital twin, machine learning

Abstract

This bachelor thesis investigates the extent to which a wind farm's performance can be maximized using a digital twin and machine learning methods, specifically artificial neural networks. A digital twin of the wind farm that can calculate its performance as a function of real environmental parameters is developed in Python and validated against real wind farm data. For 2019 data, the calculated individual power deviated from the real one by 24,05 % on average. For the total wind farm output, the mean deviation is 17,27 %. This is followed by the development of an artificial neural network, in which the digital twin is integrated as a so-called cost function, as well as its validation using test data sets. It could be proven that the developed network is capable of learning. For isolated data sets, a maximum performance increase between 4,63 % and 11,78 % (for different network parameters) could already be achieved, but not yet on average.

A U F G A B E N S T E L L U N G

der Bachelorthesis von
Frederic Hoops, [REDACTED]

Leistungsmaximierung eines Windparks mit Hilfe eines digitalen Zwillings sowie Methoden des maschinellen Lernens

Stichworte

Windparkregelung, Leistungsmaximierung, digitaler Zwilling, maschinelles Lernen

Hintergrund der Bachelorthesis

Der Übergang von der nicht-nachhaltigen Nutzung von fossilen Energieträgern sowie der Kernenergie zu einer nachhaltigen Energieversorgung mittels erneuerbarer Energien stellt heutzutage eine große Herausforderung dar. Die Nutzung von in Wind gespeicherter Energie findet dabei bereits seit Jahren Anwendung.

Dabei ist zu beobachten, dass aufgrund von begrenztem Platz Windenergieanlagen immer dichter beieinander stehen müssen. Diese beeinflussen sich in Folge dessen gegenseitig bezüglich Strömungsnachläufen (*Wakes*), weshalb es notwendig ist, einzelne Anlagen in einem Windpark in der Planungsphase bezüglich des Farm-Layouts sinnvoll zu positionieren und in der Betriebsphase „intelligent“ zu steuern, sodass eine Leistungs- bzw. Ertragsmaximierung des gesamten Parks stattfinden kann. Ertragsmindernde Faktoren wie z.B. Artenschutz oder Schallminimierung sind hierbei problematisch, müssen aber ebenfalls berücksichtigt werden.

Ziel- und Aufgabenstellung

Ziel dieser Bachelorarbeit ist es, den Windpark Curslack, bestehend aus fünf Anlagen, so optimiert zu steuern, dass eine Leistungsmaximierung des gesamten Parks erzielt werden kann. Dabei soll alternativ zu einem konventionellen physikalischen Ansatz mit Strömungssimulationen getestet werden, inwiefern ein digitaler Zwilling des Windparks mit Methoden des automatisierten maschinellen Lernens kombiniert werden kann. Grundlage der Arbeit sind dabei Messdaten des Windparks Curslack.

Die im Rahmen der Arbeit generierten Ergebnisse sind umfassend und stringent zu dokumentieren. Die Ausführungen schließen mit einer ausführlichen Diskussion samt Ausblick auf mögliche weitere Umsetzungsmaßnahmen.

Inhaltsverzeichnis

Abkürzungsverzeichnis	III
Symbolverzeichnis	IX
Tabellenverzeichnis	XI
Abbildungsverzeichnis	XIII
1 Einleitung	1
1.1 Motivation und Ziel der Arbeit	1
1.2 Struktur und Aufbau der Arbeit	2
1.3 Windpark Curslack	3
2 Theoretische Grundlagen	7
2.1 Leistungs- und Schubbeiwert	7
2.2 Axialer Induktionsfaktor	9
2.3 Turbulenzintensität	10
2.4 Analytische Modellierung der Nachlaufströmung	10
2.4.1 Jensen-Modell	11
2.4.2 Frandsen-Modell	12
2.4.3 Jensen-2D-k-Modell	13
2.4.4 Jensen-Gauss-Modell	14
2.4.5 Bastankhah-Modell	15
2.4.6 Ishihara-Qian-Modell	16
2.4.7 Weitere Ansätze	17
2.4.8 Zusammenfassung und Vergleich der Nachlaufmodelle	17
2.5 Modellierung von Mehrfachabschattungen	18
2.5.1 Prinzip der Superposition	18
2.5.2 Ansatz nach Qian-Ishihara	19
2.5.3 Rotorbasiertes Multiple-Wake-Modell für Windparks nach Qian-Ishihara	21
2.5.4 Weitere Verfahren	26
2.6 Koordinatentransformation	26
2.7 Neuronale Netze	27
2.7.1 Aufbau eines Neurons in einem neuronalen Netz	27
2.7.2 Stochastisches Gradientenabstiegsverfahren	29
2.7.3 Fehlerrückführung	30
3 Stand der Optimierungstechniken	33
4 Entwicklung eines digitalen Zwillings des Windparks	37
4.1 Annahmen	37

Inhaltsverzeichnis

4.2	Erzeugung der Anlagenobjekte	38
4.3	Leistungs- und Schubbeiwerte	39
4.4	Berücksichtigung der Windrichtung	42
4.5	Erstellung von Datenpunkten	44
4.6	Sukzessive Berechnung des Nachlaufes	45
4.7	Berechnung der Leistungen	47
4.8	Modifikation der Korrekturterm-Berechnung	48
4.9	Zusammenfassender Algorithmus	49
5	Filterung der Daten	51
5.1	Betriebszustand	51
5.2	Schallbetriebsart	51
5.3	Abschattung des Messmastes	51
6	Validierung des digitalen Zwillings	53
6.1	Vergleich der Leistungen für ausgewählte Fälle	54
6.1.1	Fall 1	54
6.1.2	Fall 2	58
6.2	Relative Abweichungen für ausgewählte Fälle	61
6.2.1	Fall 1	61
6.2.2	Fall 2	63
6.2.3	Modifikation der relativen Abweichungen	65
6.3	Relative Abweichungen in Bezug auf Windrichtung	67
6.4	Abweichung der Gesamtleistung des Windparks	69
6.5	Untersuchung des Einflusses der Varianz der Dichte	69
6.6	Untersuchung des Einflusses des Höhenwindgradienten	70
7	Entwicklung eines künstlichen neuronalen Netzes	73
7.1	Skalieren der Eingangsparameter	73
7.2	Verwendete Aktivierungsfunktion	73
7.3	Digitaler Zwilling als Kostenfunktion	74
7.4	Implementierung in Python	76
8	Validierung des neuronalen Netzes	79
9	Zusammenfassung	85
10	Kritische Reflexion und Ausblick	87
	Literaturverzeichnis	89
A	Datenträger	i
	Eigenständigkeitserklärung	iii

Abkürzungsverzeichnis

Abk.	Bezeichnung
ANN	Artificial neural network (de: Künstliches neuronales Netz)
AWG	Ausschaltwindgeschwindigkeit
CFD	Computational fluid dynamics (de: Numerische Strömungsmechanik)
DWM	Dynamic wake meandering (de: Dynamisches Mäandern des Nachlaufes)
EWG	Einschaltwindgeschwindigkeit
HAW	Hochschule für Angewandte Wissenschaften
HAWT	Horizontal axis wind turbine (de: Windenergieanlage mit horizontaler Achse)
IDE	Integrated development environment (de: Integrierte Entwicklungsumgebung)
IEC	International Electrotechnical Commission
JIT	Just in time (de: gerade rechtzeitig)
LES	Large Eddy Simulation (de: Grobstruktursimulation)
LiDAR	Light detection and ranging (de: Lichterkennung und Reichweitenmessung)
MM	Messmast
NWG	Nennwindgeschwindigkeit
RMSE	Root mean square error (de: Wurzel des mittleren quadratischen Fehlers)
WEA	Windenergieanlage

Symbolverzeichnis

Griechische Symbole

α	Maß für die Ausbreitung des Nachlaufes im Frandsen-Modell, Windrichtung als Eingabe im digitalen Zwilling
α_o	Obere Grenze einer Windrichtung
α_u	Untere Grenze einer Windrichtung
$\alpha_{i \rightarrow j}$	Windrichtung, bei der Anlage i Anlage j abschattet
β	Verhältnis zweier Windgeschwindigkeiten, Konstante im Frandsen-Modell, Korrigierte Windrichtung als Eingabe im digitalen Zwilling
$\Delta\sigma_{u,i,Korr}$	Vektor für alle Korrekturterme der Änderungen der Standardabweichung der Windgeschwindigkeit für Anlage i
δ^l	Vektor aller Fehler in der l -ten Schicht
$\nabla_a C$	Vektor aller partiellen Ableitungen der Kostenfunktion nach den Ausgängen des neuronalen Netzes
$\sigma_{u,j,alldp}$	Array der Standardabweichungen der Windgeschwindigkeit $\sigma_{u,j}$ für alle Datenpunkte
F	Allgemeine Funktion
$\Delta\sigma_u$	Änderung der Standardabweichung der Windgeschwindigkeit
$\Delta\sigma_{u,i,Korr}$	Korrekturterm der Änderung der Standardabweichung der Windgeschwindigkeit für Anlage i
$\Delta\sigma_{u,i,tip}$	Änderung der Standardabweichung der Windgeschwindigkeit an der „tip side“ von Anlage i
$\Delta\sigma_{u,ij,sum}$	Summe aller $(\Delta\sigma_{u,ij} + \Delta\sigma_{u,i,Korr})^2$
$\Delta\sigma_{u,ij}$	Änderung der Standardabweichung der Windgeschwindigkeit vor Anlage j , erzeugt von Anlage i
ΔI_a	Korrekturterm der Turbulenzintensität im Ishihara-Modell
Δu	Windgeschwindigkeitsdefizit
$\Delta u_{i,w}$	Windgeschwindigkeitsdefizit im Nachlauf, erzeugt von Anlage i
$\Delta u_{ij,sum}$	Summe aller Δu_{ij}^2
Δu_{ij}	Windgeschwindigkeitsdefizit vor Anlage j , erzeugt von Anlage i
$\Delta_{rel,ann}$	Relative Abweichung zwischen optimaler und „ $a = \frac{1}{3}$ “-Gesamtleistung des Windparks
$\Delta'_{rel,Dichte}$	Modifizierte relative Abweichung zwischen realer und berechneter Leistung einer Anlage für die Dichte-Untersuchung
$\Delta'_{rel,Gradient}$	Modifizierte relative Abweichung zwischen realer und berechneter Leistung einer Anlage für die Höhenwindgradient-Untersuchung
$\Delta'_{rel,ref}$	Referenzwert für die modifizierte relative Abweichung zwischen realer und berechneter Leistung einer Anlage
Δ_{rel}	Relative Abweichung zwischen realer und berechneter Leistung einer Anlage

Symbolverzeichnis

Δ'_{rel}	Modifizierte relative Abweichung zwischen realer und berechneter Leistung einer Anlage
δ_j^l	Fehler des j -ten Neurons in der l -ten Schicht
η	Lernrate des neuronalen Netzes
η_{CP}	Faktor zwischen realem und theoretischem Leistungsbeiwert
η_{CT}	Faktor zwischen realem und theoretischem Schubbeiwert
γ	Höhenwindexponent nach Hellmann
μ_α	Mittelwert aller Windrichtungen
Φ	Relative Luftfeuchte
σ	Standardabweichung der Windgeschwindigkeit, Breite des Nachlaufes
σ_α	Standardabweichung aller Windrichtungen
$\sigma_{u,0}$	Standardabweichung der Windgeschwindigkeit in der Umgebung
$\sigma_{u,j}$	Standardabweichung der Windgeschwindigkeit vor Anlage j
σ_i	Breite des Nachlaufes der Anlage i
ε	Breite des Nachlaufes bei $x = 0$
ϱ	Dichte
ϱ'_{Luft}	Reale Dichte der Luft
P_{sum}	Gesamtleistung des Windparks

Lateinische Symbole

dp	Array aller erzeugten Datenpunkte
A	Array aller Anlagenobjekte
a	Vektor der Induktionsfaktoren
b'	Aktualisierte Schwellenwerte im neuronalen Netz
b	Vektor der Schwellenwerte des neuronalen Netzes
C	Vektor aller Werte der Kostenfunktion des neuronalen Netzes
I_a	Array der Turbulenzintensitäten vor allen Anlagen im Windpark
P	Array der erzeugten Leistungen aller Anlagen im Windpark
Q	Matrix aller relativen Abstände der Anlagen zueinander
u	Array der Geschwindigkeiten vor allen Anlagen im Windpark
u_{j,alldp}	Array der Geschwindigkeiten u_j für alle Datenpunkte
v'	Allgemeiner aktualisierter Vektor
v	Allgemeiner Vektor
w'	Aktualisierte Gewichtungen im neuronalen Netz
w	Matrix der Gewichtungen des neuronalen Netzes
X	Zufällig generierte Menge an Datensätzen
x	Vektor der Eingänge des neuronalen Netzes
x_i	Koordinatenvektor der Anlage i
x_{i,rot}	Gedrehter Koordinatenvektor der Anlage i
y	Vektor der Ausgänge des neuronalen Netzes
y_D	Vektor der gewünschten Ausgänge des neuronalen Netzes
ⁱr	Vektor im i -System
^jp_i	Position des Ursprungs vom i -System im Bezug auf das j -System

${}^j \mathbf{R}_i$	Rotationsmatrix zwischen System i und System j
${}^j \mathbf{r}$	Vektor im j -System
${}^j \mathbf{T}_i$	Transformationsmatrix zwischen System i und System j
A	Fläche
a	Induktionsfaktor einer Anlage, Konstante im Ishihara-Modell
A_1	Fläche vor dem Rotor
A_2	Fläche hinter dem Rotor
A_R	Fläche eines Rotors
a_j^l	Ausgang des j -ten Neurons in der l -ten Schicht
$A_{F,1}$	Menge von Koordinaten, die den „full wake“ definieren
$A_{F,2}$	Menge von Koordinaten, in denen der Korrekturterm angewendet wird („full wake“)
$A_{P,1}$	Menge von Koordinaten, die den „partial wake“ definieren
$A_{P,2}$	Menge von Koordinaten, in denen der Korrekturterm angewendet wird („partial wake“)
$A_{R,j}$	Rotorfläche der Anlage j
b	Konstante im Ishihara-Modell
b_j^l	Schwellenwert des j -ten Neurons in der l -ten Schicht
C	Kostenfunktion des neuronalen Netzes
c	Konstante im Ishihara-Modell
$c_{P,real}$	Realer Leistungsbeiwert
$c_{P,theo}$	Theoretisch möglicher Leistungsbeiwert bei $a = \frac{1}{3}$
c_P	Leistungsbeiwert
$c_{T,real}$	Realer Schubbeiwert
$c_{T,theo}$	Theoretischer Schubbeiwert bei $a = \frac{1}{3}$
C_{Test}	Testset-Cost
C_{Train}	Trainingsset-Cost
c_T	Schubbeiwert
$c_{T,i}$	Schubbeiwert der Anlage i
d	Konstante im Ishihara-Modell, Durchmesser
d_R	Rotordurchmesser der Anlage
$d_{R,i}$	Rotordurchmesser der Anlage i
$d_{w,i}$	Durchmesser des Nachlaufes, erzeugt von Anlage i
$d_{w,k}$	Durchmesser des Nachlaufes, erzeugt von Anlage k
e	Konstante im Ishihara-Modell
E_{kin}	Kinetische Energie
f	Konstante im Ishihara-Modell
F_s	Schubkraft auf einen Rotor
h_{Nabe}	Nabenhöhe der Anlage
h_n	Relative Häufigkeit
$h_{Nabe,i}$	Nabenhöhe der Anlage i
$h_{Nabe,j}$	Nabenhöhe der Anlage j

Symbolverzeichnis

I	Turbulenzintensität
I_+	Induzierte Turbulenzintensität eines Rotors
I_0, I_a	Turbulenzintensität in der Umgebung
I_{wake}	Turbulenzintensität im Nachlauf einer Anlage
$I_{a,i}$	Turbulenzintensität der Umgebung vor Anlage i
$I_{a,j,\text{equiv}}$	Äquivalente Turbulenzintensität vor Anlage j
$I_{a,j}$	Turbulenzintensität der Umgebung vor Anlage j
k, k_{wake}, k^*	Wachstumsrate der Nachlaufströmung
k_1, k_2	Konstante im Ishihara-Modell
K_n	Empirische Konstante im Frandsen-Modell
$L_{w,ki}$	Vertikaler Abstand der Schnittpunkte der sich überschneidenden Nachläufe
m	Masse
P	Leistung
p	Luftdruck
P_0	In der freien Windströmung enthaltene Leistung
$P_{1/3}$	Gesamtleistung des Windparks, wenn alle Anlagen bei $a = \frac{1}{3}$ betrieben werden
P_1	Leistung vor einem Rotor
P_2	Leistung hinter einem Rotor
P_b	berechnete Leistung
P_{mech}	mechanische Leistung eines Rotors
P_n	Nennleistung der Anlage
P_{opt}	Optimale Gesamtleistung des Windparks
P_r	reale Leistung
p_w	Dampfdruck
P_j	Leistung der Anlage j
r	Radius
r_1, r_x	Charakteristische Radien in der Berechnung des Jensen- und Jensen-2d-k-Modells
R_L	Gaskonstante von Luft
r_r	Radius des Rotors
R_W	Gaskonstante von Wasserdampf
r_w	Radius des Nachlaufes
r_i	Radius des Nachlaufes der Anlage i
T	Temperatur der Luft
u	Geschwindigkeit
u^*	Windgeschwindigkeit als Zwischenschritt im Jensen-2D-k-Modell
u_∞	Windgeschwindigkeit der Umgebung
u'_∞	Korrigierte Umgebungswindgeschwindigkeit durch Varianz der Dichte / Höhenwindgradient
u_1	Windgeschwindigkeit vor einem Rotor, Windgeschwindigkeit in der Höhe h_1
u_2	Windgeschwindigkeit hinter einem Rotor, Windgeschwindigkeit in der Höhe h_2
u_{in}	Einschaltwindgeschwindigkeit der Anlage
u_n	Nennwindgeschwindigkeit der Anlage

u_{out}	Ausschaltwindgeschwindigkeit der Anlage
u_{R}	Windgeschwindigkeit in der Rotorebene
u_{w}	Windgeschwindigkeit in der Nachlaufströmung
u_{115}	Windgeschwindigkeit in einer Höhe von $z = 115$ m
$u_{64,8}$	Windgeschwindigkeit in einer Höhe von $z = 64,8$ m
$u_{i,\text{w}}$	Windgeschwindigkeit im Nachlauf, erzeugt von Anlage i
u_i	Windgeschwindigkeit vor Anlage i
$u_{j,\text{equiv}}$	Äquivalente Windgeschwindigkeit vor Anlage j
u_j	Windgeschwindigkeit vor Anlage j
v_1, v_2	Allgemeine Variablen
w_{jk}^l	Gewichtung vom k -ten Neuron in der $(l - 1)$ -ten Schicht zum j -ten Neuron in der l -ten Schicht
z_0	Bodenrauigkeit
z_j^l	Summierte Gewichtung des j -ten Neurons in der l -ten Schicht
\mathbf{R}_x	Rotationsmatrix der Elementardrehung um die x -Achse
\mathbf{R}_y	Rotationsmatrix der Elementardrehung um die y -Achse
\mathbf{R}_z	Rotationsmatrix der Elementardrehung um die z -Achse
Anderes	
$\mathbf{0}$	Nullvektor
\dot{x}	Zeitliche Ableitung einer Größe x
$\frac{dx}{dy}$	Ableitung von x nach y
$\frac{d}{dt}$	Ableitung nach der Zeit
$\frac{\partial x}{\partial y}$	Partielle Ableitung von x nach y
∇x	Gradient von x
\bar{x}	Mittelwert einer Größe x
\tilde{x}	Skalierter Wert von x mit Mittelwert 0 und Varianz 1
ξ	Substitutionsvariable
x, y, z	Koordinaten
x^{T}	Transponieren einer Größe x

Tabellenverzeichnis

1.1	Alle vorhandenen Sensoren des Messmasts inklusive ihrer Höhe und Ausrichtung	5
6.1	Gemittelte relative Abweichungen $\bar{\Delta}_{\text{rel}}$ für $\bar{u}_{\infty} = 8,51 \frac{\text{m}}{\text{s}}$, $\bar{\alpha} = 294,55^{\circ}$, $\bar{\sigma}_{u,0} = 1,26 \frac{\text{m}}{\text{s}}$ über 666 Datensätze (Fall 1)	63
6.2	Gemittelte relative Abweichungen $\bar{\Delta}_{\text{rel}}$ für $\bar{u}_{\infty} = 7,4 \frac{\text{m}}{\text{s}}$, $\bar{\alpha} = 294,38^{\circ}$, $\bar{\sigma}_{u,0} = 1,05 \frac{\text{m}}{\text{s}}$ über 398 Datensätze (Fall 2)	65
6.3	Gemittelte relative Abweichungen $\bar{\Delta}_{\text{rel}}$ und gemittelte modifizierte relative Abweichungen $\bar{\Delta}'_{\text{rel}}$ für beide Fälle	66
6.4	Gemittelte modifizierte relative Abweichungen $\bar{\Delta}'_{\text{rel}}$ für sieben Windrichtungssektoren	67
6.5	Gemittelte modifizierte relative Abweichung $\bar{\Delta}'_{\text{rel}}$ der Gesamtleistung des Windparks für Fall 1, Fall 2 und das gesamte Jahr 2019	69
6.6	Gemittelte modifizierte relative Abweichungen $\bar{\Delta}'_{\text{rel,Dichte}}$ für Fall 1 unter Berücksichtigung der variablen Dichte	70
6.7	Gemittelte modifizierte relative Abweichungen $\bar{\Delta}'_{\text{rel,Gradient}}$ für Fall 1 unter Berücksichtigung des Höhenwindgradienten	71

Abbildungsverzeichnis

1.1	Absolute und relative Positionen der Anlagen sowie des Messmastes im Windpark Curslack in Hamburg - Bergedorf, angegeben in <i>Gauß-Krüger</i> -Koordinaten . . .	4
1.2	Relevante Daten der Anlagen und des Messmastes im Windpark Curslack [9] . . .	5
2.1	Leistungs- und Schubbeiwert c_P , c_T in Abhängigkeit vom Induktionsfaktor a (eigene Darstellung in Anlehnung an [7, S. 46])	9
2.2	Darstellung des in der Arbeit verwendeten Koordinatensystems sowie der Art der Benennung von Abständen zwischen Anlagen	11
2.3	Schematische Darstellung des Geschwindigkeitsprofils der mittleren Windgeschwindigkeit (links) und dem Geschwindigkeitsdefizit (rechts) stromabwärts einer Turbine mit der Annahme einer zylindrischen Verteilung (in Anlehnung an [42])	12
2.4	Schematische Darstellung des Geschwindigkeitsprofils der mittleren Windgeschwindigkeit (links) und dem Geschwindigkeitsdefizit (rechts) stromabwärts einer Turbine mit der Annahme einer Gauss-Verteilung in Anlehnung an [42]	13
2.5	Prinzip der Superposition von Strömungsnachläufen	19
2.6	Schematische Zeichnung einer Anlage im „full wake“ (links) und im „partial wake“ (rechts) einer anderen Anlage (eigene Darstellung in Anlehnung an [59])	22
2.7	Exemplarisches Ablaufdiagramm des Algorithmus für drei hintereinander positionierte Anlagen	24
2.8	Darstellung von zwei Koordinatensystemen i und j inklusive aller relevanten Vektoren	26
2.9	Allgemeine Struktur eines künstlichen neuronalen Netzes (eigene Darstellung in Anlehnung an [8, S. 152])	27
2.10	Prinzip eines Neurons in einem künstlichen neuronalen Netz (eigene Darstellung in Anlehnung an [53, S. 52])	28
4.1	Reale Leistungs- und Schubbeiwert-Kurven der beiden Anlagentypen [49] [48]	39
4.2	Beiwerte c_P , c_T sowie Leistung P in Abhängigkeit des Induktionsfaktors a für alle Windgeschwindigkeiten $u_{in} \leq u \leq u_{out}$ an WEA 1	41
4.3	Herleitung des Zusammenhangs der beiden 0° Windrichtungen	42
4.4	Erzeugte Datenpunkte innerhalb der Rotorfläche einer Anlage	44
4.5	Darstellung der Problematik bei der Berechnung des Korrekturterms mit Hilfe einer imaginären Anlage 4'	48
5.1	Absolute Differenz zwischen im Windpark real gemessenen und berechneten Leistungen der Anlage 5 für $70^\circ \leq \alpha \leq 90^\circ$ über 124 Datensätze	52
6.1	Windpark Curslack mit eingezeichneten mittleren Windrichtungen der jeweiligen Sektoren der betrachteten Fälle	54

6.2	Reale und berechnete Leistungen P_r, P_b für $\bar{u}_\infty = 8,51 \frac{\text{m}}{\text{s}}, \bar{\alpha} = 294,55^\circ, \bar{\sigma}_{u,0} = 1,26 \frac{\text{m}}{\text{s}}$ über 667 Datensätze (Fall 1)	55
6.3	Reale und berechnete Leistungen P_r, P_b für $\bar{u}_\infty = 7,4 \frac{\text{m}}{\text{s}}, \bar{\alpha} = 294,38^\circ, \bar{\sigma}_{u,0} = 1,05 \frac{\text{m}}{\text{s}}$ über 100 Datensätze (Fall 1)	57
6.4	Reale und berechnete Leistungen P_r, P_b für $\bar{u}_\infty = 8,37 \frac{\text{m}}{\text{s}}, \bar{\alpha} = 216,55^\circ, \bar{\sigma}_{u,0} = 1,09 \frac{\text{m}}{\text{s}}$ über 398 Datensätze (Fall 2)	59
6.5	Reale und berechnete Leistungen P_r, P_b für $\bar{u}_\infty = 7,3 \frac{\text{m}}{\text{s}}, \bar{\alpha} = 216,77^\circ, \bar{\sigma}_{u,0} = 0,93 \frac{\text{m}}{\text{s}}$ über 100 Datensätze (Fall 2)	60
6.6	Relative Abweichungen Δ_{rel} für $\bar{u}_\infty = 8,51 \frac{\text{m}}{\text{s}}, \bar{\alpha} = 294,55^\circ, \bar{\sigma}_{u,0} = 1,26 \frac{\text{m}}{\text{s}}$ über 666 Datensätze (Fall 1)	62
6.7	Relative Abweichungen Δ_{rel} für $\bar{u}_\infty = 8,37 \frac{\text{m}}{\text{s}}, \bar{\alpha} = 216,55^\circ, \bar{\sigma}_{u,0} = 1,09 \frac{\text{m}}{\text{s}}$ über 398 Datensätze (Fall 2)	64
6.8	Relative Abweichung Δ_{ref} für Fall 1 + Fall 2 (666 + 398 = 1064 Datensätze) .	65
6.9	Häufigkeitsverteilungen der am Messmast ermittelten Windgeschwindigkeiten für beide Fälle	66
7.1	Auf den Anwendungsfall skalierte tanh-Funktion als Aktivierungsfunktion . . .	74
8.1	Vergleich der Gesamtleistung des Windparks mit vom Netz erzeugten Induktionsfaktoren sowie $\mathbf{a} = [\frac{1}{3}, \frac{1}{3}, \frac{1}{3}, \frac{1}{3}, \frac{1}{3}]$ mit zufälligen Netzparametern	79
8.2	Trainingsset- und Testset-Cost (<code>random2000.xlslx</code> , $\eta = 0,05$, 1 versteckte Schicht mit 16 Neuronen, Batchgröße = 10, 30 Epochen)	80
8.3	Vergleich der Gesamtleistung des Windparks mit vom Netz erzeugten Induktionsfaktoren sowie $\mathbf{a} = [\frac{1}{3}, \frac{1}{3}, \frac{1}{3}, \frac{1}{3}, \frac{1}{3}]$ (<code>random2000.xlslx</code> , $\eta = 0,05$, 1 versteckte Schicht mit 16 Neuronen, Batchgröße = 10, 30 Epochen)	81
8.4	Trainingsset- und Testset-Cost (<code>random2000.xlslx</code> , $\eta = 0,05$, 1 versteckte Schicht mit 32 Neuronen, Batchgröße = 10, 20 Epochen)	81
8.5	Vergleich der Gesamtleistung des Windparks mit vom Netz erzeugten Induktionsfaktoren sowie $\mathbf{a} = [\frac{1}{3}, \frac{1}{3}, \frac{1}{3}, \frac{1}{3}, \frac{1}{3}]$ (<code>random2000.xlslx</code> , $\eta = 0,05$, 1 versteckte Schicht mit 32 Neuronen, Batchgröße = 10, 20 Epochen)	82
8.6	Trainingsset- und Testset-Cost (<code>random8000.xlslx</code> , $\eta = 0,05$, 1 versteckte Schicht mit 32 Neuronen, Batchgröße = 10, 15 Epochen)	83
8.7	Vergleich der Gesamtleistung des Windparks mit vom Netz erzeugten Induktionsfaktoren sowie $\mathbf{a} = [\frac{1}{3}, \frac{1}{3}, \frac{1}{3}, \frac{1}{3}, \frac{1}{3}]$ (<code>random8000.xlslx</code> , $\eta = 0,05$, 1 versteckte Schicht mit 32 Neuronen, Batchgröße = 10, 15 Epochen)	83

1 Einleitung

Die vorliegende Bachelorarbeit ist im Rahmen des Bachelorstudiengangs „Maschinenbau - Entwicklung und Konstruktion“ an der HAW Hamburg entstanden.

Aufgrund des besseren Leseflusses wird in der Thesis überwiegend das generische Maskulinum verwendet. Es sei erwähnt, dass ausdrücklich alle Geschlechteridentitäten angesprochen sind.

1.1 Motivation und Ziel der Arbeit

In den letzten 40-50 Jahren ist die Abweichung der globalen Lufttemperatur vom Durchschnitt der Jahre 1850 bis 1900 konstant angestiegen [46, S. 29]. 60 % der weltweiten Treibhausgasemissionen, die maßgeblich für die Erwärmung des Planeten verantwortlich sind, entstehen durch die Nutzung und Herstellung fossiler Rohstoffe wie Erdöl, Erdgas und Kohle. [15]

Umso wichtiger ist deshalb der Übergang von der nicht-nachhaltigen Nutzung von fossilen Energieträgern zu einer nachhaltigen Energieversorgung durch erneuerbare Energien. Eine große Rolle spielt hierbei die Energiegewinnung mittels Windkraftanlagen, welche seit Jahren weltweit praktiziert wird. Allein im Jahre 2021 konnten global knapp 94 GW an Neuinstallationen verzeichnet werden, was zu einer kumulativen globalen Windenergiekapazität von 837 GW führt. [26, S. 8] Die Neuinstallationen pro Jahr müssen weitaus stärker ansteigen, als die bisherige Wachstumsrate prognostiziert, um der Energiewende gerecht zu werden. So muss sich bspw. die Anzahl der Neuinstallationen im Jahre 2030 im Vergleich zu 2021 vervierfachen. Wird die bisherige Wachstumsrate in die Zukunft projiziert, würde sich die Anzahl nicht einmal verdoppeln. [26, S. 12]

Vor diesem Hintergrund sowie der Tatsache, dass nicht unendlich Platz für neue Windkraftanlagen vorhanden ist, wird klar, dass Windparks in Zukunft auf immer engeren Platz erbaut werden müssen und sich so auch der Abstand zwischen zwei Anlagen innerhalb eines Parks verringert. Schon aktuell kann in Windparks die gegenseitige aerodynamische Abschattung zweier oder mehrerer Anlagen, der sog. *Wake-Effect* beobachtet werden, welcher durch die genannte Problematik zu einer immer größeren Herausforderung wird.

Untersuchungen und Simulationen bspw. im großen Offshore-Windpark *Horns Rev* in der östlichen Nordsee zeigten einen enormen Leistungsverlust von stromabwärts positionierten Anlagen in Bezug auf die Anlagen, die der freien Windströmung ausgesetzt sind, was unter anderem auf *Wake-Effekte* zurückzuführen ist [6, 51]. Wie eine Fallstudie von Lundquist u. a. zeigt, können Wakes in stabilen atmosphärischen Bedingungen so lange bestehen, dass sie in mehreren Millionen Euro ökonomischen Verlust (in einer Zeitperiode von 6 Jahren) resultieren [41].

Einfluss hierauf kann sowohl in der Planungsphase als auch in der Betriebsphase eines Windparks genommen werden. In der Planungsphase kann ein Windpark in Bezug auf die relativen Positionen der einzelnen Anlagen so errichtet werden, dass stromabwärts positionierte Anlagen für die am häufigsten auftretenden Windrichtungen nicht direkt im Strömungsnachlauf von anderen Anlagen stehen. In der Betriebsphase kann es sinnvoll sein, in einem Windpark einzelne Anlagen durch Variation von sog. Induktionsfaktoren so zu steuern, dass sie z.B. den Wind nicht

1 Einleitung

entsprechend der maximalen Leistungsentnahme abbremsten oder Anlagen absichtlich um die Turmachse zu drehen, so dass diese nicht in die Richtung des Windes ausgerichtet sind. Zweiteres hat zur Folge, dass die Nachlaufströmung, der Wake, abgelenkt wird.

In dieser Arbeit liegt der Fokus auf der Betriebsphase, speziell auf der erwähnten Variation der Induktionsfaktoren. Aktuell werden Anlagen in einem Windpark so geregelt, dass sie individuell alle maximale Leistung erzeugen. Dies geschieht bei einem Induktionsfaktor von $a = \frac{1}{3}$, wie zu späterem Zeitpunkt dieser Arbeit noch hergeleitet wird. Durch Variation bzw. Veränderung des Induktionsfaktors von stromaufwärts positionierten Anlagen können dann stromabwärts positionierte profitieren und so der Windparkertrag gesteigert werden. Grundlage dazu sind reale Daten aus dem Windpark aus den Jahren 2019 und 2020.

1.2 Struktur und Aufbau der Arbeit

Zu Beginn der Arbeit wird die Relevanz des Themas erläutert und der Windpark Curslack wird vorgestellt. Dazu gehören sowohl technische Informationen zu den jeweiligen Anlagen im Windpark als auch zum Messmast inklusive Sensoren, die in Abschnitt 1.3 aufgeführt sind. Außerdem werden die relativen Positionen der Anlagen zueinander anschaulich dargestellt. Es folgt in Kapitel 2 die Herleitung und Dokumentation aller notwendigen theoretischen Grundlagen, die im Laufe der Arbeit verwendet werden. Dazu zählen allgemeine technische Grundlagen zur Windenergie, speziell die analytische Beschreibung von einfachen und mehrfachen Nachlaufströmungen, Grundlagen zur Koordinatentransformation sowie Grundlagen zu künstlichen neuronalen Netzen, sprich dessen Aufbau und mathematischer Hintergrund. Es folgt in Kapitel 3 eine Analyse des Standes der Technik. Hier werden zahlreiche Publikationen untersucht, die sich mit der Maximierung der Leistung eines Windparks auf verschiedene Weisen beschäftigen. Dazu zählen unter anderem die Ablenkung der Wakes durch Drehen der Anlagen um die Turmachse und die Variation der Induktionsfaktoren der einzelnen Anlagen. Außerdem wird untersucht, inwieweit Methoden des maschinellen Lernens bereits in diesem Themengebiet Anwendung finden. In Kapitel 4 folgt die Entwicklung eines digitalen Zwillings zur möglichst genauen Leistungsberechnung im Windpark anhand von Umgebungsparametern. Nach kurzer Erläuterung der getroffenen Annahmen wird ein vorher ausgesuchtes Wake-Modell in `Python` implementiert, modifiziert und teilweise auf den Windpark Curslack angepasst. In Kapitel 5 wird erläutert, nach welchen Prinzipien die vorliegenden realen Windparkdaten gefiltert werden und welche Daten im späteren Verlauf der Arbeit verwendet werden. Schließlich folgt die Validierung des digitalen Zwillings in Kapitel 6. Nach Definition von zwei Fällen bezogen auf einen Bereich von Windrichtungen werden für diese Fälle die absoluten und relativen Abweichungen der berechneten und der realen Leistungen untersucht. Es folgt ebenfalls eine Untersuchung der relativen Abweichungen bezogen auf definierte Windrichtungssektoren in einem großen Datensatz (Jahr 2019). Am Ende des Kapitels werden zwei mögliche Optimierungen untersucht: Der Einfluss der Varianz der Dichte und der Einfluss des Höhenwindgradienten auf die Umgebungswindgeschwindigkeit. In Kapitel 7 wird schließlich unter Verwendung des digitalen Zwillings ein künstliches neuronales Netz entworfen, welches die Leistung im Windpark durch Variation der Induktionsfaktoren der einzelnen Anlagen maximiert.

Dazu wird der entwickelte digitale Zwillling modifiziert und als Kostenfunktion implementiert. Getestet wird das neuronale Netz schließlich an einem bereits definierten zur Validierung verwendeten Fall. Nach Darlegung der Ergebnisse schließt die Arbeit mit einer Zusammenfassung sowie einer kritischen Reflexion und einem Ausblick ab.

1.3 Windpark Curslack

In Kooperation mit der HAW Hamburg hat die *ReTec Zweite Betriebs UG & Co. KG* im September 2017 den Windpark Curslack in Betrieb genommen. Der Windpark ist Teil der Forschung am Technologiezentrum Energie-Campus der HAW Hamburg und dient dazu, aktuelle Fragestellungen bspw. im Bereich Windenergieerzeugung und -anlagenbetrieb zu erforschen.

Die Betriebsdauer des Windparks soll 20 Jahre betragen, wobei mit einem Gesamtenergieertrag von 33 GWh pro Jahr zu rechnen ist. Dies entspricht einer Versorgung von ca. 11000 Haushalten. Dabei läuft der Windpark ca. 3000 äquivalente Volllaststunden und vermeidet damit ca. 22000 t CO₂ -Emissionen. [10]

Der Park besteht aus fünf sog. HAWTs (Horizontal Axis Wind Turbine), also Anlagen, bei denen der Rotor um eine horizontale Achse dreht sowie einem Messmast mit verschiedenen Sensoren zum Messen von Umgebungsparametern.

Die absoluten sowie relativen Positionen der einzelnen Anlagen im Windpark sind der nachfolgenden Abbildung 1.1 zu entnehmen. Die einzelnen Anlagen sind hierbei mit WEA 1-5, der Messmast mit MM gekennzeichnet.

1 Einleitung

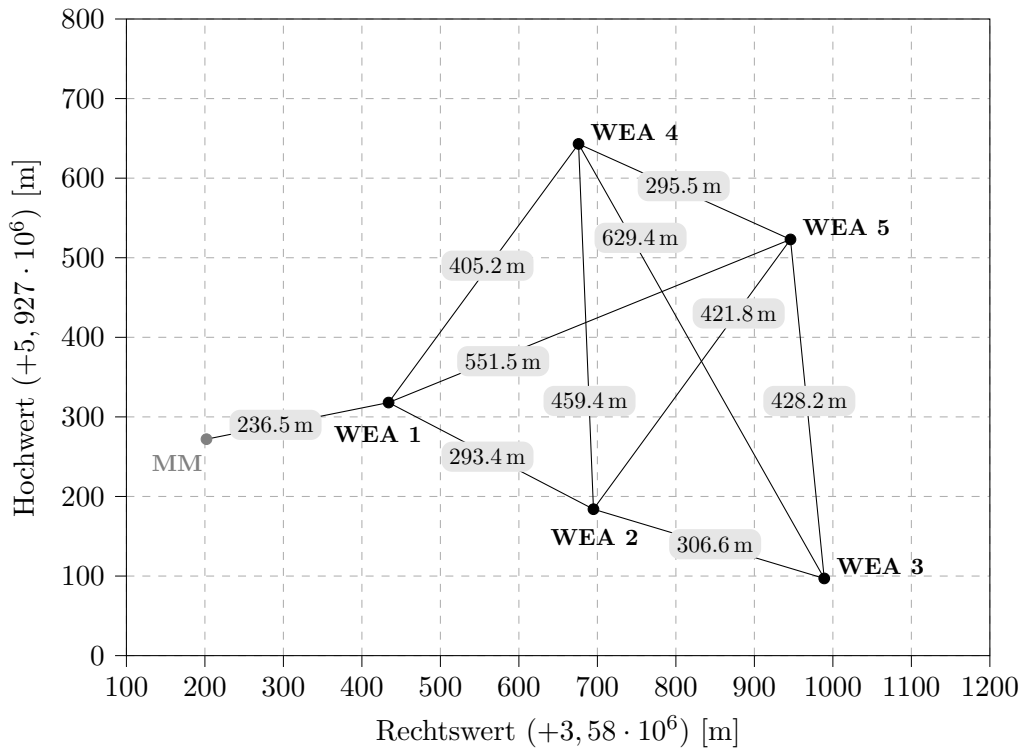


Abbildung 1.1: Absolute und relative Positionen der Anlagen sowie des Messmastes im Windpark Curslack in Hamburg - Bergedorf, angegeben in *Gauß-Krüger*-Koordinaten

Zur Darstellung wird das sog. *Gauß-Krüger*-Koordinatensystem verwendet, da mit diesem eine winkeltreue Abbildung in metrischen Koordinaten stattfinden kann. Die fünf Anlagen setzen sich aus zwei verschiedenen Typen vom Hersteller Nordex zusammen:

- 1x Nordex N117/3000 Generation Delta → WEA 1
- 4x Nordex N117/2400 Generation Gamma → WEA 2 – 5.

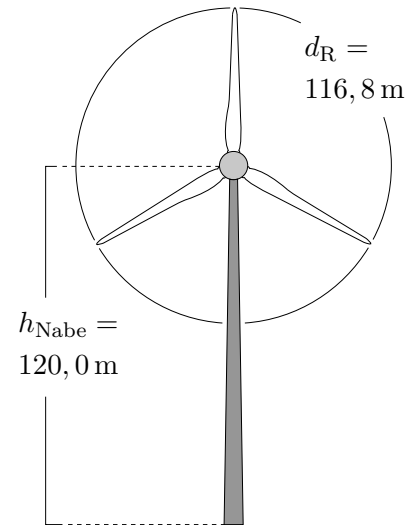
Die Werte der Anlagen im Nennbetrieb sind Abb. 1.2a zu entnehmen. Dazu zählen die Nennleistung P_n , die Nennwindgeschwindigkeit u_n sowie die Ein- und Abschaltwindgeschwindigkeit (EWG bzw. AWG) $u_{in} \wedge u_{out}$, also die Windgeschwindigkeit, bei der die Anlage beginnt, durch Rotation Leistung zu produzieren und die Windgeschwindigkeit, bei der die Anlage abgeschaltet wird, um Schäden zu vermeiden. Der Durchmesser d_R und die Nabenhöhe h_{Nabe} , die bei allen Anlagen unabhängig vom Typ identisch sind, können Abb. 1.2b entnommen werden.

Dass bei WEA 1 zwei Nennleistungen aufgelistet sind, liegt an einer Leistungserhöhung dieser Anlage von 3,0 kW auf 3,15 kW seit dem 01.01.2021. Ebenfalls in Abb. 1.2a aufgeführt sind die Sensoren des Messmastes inklusive ihrer jeweiligen Anzahl, wobei für diese Arbeit die Windgeschwindigkeit und die Windrichtung am relevantesten sein werden. Nachfolgende Tabelle zeigt die einzelnen Sensoren, die dazugehörigen Höhen und die Ausrichtung des Sensors am Messmast.

Bezeichnung	Größe	WEA 1	WEA 2-5
Nennleistung	P_n [MW]	3,0/3,15	2,4
EWG	u_{in} [$\frac{m}{s}$]	3	3
AWG	u_{out} [$\frac{m}{s}$]	25	20
NWG	u_n [$\frac{m}{s}$]	12,6	12,5

MM Sensoren	Anzahl
Windgeschw.	11
Windrichtung	5
Temperatur	2
Luftdruck	2
Luftfeuchte	2

(a) Nennleistung, Windgeschwindigkeiten, Sensoren



(b) Durchmesser und Nabhöhe

Abbildung 1.2: Relevante Daten der Anlagen und des Messmastes im Windpark Curslack [9]

Höhe	Bezeichnung (Ausrichtung)
120,5 m	V1 (313°), V2 (133°)
115 m	V3 (313°), D3 (313°)
113,1 m	T1 (133°), H1 (133°), P1 (133°)
112 m	D1 (313°)
111,8 m	V4 (133°), D4 (133°)
80 m	V5 (313°)
79,8 m	V8 (133°)
78 m	V6 (313°), D6 (313°)
65 m	V7 (313°)
64,8 m	V9 (133°)
62,5 m	D2 (313°)
44,9 m	V10 (313°)
44,7 m	V11 (133°)
11 m	P2 (133°)
10 m	T2 (133°), H2 (133°)

V $\hat{=}$ Anemometer
D $\hat{=}$ Windfahne
T $\hat{=}$ Temperatursensor
H $\hat{=}$ Luftfeuchtesensor
P $\hat{=}$ Luftdrucksensor

Tabelle 1.1: Alle vorhandenen Sensoren des Messmastes inklusive ihrer Höhe und Ausrichtung

2 Theoretische Grundlagen

Dieses Kapitel dient zur Herleitung, Erklärung, Dokumentation und späteren Referenzierung der verwendeten theoretischen Grundlagen. Das Kapitel beinhaltet sowohl Theorie zu Windenergieanlagen als auch zu verwendeten Methoden des maschinellen Lernens.

2.1 Leistungs- und Schubbeiwert

Die Zusammenhänge und dazugehörigen Herleitungen aus diesem Abschnitt sind, wenn nicht explizit anders gekennzeichnet, Hau entnommen [21, S. 81 ff.].

Die kinetische Energie einer Luftmasse m mit der Geschwindigkeit u lässt sich durch

$$E_{\text{kin}} = \frac{1}{2} m u^2 \quad (2.1)$$

ausdrücken. An einer von dieser Luft durchströmten Querschnittsfläche A kann der Massenstrom, also die in einer Zeiteinheit dt durchfließende Masse mit

$$\frac{dm}{dt} = \dot{m} = \rho u A. \quad (2.2)$$

berechnet werden. Wird die in Gl. 2.1 aufgestellte Energie nach der Zeit differenziert und Gl. 2.2 eingesetzt, ergibt sich die Leistung P nach

$$P = \frac{dE_{\text{kin}}}{dt} = \frac{1}{2} \frac{dm}{dt} u A = \frac{1}{2} \rho A u^3. \quad (2.3)$$

Sei die durchströmte Querschnittsfläche A nun der Rotor einer Windenergieanlage, kann die mechanische Leistung, die der Rotor der frei strömenden Luft entzieht, berechnet werden. Diese ergibt sich aus der Differenz der Leistungen vor (Index 1) und hinter (Index 2) dem Rotor

$$P_{\text{mech}} = P_1 - P_2 = \frac{1}{2} \rho (A_1 u_1^3 - A_2 u_2^3) \stackrel{*}{=} \frac{1}{2} \dot{m} (u_1^2 - u_2^2), \quad (2.4)$$

wobei $\stackrel{*}{=}$ die Verwendung der Kontinuitätsgleichung $\rho u_1 A_1 = \rho u_2 A_2$ kennzeichnet. Für das Verhältnis $\beta = \frac{u_2}{u_1}$, für das die Leistung maximal wird, ergibt sich die triviale Lösung $u_2 = 0$, die aber physikalisch keinen Sinn macht, da so hinter dem Rotor gar keine Strömung mehr vorhanden und die Kontinuitätsgleichung nicht erfüllt wäre.

Das *Theorem von Froude und Rankine* als weiterer Ansatz verschafft hier Abhilfe. Die (Schub-)Kraft, die der Rotor durch die freie Windströmung erfährt, ergibt sich nach

$$F_s = \dot{m} (u_1 - u_2). \quad (2.5)$$

Der Rotor muss dieser Kraft entgegenwirken und setzt dabei die mechanische Leistung

$$P_{\text{mech}} = F_s u_R \quad (2.6)$$

mit der Geschwindigkeit in der Rotorebene u_R um. Gl. 2.5 in Gl. 2.6 eingesetzt liefert

$$P_{\text{mech}} = \dot{m} (u_1 - u_2) u_R. \quad (2.7)$$

2 Theoretische Grundlagen

Die mechanische Leistung des Rotors kann also sowohl über Gl. 2.4 als auch über Gl. 2.7 bestimmt werden. Gleichsetzen der beiden Ansätze liefert

$$u_R = \frac{1}{2} (u_1 + u_2). \quad (2.8)$$

als Mittel aus der Windgeschwindigkeit vor und nach dem Rotor. Setzt man diese Beziehung in Gl. 2.7 ein und beachtet, dass der Massenstrom durch

$$\dot{m} = \rho A u_R = \rho A \frac{1}{2} (u_1 + u_2) \quad (2.9)$$

bestimmt wird, ergibt sich

$$P_{\text{mech}} = \frac{1}{4} \rho A (u_1^2 - u_2^2) (u_1 + u_2). \quad (2.10)$$

Üblich ist es, diese Leistung in Relation zu der Leistung des frei strömenden Windes P_0 mit der Geschwindigkeit u_1 (siehe Gl. 2.3) zu setzen. Es ergibt sich der *Leistungsbeiwert*

$$c_P = \frac{P_{\text{mech}}}{P_0} = \frac{\frac{1}{4} \rho A (u_1^2 - u_2^2) (u_1 + u_2)}{\frac{1}{2} \rho A u_1^3}. \quad (2.11)$$

Durch Umformungen kann jetzt der bereits eingeführte Quotient β substituiert werden:

$$c_P = f(\beta) = \frac{1}{2} \left[1 - \left(\frac{u_2}{u_1} \right)^2 \right] \cdot \left[1 + \frac{u_2}{u_1} \right]. \quad (2.12)$$

Wird c_P grafisch über β aufgetragen oder wird die Funktion differenziert, kann das Maximum des Leistungsbeiwertes bei $\beta = \frac{1}{3}$ gefunden werden. Es beträgt

$$c_P = \frac{16}{27} = 0,593 \quad (2.13)$$

und wird auch als *Betzscher Faktor* bezeichnet.

Zusammenfassend sagt er aus, dass ein Rotor maximal 59,3% der im Wind erhaltenen Leistung umwandeln kann. Dieser Fall tritt ein, wenn die Windströmung hinter dem Rotor auf ein Drittel der Windgeschwindigkeit vor dem Rotor abgebremst wird. Die Leistung, die eine Windenergieanlage produziert, kann schlussendlich über

$$P = \frac{1}{2} \rho_{\text{Luft}} A_R u^3 c_P \quad (2.14)$$

berechnet werden [7, S. 6]. Hierbei ist ρ_{Luft} die Dichte der Luft und A_R die Rotorkreisfläche.

Analog zum Leistungsbeiwert lässt sich ein *Schubbeiwert* c_T definieren. Dieser ergibt sich aus dem Verhältnis der bereits definierten Schubkraft am Rotor (Gl. 2.5) und der Axialkraft des Windes, welche sich aus dem Produkt von Staudruck der ungestörten Strömung und der Rotorkreisfläche ergibt [11]:

$$c_T = \frac{F_S}{\frac{1}{2} \rho A u_1^2}. \quad (2.15)$$

Bei einem Verhältnis von $\beta = \frac{1}{3}$ ergibt sich $c_T = \frac{8}{9} \approx 0,889$.

2.2 Axialer Induktionsfaktor

Der sog. axiale Induktionsfaktor gibt an, wie sehr sich die Strömung bei Eintritt in die Rotorebene verzögert. Formeln und Herleitungen stammen, wenn nicht explizit gekennzeichnet, aus Burton u. a. [7, S. 43]. Der Induktionsfaktor ist definiert nach

$$u_R = u_1 (1 - a) \quad \rightarrow \quad a = \frac{u_1 - u_R}{u_1}. \quad (2.16)$$

Ist der axiale Induktionsfaktor $a = 1$, so wird die Strömung theoretisch zu 100% abgebremst und existiert hinter der Rotorebene nicht mehr. Das macht physikalisch keinen Sinn. Ist der Induktionsfaktor $a = 0$, so wird kein Wind von der Anlage „abgefangen“, er strömt ungehindert durch die Rotorebene. Die Variation des Induktionsfaktors findet üblicherweise durch Drehzahlregelung oder Pitchen der Blätter statt. Die Geschwindigkeit hinter dem Rotor lässt sich ebenfalls vereinfacht durch

$$u_2 = (1 - 2a) u_1 \quad (2.17)$$

ausdrücken. Beachtet werden muss, dass die Theorie für $a \geq 0,5$ nicht mehr gültig ist, da hier $u_2 \leq 0$ wäre.

Auch die Beiwerte c_P und c_T lassen sich durch den Induktionsfaktor a ausdrücken. Es gilt

$$c_P = 4a(1 - a)^2, \quad (2.18)$$

$$c_T = 4a(1 - a). \quad (2.19)$$

Wird dieser Zusammenhang grafisch dargestellt, kann der Induktionsfaktor $a = \frac{1}{3}$, bei dem der Leistungsbeiwert sein theoretisches Maximum $c_{P,theo} = 0,593$ erreicht, gefunden werden. Der Schubbeiwert erreicht hier einen Wert von $c_{T,theo} = 0,89$.

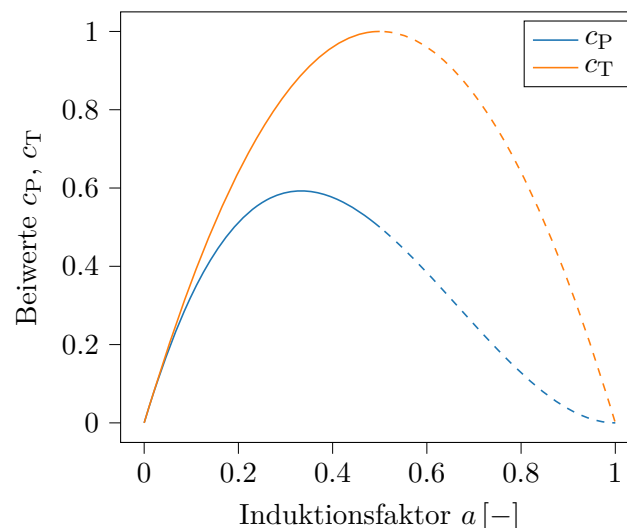


Abbildung 2.1: Leistungs- und Schubbeiwert c_P , c_T in Abhängigkeit vom Induktionsfaktor a (eigene Darstellung in Anlehnung an [7, S. 46])

2 Theoretische Grundlagen

An dieser Stelle ist es wichtig, den Unterschied zwischen dem in Abschnitt 2.1 in der Herleitung verwendeten Verhältnis β und dem Induktionsfaktor a zu erwähnen. β wird als das Verhältnis von der Geschwindigkeit vor dem Rotor und hinter dem Rotor definiert. $\beta = \frac{1}{3}$ sagt demnach aus, dass die Geschwindigkeit hinter dem Rotor nur noch ein Drittel der Geschwindigkeit vor dem Rotor beträgt. Der Induktionsfaktor a hingegen bezieht das Defizit zwischen der Geschwindigkeit vor dem Rotor und der Geschwindigkeit in der Rotorebene $u_1 - u_R$ auf die Geschwindigkeit vor dem Rotor. $a = \frac{1}{3}$ bedeutet hier also, dass die Geschwindigkeit in der Rotorebene gegenüber der Geschwindigkeit vor dem Rotor um ein Drittel reduziert wird.

Wird das Verhältnis $\beta = \frac{u_2}{u_1}$ in Gl. 2.8 und diese schließlich in Gl. 2.16 eingesetzt, so kann der Zusammenhang

$$\beta = 1 - 2a, \quad (2.20)$$

formuliert werden, welcher sich durch Gl. 2.17 bestätigt. Somit wird klar, wieso bei $a = \beta = \frac{1}{3}$ das Maximum für c_p erreicht wird.

2.3 Turbulenzintensität

Die Turbulenzintensität I allgemein ist ein Maß für die Stärke der Turbulenz im Wind und wird nach

$$I = \frac{\sigma}{\bar{u}} \quad (2.21)$$

berechnet [7, S. 17]. Hierbei ist σ die Standardabweichung der Windgeschwindigkeit mit Bezug zur mittleren Windgeschwindigkeit \bar{u} . Diese wird hierbei meist über einen Zeitraum von 10 min oder 60 min ermittelt.

Wird die Turbulenzintensität im Wake einer Windenergieanlage I_{wake} betrachtet, so setzt sich diese aus der Umgebungs- und der sog. „added“ Turbulenzintensität, eine vom Rotor induzierte, zusammen [7, S. 36]:

$$I_{\text{wake}} = \sqrt{I_0^2 + I_+^2}. \quad (2.22)$$

2.4 Analytische Modellierung der Nachlaufströmung

Wie bereits in Abschnitt 1.1 erwähnt, ist es essentiell, sich mit der mathematischen Beschreibung von Nachlaufströmungen auseinanderzusetzen. Neben Strömungssimulationen (CFD), die in der Regel genauer sind, aber durch das Lösen von nicht-trivialen Differentialgleichungen einen enorm großen Rechenaufwand nach sich ziehen, existieren zahlreiche analytische Ansätze und Ingenieurs-Modelle, die auf teilweise nur sehr wenigen teils empirischen Gleichungen basieren, die im Gegensatz zu z.B. den Navier-Stokes-Gleichungen sehr einfach gelöst werden können.

An dieser Stelle sei außerdem darauf hingewiesen, dass eine ausführliche Herleitung aller existierenden Modelle über den Rahmen dieser Arbeit hinausgehen würde. Hierzu sei auf die genannten Quellen verwiesen.

Bei Strömungsnachläufen können zwei Bereiche unterschieden werden: Der *near-wake*, also den Bereich unmittelbar hinter dem Rotor und der *far-wake*, also den Bereich weiter hinten. Die Abgrenzung dieser beiden Bereiche ist in der Literatur nicht eindeutig definiert, oft wird aber

ab einem zweifachen Rotordurchmesser als Abstand vom *far wake* gesprochen. Es ist außerdem üblich, den Abstand zwischen Anlagen nicht in absoluten Werten, sondern bezogen auf dem Rotordurchmesser d_R anzugeben, wenn es um Wakes geht. Für bspw. einen Abstand in Größe von zwei Rotordurchmessern wird folgend $2 \cdot d_R \equiv 2D$ geschrieben. Die nachfolgende Abbildung visualisiert die Angabe der Entfernung durch den Rotordurchmesser und zeigt außerdem das von diesem Zeitpunkt an verwendete Koordinatensystem. Die genannten Quellen beziehen sich auf die Gleichungen und Herleitungen des jeweils gesamten vorgestellten Modells.

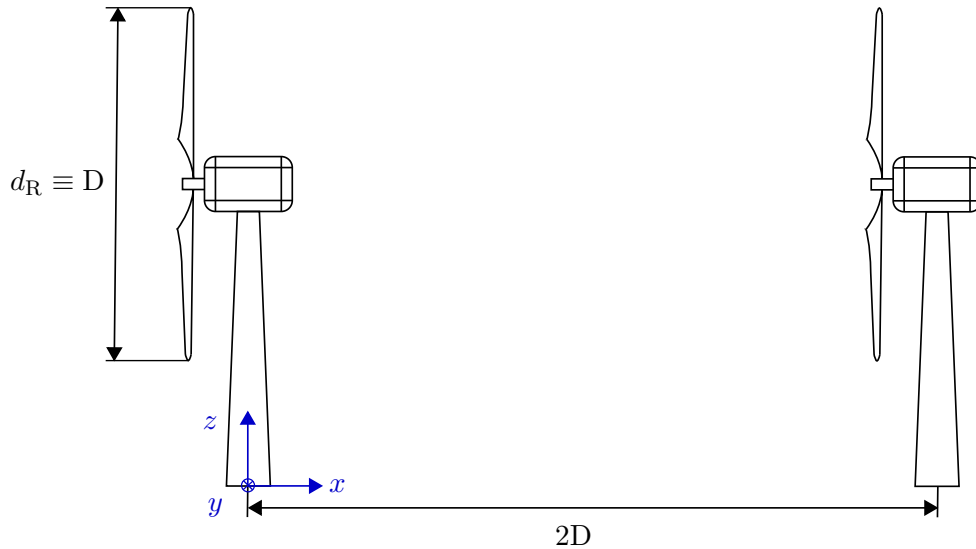


Abbildung 2.2: Darstellung des in der Arbeit verwendeten Koordinatensystems sowie der Art der Benennung von Abständen zwischen Anlagen

2.4.1 Jensen-Modell

Eines der ältesten und einfachsten Modelle ist das Modell nach Jensen, welches bereits 1983 veröffentlicht wurde [30]. Nach ihm lässt sich die Windgeschwindigkeit im Wake durch den Schubbeiwert c_T und eine Wake-Wachstumsrate k ausdrücken. Wird eine lineare Ausbreitung des Wakes als Funktion betrachtet, kann k als Steigung dieser interpretiert werden. Sie gibt also an, wie schnell sich der Wake, bezogen auf die Distanz hinter dem Rotor, ausbildet. In noch folgenden Modellen wird die Konstante auch k_{wake} oder k^* genannt. Die Geschwindigkeit an einer Position x ergibt sich nach

$$u_w(x) = u_\infty \left[\frac{1 - 2a}{\left(1 + \frac{kx}{r_1}\right)^2} \right], \quad (2.23)$$

wobei u_∞ die Windgeschwindigkeit der Umgebung ist. Das Modell ist eindimensional, es gilt: $u_w = f(x)$. Der axiale Induktionsfaktor a lässt sich durch den Schubbeiwert c_T ausdrücken, der charakteristische Radius r_1 repräsentiert den Wake-Radius unmittelbar hinter dem Rotor und lässt sich über den Rotorradius ausdrücken und die Wake-Zerfallrate ist abhängig von der

2 Theoretische Grundlagen

Nabenhöhe h_{Nabe} der Anlage sowie der lokalen Bodenrauigkeit z_0 :

$$a = \frac{1 - \sqrt{1 - c_T}}{2}, \quad r_1 = r_r \sqrt{\frac{1 - a}{1 - 2a}}, \quad k = \frac{0.5}{\ln\left(\frac{h_{\text{Nabe}}}{z_0}\right)}. \quad (2.24)$$

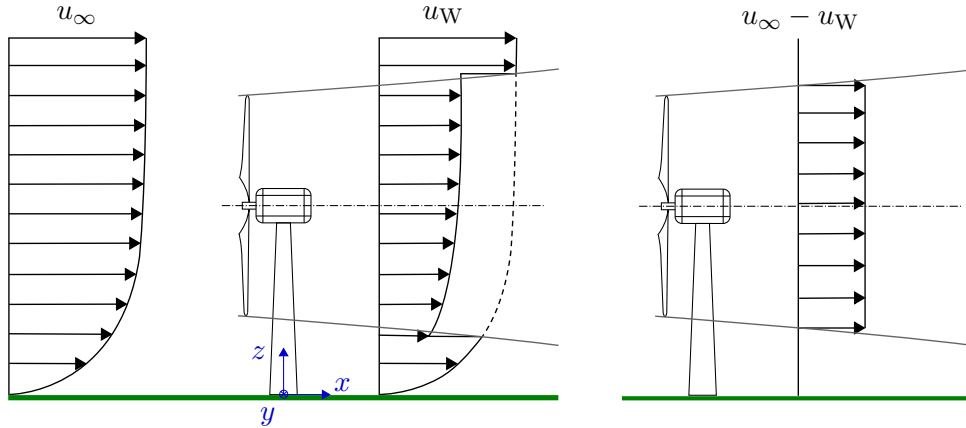


Abbildung 2.3: Schematische Darstellung des Geschwindigkeitsprofils der mittleren Windgeschwindigkeit (links) und dem Geschwindigkeitsdefizit (rechts) stromabwärts einer Turbine mit der Annahme einer zylindrischen Verteilung (in Anlehnung an [42])

Das Modell nach Jensen geht von einer zylindrischen (engl.: top hat) Verteilung des Geschwindigkeitsdefizits im Wake aus (siehe Abb. 2.3) [58, S. 91].

2.4.2 Frandsen-Modell

Ein zu dem „Standard“-Jensen-Modell alternatives Modell ist das Frandsen-Modell [16, S. 43 ff.]. Nach diesem lässt sich die Geschwindigkeit im Wake an einer Position x nach

$$u_w(x) = u_\infty \left(\frac{1}{2} \pm \frac{1}{2} \sqrt{1 - 2 \frac{A_R}{A(x)} c_T} \right) \quad (2.25)$$

berechnen. Das „+“ ist anwendbar für $a \leq 0,5$, das „-“ ist für $a > 0,5$ gültig. Es gilt $A(x) \propto d(x)$ und

$$d(x) = \left(\beta + \alpha(x) \frac{x}{d_R} \right)^{\frac{1}{2}} d_R \quad \text{mit} \quad \beta = \frac{1}{2} \frac{1 + \sqrt{1 - c_T}}{\sqrt{1 - c_T}}. \quad (2.26)$$

α ist hierbei ein Maß für die Ausbreitung des Wakes und bspw. für kleine c_T und große Distanzen hinter dem Rotor in der Größenordnung von $(10 k_{\text{wake}})$. Berechnet werden kann α nach

$$\alpha = \beta \left[\left(1 + 2 k_{\text{wake}} \frac{x}{d_R} \right)^2 - 1 \right] \left(\frac{x}{d_R} \right)^{-1}. \quad (2.27)$$

Dieses Modell ist wie das Jensen-Modell nur eindimensional und basiert ebenfalls auf einer top-hat-Verteilung des Geschwindigkeitsdefizites im Nachlauf.

Frandsen u. a. beschäftigten sich ebenfalls mit Turbulenzmodellen. Die effektive Turbulenzintensität kann nach ihnen durch

$$I_{\text{wake}}(x) = \sqrt{K_n \frac{c_T}{\left(\frac{x}{d_R}\right)^2} + I_0^2} \quad (2.28)$$

berechnet werden (vgl. Gl. 2.22), wobei im Summanden der induzierten Turbulenz K_n eine empirische Konstante ist, die von der Anlagenkonfiguration abhängig ist, sprich davon, ob die Anlage mitten im Windpark steht oder ob es sich z.B nur um zwei benachbarte Anlagen handelt. [17]

2.4.3 Jensen-2D-k-Modell

Durch Experimente in einem Windtunnel fand L. P. Chamorro heraus, dass das Geschwindigkeitsdefizit vor allem im far-wake annähernd durch eine Gauss-Funktion beschrieben werden kann [38, S. 140]. Reinwardt führte LiDAR-Messungen (Eine Methode zur Umfelderkassung mit Hilfe eines Lasers) am Windpark Curslack, welcher auch in dieser Arbeit betrachtet wird, durch und konnte schon in einer Entfernung von $2,18D$ hinter dem Rotor ebenfalls eine Gauss-Verteilung bestätigen [54, S. 77]. Diese ist in der nachstehenden Abb. 2.4 dargestellt.

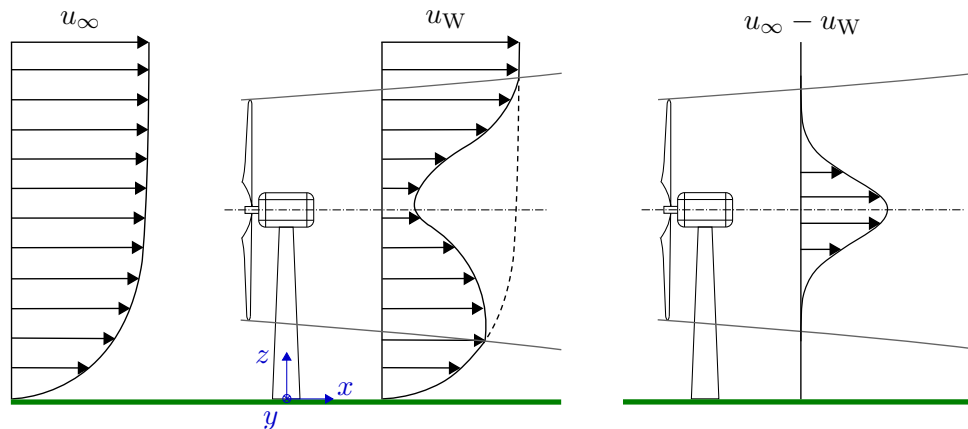


Abbildung 2.4: Schematische Darstellung des Geschwindigkeitsprofils der mittleren Windgeschwindigkeit (links) und dem Geschwindigkeitsdefizit (rechts) stromabwärts einer Turbine mit der Annahme einer Gauss-Verteilung in Anlehnung an [42]

Wird das Windgeschwindigkeitsdefizit als Gauss-Funktion betrachtet (s. Abb. 2.4 rechts), kann die Breite des Wakes an einer Position x als Standardabweichung σ der Gauss-Funktion interpretiert werden. Es gilt dann

$$\frac{\sigma(x)}{d_R} = k \frac{x}{d_R} + \varepsilon, \quad (2.29)$$

wobei k hier wie bereits beim Jensen-Modell erläutert, ebenfalls als Wachstumsrate des Wakes und ε als Offset, also als Breite des Wakes bei $x = 0$, zu sehen ist. σ ist hier nicht mit einer Turbulenzintensität wie in Abschnitt 2.3 zu verwechseln.

2 Theoretische Grundlagen

Tian u. a. verbesserten das vorgestellte Jensen-Modell hinsichtlich Dimensionalität, Geschwindigkeitsprofil und später auch bezüglich Einbindung einer Turbulenzintensität [58, S. 91 ff.]. Nach den Autoren lässt sich die Geschwindigkeit im Wake an einer Position $[x, r]^T$ in zwei Schritten berechnen, wobei der erste Schritt Gl. 2.23 entspricht:

$$u^*(x) = u_\infty \left[\frac{1 - 2a}{\left(1 + \frac{kx}{r_1}\right)^2} \right], \quad (2.30)$$

$$u_w(x, r) = (u_\infty - u^*(x)) \cos\left(\frac{\pi}{r_x} r + \pi\right) + u^*(x) \quad \text{mit} \quad r_x = kx + r_T. \quad (2.31)$$

Dieses Modell ist zweidimensional, da $u_w = f(x, r)$ gilt. Für das Geschwindigkeitsdefizit wird eine Verteilung in Form einer cos-Funktion als Vereinfachung einer Gauss-Verteilung angenommen. Zusätzlich dazu wird die effektive Turbulenzintensität I_{wake} mit einbezogen. Diese ergibt sich ähnlich wie beim Modell nach Frandsen u. a. an einer Position x nach

$$I_{\text{wake}}(x) = K_n \frac{c_T}{\left(\frac{x}{d_R}\right)} + I_0, \quad (2.32)$$

wobei K_n die bereits vorgestellte empirische Konstante darstellt. Ein Vergleich dieses Ansatzes mit dem von Frandsen u. a. (Gl. 2.28) zeigt, dass dort beim Summanden der induzierten Turbulenz $I_{\text{wake}} \propto \sqrt{c_T}$ gilt, wohingegen beim Modell nach Tian u. a. $I_{\text{wake}} \propto c_T$ gilt. Der Einfluss der Schubkraft auf den Wake soll so erhöht werden.

Die neue modifizierte Wake-Zerfallrate ergibt sich aus

$$k_{\text{wake}}(x) = k \frac{I_{\text{wake}}(x)}{I_0}. \quad (2.33)$$

Diese kann jetzt in Gl. 2.30 und Gl. 2.31 eingesetzt werden ($k \rightarrow k_{\text{wake}}(x)$), um so das Geschwindigkeitsdefizit zu berechnen.

2.4.4 Jensen-Gauss-Modell

Eine weitere Alternative, die ebenfalls auf dem Jensen-Modell basiert und stark dem Jensen-2D-k-Modell ähnelt, ist das Jensen-Gauss-Modell von Xiaoxia u. a. [60]. Anstatt eines cos-Ansatzes für das Geschwindigkeitsprofil wird hier ein Gauss-Ansatz gewählt. Dieser lässt sich im Allgemeinen nach

$$f(x) \cdot \exp\left(-\frac{r^2}{2\sigma^2}\right) \quad (2.34)$$

definieren [45, S. 34], wobei $f(x)$ durch Randbedingungen ermittelt wird. Anders als bei einer zylindrischen Verteilung ist die Grenze vom Wake zur Umgebung nicht klar getrennt, da die Gauss-Funktion gegen 0 konvergiert. Im Falle des Jensen-Gauss-Modells wird deshalb die sog. 3- σ -Regel angewandt. Diese besagt, dass mit einer 99%-igen Wahrscheinlichkeit die Werte der Gaussverteilung im Bereich von $2,58\sigma$ liegen. Der Wake-Radius ist identisch mit dem des Standard-Jensen-Modells, so dass gilt

$$r_x = 2,58\sigma. \quad (2.35)$$

Schlussendlich lässt sich die Geschwindigkeit im Wake an einer Position $[x, r]^T$ ähnlich zu Gl. 2.30 und Gl. 2.31 nach

$$u^*(x) = u_\infty \left[\frac{1 - 2a}{\left(1 + \frac{kx}{r_1}\right)^2} \right], \quad (2.36)$$

$$u_w(x, r) = u_\infty - (u_\infty - u^*(x)) \cdot \frac{5,16}{\sqrt{2\pi}} \cdot \exp \left[-\frac{r^2}{2 \left(\frac{r_x}{2,58}\right)^2} \right] \quad \text{mit} \quad r_x = kx + r_r \quad (2.37)$$

berechnen. Auch in diesem Modell kann die Turbulenzintensität ähnlich der bereits vorgestellten Ansätze von Frandsen u. a. (Gl. 2.28) und Tian u. a. (Gl. 2.32) implementiert werden. Hier ergibt diese sich nach

$$I_{\text{wake}}(x) = \left[K_n \frac{c_T}{\left(\frac{x}{d_R}\right)^{0.5}} + I_0^{0.5} \right]^2. \quad (2.38)$$

Durch diesen Ansatz soll eine bessere Balance zwischen Umgebungsturbulenz und induzierter Turbulenz geschaffen werden [60, S. 195]. Die modifizierte Wake-Zerfallrate ergibt sich dann analog zu Gl. 2.33 und kann in Gl. 2.36 und Gl. 2.37 eingesetzt ($k \rightarrow k_{\text{wake}}(x)$) werden.

2.4.5 Bastankhah-Modell

Das Bastankhah-Modell [42] basiert wie das erweiterte Jensen-Modell ebenfalls auf einer Gauss-Verteilung des Geschwindigkeitsdefizits. Es ist dreidimensional, d.h. $\Delta u = f(x, y, z)$. Wichtig zu erwähnen ist, dass hier nicht direkt die Windgeschwindigkeit im Wake, sondern das Geschwindigkeitsdefizit, berechnet wird. Basierend auf dem Ansatz einer Gauss-Funktion (Gl. 2.34) ergibt sich schlussendlich das Defizit an einer Position $[x, y, z]^T$ nach

$$\Delta u(x, y, z) = u_\infty \cdot \left[\left(1 - \sqrt{1 - \frac{c_T}{8 \left(k^* \frac{x}{d_0} + 0.2 \sqrt{\beta}\right)^2}} \right) \cdot \exp(\xi) \right]$$

$$\text{mit } \xi = -\frac{1}{2 \left(k^* \frac{x}{d_0} + 0.2 \sqrt{\beta}\right)^2} \left[\left(\frac{z - h_{\text{Nabe}}}{d_R}\right)^2 + \left(\frac{y}{d_R}\right)^2 \right]. \quad (2.39)$$

Das Modell beruht auf nur einer Gleichung, in der β wie in Gl. 2.26 berechnet wird. Zwecks Übersichtlichkeit ist die Gleichung hier noch einmal aufgeführt:

$$\beta = \frac{1}{2} \frac{1 + \sqrt{1 - c_T}}{\sqrt{1 - c_T}}. \quad (2.40)$$

Die Wachstumsrate k^* wird anhand von LES Ergebnissen gewählt. Zhan u. a. beschäftigten sich mit der optimalen Abstimmung von Parametern für verschiedene Wake-Modelle, so auch mit dem Bastankhah-Modell [61, S. 8 ff.]. Durch LiDAR-Messungen kann ein empirischer Zusammenhang zwischen k^* und der Umgebungsturbulenzintensität I_0 in der Form

$$k^* = 0,34 I_0 - 0,013 \quad (2.41)$$

hergestellt werden. Anders als bei den erweiterten Jensen-Modellen wird nur die Turbulenzintensität der Umgebung mit einbezogen, nicht aber die induzierte.

2.4.6 Ishihara-Qian-Modell

Ein weiteres analytisches Modell, basierend auf einer Gauss-Verteilung des Geschwindigkeitsdefizits, ist das Modell nach T. Ishihara [57]. Es ist wie das Bastankhah-Modell dreidimensional und berücksichtigt die Turbulenzintensität der Umgebung. Das Geschwindigkeitsdefizit an einer Position $[x, y, z]^T$ lässt sich durch

$$\Delta u(x, y, z) = u_\infty \cdot \left[\frac{1}{\left(a + \frac{bx}{d_R} + c \left(1 + \frac{x}{d_R} \right)^{-2} \right)^2} \cdot \exp \left(-\frac{r^2}{2\sigma(x)^2} \right) \right] \quad (2.42)$$

ausdrücken, wobei

$$r = \sqrt{y^2 + (z - h_{\text{Nabe}})^2} \quad (2.43)$$

gilt und die noch nicht genannten empirischen Parameter nach

$$\begin{aligned} \frac{\sigma(x)}{d_R} &= k^* \frac{x}{d_R} + \varepsilon, & k^* &= 0,11 c_T^{1,07} I_0^{0,2}, & \varepsilon &= 0,23 c_T^{-0,25} I_0^{0,17}, \\ a &= 0,93 c_T^{-0,75} I_0^{0,17}, & b &= 0,42 c_T^{0,6} I_0^{0,2}, & c &= 0,15 c_T^{-0,25} I_0^{-0,7} \end{aligned} \quad (2.44)$$

bestimmt werden. Neben dem Geschwindigkeitsdefizit existiert in diesem Modell auch ein Ansatz für die added-, also die induzierte Turbulenzintensität. Diese berechnet sich in einem Punkt $[x, y, z]^T$ hinter dem Rotor nach

$$I_+(x, y, z) = \frac{\Delta \sigma_u(x, y, z)}{u_\infty} = \frac{1}{d + e \frac{x}{d_R} + f \left(1 + \frac{x}{d_R} \right)^{-2}} \left[k_1 \exp \left(-\frac{\left(r - \frac{d_R}{2} \right)^2}{2\sigma^2} \right) + k_2 \exp \left(-\frac{\left(r + \frac{d_R}{2} \right)^2}{2\sigma^2} \right) \right] - \Delta I_a \quad (2.45)$$

und muss für die effektive Turbulenzintensität, wie bereits in Gl. 2.22 erwähnt, mit der Umgebungsturbulenzintensität verrechnet werden.

Der Korrekturterm ΔI_a für die Rotorfläche unterhalb der Nabe ergibt sich zu

$$\Delta I_a = \begin{cases} 0 & \text{sonst} \\ I_a \sin^2 \left(\pi \frac{h_{\text{Nabe}} - z}{d_R} \right) \cos^2 \left(\pi \frac{y}{d_R} \right) & z \leq 0 \wedge |y| \leq d_R. \end{cases} \quad (2.46)$$

Die weiteren, bisher noch nicht erwähnten empirischen Konstanten, ergeben sich nach

$$d = 2,3 c_T^{-1,2}, \quad e = I_a^{0,1}, \quad f = 0,7 c_T^{-3,2} I_a^{-0,45},$$

$$k_1 = \begin{cases} \cos^2 \left(\frac{\pi}{2} \left(\frac{r}{d_R} - 0,5 \right) \right) & \frac{r}{d_R} \leq 0,5 \\ 1 & \frac{r}{d_R} > 0,5 \end{cases}, \quad k_2 = \begin{cases} \cos^2 \left(\frac{\pi}{2} \left(\frac{r}{d_R} + 0,5 \right) \right) & \frac{r}{d_R} \leq 0,5 \\ 0 & \frac{r}{d_R} > 0,5. \end{cases}$$

2.4.7 Weitere Ansätze

Die in dieser Arbeit vorgestellten analytischen Ansätze sind die in der Literatur am weitesten verbreiteten. Selbstverständlich gibt es zahlreiche Modifikationen, bspw. bezüglich empirischen Konstanten zur Berechnung der Turbulenzintensitäten, oder auch weitere teils komplexe Ansätze. H. Sun bspw. entwickelte ebenfalls ein dreidimensionales Wake-Modell, welches allerdings analytisch komplexer als die bisher vorgestellten ist. Es berücksichtigt im Gegensatz zu den anderen Modellen auch den Höhenwindgradienten, sprich die unterschiedlichen Windgeschwindigkeiten in verschiedenen Höhen des Wakes von $(h_{\text{Nabe}} - r_w)$ bis $(h_{\text{Nabe}} + r_w)$. [20]

Dhiman u. a. beschäftigte sich mit der Vorhersage der Windgeschwindigkeit im Wake. Er entwickelte dazu ebenfalls ein Gauss-Modell zur Berechnung der Geschwindigkeit im Wake und verwendete dieses zusammen mit einem Algorithmus des maschinellen Lernens (Support Vector Regression), um eine Vorhersage der Windgeschwindigkeit zu treffen. [12]

Reinwardt benutzte ein sog. DWM (dynamic wake meandering)-Modell und optimierte dieses mit Daten aus LiDAR-Messungen im Windpark Curslack. Dadurch konnte die Genauigkeit bezüglich Windcharakteristik im Nachlauf von Anlagen für kleine Anlagenabstände erhöht werden. [54]

2.4.8 Zusammenfassung und Vergleich der Nachlaufmodelle

Dass Modelle wie das nach Jensen oder Frandsen u. a. für das Geschwindigkeitsdefizit eine zylindrische Verteilung annehmen, spiegelt sich auch in den Validierungsergebnissen wider. Die Modifizierung des Jensen-Modells mit einer Annäherung des Geschwindigkeitsprofils durch eine cos-Funktion sowie der Erweiterung auf den zweidimensionalen Raum und später auch mit einer Anpassung der Konstante k ermöglichen hier weitaus bessere Ergebnisse. Es sind noch immer Abweichungen zu den gemessenen Daten vorzufinden. Im Vergleich zum Standard Jensen-Modell allerdings kann das Geschwindigkeitsdefizit für verschiedene Distanzen hinter dem Rotor gut angenähert werden. [58, S. 95]

Eine weitere Verbesserung bietet hier das Jensen-Gauss-Modell. Statt einer cos-Funktion wird hier eine Gauss-Verteilung der Windgeschwindigkeit angenommen, was für manche Schubbeiwerte und Abstände hinter dem Rotor zu besseren Ergebnissen führt. [60, S. 196]

Ob das Jensen-2D-Modell oder das Jensen-2D-k-Modell bessere Ergebnisse liefern, hängt von Faktoren wie der Größe des Schubbeiwertes und der Turbulenzintensität sowie der Abgrenzung von *near wake* und *far wake* ab. [58, S. 98]

Genauso wie das Jensen-Gauss-Modell nimmt auch das Bastankhah-Modell eine Gauss-Verteilung der Windgeschwindigkeit an, was zu ähnlich guten Ergebnissen führt. Allerdings vergleicht M. Bastankhah seine Ergebnisse nur mit den Standard Modellen nach Jensen und Frandsen. Nichtsdestotrotz gibt es eine große Übereinstimmung zwischen gemessenen LES Daten und den vom Bastankhah-Modell berechneten. [42, S. 121]

Das Modell nach Ishihara ist weitaus komplexer, da sowohl das Geschwindigkeitsdefizit mit einer Gauss-Verteilung als auch die Turbulenzintensität an einer Position im Wake berechnet wird. Verglichen werden die Ergebnisse überwiegend mit LES und experimentellen Daten und liefern annehmbare Übereinstimmung. Auch die Berechnung der Turbulenz wird sowohl mit experimentellen und LES Daten, als auch mit dem Turbulenzmodell nach A. Crespo und

2 Theoretische Grundlagen

Frandsen u. a. verglichen und liefert für verschiedene Umgebungsturbulenzintensitäten gute Ergebnisse [57, S. 288 f.].

Einen sehr ausführlichen Vergleich der Wake Modelle führten Kaldellis u. a. durch. Diese stellten die Modelle tabellarisch gegenüber und zeigten auf, welche Parameter in welchen Modellen in Betracht gezogen werden. Recht deutlich wird, dass das Ishihara-Modell mit zu den komplexesten Modellen gehört, da dieses dreidimensional ist und eine Abschätzung der Turbulenzintensität im Nachlauf ermöglicht. [33, S. 4]

Auch in den Ergebnissen der drei durchgeführten Fallstudien schnitt das Ishihara-Modell am besten ab. Es konnte eine durchschnittliche Abweichung zu gemessenen Daten von weniger als 9% verzeichnen, wobei hinzugefügt werden muss, dass auch die anderen Modelle akzeptable Ergebnisse lieferten [33, S. 15].

2.5 Modellierung von Mehrfachabschattungen

Überlagern sich in einem Punkt Wakes von mehreren Anlagen, so existieren Ansätze, die resultierende Geschwindigkeit in diesem Punkt zu berechnen. Wie auch bei den Modellen der einfachen Abschattung existieren in der Literatur zahlreiche analytische Modelle, die in dieser Arbeit nicht alle ausführlich aufgeführt sind. Der Fokus liegt in diesem Abschnitt auf dem Prinzip der Superposition, speziell der rotorbasierten Superposition nach dem Qian-Ishihara-Modell [18]. Konservativere Ansätze werden am Schluss erwähnt. Auch hier gelten die genannten Quellen für den gesamten theoretischen Abschnitt.

2.5.1 Prinzip der Superposition

Ein weit verbreitetes Prinzip ist das der Superposition. Vier grundlegende Varianten werden unterschieden: [18, S. 459]

$$\text{Geometrische Summe: } \frac{u_w}{u_\infty} = \prod_{i=1}^N \frac{u_{iw}}{u_\infty}, \quad (2.47)$$

$$\text{Energiebilanz: } u_\infty^2 - u_w^2 = \sum_{i=1}^N (u_\infty^2 - u_{iw}^2), \quad (2.48)$$

$$\text{Lineare Summe: } \left(1 - \frac{u_w}{u_\infty}\right) = \sum_{i=1}^N \left(1 - \frac{u_{iw}}{u_\infty}\right), \quad (2.49)$$

$$\text{Wurzelsummenquadrat: } \left(1 - \frac{u_w}{u_\infty}\right)^2 = \sum_{i=1}^N \left(1 - \frac{u_{iw}}{u_\infty}\right)^2, \quad (2.50)$$

wobei die letzten zwei im Folgenden näher betrachtet werden. Hierbei ist u_w die gesuchte Windgeschwindigkeit an einer beliebigen Stelle im N -fach abgeschatteten Bereich und u_{iw} die Windgeschwindigkeit an dieser Stelle, erzeugt durch den Wake von Turbine i . N bezeichnet die Anzahl der Turbinen, die stromaufwärts von der betrachteten Stelle positioniert sind und somit einen Einfluss haben können.

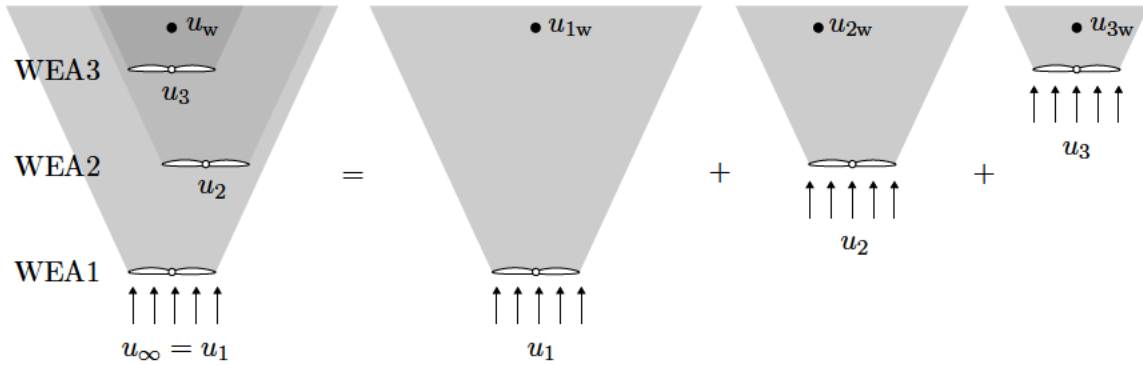


Abbildung 2.5: Prinzip der Superposition von Strömungsnachläufen

Werden bspw. drei Turbinen betrachtet, die in Bezug auf die Windrichtung hintereinander positioniert sind, lässt sich das allgemeine Superpositionsprinzip visualisieren.

Die Geschwindigkeit u_w wird vom Wake aller stromaufwärts positionierten Anlagen beeinflusst (WEA1, WEA2, WEA3). Zur Berechnung wird daher zuerst separat der Einfluss von Turbine 1, Turbine 2 und Turbine 3 auf u_w betrachtet und schließlich superpositioniert.

Für die Berechnung des Einflusses, also des Wakes, von z.B. Turbine 3, wird nicht die wirkliche Geschwindigkeit u_3 verwendet, sondern die Umgebungswindgeschwindigkeit u_1 . Selbiges gilt für die Turbulenzintensität. Auch hier wird nicht die jeweilige Intensität vor jedem Rotor i , sondern die Umgebungsturbulenzintensität I_0 betrachtet. [18, S. 460]

2.5.2 Ansatz nach Qian-Ishihara

G.-W. Qian hingegen argumentiert, dass der Bezug zu den Umgebungsparametern nur gezogen werden darf, wenn die Anlagen sehr weit auseinander stehen, da sonst der gegenseitige Einfluss zweier Anlagen verloren geht. Aus diesem Grund differenziert G.-W. Qian zwischen umgebungsbasierter Superposition (wie im vorherigen Abschnitt vorgestellt) und rotorbasierter Superposition [18] anhand der linearen Summe und des Wurzelsummenquadrates.

Umgebungsbasierte Superposition

Werden Gl. 2.49 und Gl. 2.50 nach der gesuchten Geschwindigkeit u_w umgestellt, so ergeben sich:

$$\text{Lineare Summe:} \quad u_w = u_\infty - \sum_{i=1}^N \left(u_\infty \left(1 - \frac{u_{iw}}{u_\infty} \right) \right), \quad (2.51)$$

$$\text{Wurzelsummenquadrat:} \quad u_w = u_\infty - \sqrt{\sum_{i=1}^N \left(u_\infty \left(1 - \frac{u_{iw}}{u_\infty} \right)^2 \right)}. \quad (2.52)$$

2 Theoretische Grundlagen

Alternativ formuliert ergibt sich für die umgebungs-basierte Superposition (linear und quadratisch) schlussendlich

$$u_w = u_\infty - \sum_{i=1}^N (\Delta u_{iw}) \quad \text{mit} \quad \Delta u_{iw} = u_\infty - u_{iw} = u_\infty \left(1 - \frac{u_{iw}}{u_\infty}\right) \quad (2.53)$$

sowie

$$u_w = u_\infty - \sqrt{\sum_{i=1}^N (\Delta u_{iw}^2)} \quad \text{mit} \quad \Delta u_{iw} = u_\infty - u_{iw} = u_\infty \left(1 - \frac{u_{iw}}{u_\infty}\right). \quad (2.54)$$

Eine Substitution wie diese ist sinnvoll, da z.B. im Bastankhah- oder Ishihara-Modell bereits ein Δu_{iw} berechnet wird. Δu_{iw} wird hierbei nach dem bereits vorgestellten Single-Wake-Modell nach Ishihara berechnet.

Rotorbasierte Superposition

Bei der rotorbasierten Superposition werden die Geschwindigkeiten u_{iw} nicht auf die Umgebungsgeschwindigkeit u_∞ , sondern auf die real vorherrschende Geschwindigkeit u_i am Rotor i bezogen. Selbiges gilt für die Turbulenzintensität.

Damit ergibt sich dann für die Definition von Δu_{iw} folgende Gleichung:

$$\Delta u_{iw} = u_i - u_{iw} = u_i \left(1 - \frac{u_{iw}}{u_i}\right). \quad (2.55)$$

Hierbei ist u_i nun die Windgeschwindigkeit vor dem Rotor i und u_{iw} die Geschwindigkeit an der, aufgrund des Wakes, zu untersuchenden Stelle von Turbine i .

Δu_{iw} ist prinzipiell genau so wie bei der umgebungs-basierten Superposition definiert, nur mit Bezug auf die Geschwindigkeit u_i vor dem jeweiligen Rotor i statt der Umgebungsgeschwindigkeit u_∞ . Aus diesem Grund wird für die Berechnung der Einzel-Wakes (siehe z.B. Gl. 2.42) statt u_∞ auch mit u_i gerechnet.

Letztendlich kann also für die rotorbasierte Superposition (linear und quadratisch)

$$u_w = u_\infty - \sum_{i=1}^N (\Delta u_{iw}) \quad \text{mit} \quad \Delta u_{iw} = u_i - u_{iw} = u_i \left(1 - \frac{u_{iw}}{u_i}\right) \quad (2.56)$$

sowie

$$u_w = u_\infty - \sqrt{\sum_{i=1}^N (\Delta u_{iw}^2)} \quad \text{mit} \quad \Delta u_{iw} = u_i - u_{iw} = u_i \left(1 - \frac{u_{iw}}{u_i}\right) \quad (2.57)$$

geschrieben werden. Analog dazu wird anstelle der Umgebungsturbulenzintensität I_0 die jeweilige Turbulenzintensität $I_{a,i}$ benutzt, die sich aus Gl. 2.45 und der Umgebungsturbulenz berechnet.

2.5.3 Rotorbasiertes Multiple-Wake-Modell für Windparks nach Qian-Ishihara

Abb. 2.5 zeigt, dass für die Berechnung des Einflusses von WT3 auf u_w in der rotorbasierten Variante die Parameter (Geschwindigkeit und Turbulenzintensität) u_3 und $I_{a,3}$ vor WEA3 benötigt werden, welche wiederum von WEA1 und WT2 abhängen.

Aus diesem Grund ist es in dieser Variante wichtig, alle Parameter sukzessiv von der in Windrichtung ersten Turbine nach hinten zu berechnen. Hierfür stellt G.-W. Qian einen Algorithmus vor [18], der es ermöglicht, im Hinblick auf das spätere Ziel dieser Arbeit, die jeweilige Geschwindigkeit vor jedem Rotor j eines Windparks und somit die Leistungen zu berechnen.

Da das bisher vorgestellte Ishihara-Modell (vgl. Abschnitt 2.4.6) nur für einzelne Wakes galt, bedarf es einiger Anpassungen der Indizes, um es konsequent in einem Windpark anwenden zu können. Die im Folgenden hergeleiteten Gleichungen sind, wenn nicht ausdrücklich anders gekennzeichnet, G.-W. Qian entnommen [18].

Die Geschwindigkeiten vor einem Rotor j werden als u_j definiert. Das Geschwindigkeitsdefizit, welches von Turbine i erzeugt wird und Einfluss auf die Geschwindigkeit vor Turbine j hat, wird mit Δu_{ij} bezeichnet. Es hängt von den Parametern der Turbine i ab.

Somit modifiziert sich Gl. 2.42 zur Berechnung eines Single-Wakes innerhalb eines Windparks zu der Form

$$\Delta u_{ij} = u_i \cdot f(I_{a,i}, c_{T,i}, (x - x_i), r_i, \sigma_i). \quad (2.58)$$

Die Geschwindigkeit u_j vor einem Rotor j kann dann über das rotorbasierte Superpositionsprinzip (vgl. Gl. 2.57)

$$u_j = u_\infty - \sqrt{\sum_{i=1}^N (\Delta u_{ij}^2)} \quad (2.59)$$

berechnet werden. N ist hierbei die Anzahl aller Anlagen, die vor der zu untersuchenden Anlage j stehen.

Analog zum Geschwindigkeitsdefizit lässt sich auch die Änderung der Standardabweichung der Windgeschwindigkeit (vgl. Gl. 2.45) betrachten. In einem Windpark modifiziert sie sich zu

$$\Delta \sigma_{u,ij} = u_i \cdot f(I_{a,i}, c_{T,i}, (x - x_i), r_i, \sigma_i). \quad (2.60)$$

Hierbei ist demnach $\Delta \sigma_{u,ij}$ die Änderung der Standardabweichung der Windgeschwindigkeit am betrachteten Punkt vor Turbine j , erzeugt von Rotor i . An dieser Stelle sei erneut darauf hingewiesen, dass die Breite des Wakes nicht mit der Standardabweichung der Turbulenz zu verwechseln ist.

Um die effektive Standardabweichung der Windgeschwindigkeit $\sigma_{u,j}$ vor Turbine j zu bestimmen, bedarf es ebenfalls der Superposition

$$\sigma_{u,j}^2 = \sigma_{u,0}^2 + \sum_{i=1}^n (\Delta \sigma_{u,ij} + \Delta \sigma_{u,i,\text{Korr}})^2. \quad (2.61)$$

Hierbei ist $\sigma_{u,0}$ die Standardabweichung der Windgeschwindigkeit in der Umgebung und $\Delta \sigma_{u,i,\text{Korr}}$ ein Korrekturterm. Er dient dazu, die Änderung der Standardabweichung der Windgeschwindigkeit $\Delta \sigma_{u,ij}$ von Turbine i zu erhöhen oder zu verringern, je nachdem, wie Anlage i von der am

2 Theoretische Grundlagen

nächsten stromaufwärts positionierten Anlage (im Folgenden Anlage k genannt) abgeschattet wird. Unterschieden werden der sog. „full wake“, also eine komplette Abschattung der Anlage und der „partial wake“, also eine teils abgeschattete Anlage. Die nachfolgende Abbildung veranschaulicht die beiden möglichen Fälle.

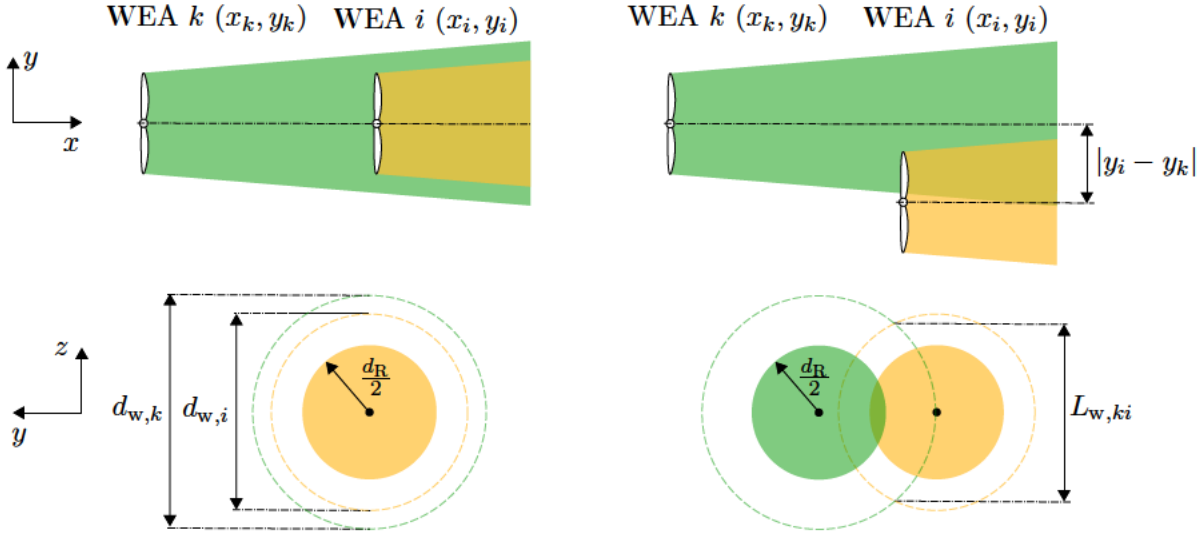


Abbildung 2.6: Schematische Zeichnung einer Anlage im „full wake“ (links) und im „partial wake“ (rechts) einer anderen Anlage (eigene Darstellung in Anlehnung an [59])

Zunächst lassen sich „full wake“ und „partial wake“ für die betrachtete Anlage i in Relation zur Position der nächsten Anlage stromaufwärts k über

$$A_{F,1} = \left\{ (x_i, y_i) \mid 0 < x_i - x_k < 15 d_R, |y_i - y_k| \leq \frac{d_{w,k}(x_i) - d_R}{2} \right\} \quad (2.62)$$

und

$$A_{P,1} = \left\{ (x_i, y_i) \mid 0 < x_i - x_k < 15 d_R, \frac{d_{w,k}(x_i) - d_R}{2} < |y_i - y_k| \leq \frac{d_{w,k}(x_i) + d_R}{2} \right\} \quad (2.63)$$

definieren. $A_{F,1}$ sowie $A_{P,1}$ (und im Folgenden $A_{F,2}$ sowie $A_{P,2}$) sind Mengen von Koordinatenwertepaaren (F für full wake, P für partial wake). (x_i, y_i) und (x_k, y_k) sind demnach die Koordinaten des Rotormittelpunktes von Anlage i und k . $d_{w,k}(x_i)$ bezeichnet den Durchmesser des Wakes, erzeugt von Anlage k , an der Position der Anlage i . Er ergibt sich nach

$$d_{w,k}(x_i) = 4 \sqrt{2 \ln(2)} \sigma \quad \text{mit} \quad \sigma(x_i - x_k). \quad (2.64)$$

σ wird hier nach Gl. 2.44 aus dem Single-Wake-Modell berechnet. Wie in den Bedingungen zu erkennen ist, wird die Interaktion der Wakes vernachlässigt, wenn die Anlagen in Strömungsrichtung mehr als $15D$ auseinander stehen. Für die Distanz der Anlagen in y -Richtung ($|y_i - y_k|$) müssen die aufgeführten geometrischen Bedingungen gelten, damit die Anlage i entweder im „full wake“ oder im „partial wake“ der Anlage k steht.

Des Weiteren bedarf es einer definierten Region, in der der eingeführte Korrekturterm angewendet werden darf. Diese ergeben sich für den „full wake“ nach

$$A_{F,2} = \left\{ (y, z) \mid r_i \leq \frac{d_R}{2} \right\} \quad (2.65)$$

und für den „partial wake“ nach

$$A_{P,2} = \left\{ (y, z) \mid |y - y_i| \leq \frac{d_{w,i}}{2}, |z - h_{\text{Nabe}}| \leq \frac{L_{w,ki}}{2}, \text{sgn}(y - y_i) \text{sgn}(y_k - y_i) = 1 \right\}. \quad (2.66)$$

Dabei lässt sich der Radius r_i , also die Entfernung vom Rotormittelpunkt des Rotors i , wie bereits in vorherigen Abschnitten erwähnt über

$$r_i = \sqrt{(y - y_i)^2 + (z - h_{\text{Nabe}})^2} \quad (2.67)$$

berechnen. Der vertikale Abstand der beiden Schnittpunkte der sich überschneidenden Wakes von Turbine k und i lässt sich geometrisch über

$$L_{w,ki} = \frac{1}{|y_i - y_k|} \sqrt{(y_i - y_k)^2 d_{w,k}^2 - \left((y_i - y_k)^2 - \frac{d_{w,i}^2}{4} + \frac{d_{w,k}^2}{4} \right)^2} \quad (2.68)$$

ausdrücken. $d_{w,k}$ und analog $d_{w,i}$ lassen sich über

$$d_{w,k} = 4 \sqrt{2 \ln(2)} \sigma \quad \text{mit} \quad \sigma(x - x_k) \quad (2.69)$$

berechnen. Letztendlich kann der Korrekturterm dann abhängig von diesen Bedingungen definiert werden. Es wird angenommen, dass dieser halb so groß wie die Änderung der Standardabweichung der Windgeschwindigkeit $\Delta\sigma_{u,i,\text{tip}}$ an der Spitze des Rotors i bei $y = 0$ und $z = \frac{d_{R,i}}{2}$ ist. Außerdem wird er durch eine trigonometrische Funktion geglättet:

$$\Delta\sigma_{u,i,\text{Korr}} = \begin{cases} \frac{1}{2} \Delta\sigma_{u,i,\text{tip}} \cos^2 \left(\frac{\pi r_i}{d_{R,i}} \right) & (x_i, y_i) \in A_{F,1} \wedge (y, z) \in A_{F,2} \\ -\frac{1}{2} \Delta\sigma_{u,i,\text{tip}} \sin^2 \left(\frac{\pi(y-y_i)}{d_{w,i}} \right) \cos^2 \left(\frac{\pi(z-h_{\text{Nabe}})}{L_{w,ki}} \right) & (x_i, y_i) \in A_{P,1} \wedge (y, z) \in A_{P,2} \\ 0 & \text{sonst.} \end{cases} \quad (2.70)$$

Mit Hilfe von $\sigma_{u,j}$ (Gl. 2.61) kann jetzt die Turbulenzintensität $I_{a,j}$ über

$$I_{a,j} = \frac{\sigma_{u,j}}{u_j} \quad (2.71)$$

berechnet werden. Für den Fall, dass $I_{a,j} \leq 0,03$ ist, gilt $I_{a,j} = 0,03$, da das vorgestellte Modell nur für $I_{a,j} \geq 0,03$ gültig ist.

Es sei erwähnt, dass dieses Vorgehen nur für einen Iterationsschritt, also für eine betrachtete Anlage j gilt. Die errechnete Geschwindigkeit u_j und Turbulenzintensität $I_{a,j}$ vor Rotor j werden im nächsten Schritt, also quasi beim $(j + 1)$ -ten Rotor selbstverständlich mit dem Index i ersetzt.

2 Theoretische Grundlagen

Zwecks besserem Verständnis zeigt die nachfolgende Abbildung den bisher beschriebenen Algorithmus als Ablaufdiagramm exemplarisch für drei hintereinander stehende Anlagen.

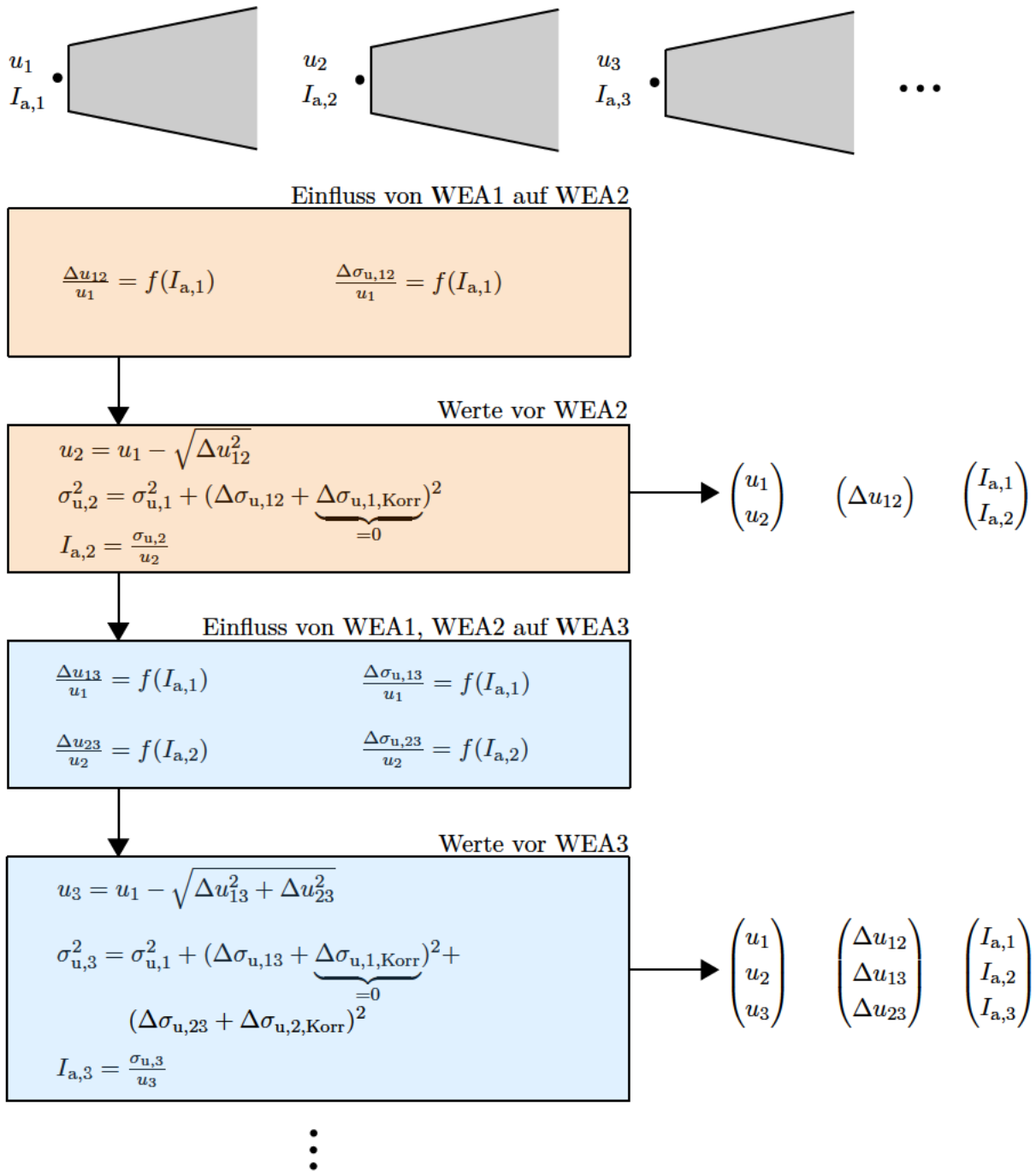


Abbildung 2.7: Exemplarisches Ablaufdiagramm des Algorithmus für drei hintereinander positionierte Anlagen

Die Windgeschwindigkeit vor Anlage 1 ist die Umgebungswindgeschwindigkeit. Sie muss nicht berechnet werden, da Anlage 1 die vorderste Anlage ist. Der orange gefärbte Bereich kennzeichnet die Berechnung des Einflusses aller relevanten Anlagen auf Anlage 2, in dem Fall nur von Anlage

1 sowie die anschließende Berechnung der Windgeschwindigkeit und der Turbulenzintensität vor Anlage 2. Danach folgt der blau gefärbte Bereich. Er kennzeichnet analog die Berechnung aller Einflüsse auf Anlage 3, in dem Fall von Anlage 1 und 2, sowie der genannten Parameter vor Anlage 3.

Eine letzte noch nicht erwähnte Modifikation des Modells ist die Mittelung der Parameter über die Rotorfläche. Die meisten Modelle gehen von einer gleichförmigen Strömung in y und z -Richtung aus, die in der Rotorebene herrscht, was aber auf einen realen Windpark nicht zutrifft, da auf Grund des Nachlaufes von Anlagen eine nicht einheitliche Strömung herrscht. [18, S. 3] Aus diesem Grund werden äquivalente Geschwindigkeiten und Turbulenzintensitäten gebildet. Die äquivalente Geschwindigkeit ergibt sich durch eine geometrische Mittelung

$$u_{j,\text{equiv}} = \frac{1}{A_{R,j}} \int_{\text{Rotor}} u_j(x_j, y, z) \, dA = \frac{1}{A_{R,j}} \int_0^{2\pi} \int_0^{\frac{d_{R,j}}{2}} u_j(x_j, y, z) \, dr \, d\varphi \quad (2.72)$$

der Geschwindigkeiten über die Rotorfläche. Hierbei sind u_j die errechneten Geschwindigkeiten in der Rotorfläche und $A_{R,j}$ die Rotorfläche selbst. Die äquivalente Turbulenzintensität ergibt sich über das quadratische Mittel der effektiven Standardabweichungen der Windgeschwindigkeit dividiert durch die äquivalente Geschwindigkeit $u_{j,\text{equiv}}$ nach

$$\begin{aligned} I_{a,j,\text{equiv}} &= \frac{1}{A_{R,j} u_{j,\text{equiv}}} \sqrt{\int_{\text{Rotor}} \sigma_{u,j}^2(x_j, y, z) \, dA} \\ &= \frac{1}{A_{R,j} u_{j,\text{equiv}}} \sqrt{\int_0^{2\pi} \int_0^{\frac{d_{R,j}}{2}} \sigma_{u,j}^2(x_j, y, z) \, dr \, d\varphi}. \end{aligned} \quad (2.73)$$

Hierbei sind $\sigma_{u,j}^2$ die errechneten quadrierten effektiven Standardabweichungen der Windgeschwindigkeit in der Rotorfläche (vgl. Gl. 2.61).

Bei der sukzessiven Berechnung der Geschwindigkeiten und Turbulenzintensitäten vor jedem Rotor j werden demnach in der gesamten Rotorfläche Werte für u_j und $\sigma_{u,j}$ nach Gl. 2.59 und Gl. 2.61 errechnet und schließlich äquivalente Werte gebildet, mit denen anschließend weiter gerechnet wird.

2.5.4 Weitere Verfahren

Neben den erwähnten Verfahren gibt es weitere konservativere Ansätze. Die IEC (International Electrotechnical Commission) Richtlinie 61400-1 Ed. 4 [24, E.2.5] schlägt eine Methode vor, die zwischen Windgeschwindigkeiten unter der Nennwindgeschwindigkeit und oberhalb dieser differenziert. Für den zweiten Fall sieht die Richtlinie vor, die Geschwindigkeitsdefizite linear zu superpositionieren (siehe Gl. 2.49). Für den Fall unterhalb der Nenngeschwindigkeit allerdings schlägt die Richtlinie vor, den Einfluss aller stromaufwärts positionierten Anlagen an einem Punkt separat zu betrachten und die niedrigste Windgeschwindigkeit zu benutzen. Es gilt dann

$$u_w(x, y, z) = \min_i(u_i(x, y, z)), \quad (2.74)$$

wobei u_i die Windgeschwindigkeit am zu untersuchenden Punkt, beeinflusst von Anlage i , ist.

Ein weiterer Ansatz ist das Modell nach Keck [35, S. 87 ff.]. Es wird unter anderem in Bezug auf das in Abschnitt 2.4.7 bereits erwähnte DWM-Modell betrachtet. Im Modell nach Keck werden die Geschwindigkeiten vor den Turbinen sukzessive von stromaufwärts zu stromabwärts berechnet, allerdings ohne Einflüsse separat zu betrachten und diese schließlich zu superpositionieren. Die Bedingungen für die Strömung vor jeder Anlage j ändern sich somit bei jedem Berechnungsschritt und sind dann jeweils abhängig von der stromaufwärts positionierten Anlage $j - 1$.

2.6 Koordinatentransformation

Die Herleitung sowie die Gleichungen der Koordinatentransformation stammen aus Siciliano u. a. [56, S. 10 ff.].

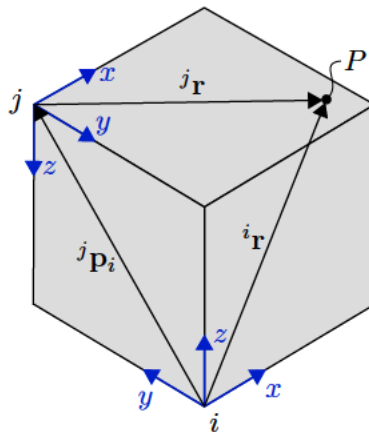


Abbildung 2.8: Darstellung von zwei Koordinatensystemen i und j inklusive aller relevanten Vektoren

Gegeben sei ein Vektor ${}^i r$ im Koordinatensystem i . Um diesen Vektor in einem zweiten Koordinatensystem j auszudrücken, muss die relative Position (translatorische Verschiebung) und Orientierung (Drehung um die Koordinatenachsen) bekannt sein.

Die Position des Ursprungs vom i -System in Bezug auf das j -System wird durch einen Vektor

$${}^j p_i = \left[{}^j p_i^x, {}^j p_i^y, {}^j p_i^z \right]^T \quad (2.75)$$

angegeben. Die relative Orientierung wird in Form einer Rotationsmatrix ${}^j\mathbf{R}_i$ angegeben. Diese setzt sich nach

$${}^j\mathbf{R}_i = \mathbf{R}_x(\theta) \cdot \mathbf{R}_y(\theta) \cdot \mathbf{R}_z(\theta) \quad (2.76)$$

zusammen, wobei $\mathbf{R}_x(\theta)$, $\mathbf{R}_y(\theta)$, $\mathbf{R}_z(\theta)$ die Elementardrehungen um x , y und z sind. Diese berechnen sich nach

$$\begin{aligned} \mathbf{R}_x(\theta) &= & \mathbf{R}_y(\theta) &= & \mathbf{R}_z(\theta) &= \\ \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta) & -\sin(\theta) \\ 0 & \sin(\theta) & \cos(\theta) \end{bmatrix}, & & \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{bmatrix}, & & \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}. \end{aligned} \quad (2.77)$$

Der Vektor ${}^j\mathbf{r}$ im j -System lässt sich dann mittels

$$\begin{bmatrix} {}^j\mathbf{r} \\ 1 \end{bmatrix} = {}^j\mathbf{T}_i \begin{bmatrix} {}^i\mathbf{r} \\ 1 \end{bmatrix} = \begin{bmatrix} {}^j\mathbf{R}_i & {}^j\mathbf{p}_i \\ \mathbf{0}^T & 1 \end{bmatrix} \cdot \begin{bmatrix} {}^i\mathbf{r} \\ 1 \end{bmatrix} \quad (2.78)$$

berechnen. ${}^j\mathbf{T}_i$ wird Transformationsmatrix genannt und enthält alle Informationen zwischen dem i - und dem j -System.

2.7 Neuronale Netze

Künstliche neuronale Netze sind ein Teilgebiet des maschinellen Lernens und Forschungsgegenstand im Bereich der künstlichen Intelligenz. Sie lernen anhand von meist sehr großen Trainingsdatensätzen (Eingänge und korrespondierende Ausgänge) selbstständig, um später anhand einer für das Netz völlig unbekanntem Eingabe die richtige Ausgabe zu finden.

2.7.1 Aufbau eines Neurons in einem neuronalen Netz

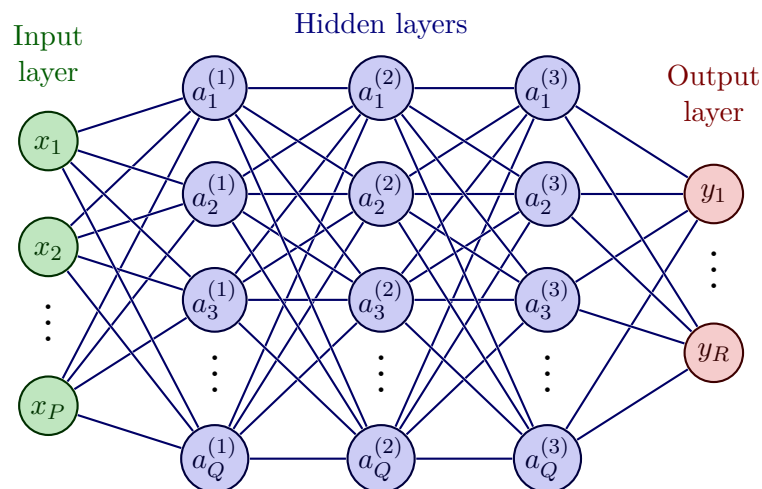


Abbildung 2.9: Allgemeine Struktur eines künstlichen neuronalen Netzes (eigene Darstellung in Anlehnung an [8, S. 152])

2 Theoretische Grundlagen

Zu Beginn werden einige Variablen und Notationen definiert. Die allgemeine Struktur eines solchen künstlichen neuronalen Netzes kann Abb. 2.9 entnommen werden. Es besteht aus mehreren Schichten, welche mit sog. Neuronen gefüllt sind. Jedes Neuron einer Schicht l ist verbunden mit jedem Neuron der Schicht $(l - 1)$ sowie $(l + 1)$.

Die erste Schicht ist hierbei die Eingabeschicht und beinhaltet die Eingabevariablen in der Form $[x_1, x_2, \dots, x_P]^T$. Die Ausgabeschicht enthält die vom neuronalen Netz berechneten Ausgaben in der Form $[y_1, y_2, \dots, y_R]^T$.

Dazwischen befinden sich sog. versteckte Schichten, dessen Anzahl variabel ist. Der Ausgang des j -ten Neurons in der l -ten versteckten Schicht, welcher als Eingang für jedes Neuron in der $(l + 1)$ -ten Schicht fungiert, wird mit a_j^l bezeichnet. Die Verbindungen zwischen zwei Neuronen werden Gewichtungen w genannt. Hierbei bezeichnet w_{jk}^l die Gewichtung vom k -ten Neuron in der $(l - 1)$ -ten Schicht zum j -ten Neuron in der l -ten Schicht.

Wird ein Neuron des Netzes genauer betrachtet (bspw. das zweite Neuron ($j = 2$) aus der ersten versteckten Schicht ($l = 1$)) und wird von drei Eingängen x_1, x_2 und x_3 sowie den dazugehörigen Gewichtungen w_{21}^1, w_{22}^1 und w_{23}^1 und einem Schwellenwert b^1 ausgegangen,

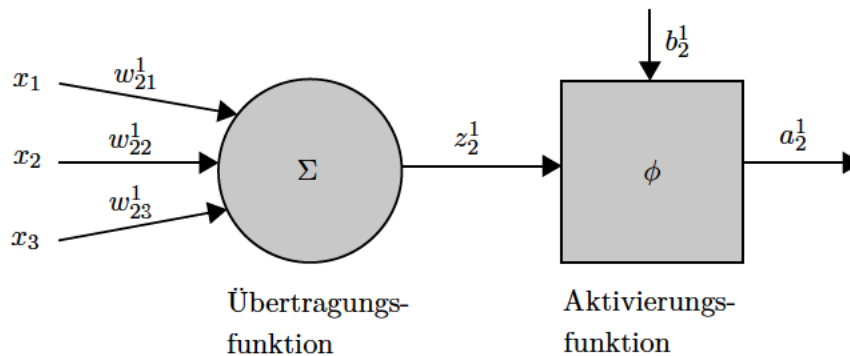


Abbildung 2.10: Prinzip eines Neurons in einem künstlichen neuronalen Netz (eigene Darstellung in Anlehnung an [53, S. 52])

so kann zunächst mit Hilfe einer Übertragungsfunktion eine summierte Gewichtung

$$z_2^1 = \sum_{i=1}^3 w_{2i}^1 x_i \quad (2.79)$$

berechnet werden [45, S. 41]. Mit Hilfe der Aktivierungsfunktion ϕ kann nun bestimmt werden, ob das Perzeptron aktiviert wird oder nicht. Hierzu bedarf es den Schwellenwert b_2^1 . Für die Aktivierung gilt [45, S. 42]

$$a_2^1 = f(z) = \begin{cases} 0 & \text{wenn } z > b_2^1 \\ 1 & \text{wenn } z \leq b_2^1 \end{cases} \quad (2.80)$$

Die Ausgabe kann dann ebenfalls als

$$a_2^1 = \begin{cases} 1 & \text{wenn } \sum_{i=1}^3 w_{2i}^1 x_i \leq 0 \\ 0 & \text{wenn } \sum_{i=1}^3 w_{2i}^1 x_i > 0 \end{cases} \quad (2.81)$$

formuliert werden.

Neben dem Prinzip eines Perzeptrons, welches nur Werte von entweder 0 oder 1 ausgeben kann, gibt es weitere Arten von Neuronen wie z.B. das tanh-Neuron [47, S. 121]. Die Ausgabe eines solchen Neurons ist als

$$a_2^1 = \tanh \left(\sum_{i=1}^3 w_{2i}^1 x_i + b \right) \quad (2.82)$$

definiert. Im Gegensatz zum Perzeptron kann das tanh-Neuron Werte im Bereich $[-1; 1]$ ausgeben. Eine minimale Änderung in den Gewichtungen oder im Schwellenwert erzeugt somit nur eine minimale Änderung des Ausgangs und springt nicht ab einer gewissen Schwelle von z.B 0 auf 1.

2.7.2 Stochastisches Gradientenabstiegsverfahren

Um den Lernprozess eines neuronalen Netzes zu quantifizieren, bedarf es einer sog. Kostenfunktion. Diese gibt an, wie groß die Abweichung der Ausgabe des Netzes im Vergleich zur gewünschten Ausgabe ist. Ein Beispiel ist die quadratische Kostenfunktion

$$C(\mathbf{w}, \mathbf{b}) \equiv \frac{1}{2n} \sum_x \|\mathbf{y}_D(\mathbf{x}) - \mathbf{y}(\mathbf{x}, \mathbf{w}, \mathbf{b})\|^2 \quad (2.83)$$

[47, S. 16]. Hierbei beinhaltet die Matrix \mathbf{w} alle Gewichtungen des Netzwerkes, der Vektor \mathbf{b} alle Schwellenwerte und n ist die Anzahl der Trainingseingänge. \mathbf{y}_D ist ein Vektor aller gewünschten Ausgänge korrespondierend zum Eingangsvektor \mathbf{x} . \mathbf{y} beinhaltet alle vom Netzwerk berechneten Ausgänge.

Um das Minimum dieser Funktion zu finden, wird das Gradientenabstiegsverfahren angewandt. Generell kann mit diesem das Minimum einer Funktion $F(v_1, v_2) = F(\mathbf{v})$ gefunden werden, indem iterativ die Aktualisierungsregel

$$\mathbf{v} \rightarrow \mathbf{v}' = \mathbf{v} - \eta \nabla F \quad \text{mit} \quad \nabla F \equiv \left(\frac{\partial F}{\partial v_1}, \frac{\partial F}{\partial v_2} \right) \quad (2.84)$$

angewandt wird [47, S. 19]. η ist hier die Lernrate, eine Konstante, die angibt, mit welcher Schrittweite das Minimum der Funktion gesucht wird.

Übertragen auf die Kostenfunktion des neuronalen Netzes und mit komponentenweiser Formulierung ergeben sich dann die Aktualisierungsregeln für die Gewichtungen und die Schwellenwerte [47, S. 22]:

$$\mathbf{w} \rightarrow \mathbf{w}' = \mathbf{w} - \eta \frac{\partial C}{\partial \mathbf{w}}, \quad (2.85)$$

$$\mathbf{b} \rightarrow \mathbf{b}' = \mathbf{b} - \eta \frac{\partial C}{\partial \mathbf{b}}. \quad (2.86)$$

2 Theoretische Grundlagen

Wie in Gl. 2.83 zu sehen, wird die Kostenfunktion über die Bildung des Durchschnitts von allen Trainingseingängen \mathbf{x} berechnet, was aufgrund der partiellen Ableitungen zu enorm hohen Rechenaufwand führt. Aus diesem Grund ist es üblich, eine stochastische Variante des Gradientenabstiegsverfahrens anzuwenden, welche im Folgenden kurz erläutert werden soll. Gegeben sei eine zufällig generierte Menge (auch *batch* genannt)

$$\mathbf{X} = [\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_m]^T, \quad (2.87)$$

wobei \mathbf{X}_i zufällig ausgewählte Datensätze aus der Menge aller Trainingseingänge \mathbf{x} ist. Mit einer hinreichend großen Teilmenge \mathbf{X} kann der Gradient ∇C sowohl über den Durchschnitt der Teilmenge \mathbf{X} als auch über den Durchschnitt aller Trainingseingänge \mathbf{x} berechnet werden. Es gilt daher

$$\nabla C = \frac{\sum_{\mathbf{x}} \nabla C_{\mathbf{x}}}{n} \approx \frac{\sum_{i=1}^m \nabla C_{\mathbf{X}_i}}{m}. \quad (2.88)$$

Die Aktualisierungsregeln lassen sich dann zu

$$\mathbf{w} \rightarrow \mathbf{w}' = \mathbf{w} - \frac{\eta}{m} \sum_i \frac{\partial C_{\mathbf{X}_i}}{\partial \mathbf{w}}, \quad (2.89)$$

$$\mathbf{b} \rightarrow \mathbf{b}' = \mathbf{b} - \frac{\eta}{m} \sum_i \frac{\partial C_{\mathbf{X}_i}}{\partial \mathbf{b}} \quad (2.90)$$

modifizieren [47, S. 22]. Durch ständige Anpassung der Gewichtungen und Schwellenwerte im Hinblick auf eine Minimierung der Kostenfunktion kann das Netzwerk nun lernen.

2.7.3 Fehlerrückführung

Die Fehlerrückführung ist ein Algorithmus, der dazu dient, die partiellen Ableitungen der Kostenfunktion in Bezug auf die Ausgangsvariablen des ANN und schließlich jedes Gewicht und jeden Schwellenwert im Netzwerk zu berechnen, da diese in Gl. 2.89 sowie Gl. 2.90 benötigt werden.

Definiert wird dazu zunächst ein Fehler δ_j^L , also ein Fehler, der dem Neuron j in der Ausgangsbeschriftung (L) zugehörig ist. Dieser berechnet sich nach

$$\delta_j^L = \frac{\partial C}{\partial a_j^L} \sigma'(z_j^L). \quad (2.91)$$

Dabei ist σ' die Ableitung der Aktivierungsfunktion, also ein Maß dafür, wie schnell diese sich bei einem Wert z_j^L (also z beim j -ten Neuron in der Ausgangsbeschriftung L) ändert und $\frac{\partial C}{\partial a_j^L}$ ein Maß dafür, wie schnell sich die Kostenfunktion in Bezug auf die Ausgabe des j -ten Neurons der Ausgangsbeschriftung ändert. In Matrix-Schreibweise lässt sich dann

$$\boldsymbol{\delta}^L = \nabla_a C \circ \sigma'(z^L) \quad (2.92)$$

schreiben. „ \circ “ kennzeichnet das Hadamard-Produkt. z^L ist der Vektor aller Werte z in der Ausgangsbeschriftung und $\nabla_a C$ ein Vektor, der alle partiellen Ableitungen der Kostenfunktion C nach

den einzelnen Aktivierungen a_j^L der Ausgabeschicht enthält [47, S. 45]:

$$\nabla_a C = \begin{bmatrix} \frac{\partial C}{\partial a_1^L} \\ \vdots \\ \frac{\partial C}{\partial a_R^L} \end{bmatrix}. \quad (2.93)$$

Damit der Fehler beginnend von der Ausgabeschicht L durch alle Schichten l des Netzwerkes rückgeführt werden kann, bedarf es einen Zusammenhang zwischen dem Fehler der l -ten und der $(l + 1)$ -ten Schicht. Dieser ist durch

$$\delta^l = ((\mathbf{w}^{l+1})^T \delta^{l+1}) \circ \sigma'(z^l) \quad (2.94)$$

[47, S.45] gegeben. $(\mathbf{w}^{l+1})^T$ ist hierbei die transponierte Matrix der Gewichtungen aus der $(l + 1)$ -ten Schicht.

Die partielle Ableitung der Kostenfunktion C nach einem beliebigen Schwellenwert b_j^l in der l -ten Schicht am j -ten Neuron kann nun direkt bestimmt werden, da diese exakt gleich dem Fehler an diesem Neuron ist. Es gilt daher

$$\frac{\partial C}{\partial b_j^l} = \delta_j^l. \quad (2.95)$$

Die partielle Ableitung der Kostenfunktion nach einem beliebigen Gewicht w_{jk}^l ergibt sich nach

$$\frac{\partial C}{\partial w_{jk}^l} = a_k^{l-1} \delta_j^l. \quad (2.96)$$

Mit der Notation w_{jk}^l und der Tatsache, dass diese Gewichtung das Neuron k in der $(l - 1)$ -ten Schicht mit dem Neuron j in der l -ten Schicht verbindet, wird klar, dass die Aktivierung a_k^{l-1} die des Neurons ist, welche als Eingang der Gewichtung zu sehen ist und δ_j^l der Fehler des Neurons, welcher als Ausgang der Gewichtung zu sehen ist [47, S. 46]. Mit Hilfe dieser beiden partiellen Ableitungen können die im vorherigen Abschnitt vorgestellten Aktivierungsregeln für die Gewichtungen und Schwellenwerte angewandt und das Netzwerk so trainiert werden.

3 Stand der Optimierungstechniken

Die Regelung einzelner Anlagen in einem Windpark im Hinblick auf eine Leistungsmaximierung des gesamten Parks durch Beeinflussung der Nachlaufströmung der Anlagen, manchmal auch koordinierte Regelung genannt, wird heutzutage immer noch erforscht. Dennoch existieren bereits einige Ansätze in der Literatur, die im Folgenden vorgestellt werden sollen. Im Folgenden wird entweder von einer, meist jährlichen, Ertragssteigerung oder von einer Leistungssteigerung für gewisse Umgebungsparameter bzw. Ausgangsbedingungen, gesprochen.

Annoni u. a. validierten verschiedene Nachlaufmodelle, darunter das Jensen-Modell, aber auch komplexe Modelle höherer Ordnung, wie das aus der Simulationssoftware *SOFWA*. Die Autoren kamen zu dem Entschluss, dass sich Genauigkeit und Rechenzeit antiproportional verhalten. Für eine koordinierte Regelung, die im Optimalfall auch noch in Echtzeit stattfindet, ist allerdings sowohl Genauigkeit, als auch eine geringe Rechenzeit notwendig. [5]

Außerdem untersuchten Annoni u. a. ein Modell, das *FLORIS*-Modell, welches eine Kombination aus dem Jensen-Modell und einem Modell für Nachlaufablenkung sowie Modifikationen darstellt. Sie passten dieses Modell an das komplexe *SOFWA*-Modell an, so dass Abweichungen zwischen den Modellen minimiert werden. [4]

Überwiegend werden für eine koordinierte Regelung zwei Parameter der Anlagen variiert, der Induktionsfaktor und der Gierwinkel, also der Winkel um die Turmachse.

G.-W. Qian beschäftigte sich mit der Ablenkung der Nachlaufströmung durch Veränderung des Gierwinkels. Er testete sein Wake-Modell beispielhaft an einem Anlagen-Array der Größe 4×5 mit einem Abstand von $5D$ und $7D$. Umgebungsparameter bezog er vom Messmast des japanischen Windparks Eurus Tomamae. Der Autor konnte einen jährlichen Zuwachs des Energieertrages von ca. $1,25\%$ bei einem Gierwinkel-Offset von 30° verzeichnen. Im selben Zuge allerdings schränkte er den Offset des Winkels auf $\pm 15^\circ$ ein, um Anforderungen in Bezug auf Ermüdungsbelastungen gerechnet zu werden. [18]

Howland u. a. zeigten ebenfalls das Potential der Ablenkung von Nachlaufströmungen. Die Autoren konnten 7% - 13% Leistungszuwachs für Windgeschwindigkeiten nahe der Durchschnittswindgeschwindigkeit des Windparks und für Windrichtungen, die weniger als 10% des nächtlichen Betriebs auftreten, verzeichnen. Bei niedrigen Windgeschwindigkeiten und denselben Windrichtungen waren es 28% - 47% . Außerdem fanden sie heraus, dass die Variabilität der Leistungsproduktion um 72% verringert werden konnte. [23]

Auch die Variation der Induktionsfaktoren der einzelnen Anlagen kann einen signifikanten Einfluss auf die Leistungsproduktion haben. K. E. Johnson konnte für den einfachsten Fall von zwei hintereinander positionierten Anlagen zeigen, dass durch intelligente Wahl des Induktionsfaktors der stromaufwärts stehenden Anlage in Summe mehr Leistung erzeugt werden kann, als wenn beide Anlagen mit einem für sich optimalen Induktionsfaktor betrieben werden. [32]

E. Ebegebulem entwickelte einen Extremwertregler, also einen Regler, der eine Zielgröße maximieren soll. Untersucht wurden drei Fallbeispiele, in denen sich nach definierter Zeit die Windrichtung ändert. Der entwickelte Regler ist im Stande, die Induktionsfaktoren in Echtzeit so anzupassen,

3 Stand der Optimierungstechniken

dass eine Leistungsmaximierung stattfinden kann. [14]

Ebenfalls mit Hilfe eines Extremwertreglers versuchte K. E. Johnson, die Leistung eines Windparks zu maximieren. Dabei fand der Autor heraus, dass für konstanten Wind und niedrige Turbulenzintensitäten mit dem entwickelten Regler gute Ergebnisse erzielt werden konnten. Für höhere Turbulenzintensitäten ist dies nicht der Fall. [31]

Salamah teilte das Vorgehen in zwei Schritte. Sie verwendeten zunächst einen Algorithmus für die Vorhersage der Leistung im Windpark und schließlich eine sog. Gleitregimeregelung, um ein optimales Set an Induktionsfaktoren zu finden und die Leistung so zu maximieren. [55]

E. Bitar formulierte die koordinierte Regelung als mathematisches Optimierungsproblem. Der Autor betrachtete ein eindimensionales Array von Anlagen. Er unterschied den *far wake* und den *near wake*. Für ersteres benutzte er das Jensen-Modell und suchte den optimalen Abstand der Anlagen zueinander, um die Leistung des Anlagen-Arrays zu maximieren. Für den Bereich des *near wakes* benutzte er ein weiteres analytisches Modell. Mit Hilfe von diesem ermittelte er ein optimales Set von Induktionsfaktoren für das Anlagen-Array und konnte beweisen, dass ein Leistungszuwachs erreicht werden kann. [13]

Hoek u. a. benutzten das Tool **FarmFlow**, welches auf CFD basiert. Mit Hilfe des Tools suchten die Autoren einen optimalen Offset der Pitch-Winkel, allerdings nur abhängig von der Windrichtung und nur für die stromaufwärts positionierte Anlage. Zur Validierung verwendeten sie Daten aus dem Goole Fields Windpark in England, bestehend aus 16 Anlagen. Untersucht wurde allerdings nur eine Reihe, bestehend aus fünf Anlagen. Die Autoren konnten keinen signifikanten Anstieg der Leistung feststellen. Eine Hochrechnung zeigte, dass für den gesamten Windpark, also unter Berücksichtigung aller Windrichtungen, der Ertrag um weniger als 0,37% gesteigert werden konnte, was aber auch auf ungenaue und grobe Annahmen zurückzuführen sei. [22]

Weitere Publikationen untersuchten die Windparkertragssteigerung durch Variation von sowohl Induktionsfaktoren, als auch der Gierwinkel. Park u. a. kombinierten beide Methoden, formulierten ein Optimierungsproblem und wandten ein Gradientenabstiegsverfahren an, um ein optimales Set an Induktionsfaktoren und Gierwinkeln zu finden. Numerische Simulationen zeigten hier ebenfalls, dass eine Leistungssteigerung möglich ist. [50]

Giebel u. a. behauptet, die momentan effizientere Methode bezüglich der Jahresertragssteigerung ist das Ablenken des Wakes. [19]. Nichtsdestotrotz führen beide Ansätze zu guten Ergebnissen, weshalb nicht behauptet werden kann, eine der beiden Ansätze sei signifikant besser als der andere.

J. Park verfolgte außerdem einen Ansatz, bei dem die Leistung des Windparks als Funktion der Regelgrößen ausgedrückt wird und schließlich sog. *Sequential Convex Programming* angewandt wird, um die Funktion zu maximieren. Getestet wurde die Herangehensweise im Windpark Horns Rev, in dem eine über alle Windrichtungen gemittelte Leistungssteigerung von rund 7% verzeichnet werden konnte. [28]

Datengesteuerte Optimierungen und Methoden aus dem Bereich des maschinellen Lernens sind in der Literatur zwar weniger präsent, aber dennoch zu finden. So formulierte A. Menon ein datengesteuertes Optimierungsproblem. Der Autor entwickelte einen Lernalgorithmus und implementierte ihn in **SIMULINK**. [3]

Marden u. a. untersuchten zwei verschiedene Lern-Algorithmen aus der Spieltheorie, die jeweils

modellunabhängig sind. Sie basieren also auf keinem physikalischen oder aerodynamischen Modell. Der Algorithmus wurde in einer Replikation des Windparks Horns Rev angewandt und es konnte eine Leistungssteigerung von bis zu 25 % erreicht werden (Bei $u_\infty = 8 \frac{\text{m}}{\text{s}}$ und konstanter Windrichtung). Allerdings merken die Autoren an, dass dieser Wert kritisch zu betrachten sei, da unrealistische Annahmen, wie die eines stationären Zustandes, angenommen wurden. [44]

J. Park wandte ein System der sog. Bayes'schen Optimierung aus dem Bereich des maschinellen Lernens an. Dieses entwickelte System besteht aus zwei Phasen: Die Lern-Phase, in der das System mittels Regression den Zusammenhang der Input-Parameter und dem dazugehörigen Output lernt, und der Optimierungsphase, in der das System die Input-Parameter zu Gunsten der Leistungsmaximierung optimiert. Getestet wurde es an einem Modell eines eindimensionalen Anlagen-Arrays (vier Modell-Anlagen mit einem Abstand von 7D). Eine Leistungssteigerung von 27 % wurde erreicht. [27]

L. E. Andersson hält fest, dass eine modellbasierte Regelung eines Windparks umso erfolgreicher ist, je genauer das entwickelte Modell ist. Aus diesem Grund untersuchten sie, inwieweit mit Hilfe von gaußscher Regression der Output des Modells auf den Output des realen Windparks angepasst werden kann. In verschiedenen Fallstudien validierte der Autor sein Vorgehen und fand heraus, dass der Error mit der Größe des Windparks und somit der Anzahl der Anlagen zunimmt. [37]

Primär werden Methoden des maschinellen Lernens und speziell neuronale Netze allerdings im Bereich der Vorhersagen von Parametern und nicht in der Optimierung genutzt. So beschäftigten sich Khosravi u. a. damit, mit verschiedenen Algorithmen des maschinellen Lernens sowie Daten aus der Vergangenheit eine Vorhersage für die Umgebungswindgeschwindigkeit zu treffen. Alle Algorithmen lieferten sehr gute Ergebnisse. Das neuronale Netz bspw. wies hier einen RMSE von $0,012 \frac{\text{m}}{\text{s}}$ in Relation zur gemessenen Windgeschwindigkeit auf. [36]

Karami u. a. entwickelten ein System aus zwei neuronalen Netzen, welches für die Vorhersage von Windrichtungen entwickelt wurde. Das System basiert auf Daten der Drehzahlen der Rotoren. Die Optimierung findet in Echtzeit statt. Untersucht wurden Windrichtungen zwischen 0° und 90° , wobei die Genauigkeit zwischen ca. 74 % und ca. 97 % liegt. [34]

A. Marvuglia erstellte ein stationäres Modell eines Windparks mit neuronalen Netzen. Drei verschiedene Arten von neuronalen Netzen wurden getestet und verglichen. Der Autor erzeugte schließlich datengesteuerte Leistungskurven, die später zu Überwachungs- oder Vorhersagezwecken genutzt werden können. [2]

M. F. Howland verglich ein physikalisch basiertes Modell mit einem zweischichtigen neuronalen Netz in Bezug auf die Leistungsvorhersage. Dabei beinhaltete das statistische Modell auch physikalische Informationen über die Strömungsdynamik. Die Autoren konnten feststellen, dass die Abweichung für eine Windrichtung von 250° um 38 % und für eine Windrichtung von 330° um 6 % mit dem neuronalen Netz reduziert werden kann. Betrachtet wurden hierbei reale Daten über einen Zeitraum von 5 Jahren. [43]

Die Ergebnisse der Leistungsmaximierung anhand von Variation der Induktionsfaktoren streuen je nach Methodik der Optimierung stark. Eine Maximierung der Leistung anhand von Variation der Induktionsfaktoren wurde dennoch bereits vielfach getestet und als erfolgreich eingestuft.

3 Stand der Optimierungstechniken

Wie anhand der Literaturrecherche außerdem zu erkennen ist, kann der Einsatz von neuronalen Netzen zu Optimierungszwecken als eher unkonventionell bezeichnet werden. Methoden des maschinellen Lernens werden eher im Sinne einer möglichst genauen Vorhersage von Parametern genutzt. Die Effizienz der Kombination dieser beiden Methodiken soll in dieser Arbeit daher genauer untersucht werden.

4 Entwicklung eines digitalen Zwillings des Windparks

In diesem Kapitel wird ein digitaler Zwilling des Windparks entwickelt, mit dem für ein gegebenes Set an Eingangsparametern die Leistung der einzelnen Anlagen und damit des gesamten Windparks unter Berücksichtigung von Verlusten, primär von Wake-Verlusten, berechnet werden kann. Zwecks Übersichtlichkeit wird im Fließtext kein Python-Code eingefügt. Dieser ist auf dem beigefügten Datenträger zu finden. Das dazugehörige Skript heißt `digital_twin.py`.

4.1 Annahmen

Im Zuge der Entwicklung des digitalen Zwillings werden und wurden bereits einige Annahmen getroffen, um die Komplexität einzugrenzen. Diese sind im Folgenden aufgeführt.

- Die Sensoren aus Tab. 1.1 sind am Messmast entweder in südöstlicher (133°) oder in nordwestlicher (313°) Richtung angebracht. Es kommt also bei Sensoren, die nicht bei $z = 120,5$ m angebracht sind, vor, dass für bestimmte Windrichtungen der Messmast selbst bzw. Teile davon den Sensor abschatten und zu minimal verfälschten Messwerten führen. Im weiteren Verlauf der Arbeit werden ebenfalls Messwerte für z.B. die Geschwindigkeit in einer bestimmten Höhe benötigt (vgl. Abschnitt 6.6), die nur für eine Ausrichtung vorhanden sind. Es wird daher in dieser Arbeit nicht zwischen den Ausrichtungen differenziert.
- Es wird überall eine konstante Dichte der Luft von $\rho_{\text{Luft}} = 1,225 \frac{\text{kg}}{\text{m}^3}$ angenommen. Im späteren Verlauf der Arbeit wird der Einfluss der Varianz der Dichte auf die Windgeschwindigkeit und somit die produzierte Leistung untersucht (vgl. Abschnitt 6.5).
- Der Boden wird bezogen auf die z -Koordinate als eben angenommen. Alle Anlagen stehen bei $z = 0$.
- Die Umgebungswindgeschwindigkeit u_∞ wird am Messmast von Sensor V3 und die Windrichtung vom Sensor D3 entnommen. Sensor V1 befindet sich zwar mit $z = 120,5$ m näher an der Nabenhöhe von $h_{\text{Nabe}} = 120$ m, dennoch werden formalerweise alle Umgebungsparameter von der selben Höhe entnommen. Nichtsdestotrotz ist dies nur eine punktuelle Geschwindigkeit, die nicht repräsentativ für eine Umgebungswindgeschwindigkeit ist. Ein weiterer Ansatz mit Berücksichtigung des Höhenwindgradienten findet in Abschnitt 6.6 statt, allerdings ebenfalls auf Basis von zwei punktuellen Geschwindigkeiten am Messmast. Dazu kommt, dass der Messmast nah an Anlage 1 positioniert ist. Es existieren Windrichtungen, in denen Anlagen stromaufwärts positioniert sind, aber eine große Entfernung zum Messmast aufweisen. Für diese Anlagen ist die Annahme, dass die Umgebungswindgeschwindigkeit die des Messmastes ist, sehr grob.

4.2 Erzeugung der Anlagenobjekte

Der digitale Zwilling wird in der objektorientierten Programmiersprache Python implementiert. Die Anlagen des Windparks sind erzeugte Objekte und deren Eigenschaften und Parameter sind Attribute der jeweiligen Klasse. Für die N117/3000 Anlage wird die Klasse `Nordex3000` und analog für die N117/2400 die Klasse `Nordex2400` erzeugt. Beide Klassen sind der übergeordneten Klasse `Anlage` vererbt. Diese enthält die folgenden Attribute, die über den Konstruktor erzeugt werden:

- `x`: x -Position der Anlage in Gauss-Krüger-Koordinaten
- `y`: y -Position der Anlage in Gauss-Krüger-Koordinaten
- `z`: z -Position der Anlage ($z = 0$ entspricht dem Boden)
- `z_Hub`: Nabenhöhe der Anlage ($= h_{\text{Nabe}}$)
- `d_rotor`: Rotordurchmesser der Anlage ($= d_{\text{R}}$)
- `a`: Induktionsfaktor der Anlage
- `u_in`: Einschaltwindgeschwindigkeit
- `u_out`: Abschaltwindgeschwindigkeit
- `name`: Name der Anlage (z.B. WEA1)
- `P`: Produzierte Leistung der Anlage

Die Erzeugung eines Anlagenobjektes erfolgt für WEA1 bspw. durch

```
WEA1 = Nordex3000(3580434, 5927318, 0, 1/3, 3, 25, „WEA1“, 0)
```

Dabei sind $[3580434, 5927318, 0]^T$ die Koordinaten der Anlage, $a = \frac{1}{3}$, $u_{\text{in}} = 3 \frac{\text{m}}{\text{s}}$, $u_{\text{out}} = 25 \frac{\text{m}}{\text{s}}$ und WEA1 der Name der Anlage. Zu beachten ist, dass mit $z = 0$ der Nullpunkt jeder Anlage im Fuß des Turms liegt und nicht im Rotormittelpunkt. Die Leistung P wird als 0 kW initialisiert und im späteren Verlauf überschrieben.

4.3 Leistungs- und Schubbeiwerte

Die realen Werte für die Beiwerte und die daraus resultierenden Leistungskurven der beiden Anlagentypen sind der folgenden Abb. 4.1 zu entnehmen. Die bereits angesprochene Leistungserhöhung der Delta-Anlage auf 3150 kW ab dem 01.01.2021 ist hier nicht abgebildet, da diese Daten in der Arbeit auch nicht verwendet werden.

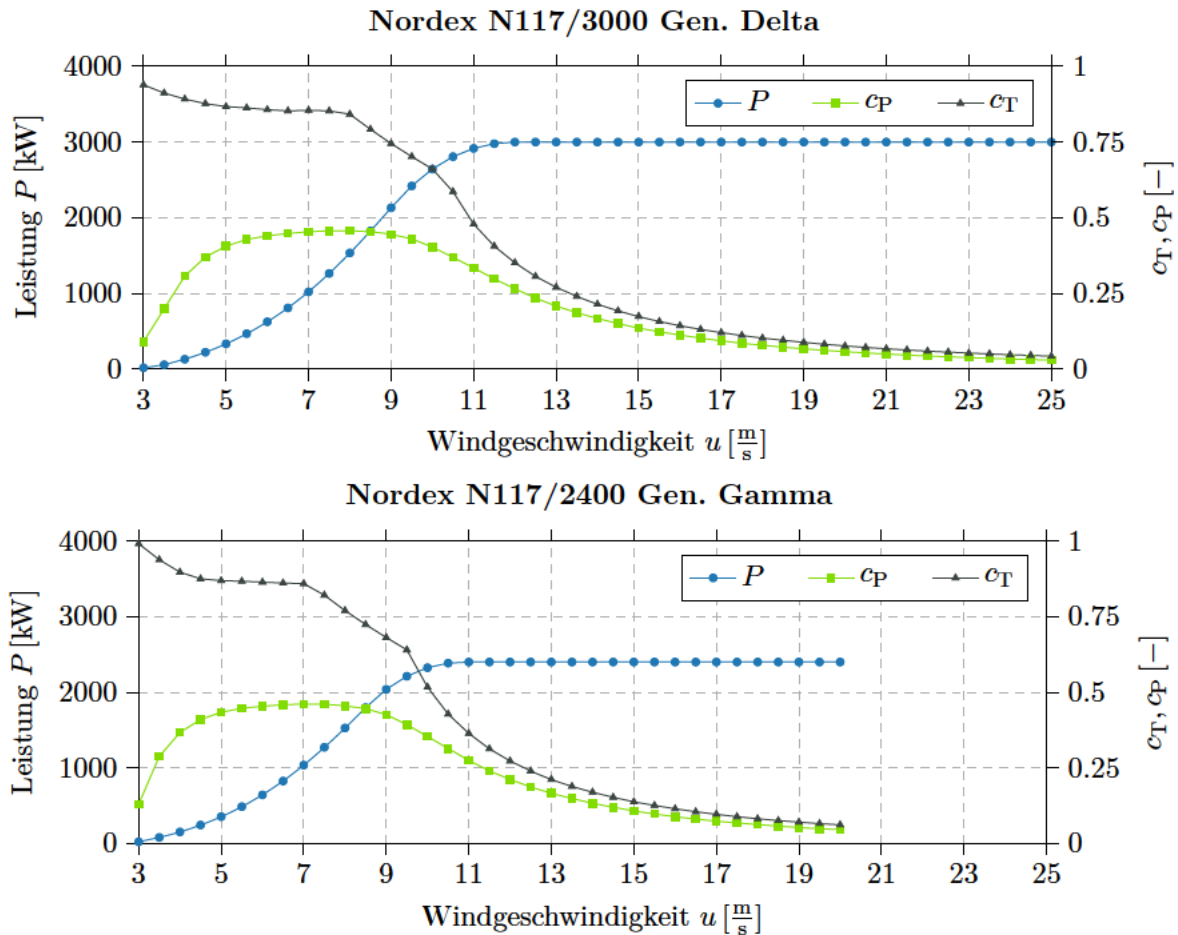


Abbildung 4.1: Reale Leistungs- und Schubbeiwert-Kurven der beiden Anlagentypen [49] [48]

Diese Werte gelten im Teillastbereich für einen Induktionsfaktor von $a = \frac{1}{3}$, da die Anlagen näherungsweise in diesem Bereich so geregelt werden, dass sie maximale Leistung erzeugen. Im Volllastbereich ergibt sich durch Pitchen der Rotorblätter zwangsläufig ein kleinerer Induktionsfaktor. Da später genau dieser Induktionsfaktor variiert werden soll, werden die Leistungs- und Schubbeiwerte nicht nur in Abhängigkeit der Windgeschwindigkeit, sondern auch in Abhängigkeit des Induktionsfaktors benötigt. Hierzu werden die zwei Faktoren η_{c_P} und η_{c_T} eingeführt, die als Wirkungsgrade interpretiert werden können. Sie sollen das Verhältnis der realen Beiwerte aus Abb. 4.1 bei $a = \frac{1}{3}$ zu dem theoretisch maximalen Wert für die Beiwerte nach Gl. 2.18 und Gl.

4 Entwicklung eines digitalen Zwillings des Windparks

2.19 für $a = \frac{1}{3}$ angeben. Sie berechnen sich nach

$$\eta_{c_P}(u) = \frac{c_{P,\text{real}}(u)}{c_{P,\text{theo}}} = \frac{c_{P,\text{real}}(u)}{\frac{16}{27}}, \quad (4.1)$$

$$\eta_{c_T}(u) = \frac{c_{T,\text{real}}(u)}{c_{T,\text{theo}}} = \frac{c_{T,\text{real}}(u)}{\frac{8}{9}}. \quad (4.2)$$

Für eine Windgeschwindigkeit von bspw. $6 \frac{\text{m}}{\text{s}}$ gilt nach Abb. 4.1 für die Nordex N177/3000 Anlage $c_{P,\text{real}} = 0,44$ und $c_{T,\text{real}} = 0,857$. Damit ergeben sich $\eta_{c_P} = 0,743$ und $\eta_{c_T} = 0,964$. Bei dieser Windgeschwindigkeit beträgt der reale Leistungsbeiwert bei $a = \frac{1}{3}$ also ca. 74% des theoretischen Wertes und der Schubbeiwert ca. 96% des theoretischen Wertes. Zurückzuführen sind diese Differenzen auf Verluste, bspw. aufgrund der nicht perfekten Blattgeometrie.

Schlussendlich können somit die Gleichungen 2.18 und 2.19 zu

$$c_P(a, u) = 4 a (1 - a)^2 \eta_{c_P}(u), \quad (4.3)$$

$$c_T(a, u) = 4 a (1 - a) \eta_{c_T}(u) \quad (4.4)$$

modifiziert werden. So können zu jeder Windgeschwindigkeit und für jeden Induktionsfaktor der reale Leistungsbeiwert und der reale Schubbeiwert berechnet werden.

Die beiden vererbten Klassen enthalten deshalb außerdem jeweils das Attribut `cPcT_matrix`. Dieses Array wird durch Einlesen einer Excel-Datei erzeugt und enthält die Faktoren η_{c_P} und η_{c_T} zu jeder Windgeschwindigkeit $u_{\text{in}} \leq u \leq u_{\text{out}}$. Die beiden Dateien sind in `Nordex3000.xlsx` für die Delta-Anlage und `Nordex2400.xlsx` für die Gamma-Anlage zu finden.

Über eine klasseninterne Funktion `calculate_cPcT` wird später die Windgeschwindigkeit sowie der Induktionsfaktor eingegeben und die dazugehörigen Beiwerte ausgegeben. Befindet sich die Windgeschwindigkeit zwischen zwei Randwerten, so werden die Faktoren η_{c_P} , η_{c_T} linear interpoliert. Befindet sich die Windgeschwindigkeit unter der Einschalt- oder über der Ausschaltwindgeschwindigkeit, werden die entsprechenden Beiwerte c_P , $c_T = 0$ übergeben, da die Anlage unter diesen Bedingungen nicht aktiv ist.

Beispielhaft sind nachfolgend die Beiwerte-Kurven für ein ausgewähltes Set an Induktionsfaktoren dargestellt.

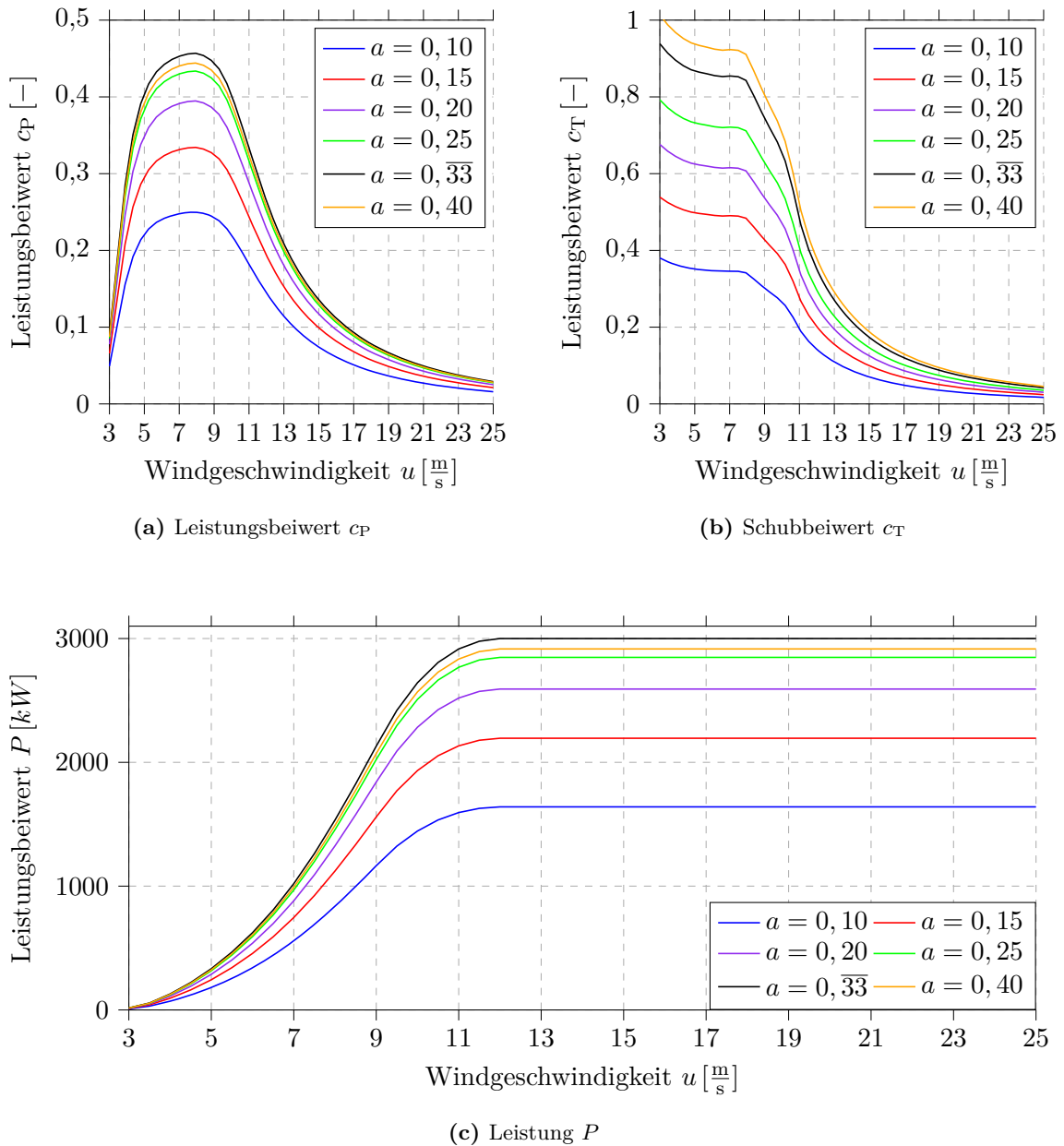


Abbildung 4.2: Beiwerte c_p , c_T sowie Leistung P in Abhängigkeit des Induktionsfaktors a für alle Windgeschwindigkeiten $u_{in} \leq u \leq u_{out}$ an WEA 1

Für den Fall $a = 0,33$ (schwarz) entsprechen die Kurven der Beiwerte denen aus Abb. 4.1. Bei den Leistungskurven wird deutlich, dass auch bei nicht optimalen Induktionsfaktoren im Volllastbetrieb, also für Windgeschwindigkeiten über u_n , die Leistung geringer, aber auch nahezu konstant ist.

4.4 Berücksichtigung der Windrichtung

Die Windrichtung ist nach allgemeinem Standard so definiert, dass angegeben wird, aus welcher Richtung der Wind kommt. Dieser Standard wird auch im Windpark Curslack verwendet. Eine Windrichtung von 0° entspricht also einem Wind von Norden nach Süden, eine Windrichtung von 90° entspricht einem Wind von Osten nach Westen. Die „Drehrichtung“ ist also im Uhrzeigersinn, mathematisch negativ.

Programmetechnisch soll eine Windrichtung von 0° allerdings einem Wind in x -Richtung entsprechen. Diese wird nach einem konventionellen Koordinatensystem von links nach rechts definiert. Außerdem soll eine mathematisch positive Drehrichtung verwendet werden. Aus diesem Grund ist eine Anpassung des eingegebenen Winkels der Windrichtung erforderlich. Anhand nachstehender Abbildung kann diese Modifizierung hergeleitet werden.

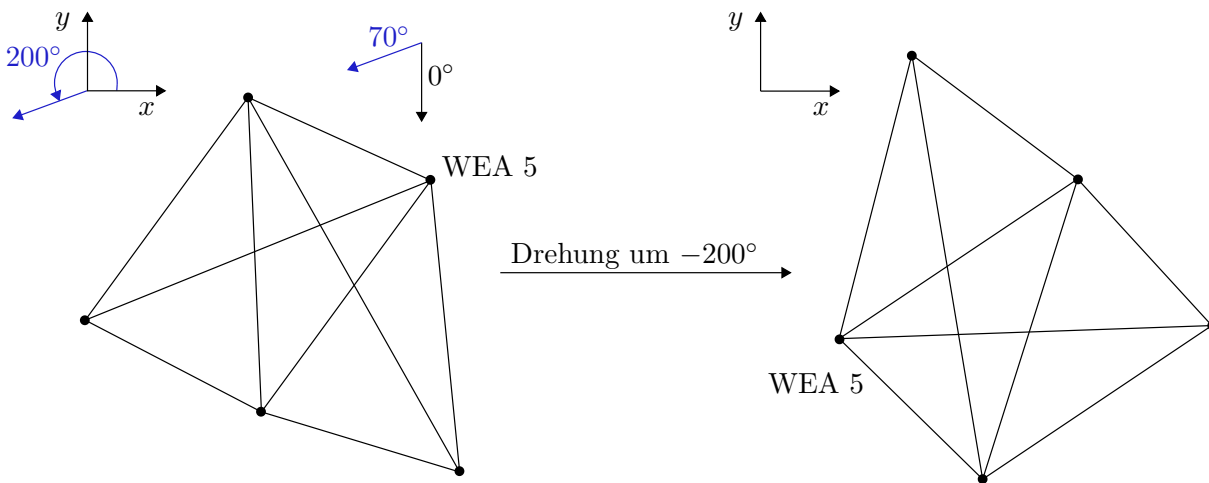


Abbildung 4.3: Herleitung des Zusammenhangs der beiden 0° Windrichtungen

Oben mittig zu sehen ist die 0° Richtung, wie sie konventionell definiert ist. Ein Wind aus 70° entspricht demnach dem blauen Pfeil, einer mathematisch negativen Drehung. Wird die x -Richtung von Westen nach Osten betrachtet, so entspricht der selbe blaue Pfeil einer Drehung von 200° , allerdings mathematisch positiv.

Allgemein lässt sich also sagen, dass durch eine Drehung von 270° mathematisch negativ der 0° Pfeil in den Pfeil der definierten x -Richtung überführt werden kann. Da die konventionelle Drehung mathematisch negativ ist und im Programm eine mathematisch positive Drehung definiert wird, wird ein Winkel α nicht addiert, sondern subtrahiert.

Da im digitalen Zwilling kein Vektor existiert, der die Windrichtung definiert, wird der gesamte Windpark, also alle Koordinaten der einzelnen Anlagen, um den negativen Winkel der Windrichtung gedreht. Schlussendlich gilt

$$\beta = -(270 - \alpha), \quad (4.5)$$

wobei α der Winkel aus den realen Windparkdaten ist und β der modifizierte Winkel, um den der gesamte Windpark gedreht werden muss, sodass eine Windrichtung aus x -Richtung den selben

Effekt hat. In Abb. 4.3 entspricht dies also einer Drehung um -200° . WEA 5 steht somit in beiden Fällen in Bezug zur Windrichtung ganz vorne.

Die Drehung des Windparks erfolgt nach Abschnitt 2.6. Für eine Drehung in der x - y -Ebene um die z -Achse ergibt sich die Rotationsmatrix nach Gl. 2.77 zu

$$\mathbf{R}_z(\beta) = \begin{bmatrix} \cos(\beta) & -\sin(\beta) & 0 \\ \sin(\beta) & \cos(\beta) & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (4.6)$$

Die Koordinaten der einzelnen Anlagen lassen sich als

$$\mathbf{x}_i = \begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix}, \quad i \in \{1, \dots, 5\} \quad (4.7)$$

formulieren. Die Koordinaten des gedrehten Windparks lassen sich dann über

$$\mathbf{x}_{i,\text{rot}} = \mathbf{R}_z(\beta) \cdot \mathbf{x}_i \quad (4.8)$$

berechnen. Die Drehung um z erfolgt um einen Punkt, weshalb der gedrehte Park bei anderen absoluten Koordinaten steht. Da für die spätere Berechnung aber nur relative Positionen der Anlagen verwendet werden, ist dies irrelevant.

Nach Transformation des Windparks in Windrichtung folgt eine Sortierung des Anlagen-Arrays \mathbf{A} (**turbines**) nach der x -Koordinate der jeweiligen Anlagen. So kann später sukzessiv durch das Array iteriert werden. Ab hier bedeutet der Index $i = 1$, dass es sich um die in Windrichtung erste Anlage handelt, nicht zwingend um WEA 1. Eine eindeutige Zuordnung der Anlage findet über das Attribut **name** statt.

Schlussendlich werden die noch sehr großzahligen Gauss-Krüger-Koordinaten der Anlagen alle auf die in Windrichtung erste Anlage bezogen. Es gilt dann

$$\mathbf{x}_{i,\text{rot}} = \mathbf{x}_{i,\text{rot}} - \mathbf{x}_{1,\text{rot}} \quad \text{und am Ende} \quad \mathbf{x}_{1,\text{rot}} = \mathbf{0}. \quad (4.9)$$

Dieser Schritt ist nicht zwingend notwendig, hilft aber beim Verständnis der Ausgabe von den relativen Positionen, da die Zahlenwerte klein sind und direkt den relativen Abstand zur ersten Anlage in Windrichtung repräsentieren. Die neuen Koordinaten werden den Attributen **x**, **y** und **z** jeder einzelnen Anlage zugewiesen.

4.5 Erstellung von Datenpunkten

Um die geometrische Mittelung der Windgeschwindigkeiten sowie die quadratische Mittelung der Standardabweichungen der Windgeschwindigkeit vor jedem Rotor j (vgl. Gl. 2.72 und Gl. 2.73) umzusetzen, wird eine Punktwolke innerhalb der jeweiligen Rotorfläche definiert. Die Verteilung der Datenpunkte erfolgt gleichmäßig über die gesamte Rotorfläche.

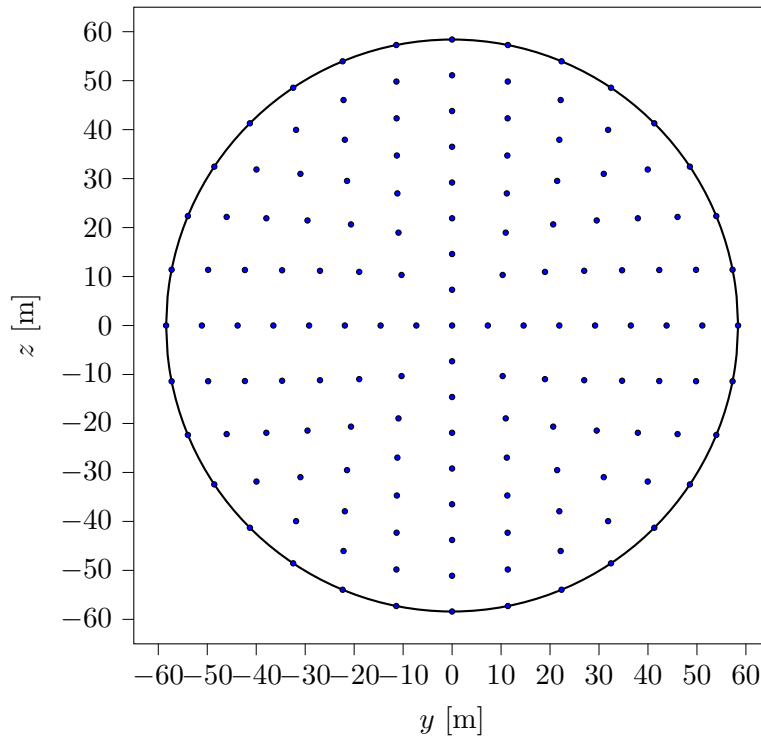


Abbildung 4.4: Erzeugte Datenpunkte innerhalb der Rotorfläche einer Anlage

Es muss beachtet werden, dass der Rotormittelpunkt hier bei $(0, 0)$ (und nicht bei $(0, 120)$) liegt, da später die Koordinaten der Datenpunkte auf die relativen Abstände der betrachteten Anlagen addiert werden. Als radiale Schrittweite werden $7,3\text{ m}$ gewählt, so ergeben sich beim Rotordurchmesser von $116,8\text{ m}$ exakt 8 Ringe, die mit Datenpunkten besetzt sind. Die tangentielle Schrittweite wird adaptiv in Abhängigkeit des Ringes gewählt, sodass eine gleichmäßige Verteilung der Datenpunkte bezogen auf die Ringe gewährleistet ist. Verdoppelt sich der radiale Abstand zum Rotormittelpunkt, so wird die tangentielle Schrittweite halbiert, sodass, auf den jeweiligen Durchmesser jeden Ringes bezogen, gleich viele Datenpunkte vorhanden sind. Als Startwert für den ersten Ring bei $r = 7,3\text{ m}$ werden vier Datenpunkte gewählt, was einer tangentialen Schrittweite von 90° entspricht. Die Datenpunkte werden schließlich von Polarkoordinaten in kartesische Koordinaten umgerechnet

$$y_{\text{dp}} = r \cos(\varphi), \quad z_{\text{dp}} = r \sin(\varphi), \quad (4.10)$$

um später besser mit ihnen rechnen zu können, und in ein Array \mathbf{dp} gespeichert. Der Datenpunkt $(0, 0)$ (Rotormittelpunkt) wird ebenfalls angehängen.

4.6 Sukzessive Berechnung des Nachlaufes

Nachstehende Matrix \mathbf{Q} enthält die in Abb. 1.1 bereits aufgezeigten relativen Abstände der Anlagen zueinander, bezogen auf den Rotordurchmesser d_R . Das Element q_{ij} ist hierbei der Abstand zwischen der i -ten und der j -ten WEA. Element $q_{23} = 2,62 d_R$ kennzeichnet somit den Abstand zwischen WEA 2 und WEA 3.

$$\mathbf{Q} = \{q_{ij}\} = \begin{bmatrix} 0 & 2,51 & 5,11 & 3,46 & 4,71 \\ & 0 & 2,62 & 3,93 & 3,61 \\ & & 0 & 5,38 & 3,66 \\ & & & 0 & 2,53 \\ \text{sym.} & & & & 0 \end{bmatrix} \cdot d_R$$

Wie Reinwardt für den Windpark Curslack mit Hilfe von experimentellen Messungen bestätigen konnte (vgl. Abschnitt 2.4.3), lässt sich bereits in einem Abstand von $2,18D$ hinter der Anlage ein Geschwindigkeitsdefizit in Form einer Gauss-Verteilung wiederfinden [54, S. 77]. Da $\min(q_{ij}) = 2,51$ gilt, kann angenommen werden, dass das Geschwindigkeitsprofil vor jeder Anlage, beeinflusst von einer oder mehreren Anlagen davor, gaußförmig anzunehmen ist, weshalb die vorgestellten trivialen Modelle nach z.B. Jensen oder Frandsen mit einer angenommenen zylindrischen Verteilung des Geschwindigkeitsdefizits für den digitalen Zwillings nicht in Frage kommen.

Aufgrund der in Abschnitt 2.4.8 aufgeführten Ergebnisse wird das Single-Wake-Modell nach Ishihara-Qian implementiert. Wie Tian u. a. herausfanden, können mit der Methode des Wurzelsummenquadrates (vgl. Gl. 2.50) im Vergleich zu den anderen vorgestellten Superpositionsmethoden die besten Ergebnisse erzielt werden. Es konnte herausgefunden werden, dass die geometrische Summe und die lineare Superposition das Geschwindigkeitsdefizit überschätzen [59, S. 460].

Des Weiteren fand G.-W. Qian heraus, dass die in Abschnitt 2.5.2 erläuterte rotorbasierte Superposition im Gegensatz zur umgebungsbasierten Superposition bessere Ergebnisse liefert [18, S. 4]. Aus den genannten Gründen findet im Folgenden eine Implementierung des rotorbasierten Multiple-Wake-Modells aus Abschnitt 2.5.3 mit Mittelung der Parameter über die Rotorfläche statt, welche im Folgenden vorgestellt wird.

Begonnen wird mit der Initialisierung zweier Arrays \mathbf{u} und \mathbf{I}_a . Die iterativ berechneten Geschwindigkeiten u_j und Turbulenzintensitäten $I_{a,j}$ vor jedem Rotor j werden an die Arrays angehängt, um so in jedem Iterationsschritt auf die Werte vor den Anlagen zugreifen zu können. Ein weiteres Array \mathbf{P} wird später mit allen Leistungen P_j jeder Anlage j gefüllt.

In der äußersten Schleife erfolgt eine Iteration über alle Anlagen j im Windpark, genauer gesagt über das Anlagen-Array \mathbf{A} . Für die erste Anlage können die Umgebungswindgeschwindigkeit u_∞ und dessen Standardabweichung $\sigma_{u,0}$ (welche mit u_∞ noch zu einer Turbulenzintensität I umgerechnet wird) direkt an die Arrays \mathbf{u} und \mathbf{I}_a angehängt werden.

Für alle Anlagen $j \neq 0$ folgt eine Initialisierung der Datenpunkte nach Abschnitt 4.5. Die kartesischen Datenpunkte werden in einem Array \mathbf{dp} gespeichert. Dieses hat die Form $[n \times 2]$, wobei n die Anzahl der Datenpunkte ist. Es folgt in einer inneren Schleife eine Iteration über alle

4 Entwicklung eines digitalen Zwillings des Windparks

Datenpunkte. Für jeden Datenpunkt, also für jede Position innerhalb der Rotorfläche von Anlage j folgt nun die Berechnung der Einflüsse von allen Anlagen i , die stromaufwärts von Anlage j positioniert sind.

Für diese Berechnung ist eine weitere Schleife über alle Anlagen i im Bereich $0 \leq i < j$ notwendig. Folgende Berechnungen erfolgen demnach für jede Anlage i , die stromaufwärts von Anlage j positioniert ist, und für einen bestimmten Datenpunkt. Die relative Position von Anlage i zu Anlage j ergibt sich durch

$$x_{\text{diff}} = x_j - x_i \quad (4.11)$$

$$y_{\text{diff}} = y_j - y_i + y_{\text{dp}} \quad (4.12)$$

$$z_{\text{diff}} = h_{\text{Nabe},j} + z_{\text{dp}} \quad (4.13)$$

Das Geschwindigkeitsdefizit an einem Datenpunkt vor Anlage j , erzeugt von Anlage i , wird über Gl. 2.58 berechnet. Analog dazu wird die Änderung der Standardabweichung der Windgeschwindigkeit an einem Datenpunkt vor Anlage j , erzeugt durch Anlage i über Gl. 2.60 berechnet. Dies erfolgt über die beiden Funktionen `calculate_delta_u` und `calculate_delta_sigma_u`, die die Parameter Δu_{ij} und $\Delta \sigma_{u,ij}$ in der Form

$$\Delta u_{ij}, \Delta \sigma_{u,ij} = f(u_i, I_{a,i}, x_{\text{diff}}, y_{\text{diff}}, z_{\text{diff}}, d_{R,i}, c_{T,i}, h_{\text{Nabe},i}) \quad (4.14)$$

berechnen. $c_{T,i}$ wird über die Funktion `calculate_PcT` ermittelt. $d_{R,i}, h_{\text{Nabe},i}$ sind Attribute der jeweiligen Anlage. $u_i, I_{a,i}$ werden den Arrays \mathbf{u} und \mathbf{I}_a entnommen.

Für alle $i \neq 0$ folgt noch die Berechnung des Korrekturterms $\Delta \sigma_{u,i,\text{Korr}}$. Für $i = 0$ ist diese Berechnung nicht notwendig, da keine Anlage $i - 1$ existiert. Hierzu werden alle notwendigen Parameter von Gl. 2.62 bis Gl. 2.69 berechnet. Diese hängen von den Werten der Anlage i und Anlage $i - 1$ ab, welche ebenfalls Attribute der jeweiligen Anlage sind oder aus den Geschwindigkeits- und Turbulenzintensitäts-Arrays stammen. Die für den Korrekturterm notwendige Abweichung der Standardabweichung der Windgeschwindigkeit an der „tip side“ von Anlage i , also $\Delta \sigma_{u,i,\text{tip}} = \Delta \sigma_{u,i}(y = 0, z = \frac{d_{R,i}}{2})$ wird ebenfalls über die Funktion `calculate_delta_sigma_u` berechnet. Jetzt in der Form

$$\Delta \sigma_{u,i,\text{tip}} = f\left(u_{i-1}, I_{a,i-1}, (x_i - x_{i-1}), (y_i - y_{i-1}), \left(h_{\text{Nabe},i} + \frac{d_{R,i}}{2}\right), d_{R,i-1}, c_{T,i-1}, h_{\text{Nabe},i-1}\right). \quad (4.15)$$

Die Berechnung des Korrekturterms erfolgt dann nach Gl 2.70.

Schlussendlich werden die berechneten Werte aller Anlagen i nach Gl. 2.59 und Gl. 2.61 für einen Datenpunkt superpositioniert. Dazu wird bereits in der innersten Schleife (Iteration über alle Anlagen i) die Summe der quadrierten Werte nach

$$\Delta u_{ij,\text{sum}} = \sum (\Delta u_{ij}^2) \quad (4.16)$$

und

$$\Delta \sigma_{u,ij,\text{sum}} = \sum (\Delta \sigma_{u,ij} + \Delta \sigma_{u,i,\text{Korr}})^2 \quad (4.17)$$

gebildet und in der Schleife über alle Datenpunkte dann nach den Gleichungen superpositioniert:

$$u_j = u_\infty - \sqrt{\Delta u_{ij,\text{sum}}}, \quad (4.18)$$

$$\sigma_{u,j}^2 = \sigma_{u,0}^2 + \Delta \sigma_{u,ij,\text{sum}}. \quad (4.19)$$

Die dazu notwendigen Anfangswerte u_∞ , $\sigma_{u,0}$ sind vorhandene Eingangsparameter. Die Ergebnisse für u_j und $\sigma_{u,j}^2$ werden schließlich in Arrays $\mathbf{u}_{j,\text{alldp}}$ und $\boldsymbol{\sigma}_{u,j,\text{alldp}}$ gespeichert. Nach Durchlauf der Schleife über alle Datenpunkte wird dann zum einen das geometrische Mittel aller Geschwindigkeiten, also der Mittelwert, gebildet (vgl. Gl. 2.72), um die äquivalente Geschwindigkeit $u_{j,\text{equiv}}$ zu erhalten und zum anderen das quadratische Mittel aller Standardabweichungen gebildet und durch die gemittelte Geschwindigkeit dividiert (vgl. Gl. 2.73), um die äquivalente Turbulenzintensität $I_{a,j,\text{equiv}}$ zu erhalten. Diese beiden Werte werden schließlich an die Arrays \mathbf{u} und \mathbf{I}_a angehängt. Davor erfolgt die bei Gl. 2.71 angesprochene Überprüfung, ob $I_{a,j,\text{equiv}} > 0,03$ ist.

4.7 Berechnung der Leistungen

Es folgt eine Überprüfung, ob die jeweilige Geschwindigkeit zwischen Einschalt- und Abschaltgeschwindigkeit der jeweiligen Anlage liegt. Ist dies der Fall, wird schlussendlich mit der Geschwindigkeit $u_{j,\text{equiv}}$, der Standarddichte von Luft $\rho_{\text{Luft}} = 1,225 \frac{\text{kg}}{\text{m}^3}$, der Rotorfläche und dem Leistungsbeiwert aus der Funktion `calculate_cPcT` mit Gl. 2.14 die dazugehörige Leistung P_j ermittelt und dem Attribut `P` des jeweiligen Objektes zugeordnet. Danach wird mit der nächsten äußeren Iteration $j + 1$ begonnen.

Sind alle Leistungen der Turbinen berechnet, wird das Anlagen-Array \mathbf{A} nach dem Attribut `name` sortiert, um wieder eine aufsteigende Reihenfolge der Anlagen im Array zu haben, die nicht mehr abhängig von der Windrichtung ist. Danach wird das Array \mathbf{P} durch Iteration über das Anlagen-Array \mathbf{A} mit den Leistungen P_j gefüllt. So können später die Leistungen einfacher mit realen Daten verglichen werden.

4.8 Modifikation der Korrekturterm-Berechnung

In der Publikation von G.-W. Qian heißt es, dass für die Berechnung des Korrekturterms $\Delta\sigma_{u,i,Korr}$ für Anlage i die am nächsten stromaufwärts gelegene Anlage betrachtet werden soll. Diese hat aber nicht zwingend Einfluss auf die Anlage i . Die nachfolgende Abbildung veranschaulicht die Problematik.

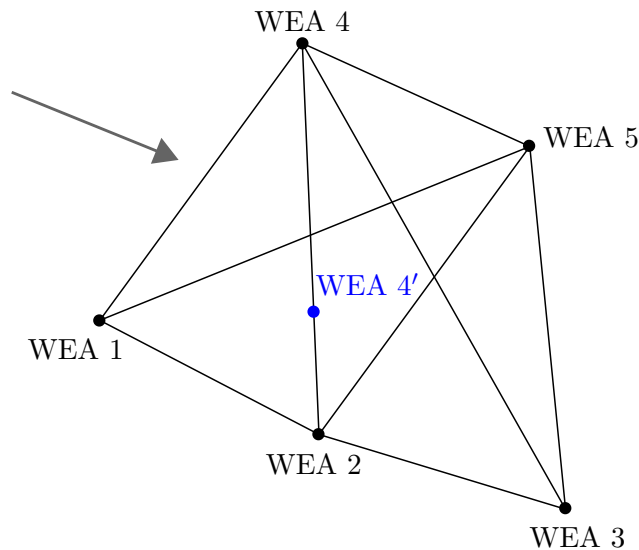


Abbildung 4.5: Darstellung der Problematik bei der Berechnung des Korrekturterms mit Hilfe einer imaginären Anlage 4'

Betrachtet wird die Anlagen-Konfiguration inklusive eingezeichneter Windrichtung (grauer Pfeil) und der imaginären Anlage 4' in Abb. 4.5. Wird Anlage 3, mit dem Ziel der Berechnung des Einflusses von Anlage 2 auf diese zu berechnen, betrachtet, so wäre die am nächsten positionierte Anlage stromaufwärts von Anlage 2 die Anlage 4'. Diese wird voraussichtlich keinen Einfluss auf Anlage 2 haben, was zu einem Korrekturfaktor von 0 führt. Physikalisch ist es jedoch sinnvoll, für die gegebene Windrichtung den Einfluss von Anlage 1 auf Anlage 2 zu betrachten, auch wenn Anlage 1 nicht die am nächsten positionierte ist.

Aus diesem Grund wird die Berechnung insofern modifiziert, als dass für eine Anlage i (in dem Fall Anlage 2) der Einfluss von jeder Anlage p stromaufwärts $[0, \dots, i - 1]$ (In dem Fall Anlage 1, Anlage 4 und Anlage 4') betrachtet und somit 3 Korrekturfaktoren berechnet werden. Für das weitere Vorgehen wird dann der Korrekturfaktor mit dem größten Wert benutzt. Im vorgestellten Fall wäre dies somit der Korrekturfaktor, der sich ergibt, wenn der Einfluss von Anlage 1 auf Anlage 2 betrachtet wird. Konkret werden alle Korrekturfaktoren in einem Array $\Delta\sigma_{u,i,Korr}$ gespeichert und am Ende wird der maximale Wert für die weitere Berechnung verwendet.

4.9 Zusammenfassender Algorithmus

Der nachfolgende visualisierte Algorithmus zeigt das beschriebene Vorgehen. Zwecks Übersichtlichkeit und Verständlichkeit sind nicht alle Berechnungen / Bedingungen im Schema aufgeführt. Bspw. wird auf die Darstellung der Abfragen, ob $I_{a,j} \leq 0,03$ ist oder ob die ermittelte Geschwindigkeit im Bereich $u_{in} \leq u_j \leq u_{out}$ liegt, verzichtet. Ebenfalls wird die Berechnung des Korrekturterms für jede Anlage i in einer einzigen Zeile dargestellt, obwohl dazu selbstverständlich die gesamte Berechnung der mathematischen Mengen und geometrischen Größen (vgl. Gl. 2.62 bis Gl. 2.70) notwendig ist.

Das Schema dient dazu, die Umsetzung der Implementierung des Modells in seinen wichtigsten Punkten zu zeigen und lässt sich in zwei Bereiche aufteilen. Der erste Bereich beinhaltet das Einlesen der Umgebungsparameter und die Initialisierung der notwendigen Arrays. Außerdem werden die Koordinaten der Anlagen bezogen auf die Windrichtung wie beschrieben modifiziert.

Algorithm 1: Algorithmus der Leistungsberechnung nach dem Qian-Ishihara-Modell

Data: Eingangsparameter: $\alpha, u_\infty, \sigma_{u,0}, \mathbf{a}$

Berechne $I_{a,0}$ ▷ Gl. 2.71

Initialisiere $\mathbf{u}, \mathbf{I}_a, \mathbf{P}$

u_∞ an \mathbf{u} , $I_{a,0}$ an \mathbf{I}_a anhängen

\mathbf{T} erstellen

β als $f(\alpha)$ berechnen ▷ Gl. 4.5

Windpark drehen ▷ Gl. 4.8

\mathbf{T} nach kleinstem x -Wert sortieren

Windpark-Koordinaten relativ setzen ▷ Gl. 4.9

Der zweite Teil beinhaltet die eigentliche Berechnung der Nachlaufströmung inklusive aller notwendigen Schleifen sowie die schlussendliche Berechnung und Ausgabe der Leistungen.

4 Entwicklung eines digitalen Zwillings des Windparks

```
for  $j$  in range(len( $T$ )) do
    Erstelle  $\mathbf{dp}$ 
    Initialisiere  $\mathbf{u}_{j,\text{alldp}}$ ,  $\sigma_{\mathbf{u},j,\text{alldp}}$ 
    for datapoint in  $\mathbf{dp}$  do
        Initialisiere  $\Delta u_{ij,\text{sum}} = 0$ ,  $\Delta \sigma_{\mathbf{u},ij,\text{sum}} = 0$ 
        for  $i$  in range(0,  $j$ , 1) do
            Berechne  $x_{\text{diff}}$ ,  $y_{\text{diff}}$ ,  $z_{\text{diff}}$  ▷ Gl. 4.11, 4.12, 4.13
            Berechne  $\Delta u_{ij}$  ▷ Gl. 2.42, 4.14
            Berechne  $\Delta \sigma_{\mathbf{u},ij}$  ▷ Gl. 2.45, 4.14
            Initialisiere  $\Delta \sigma_{\mathbf{u},i,\text{Korr}} = 0$ ,  $\Delta \sigma_{\mathbf{u},i,\text{Korr}}$ 
            if  $i = 0$  then
                | pass
            else
                for  $p$  in range( $i - 1, -1, -1$ ) do
                    | Berechne  $\Delta \sigma_{\mathbf{u},i,\text{tip}}$  ▷ Gl. 2.45, 4.15
                    | Berechne  $\Delta \sigma_{\mathbf{u},i,\text{Korr}}$  ▷ Gl. 2.70
                end
                 $\Delta \sigma_{\mathbf{u},i,\text{Korr}} = \max(\Delta \sigma_{\mathbf{u},i,\text{Korr}})$ 
            end
             $\Delta u_{ij,\text{sum}} += \Delta u_{ij}^2$ 
             $\Delta \sigma_{\mathbf{u},ij,\text{sum}} += (\Delta \sigma_{\mathbf{u},ij} + \Delta \sigma_{\mathbf{u},i,\text{Korr}})^2$ 
        end
        Berechne  $u_j$  durch Superposition ▷ Gl. 4.18
        Berechne  $\sigma_{\mathbf{u},j}^2$  durch Superposition ▷ Gl. 4.19
         $\mathbf{u}_j \rightarrow \mathbf{u}_{j,\text{alldp}}$ 
         $\sigma_{\mathbf{u},j}^2 \rightarrow \sigma_{\mathbf{u},j,\text{alldp}}$ 
    end
    Berechne  $u_{j,\text{equiv}}$ ,  $I_{\mathbf{a},j,\text{equiv}}$  ▷ Gl. 2.72, 2.73
     $u_{j,\text{equiv}} \rightarrow \mathbf{u}$ 
     $I_{\mathbf{a},j,\text{equiv}} \rightarrow \mathbf{I}_{\mathbf{a}}$ 
    Berechne Leistung  $P_j$  und ordne es Attribut zu
end
 $T$  nach Name aufsteigend zurücksortieren
 $P$  mit allen Leistungen  $P_j$  füllen
```

Der gesamte Code ist als `digital_twin.py` auf dem Datenträger zu finden.

5 Filterung der Daten

Alle realen Daten des Windparks, also der Anlagen sowie des Messmastes, liegen in Excel-Listen vor. Sie werden vom CC4E der HAW Hamburg bereitgestellt. Die Daten liegen in einem Zeitintervall von 10 min vor. Aus den verschiedenen Excel-Listen werden sog. Pivot-Tabellen für die einzelnen Jahre 2019 und 2020 erstellt. In diesen kann dann relativ simpel nach gewissen Kriterien gefiltert werden. Eine solche Filterung der Daten ist notwendig, da es z.B. Zustände der Anlagen gibt, die nicht berücksichtigt werden sollen. Alle vorgenommenen Filterungen werden im Folgenden kurz aufgeführt.

5.1 Betriebszustand

Der Betriebszustand ist durch Zahlen von 1 bis 26 definiert, wovon Zustand 11-15 den Produktionsbetrieb kennzeichnen. Alle anderen Zustände beschreiben Spezialfälle wie z.B. verschiedene Pitchtests, Trudeln der Anlage, Bremsentests und Sensorenkalibrierungen. Alle vorhandenen Daten werden so gefiltert, dass nur Betriebszustände des Produktionsbetriebes verwendet werden, da die Ergebnisse sonst verfälscht werden würden.

5.2 Schallbetriebsart

Die Schallbetriebsart definiert, ob eine Anlage in der Nennbetriebsart (Modus 0) oder in einem schallreduzierten Modul betrieben wird (Modus 5 oder 7 je nach Anlagentyp). Dieser schallreduzierte Modus ist für die Nacht eingeführt worden, um die Lautstärke der Anlagen zu reduzieren. In dieser Arbeit soll eine Leistungsmaximierung beispielhaft an den Tages-Daten durchgeführt werden, weshalb auch die Validierung nur an den Tages-Daten erfolgt. Alle Zustände $\neq 0$ werden deshalb verworfen.

5.3 Abschattung des Messmastes

Durch die Lage der Anlagen in Bezug auf den Messmast (vgl. Abb. 1.1) wird deutlich, dass es einen Sektor an Windrichtungen gibt, in dem der Windpark den Messmast abschattet. Dies führt dazu, dass der digitale Zwillings mit niedrigeren Umgebungsgeschwindigkeiten rechnet, als an den entsprechenden Anlagen vorherrschen.

Die realen sowie korrespondierenden berechneten Leistungsdaten für den Bereich $70^\circ \leq \alpha \leq 90^\circ$ (in diesem Bereich wird der Messmast definitiv abgeschattet) mit allen bisher genannten Filterungen für die Anlage 5, also die für diesen Windrichtungssektor vorderste Anlage, sind in nachfolgender Abbildung dargestellt.

5 Filterung der Daten

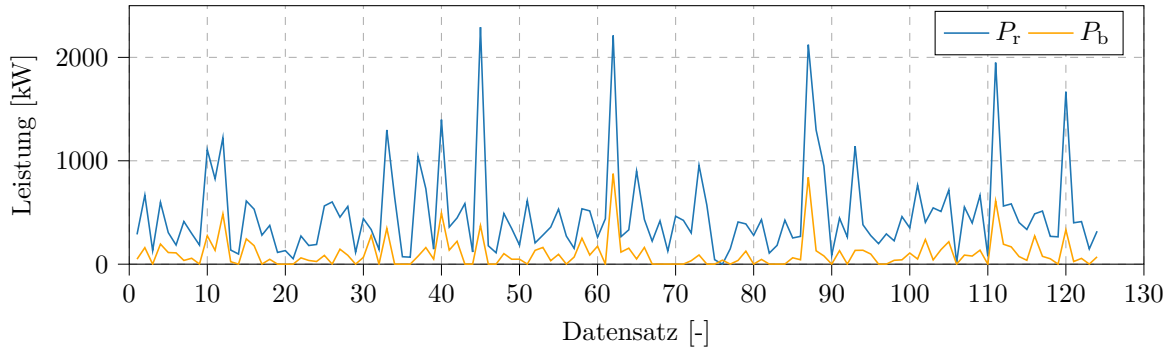


Abbildung 5.1: Absolute Differenz zwischen im Windpark real gemessenen und berechneten Leistungen der Anlage 5 für $70^\circ \leq \alpha \leq 90^\circ$ über 124 Datensätze

Die Ergebnisse entsprechen den Erwartungen und bestätigen die Abschattung des Messmastes. Die realen Leistungsdaten sind immer größer als die berechneten, da die realen Windgeschwindigkeiten vor Anlage 5 immer höher sind als die vom Messmast gemessenen.

Da die Abschattung des Windparks nicht exakt zu ermitteln ist und der Windrichtungssektor auch keine klare Grenze hat, wird keine Korrektur mit z.B. einem Faktor durchgeführt, der die Windgeschwindigkeit künstlich erhöht. Stattdessen wird der Sektor, in dem die Abschattung stattfindet, aus den Daten herausgefiltert.

Die Windrichtung, bei der der Messmast in einer Linie mit WEA 4 steht, wird als untere Grenze definiert, ergibt sich geometrisch nach

$$\alpha_u = 90^\circ - \arctan\left(\frac{y_4 - y_M}{x_4 - x_M}\right) \quad (5.1)$$

und beträgt $\alpha_u = 51,95^\circ$. (x_M, y_M) sind hierbei die Koordinaten des Messmastes.

Die Windrichtung, bei der WEA 3 in einer Linie mit dem Messmast steht, kann als obere Grenze definiert werden, ergibt sich geometrisch nach

$$\alpha_o = 90^\circ - \arctan\left(\frac{y_3 - y_M}{x_3 - x_M}\right) \quad (5.2)$$

und beträgt $\alpha_o = 102,54^\circ$.

Da sich die Nachlaufströmung linear ausbreitet, werden die Grenzen großzügig gewählt und jeweils ca. 15° addiert bzw. subtrahiert. Somit ergibt sich ein Bereich von

$$37^\circ < \alpha < 117^\circ, \quad (5.3)$$

der aus den Daten rausgefiltert wird. Es bleibt somit ein Bereich von

$$117^\circ \leq \alpha \leq 360^\circ \wedge 0^\circ \leq \alpha \leq 37^\circ, \quad (5.4)$$

der betrachtet wird.

Nach allen erwähnten Filterungen bleiben bspw. für das Jahr 2019 insgesamt 12401 Datensätzen übrig. Im Jahr 2020 sind es 17794 Datensätze. Es stehen damit insgesamt 30195 Datensätze zur Verfügung.

6 Validierung des digitalen Zwillings

Dieses Kapitel dient zur Validierung des digitalen Zwillings mit ausgewählten realen Windparkdaten. Die Validierung erfolgt exemplarisch mit Daten aus dem Jahr 2019. Diese Daten sind in der Datei `2019_mod.xlsx` zu finden. Die Validierung erfolgt über das Skript `digital_twin_valid.py`.

Da jede Anlage des Windparks aktuell immer bei einem Induktionsfaktor von $a = \frac{1}{3}$ betrieben wird (wie bereits erwähnt streng genommen im Volllastbereich bei Windgeschwindigkeiten $u > u_n$ nicht mehr), kann die Validierung auch nur für diesen Induktionsfaktor erfolgen. Es ist sinnvoll, die Validierung bei Windrichtungen durchzuführen, bei denen Anlagen im Nachlauf von anderen Anlagen stehen und somit die komplexe Nachlauf-Berechnung im digitalen Zwilling relevant wird. Neben einem qualitativen Vergleich der berechneten und realen Leistungskurven sowie ausgewählten Datensätzen für zwei Fälle, folgt ebenfalls eine quantitative Betrachtung in Form von gemittelten relativen Abweichungen für diese zwei Fälle sowie den gesamten Datensatz aus dem Jahr 2019.

Im ersten Fall werden Windrichtungen betrachtet, in denen WEA 2 und WEA 3 im Nachlauf von WEA 1 und somit auch WEA 5 im Nachlauf von WEA 4 steht (Datei: `2019_fall1.xlsx`). Im zweiten Fall werden Windrichtungen betrachtet, in denen WEA 4 im Nachlauf von WEA 1 und WEA 5 im Nachlauf von WEA 2 steht (Datei: `2019_fall2.xlsx`).

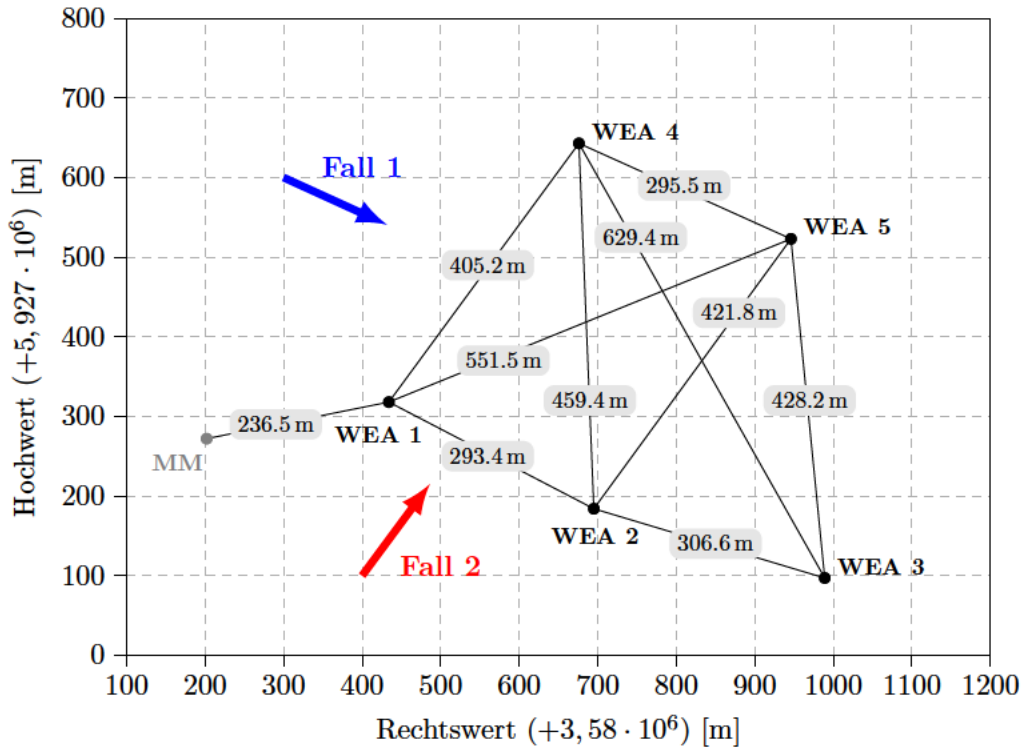


Abbildung 6.1: Windpark Curslack mit eingezeichneten mittleren Windrichtungen der jeweiligen Sektoren der betrachteten Fälle

6.1 Vergleich der Leistungen für ausgewählte Fälle

Im Folgenden werden die realen und berechneten Leistungen für die ausgewählten Fälle direkt miteinander verglichen.

6.1.1 Fall 1

Der exakte Windrichtungssektor für Fall 1 ergibt sich analog zur Berechnung des Windrichtungssektors für die Abschattung des Messmastes (vgl. Gl. 5.1) über die arctan-Funktion. Es gelten somit folgende Windrichtungen:

$$\alpha_{1 \rightarrow 2} = 297,18^\circ, \quad (6.1)$$

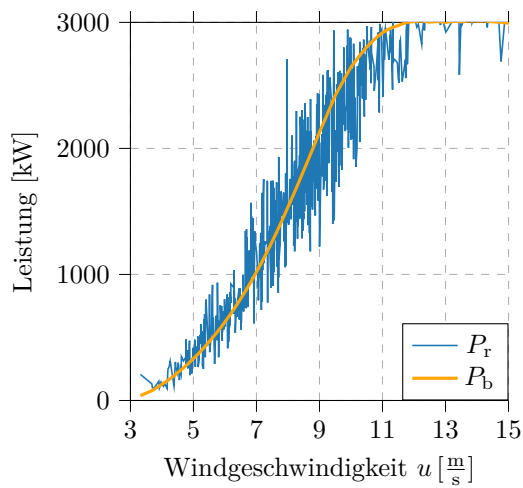
$$\alpha_{1 \rightarrow 3} = 291,71^\circ, \quad (6.2)$$

$$\alpha_{4 \rightarrow 5} = 293,96^\circ. \quad (6.3)$$

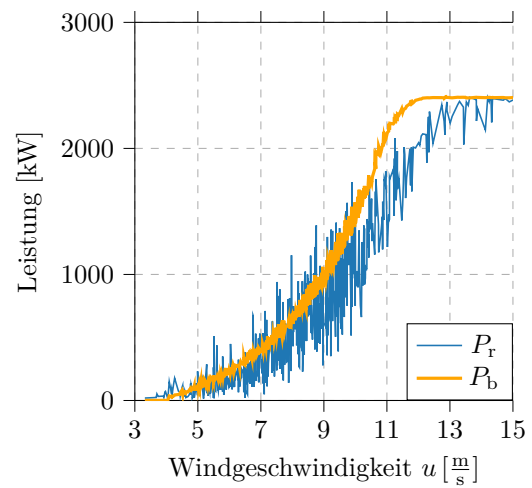
$\alpha_{1 \rightarrow 2}$ ist somit die Windrichtung, bei der Anlage 1 Anlage 2 abschattet. Es erfolgt daher eine Filterung der Datensätze nach

$$291^\circ \leq \alpha \leq 298^\circ, \quad (6.4)$$

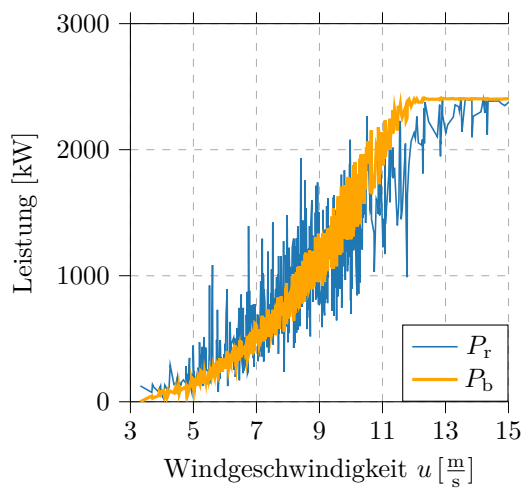
welche zu 667 Datensätzen führt. Nachfolgende Abbildungen zeigen die realen Leistungen P_r und die korrespondierenden berechneten Leistungen P_b dieser Datensätze für jede Anlage in Abhängigkeit der Windgeschwindigkeit u .



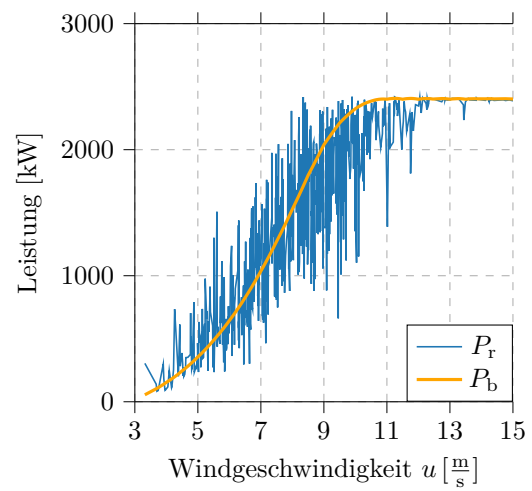
(a) Anlage 1



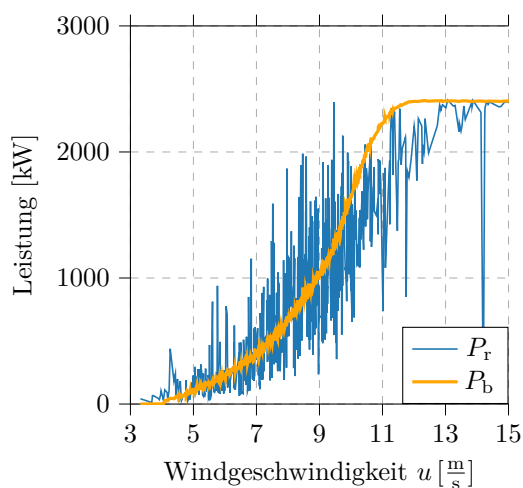
(b) Anlage 2



(c) Anlage 3



(d) Anlage 4



(e) Anlage 5

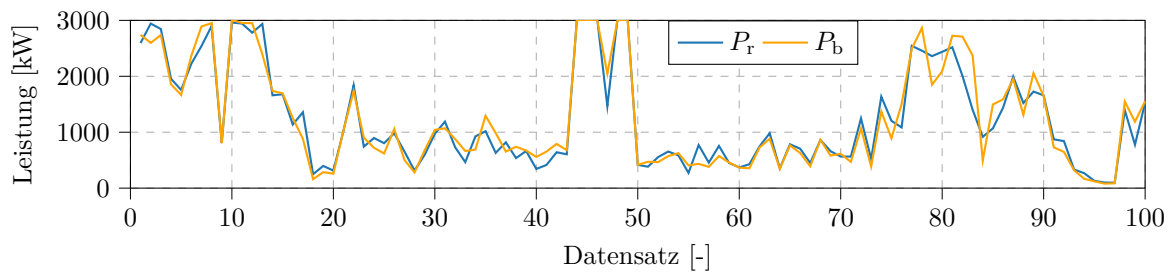
Abbildung 6.2: Reale und berechnete Leistungen P_r, P_b für $\bar{u}_\infty = 8,51 \frac{\text{m}}{\text{s}}$, $\bar{\alpha} = 294,55^\circ$, $\bar{\sigma}_{u,0} = 1,26 \frac{\text{m}}{\text{s}}$ über 667 Datensätze (Fall 1)

Anhand der Ergebnisse lassen sich einige Zusammenhänge erklären. Die realen Leistungswerte schwanken bei allen Anlagen stark. Ein Grund dafür sind die verschiedenen Windrichtungen, allerdings beträgt die Größe des Sektors nur 7° (vgl. Gl. 6.4). Daher lässt sich dennoch sagen, dass die Leistungswerte auch bei näherungsweise gleichen Umgebungsparametern teilweise stark schwanken. Auch Turbulenz des Windes sowie dessen Höhenwindgradient haben großen Einfluss auf Messunsicherheiten und resultieren in Schwankungen. [25, K.9.2] Die errechnete Leistung bildet dennoch für eine gewisse Windgeschwindigkeit näherungsweise den Mittelwert des für diese Windgeschwindigkeit vorliegenden Spektrums an real gemessenen Leistungen ab.

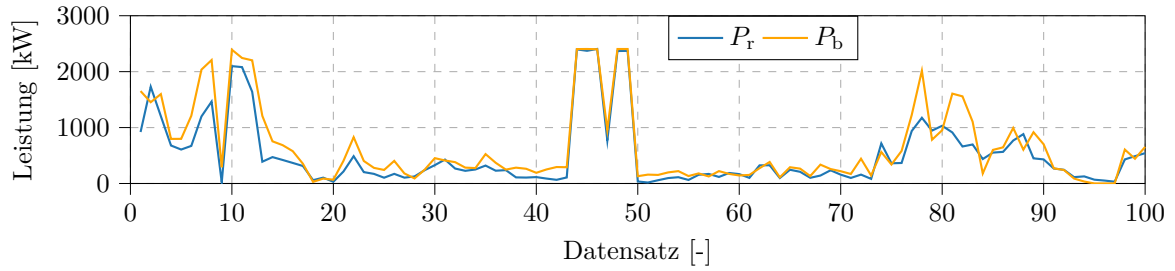
Ebenfalls zu beobachten ist, dass die berechneten Leistungen bei Anlage 2, 3 und 5 streuen, während dies bei Anlage 1 und 4 quasi nicht der Fall ist. Dies ist auf den betrachteten Fall zurückzuführen. Anlage 2 und 5 stehen im Nachlauf von Anlage 1 und 4, Anlage 3 sogar im Nachlauf von Anlage 1 und 2. Es wird festgehalten, dass die errechneten Leistungen für eine gewisse Windgeschwindigkeit umso stärker streuen, je mehr die Anlage im Nachlauf von anderen Anlagen steht. Das liegt daran, dass der Einfluss der Windrichtung auf diese Anlagen viel größer ist als auf die vorderen „frei“ stehenden Anlagen. Ob Anlage 1 aus einer Richtung von 291° oder 298° angeströmt wird, hat keinen Einfluss auf die Leistung, wohingegen bei Anlage 3 sehr wohl Relevanz der Richtung besteht, da diese den Nachlauf und somit die Berechnung im digitalen Zwillings stark beeinflusst.

Die berechneten Leistungen werden für niedrige Windgeschwindigkeiten $3 \frac{\text{m}}{\text{s}} \leq u \leq 7 \frac{\text{m}}{\text{s}}$ tendenziell eher unterschätzt und für größere Windgeschwindigkeiten $10 \frac{\text{m}}{\text{s}} \leq u \leq 13 \frac{\text{m}}{\text{s}}$ überschätzt. Dies ist schlüssig, wenn beachtet wird, dass aus diesem Grund im DWM Modell auch das Mäandern des Wakes mit einbezogen wird. [52]

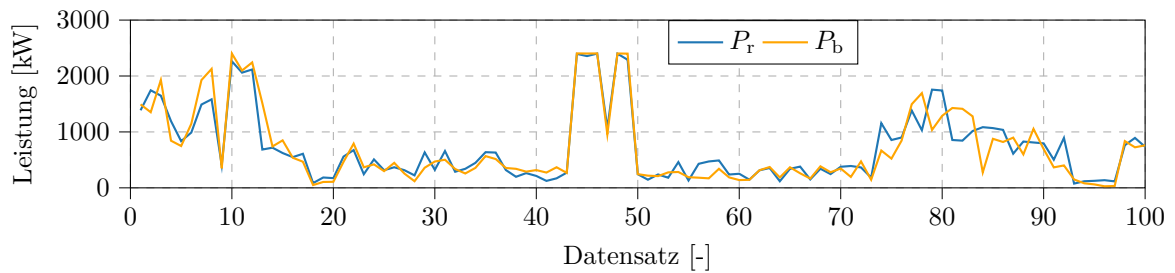
Um die Abweichungen visuell besser darzustellen, werden die realen und berechneten Leistungen auch über 100 ausgewählte Datensätze geplottet. In diesem Fall dargestellt sind die Datensätze 100 - 200 aus den gesamten 667 Datensätzen, da in den Datensätzen 0 - 100 oft eine Windgeschwindigkeit über Nennwindgeschwindigkeit vorzufinden ist und somit nahezu jede Anlage Nennleistung erzeugt.



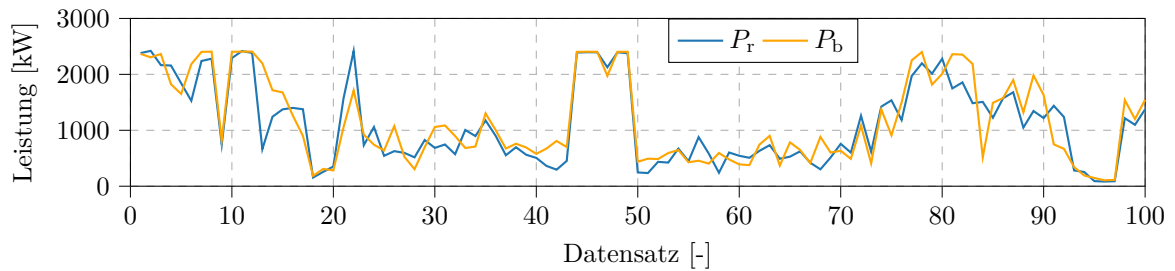
(a) Anlage 1



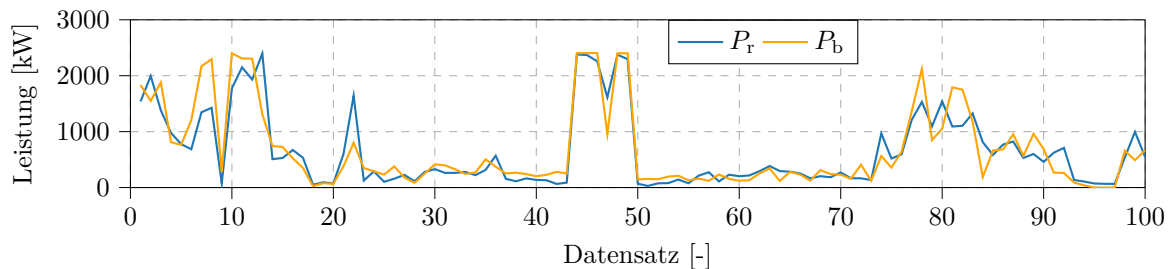
(b) Anlage 2



(c) Anlage 3



(d) Anlage 4



(e) Anlage 5

Abbildung 6.3: Reale und berechnete Leistungen P_r, P_b für $\bar{u}_\infty = 7,4 \frac{\text{m}}{\text{s}}, \bar{\alpha} = 294,38^\circ, \bar{\sigma}_{u,0} = 1,05 \frac{\text{m}}{\text{s}}$ über 100 Datensätze (Fall 1)

6 Validierung des digitalen Zwillings

Allgemein lässt sich sagen, dass die berechneten Leistungen aller Anlagen in akzeptabler Übereinstimmung mit den realen Werten sind. Quantitativ sind Abweichungen vorhanden, doch vor allem qualitativ stimmen die beiden Graphen für die Anlagen gut überein.

6.1.2 Fall 2

Das Vorgehen in Fall 2 ist identisch zu dem in Fall 1. Auch hier ergeben sich die Windrichtungen über die arctan-Funktion. Es gilt diesmal

$$\alpha_{1 \rightarrow 4} = 216,67^\circ, \quad (6.5)$$

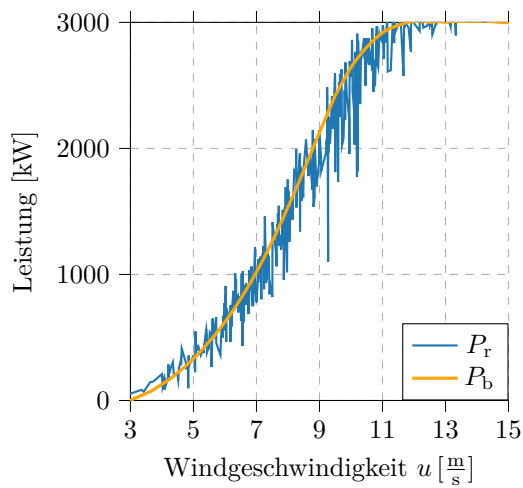
$$\alpha_{2 \rightarrow 5} = 216,52^\circ. \quad (6.6)$$

Es erfolgt diesmal eine Filterung der Datensätze nach

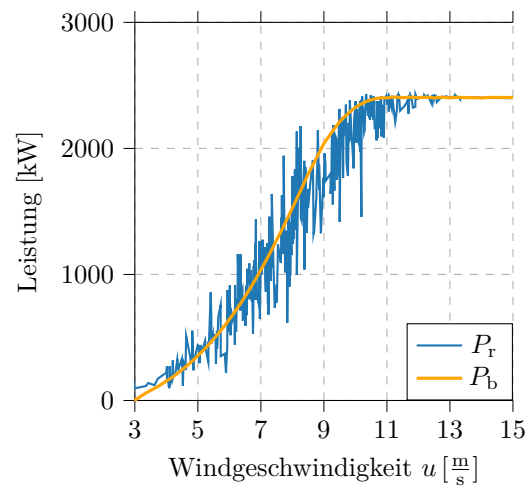
$$214^\circ \leq \alpha \leq 219^\circ, \quad (6.7)$$

welche zu 398 Datensätzen führt. Nachfolgende Abbildungen zeigen auch hier die reale Leistungen P_r und die korrespondierenden berechneten Leistungen P_b dieser Datensätze für jede Anlage in Abhängigkeit der Windgeschwindigkeit u .

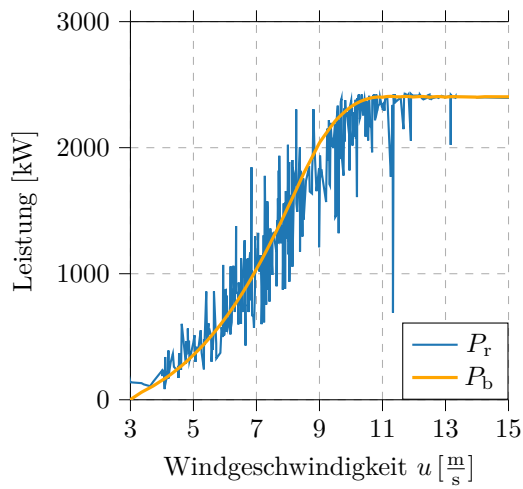
Die Erkenntnisse von Fall 1 lassen sich hier exakt übertragen. Gut zu erkennen ist die Streuung der berechneten Werte bei Anlage 4 und 5, also wieder den Anlagen, die im Nachlauf von anderen Anlagen stehen. Auch die Stärke der Streuung von Anlage 4 und 5 stimmt mit der von Anlage 2 und 5 aus Fall 1 (vgl. Abb 6.2b und Abb. 6.2e) überein. Dies macht Sinn, da Anlage 4 und 5 in Fall 2 sowie Anlage 2 und 5 in Fall 1 jeweils im Nachlauf von exakt einer Anlage stehen. Analog zu Fall 1 sind auch hier die realen und berechneten Leistungen über die Datensätze 100 - 200 von den insgesamt 398 Datensätzen geplottet. Auch hier lässt sich bei allen Anlagen zwar eine quantitative Abweichung feststellen, dennoch folgt der Graph der berechneten Leistungen qualitativ dem der realen Leistungen.



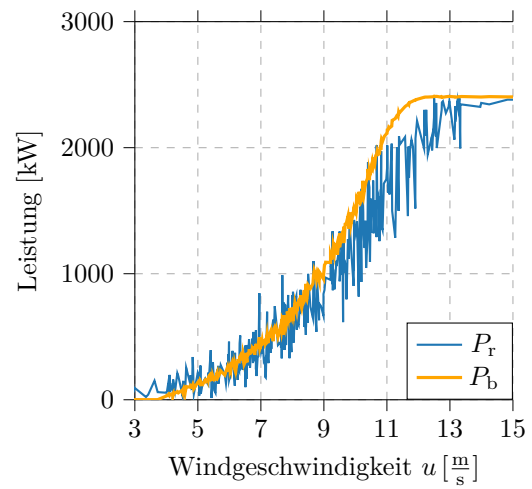
(a) Anlage 1



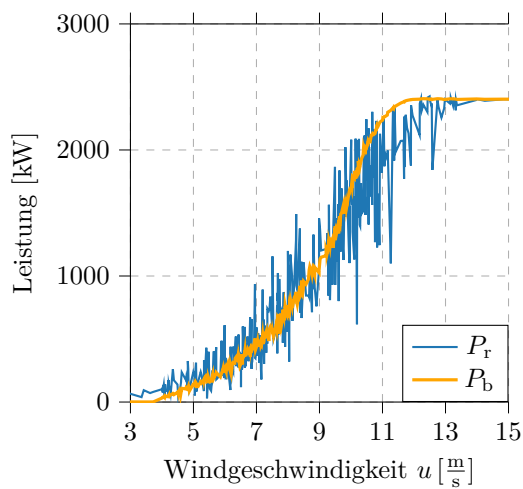
(b) Anlage 2



(c) Anlage 3



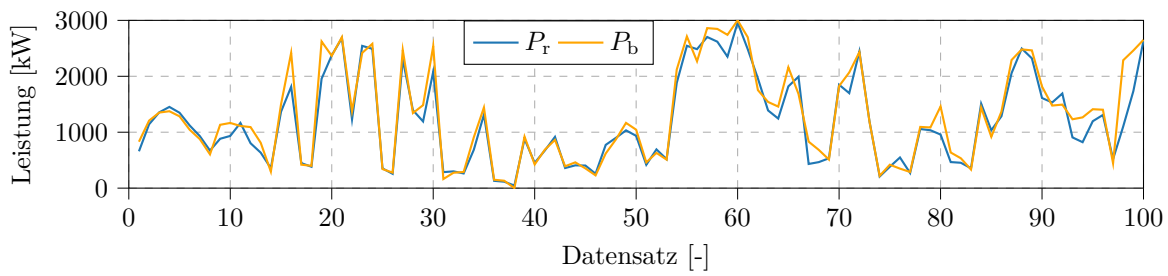
(d) Anlage 4



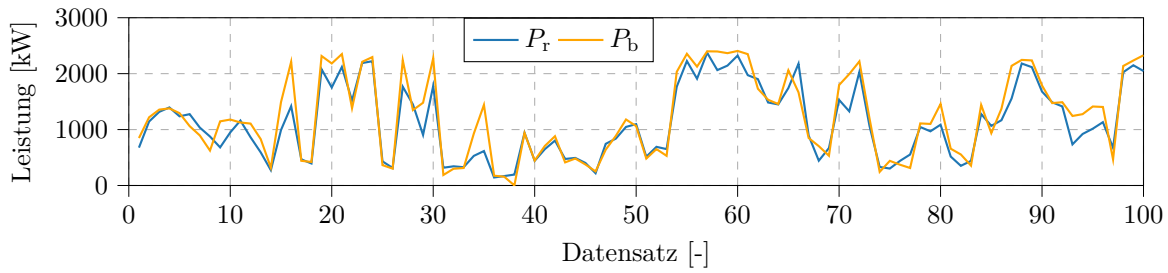
(e) Anlage 5

Abbildung 6.4: Reale und berechnete Leistungen P_r, P_b für $\bar{u}_\infty = 8,37 \frac{\text{m}}{\text{s}}$, $\bar{\alpha} = 216,55^\circ$, $\bar{\sigma}_{u,0} = 1,09 \frac{\text{m}}{\text{s}}$ über 398 Datensätze (Fall 2)

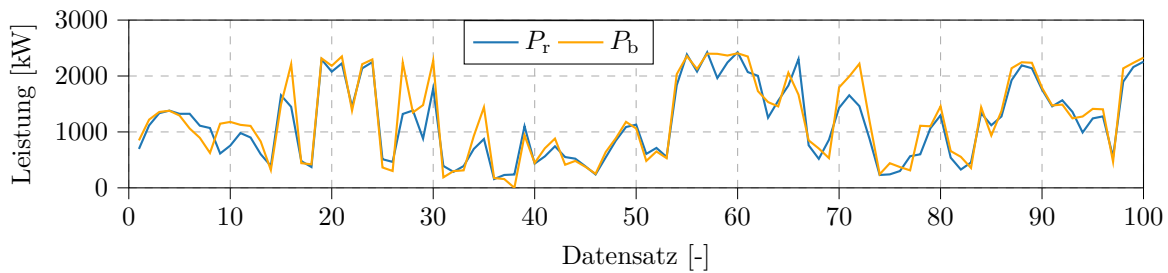
6 Validierung des digitalen Zwillings



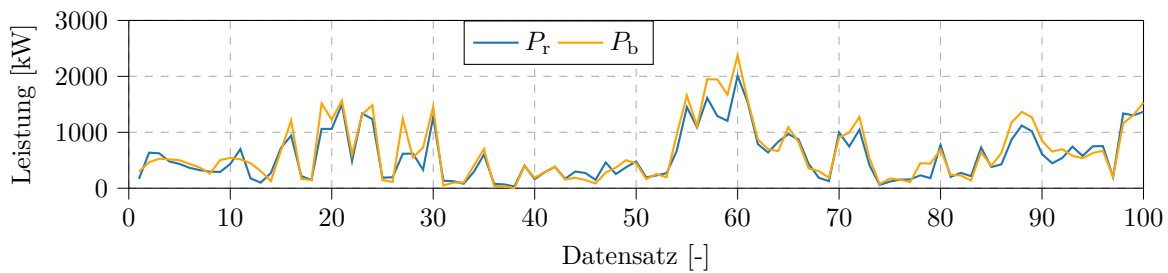
(a) Anlage 1



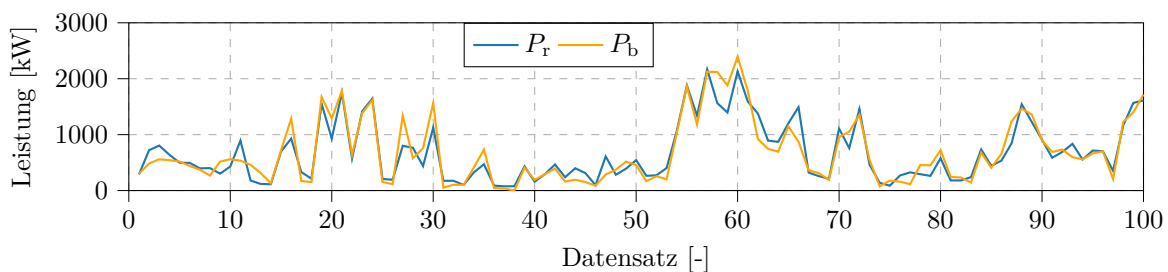
(b) Anlage 2



(c) Anlage 3



(d) Anlage 4



(e) Anlage 5

Abbildung 6.5: Reale und berechnete Leistungen P_r, P_b für $\bar{u}_\infty = 7,3 \frac{\text{m}}{\text{s}}$, $\bar{\alpha} = 216,77^\circ$, $\bar{\sigma}_{u,0} = 0,93 \frac{\text{m}}{\text{s}}$ über 100 Datensätze (Fall 2)

6.2 Relative Abweichungen für ausgewählte Fälle

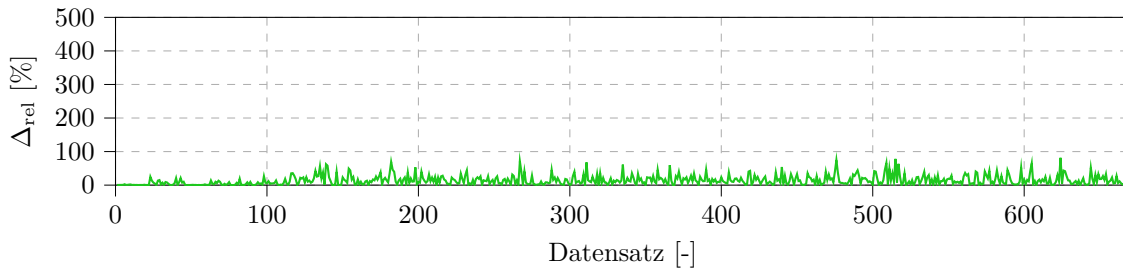
Absolute Abweichungen, so wie sie bisher betrachtet wurden, haben statistisch wenig Aussagekraft, da eine absolute Abweichung von 100 kW bei einer realen Leistung von 200 kW natürlich gravierender als bei einer realen Nennleistung von 2400 kW ist. Aus diesem Grund sollen in diesem Abschnitt die relativen Abweichungen für die beiden bereits vorgestellten Fälle betrachtet und analysiert werden. Die relative Abweichung ergibt sich allgemein nach

$$\Delta_{\text{rel}} = \frac{|P_b - P_r|}{P_r} \cdot 100\%. \quad (6.8)$$

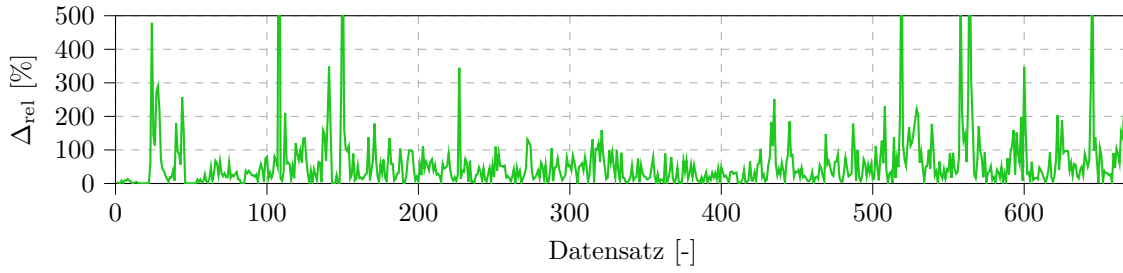
6.2.1 Fall 1

Nachfolgend sind die relativen Abweichungen für alle Anlagen für die 667 Datensätze aus Fall 1 aufgeführt. Genau genommen sind es nur 666 Datensätze, da in einem Datensatz Anlage 5 eine Leistung von 0 kW aufweist, obwohl alle anderen Anlagen Leistung erzeugen. Da dies zu einer unendlich hohen relativen Abweichung führt, wird dieser Datensatz gelöscht (Datei: 2019_fall11_ohne_0_datensatz.xlsx).

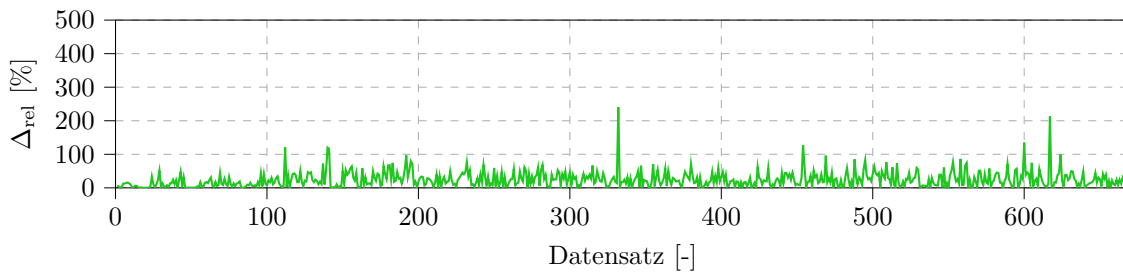
6 Validierung des digitalen Zwillings



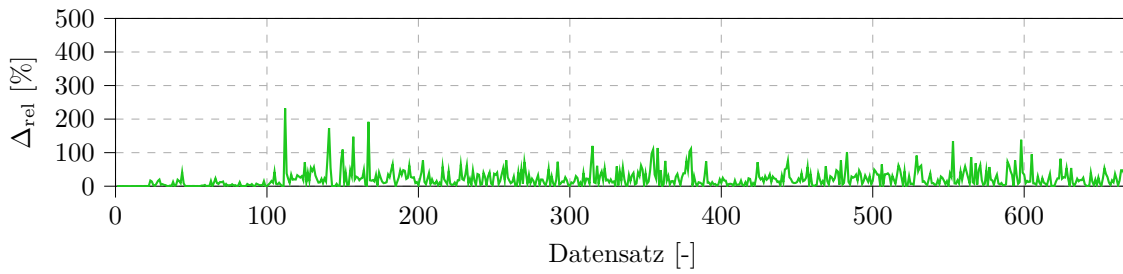
(a) Anlage 1



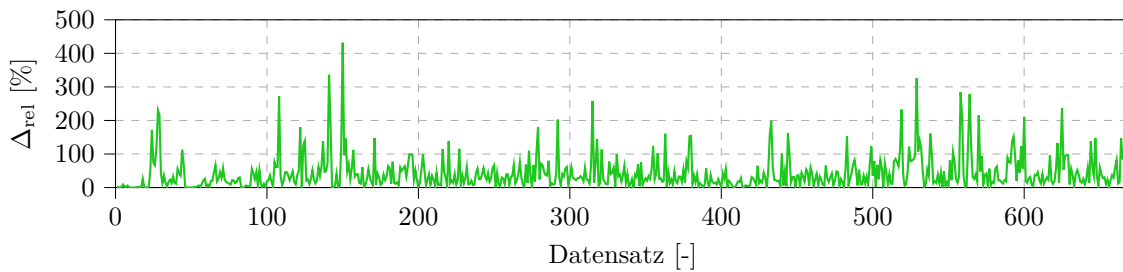
(b) Anlage 2



(c) Anlage 3



(d) Anlage 4



(e) Anlage 5

Abbildung 6.6: Relative Abweichungen Δ_{rel} für $\bar{u}_{\infty} = 8,51 \frac{\text{m}}{\text{s}}$, $\bar{\alpha} = 294,55^{\circ}$, $\bar{\sigma}_{u,0} = 1,26 \frac{\text{m}}{\text{s}}$ über 666 Datensätze (Fall 1)

Die gemittelten relativen Abweichungen über alle Datensätze aus Fall 1 sind der nachfolgenden Tabelle zu entnehmen.

	WEA 1	WEA 2	WEA 3	WEA 4	WEA 5
$\bar{\Delta}_{\text{rel}} [\%]$	14,19	58,45	24,48	22,11	42,47

Tabelle 6.1: Gemittelte relative Abweichungen $\bar{\Delta}_{\text{rel}}$ für $\bar{u}_{\infty} = 8,51 \frac{\text{m}}{\text{s}}$, $\bar{\alpha} = 294,55^{\circ}$, $\bar{\sigma}_{u,0} = 1,26 \frac{\text{m}}{\text{s}}$ über 666 Datensätze (Fall 1)

Die gemittelten Abweichungen der Leistungen für diese 666 Datensätze sind mit der Abweichung des Ertrags über diesen Zeitraum gleichzusetzen, da sich der Ertrag aus dem Produkt von Leistung und Zeit ergibt und die Zeit bei einer Betrachtung der Abweichung der Erträge gekürzt werden kann. Von einem definierten Stunden- oder Tagesertrag kann jedoch nicht gesprochen werden, da die Leistungsdaten durch die Filterung bezogen auf die Zeit Lücken aufweisen. Wie Tabelle 6.1 zu entnehmen ist, weist Anlage 1 die geringste mittlere Abweichung auf. Dies lässt sich zum einen darauf zurückführen, dass diese Anlage nicht im Nachlauf von anderen Anlagen steht und die Berechnung der Leistung somit nicht auf einer komplexen Berechnung beruht, und zum anderen darauf, dass der Messmast am nächsten an Anlage 1 steht und in der Berechnung die Windgeschwindigkeit des Messmastes als Umgebungsgeschwindigkeit verwendet wird. Dass die Abweichung von Anlage 4 größer ist als die von Anlage 1, obwohl Anlage 4 für den Windrichtungssektor ebenfalls nicht im Nachlauf von anderen Anlagen steht, ist darauf zurückzuführen, dass für diese Anlage im digitalen Zwillings mit der Windgeschwindigkeit vom Messmast gerechnet wird. Da Anlage 4 mehr als 600 m von Messmast entfernt ist, ist die Annahme, dass die Geschwindigkeit vor Anlage 4 der vom Messmast gleicht, sehr grob. Anlage 2, 3 und 5 weisen größere mittlere Abweichungen auf, was vermutlich auf den Fakt zurückzuführen ist, dass diese im Nachlauf von anderen Anlagen stehen und die Ungenauigkeiten des digitalen Zwillings hier relevant werden. Auffällig ist, vor allem bei Anlage 2 und Anlage 5, dass die meisten relativen Abweichungen konstant und in einem vertretbaren Bereich sind, aber bei vereinzelt Datensätzen enorm große Peaks vorzufinden sind (vgl. Abb. 6.6b, Abb. 6.6e). Diese sind bei Anlage 2 teilweise größer als 500 %, dennoch wird die y -Achse zwecks Vergleichbarkeit gleich der der anderen Anlagen gewählt. Die Abweichungen nehmen hier Werte bis zu 1400 % an. Bei einer Anzahl an Datensätzen von „nur“ 666 verfälschen diese Außreißer den Mittelwert signifikant, was sich in der gemittelten relativen Abweichung von 58,45 % widerspiegelt.

6.2.2 Fall 2

Analog zu Fall 1 werden auch die relativen Abweichungen für die 398 Datensätze aus Fall 2 im Folgenden aufgeführt.

6 Validierung des digitalen Zwillings

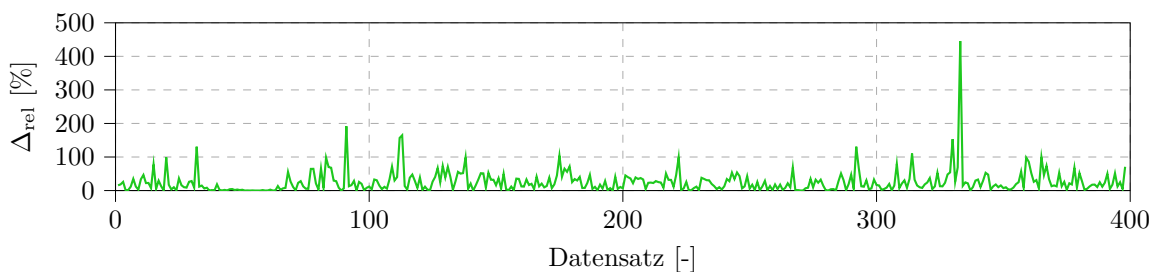
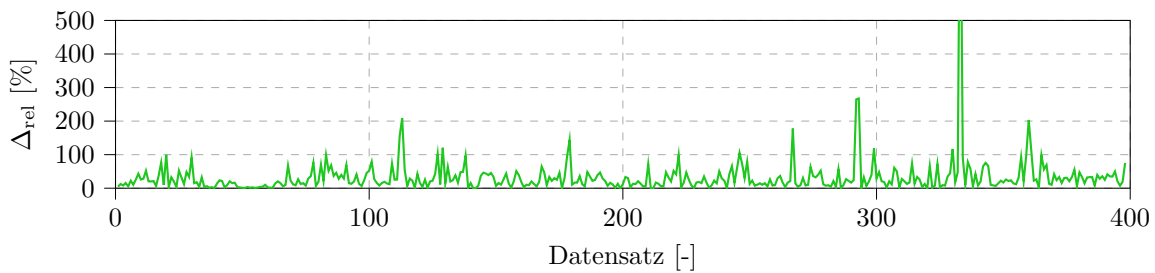
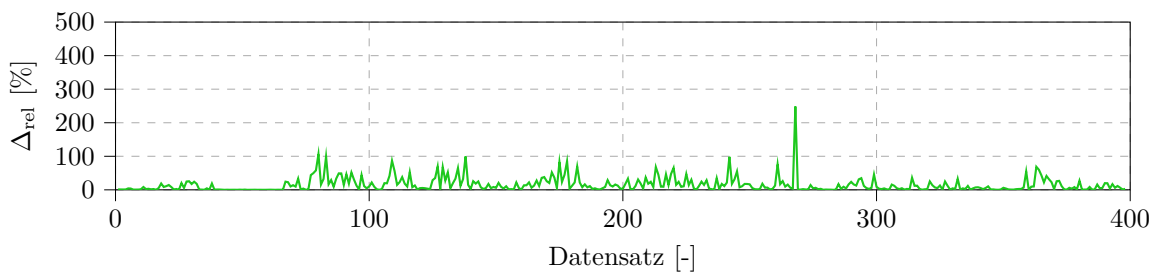
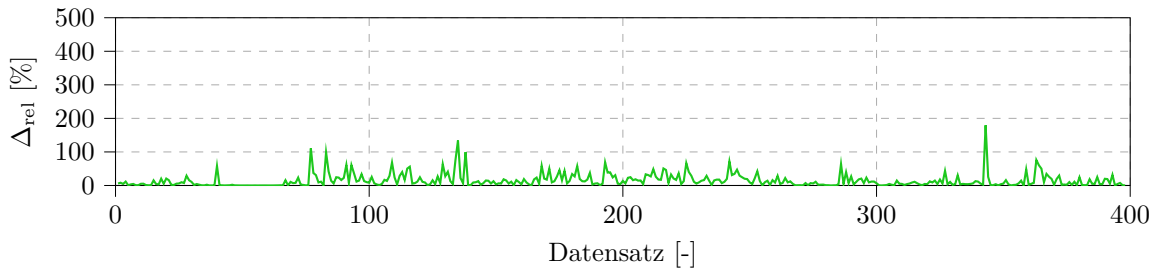
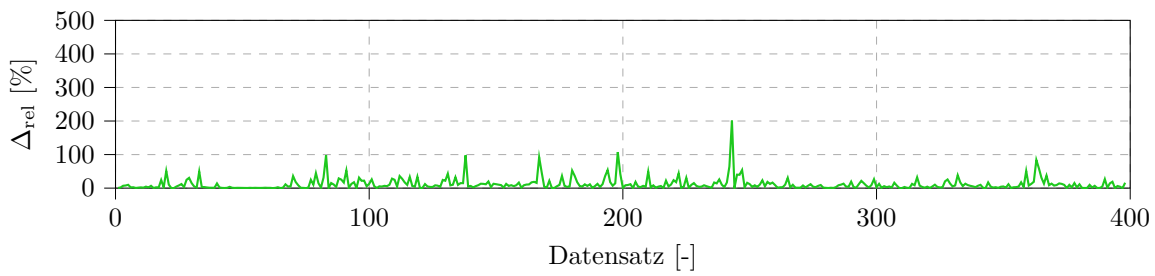


Abbildung 6.7: Relative Abweichungen Δ_{rel} für $\bar{u}_\infty = 8,37 \frac{\text{m}}{\text{s}}$, $\bar{\alpha} = 216,55^\circ$, $\bar{\sigma}_{u,0} = 1,09 \frac{\text{m}}{\text{s}}$ über 398 Datensätze (Fall 2)

Die gemittelten relativen Abweichungen sind in der nachfolgenden Tabelle aufgeführt.

	WEA 1	WEA 2	WEA 3	WEA 4	WEA 5
$\overline{\Delta}_{\text{rel}} [\%]$	11,95	15,04	15,38	32,58	25,72

Tabelle 6.2: Gemittelte relative Abweichungen $\overline{\Delta}_{\text{rel}}$ für $\bar{u}_{\infty} = 7,4 \frac{\text{m}}{\text{s}}$, $\bar{\alpha} = 294.38^\circ$, $\bar{\sigma}_{u,0} = 1.05 \frac{\text{m}}{\text{s}}$ über 398 Datensätze (Fall 2)

Auch diese Werte decken sich mit den Erwartungen und bestätigen die bereits aufgeführten Schlussfolgerungen aus Fall 1. Anlage 1, 2 und 3 stehen für den Windrichtungssektor in keinem Nachlauf von anderen Anlagen, weshalb die gemittelte relative Abweichung dort am geringsten ist. Anlage 1 weist aufgrund der kleinsten Distanz zum Messmast wieder die geringste Abweichung auf. Wie auch in Fall 1 können einzelne enorm hohe Abweichungen beobachtet werden, wie z.B. bei Anlage 4 und 5 (vgl. Abb. 6.7d, Abb. 6.7e). Diese haben auch hier signifikanten Einfluss auf den Mittelwert, durch den kleineren Datensatz sogar mehr als in Fall 1.

6.2.3 Modifikation der relativen Abweichungen

Werden die relativen Abweichungen der beiden Fälle zusammen über die realen Leistungen statt den Datensätzen aufgetragen, kann ein exponentieller Verlauf erkannt werden.

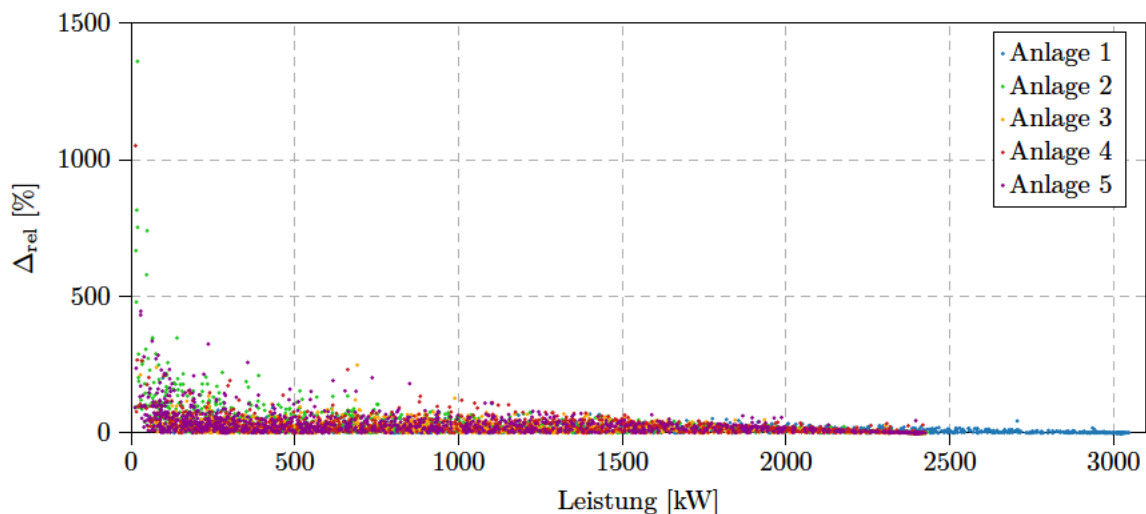


Abbildung 6.8: Relative Abweichung Δ_{ref} für Fall 1 + Fall 2 (666 + 398 = 1064 Datensätze)

Je kleiner die reale Leistung P_r ist, desto größer fallen die relativen Abweichungen aus. Der Verlauf ist schlüssig, da die selbe absolute Abweichung zwischen realer und berechneter Leistung bei einer kleineren realen Leistung mathematisch zu einer größeren relativen Abweichung führt. Durch Visualisieren der Verteilungen der Windgeschwindigkeiten für Fall 1 und Fall 2 wird deutlich, dass die geringen Windgeschwindigkeiten $3 \frac{\text{m}}{\text{s}} \leq u < 5 \frac{\text{m}}{\text{s}}$, die in kleinen realen Leistungen resultieren, nur einen sehr geringen Anteil von allen Datensätzen ausmachen. Für Fall 1 sind es 5,64% und Fall 2 8,79%.

6 Validierung des digitalen Zwillings

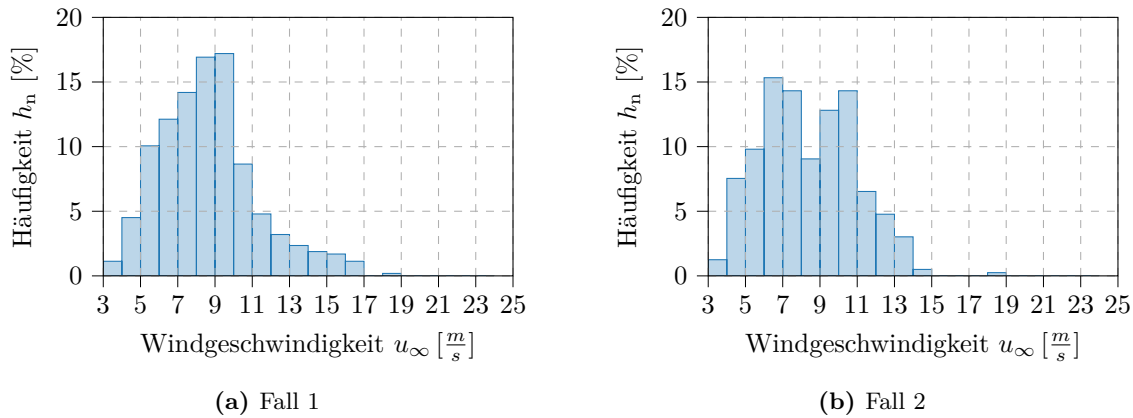


Abbildung 6.9: Häufigkeitsverteilungen der am Messmast ermittelten Windgeschwindigkeiten für beide Fälle

Bei einer Betrachtung des Ertrags über einen längeren Zeitraum würden die kleinen Leistungen daher kaum ins Gewicht fallen. Aus diesem Grund wird die Berechnung der relativen Abweichung modifiziert. Wenn die reale Leistung unter 5 % der Nennleistung liegt, wird für den betrachteten Datensatz und die relevante Anlage keine relative Abweichung berechnet bzw. dieser Wert verworfen. Dies führt zu unterschiedlichen Anzahlen von Abweichungen, die zur Berechnung des Mittelwertes genutzt werden. Diese Differenz ist aber vernachlässigbar. Die mittleren Abweichungen haben trotzdem näherungsweise die selbe Aussagekraft.

Natürlich muss erwähnt werden, dass es auch vorkommt, dass die Umgebungswindgeschwindigkeit mittelhoch bis hoch ist, eine Anlage aber im exakten Nachlauf einer anderen steht und diese Anlage dann eine sehr geringe Windgeschwindigkeit erfährt und somit wenig Leistung produziert. Die Rechtfertigung der Modifikation anhand der Häufigkeit der Umgebungswindgeschwindigkeit funktioniert daher nur bedingt.

Schlussendlich ergeben sich folgende gemittelte modifizierte relative Abweichungen, welche mit $\overline{\Delta}'_{\text{rel}}$ gekennzeichnet sind. Zwecks Vergleichbarkeit sind die bereits ermittelten Abweichungen ebenfalls mit aufgeführt.

Fall	Abweichung	WEA 1	WEA 2	WEA 3	WEA 4	WEA 5
1	$\overline{\Delta}_{\text{rel}}$ [%]	14,19	58,45	24,48	22,11	42,47
	$\overline{\Delta}'_{\text{rel}}$ [%]	14,00	41,14	23,51	21,85	34,90
2	$\overline{\Delta}_{\text{rel}}$ [%]	11,95	15,04	15,38	32,58	25,72
	$\overline{\Delta}'_{\text{rel}}$ [%]	10,65	14,36	15,18	25,11	21,32

Tabelle 6.3: Gemittelte relative Abweichungen $\overline{\Delta}_{\text{rel}}$ und gemittelte modifizierte relative Abweichungen $\overline{\Delta}'_{\text{rel}}$ für beide Fälle

Wie erwartet, verringern sich die Werte durch die Modifikation enorm. Bspw. in Fall 1 beträgt die relative Abweichung nur noch 41,14 % statt 58,45 %. Dieser Wert ist gleichzeitig die größte Abweichung unter allen Anlagen und den zwei betrachteten Fällen. Wird dazu noch beachtet, dass

auch Anlage 1, die die besten Bedingungen für eine geringe relative Abweichung hat (geringster Abstand zum Messmast, Anlage steht in keinem Nachlauf für die beiden betrachteten Fälle), eine minimale Abweichung von 10,65 % aufweist, sind die Ergebnisse aus Tabelle 6.3 durchaus akzeptabel.

6.3 Relative Abweichungen in Bezug auf Windrichtung

In diesem Abschnitt sollen die relativen Abweichungen in Bezug auf alle Windrichtungen für das ganze Jahr 2019 untersucht und diskutiert werden. Dazu werden Windrichtungssektoren definiert und die 12401 Datensätze werden in die jeweiligen Sektoren aufgeteilt. Nach Gl. 5.3 ergibt sich ein Bereich von 280° , der in sieben Sektoren wie folgt gleichmäßig aufgeteilt wird:

S1 = $117^\circ - 157^\circ$	1369 Datensätze
S2 = $157^\circ - 197^\circ$	1041 Datensätze
S3 = $197^\circ - 237^\circ$	2716 Datensätze
S4 = $237^\circ - 277^\circ$	2967 Datensätze
S5 = $277^\circ - 317^\circ$	2975 Datensätze
S6 = $317^\circ - 357^\circ$	672 Datensätze
S7 = $357^\circ - 37^\circ$	661 Datensätze
<hr/>	
Σ	12401 Datensätze

Die dazugehörigen Dateien sind unter den Namen S1.xlsx, S2.xlsx, ... zu finden. Die nachfolgende Tabelle zeigt die gemittelten modifizierten relativen Abweichungen über alle sieben Windrichtungssektoren.

	Sektor	WEA 1	WEA 2	WEA 3	WEA 4	WEA 5
$\overline{\Delta}'_{\text{rel}} [\%]$	S1	28,06	21,50	24,25	30,44	26,15
	S2	11,48	15,72	18,65	30,43	35,71
	S3	12,00	16,80	17,95	26,47	29,72
	S4	9,48	16,13	24,57	20,56	24,21
	S5	14,39	31,91	28,32	21,66	32,15
	S6	21,15	23,05	26,28	22,84	24,82
	S7	25,81	33,44	35,43	27,99	32,06

Tabelle 6.4: Gemittelte modifizierte relative Abweichungen $\overline{\Delta}'_{\text{rel}}$ für sieben Windrichtungssektoren

An dieser Stelle sei auf die unterschiedlichen Anzahlen der Datensätze in den verschiedenen Sektoren hingewiesen. Diese Differenz ist auf die Verteilung der Windrichtung für den Standort zurückzuführen und somit nicht beeinflussbar. Für die Auswertung wird angenommen, dass

6 Validierung des digitalen Zwillings

sowohl die Ergebnisse aus S7 mit nur 661 Datensätzen als auch die aus S5 mit 2975 Datensätzen die gleiche Aussagekraft haben.

Die ermittelten Abweichungen decken sich grundlegend mit den Erwartungen. Die kleinste relative Abweichung weist Anlage 1 mit 9,48 % in Sektor 4 auf. Das deckt sich mit den bereits gewonnenen Erkenntnissen, da zum einen in diesem Sektor der Wind aus Südwesten strömt und Anlage 1 stromaufwärts positioniert ist und zum anderen die Umgebungsgeschwindigkeit am Messmast am ehesten der vor Anlage 1 entspricht.

Die größte relative Abweichung weist Anlage 5 mit 35,71 % in Sektor 2 auf. In diesem Sektor steht Anlage 5 aufgrund eines Windes aus Süd / Südost im Nachlauf von teilweise Anlage 2 und teilweise Anlage 3. Dazu kommt, dass Anlage 3 mit grob 800 m am weitesten vom Messmast entfernt positioniert ist. Die Umgebungswindgeschwindigkeit vom Messmast ist also bei Anlage 3 am wenigsten repräsentativ, da durch die große Entfernung andere Bedingungen herrschen können. Die gemittelte Abweichung über den gesamten Datensatz aus dem Jahr 2019 beträgt 24,05 %.

Zusammenfassend kann festgehalten werden, dass immer Abweichungen auftreten, diese aber in einer vertretbaren Größenordnung liegen. Wie erwartet, weisen Anlagen, die im Nachlauf von anderen Anlagen stehen und somit die Berechnung im digitalen Zwillings relevant wird, tendenziell eine größere Abweichung der Leistungen auf als Anlagen, die bezogen auf die Windrichtung stromaufwärts positioniert sind und somit keiner Berechnung der Nachlaufströmung unterliegen. Allerdings ist auch in diesen Fällen eine Abweichung vorhanden, die nicht signifikant kleiner ist. Das ist auf die allgemeinen Schwankungen der gemittelten 10 min-Werte zurückzuführen, wie in Abb. 6.2 und Abb. 6.4 bereits gezeigt wurde. Es wird festgehalten, dass die allgemeine Schwankung der 10 min-Werte sowie die Tatsache, dass für die Umgebungswindgeschwindigkeit immer die Geschwindigkeit am Messmast bei $z = 115$ m benutzt wird, die zwei größten Ursachen für die Abweichungen sind.

6.4 Abweichung der Gesamtleistung des Windparks

Da die realen Leistungen teilweise stark schwanken, soll in diesem Abschnitt die Gesamtleistung des Windparks mit der berechneten Gesamtleistung verglichen werden. Der Vergleich wird für die beiden bereits vorgestellten Fälle sowie für das gesamte Jahr 2019 durchgeführt.

Wie auch bei der Betrachtung der Einzelleistungen werden hier sehr kleine reale Gesamtleistungen nicht in die relativen Abweichungen mit einbezogen (vgl. Abschnitt 6.2.3). Liegt die reale Gesamtleistung des Windparks unter 5 % der Gesamtnennleistung von 12600 kW, wird der Wert verworfen bzw. nicht mit in die Berechnung des Mittelwertes einbezogen.

	Fall 1	Fall 2	Jahr 2019
$\overline{\Delta}'_{\text{rel}} [\%]$	18,58	12,70	17,27

Tabelle 6.5: Gemittelte modifizierte relative Abweichung $\overline{\Delta}'_{\text{rel}}$ der Gesamtleistung des Windparks für Fall 1, Fall 2 und das gesamte Jahr 2019

Wie in Tabelle 6.5 zu sehen ist, sind die Abweichungen der Gesamtleistung des Windparks durchaus geringer als die der Einzelanlagen. Natürlich sind diese Werte kritisch zu betrachten, da sich Abweichungen von Einzelanlagen hier kompensieren und so zu einer geringeren Gesamtabweichung führen. Dennoch kann aufgrund aller vorliegenden Ergebnisse die Validierung des digitalen Zwillings zur Berechnung der Leistungen im Windpark Curslack in Abhängigkeit der Umgebungsparameter als erfolgreich eingestuft werden.

6.5 Untersuchung des Einflusses der Varianz der Dichte

Die IEC-Richtlinie 61400-12 schlägt eine Normierung der Windgeschwindigkeit in Abhängigkeit der Dichte der Luft vor [25, K.9.1.5], welche zunächst ermittelt werden muss. Diese ergibt sich nach

$$\rho'_{\text{Luft}} = \frac{1}{T} \left[\frac{p}{R_L} - \Phi p_W \left(\frac{1}{R_L} - \frac{1}{R_W} \right) \right]. \quad (6.9)$$

Hierbei sind p , T und Φ der Luftdruck, die Temperatur und die relative Luftfeuchte an der Stelle, an der auch die Dichte ermittelt werden soll. Da die Umgebungswindgeschwindigkeit bei $z = 115$ m (Sensor V3) ermittelt wird, auf dieser Höhe aber nicht jede Sensorart vorzufinden ist (vgl. Tabelle 1.1), werden die Größen von den Sensoren P1, T1, H1 bei $z = 113,1$ m entnommen. $R_L = 287,05 \frac{\text{J}}{\text{kg K}}$ ist die Gaskonstante von trockener Luft, $R_W = 461,5 \frac{\text{J}}{\text{kg K}}$ die von Wasserdampf. p_W ist der Dampfdruck. Dieser ergibt sich abhängig von der Temperatur T nach

$$p_W = 2,05 \cdot 10^{-5} \cdot \exp(0,0631846 T). \quad (6.10)$$

Die Normierung bzw. Korrektur der Umgebungswindgeschwindigkeit u_∞ ergibt sich dann nach

$$u'_\infty = u_\infty \left(\frac{\rho'_{\text{Luft}}}{\rho_{\text{Luft}}} \right)^{\frac{1}{3}} \quad (6.11)$$

6 Validierung des digitalen Zwillings

mit einer Standarddichte von $\rho_{\text{Luft}} = 1,225 \frac{\text{kg}}{\text{m}^3}$.

Die Normierung der Geschwindigkeit wird für die 667 Datensätze aus Fall 1 vorgenommen (Datei: `dichte_untersuchung.xlsx`) und die Leistungen werden mit der neu berechneten Geschwindigkeit u'_{∞} ermittelt. Die Abweichungen sind der nachfolgenden Tabelle zu entnehmen.

	WEA 1	WEA 2	WEA 3	WEA 4	WEA 5
$\overline{\Delta}'_{\text{rel,ref}}$	14,00	41,14	23,51	21,85	34,90
$\overline{\Delta}'_{\text{rel,Dichte}}$	13,87	41,16	23,51	21,77	34,86

Tabelle 6.6: Gemittelte modifizierte relative Abweichungen $\overline{\Delta}'_{\text{rel,Dichte}}$ für Fall 1 unter Berücksichtigung der variablen Dichte

Zwecks Vergleichbarkeit sind ebenfalls die Abweichungen für den gesamten Datensatz aus Fall 1 ohne Berücksichtigung der Dichte mit aufgeführt und als $\overline{\Delta}_{\text{rel,ref}}$ gekennzeichnet. Wie zu erkennen ist, hat die Berücksichtigung der Dichte für die ausgewählten Datensätze so gut wie keinen Einfluss auf die gemittelten Abweichungen.

6.6 Untersuchung des Einflusses des Höhenwindgradienten

Neben der Normierung der Geschwindigkeit mit Bezug auf die Dichte kann auch der Höhenwindgradient berücksichtigt werden. Da Windgeschwindigkeiten abhängig von der Höhe stark variieren können, hat Hellmann das Potenzgesetz

$$u_2 = u_1 \left(\frac{h_2}{h_1} \right)^\gamma \quad (6.12)$$

entwickelt, mit Hilfe dessen mit einer bekannten Geschwindigkeit u_1 in einer Höhe h_1 die Geschwindigkeit u_2 in einer Höhe h_2 geschätzt werden kann [21, S. 557]. Der Exponent γ ist bei gewöhnlichen Bedingungen $\gamma = 0,15$.

Auch diese Anpassung wird mit dem Datensatz aus Fall 1 untersucht (Datei: `gradient_untersuchung.xlsx`). Hierzu wird für jeden Datensatz nicht nur die Windgeschwindigkeit bei $z = 115,0 \text{ m}$ (Sensor V3) exportiert, sondern auch die Windgeschwindigkeit bei $z = 64,8 \text{ m}$ (Sensor V9). Aus diesen beiden Windgeschwindigkeiten und den korrespondierenden Höhen wird für jeden Datensatz ein individueller Höhenwindexponent

$$\gamma = \log_{\frac{115}{64,8}} \left(\frac{u_{115}}{u_{64,8}} \right) \quad (6.13)$$

berechnet. Mit diesem kann dann nach Gl. 6.12 in jeder Höhe eine Geschwindigkeit ermittelt werden. Anschließend wird ein Array definiert, welches ausreichend viele Höhen über die Rotorfläche beinhaltet:

$$\left[h_{\text{Nabe}} - \frac{d_{\text{R}}}{2}, \dots, h_{\text{Nabe}} + \frac{d_{\text{R}}}{2} \right].$$

Es wird eine Schrittweite von 1,6 m gewählt. Für jede Höhe wird dann eine Geschwindigkeit über das Potenzgesetz ermittelt und schließlich eine gemittelte Geschwindigkeit u'_∞ berechnet, mit der dann die Berechnung im digitalen Zwilling erfolgt. Die Abweichungen zu den realen Leistungsdaten sind auch hier nachfolgender Tabelle zu entnehmen.

	WEA 1	WEA 2	WEA 3	WEA 4	WEA 5
$\overline{\Delta}'_{\text{rel,ref}}$	14,00	41,14	23,51	21,85	34,90
$\overline{\Delta}'_{\text{rel,Gradient}}$	14,17	42,69	23,75	22,18	35,90

Tabelle 6.7: Gemittelte modifizierte relative Abweichungen $\overline{\Delta}'_{\text{rel,Gradient}}$ für Fall 1 unter Berücksichtigung des Höhenwindgradienten

Zwecks Vergleichbarkeit sind auch hier die Abweichungen für den gesamten Datensatz aus Fall 1 ohne Berücksichtigung des Höhenwindgradienten aufgeführt und als $\overline{\Delta}'_{\text{ref}}$ gekennzeichnet. Ein Vergleich der Abweichungen zeigt, dass auch unter Berücksichtigung des Höhenwindgradienten keine genaueren Ergebnisse erzielt werden können. Das könnte daran liegen, dass der Höhenwindgradient die Geschwindigkeit immer in eine Richtung korrigiert, bspw. immer zu einer etwas höheren Geschwindigkeit, mit der dann gerechnet wird. Da die berechneten Leistungen aber nie konstant unter oder konstant über den realen Leistungen liegen, sind die Abweichungen zwangsläufig mindestens genauso groß.

Es erfolgt daher keine Implementierung des Höhenwindgradienten und der in Abschnitt 6.5 vorgestellten Varianz der Dichte. Es sei aber ausdrücklich erwähnt, dass die zwei vorgestellten Methoden zum einen nur grobe Annäherungen sind und die Realität nur bedingt widerspiegeln und zum anderen noch viele weitere Ansätze in der Literatur existieren, dessen Umsetzung aber über den Rahmen dieser Arbeit hinausgehen würde.

7 Entwicklung eines künstlichen neuronalen Netzes

Nachdem die Berechnung der Leistung im Windpark durch den digitalen Zwilling als erfolgreich eingestuft wurde, soll nun mit Hilfe von diesem und einem künstlichen neuronalen Netz die Leistung maximiert werden. Die grundlegende Idee ist, den digitalen Zwilling als Kostenfunktion zu implementieren. Hierbei bilden die Umgebungsparameter $(\alpha, u_\infty, \sigma_{u,0})$ die Eingangsparameter des neuronalen Netzes und die Induktionsfaktoren der Anlagen $(a_1 \dots a_5)$ die Ausgangsparameter. Aus den Ausgängen des neuronalen Netzes sowie den korrespondierenden Umgebungsparametern wird dann mit Hilfe der Kostenfunktion bewertet, wie gut oder schlecht die Wahl der Induktionsfaktoren erfolgte.

7.1 Skalieren der Eingangsparameter

Gemäß der eingeführten Notation in Abschnitt 2.7 gilt

$$\mathbf{x} = [\alpha, u_\infty, \sigma_{u,0}]^T. \quad (7.1)$$

Dabei nimmt α Werte von 0° bis 359° an, während u_∞ sich größtenteils in einer Spanne von $6 \frac{\text{m}}{\text{s}} - 11 \frac{\text{m}}{\text{s}}$ bewegt (vgl. Abb. 6.9) und $\sigma_{u,0}$ noch kleinere Werte als u_∞ aufweist. Es ist üblich, die Eingangsparameter zu skalieren, damit die Größenordnungen identisch ist [40]. Die Skalierung erfolgt durch eine Funktion analog zum `StandardScaler` der open source Programmbibliothek `scikit-learn` [40]. Mathematisch erfolgt diese für bspw. die Windrichtung nach

$$\tilde{\alpha} = \frac{\alpha - \mu_\alpha}{\sigma_\alpha}, \quad (7.2)$$

wobei μ_α den Mittelwert aller Windrichtungen und σ_α die Standardabweichung aller Windrichtungen im Datensatz darstellt. Nach der Skalierung weist die Menge aller α einen Mittelwert von 0 und eine Varianz von 1 auf. Eine Rückskalierung ist dementsprechend ebenfalls möglich. Analog werden alle Werte für u_∞ und $\sigma_{u,0}$ skaliert. Programmtechnisch erfolgt die Skalierung über die Funktion `scale()`.

7.2 Verwendete Aktivierungsfunktion

Als Aktivierungsfunktion wird die tanh-Funktion verwendet (vgl. Gl. 2.82). Da die Induktionsfaktoren einen physikalisch sinnvollen Wertebereich von $[0; 0,5]$ besitzen (vgl. Abb. 2.1), wird die tanh-Funktion skaliert. Als obere Grenze wird daher $a = 0,5$ gewählt, als untere Grenze $a = 0,1$. Werte $a < 0,1$ werden als nicht sinnvoll eingestuft, da die Anlage damit nahezu keinen Wind abfangen würde und keine Leistung produziert.

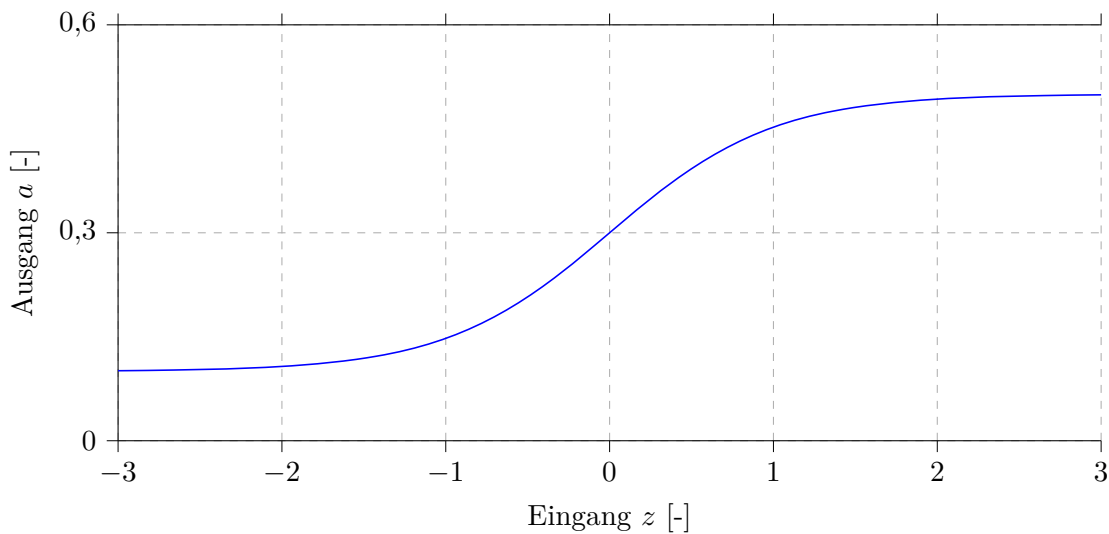


Abbildung 7.1: Auf den Anwendungsfall skalierte tanh-Funktion als Aktivierungsfunktion

Die abgebildete Funktion lässt sich mathematisch dann als

$$a = \tanh(z) \cdot \frac{0,5 - 0,1}{2} + \frac{0,5 - 0,1}{2} + 0,1 \quad (7.3)$$

schreiben.

7.3 Digitaler Zwilling als Kostenfunktion

Der entwickelte digitale Zwilling soll im neuronalen Netz als Kostenfunktion implementiert werden. Dazu bedarf es allerdings einiger Änderungen.

Anders als bei der vorgestellten Kostenfunktion (vgl. Gl. 2.83) gibt es hier allerdings keinen gewünschten Ausgang, der mit dem Ausgang des Netzes verglichen werden kann. Aus diesem Grund wird die berechnete Leistung des Windparks direkt als „Cost“ definiert. Da mit Hilfe des Gradientenabstiegsverfahrens (vgl. Abschnitt 2.7.2) die Kostenfunktion immer minimiert werden soll, wird nicht die summierte Leistung aller Anlagen ausgegeben, sondern die Differenz zur maximal möglichen Leistung. Diese beträgt

$$P_{\text{opt}} = 4 \cdot 2400 \text{ kW} + 3000 \text{ kW} = 12600 \text{ kW}. \quad (7.4)$$

Die Ausgabe aus der Kostenfunktion lässt sich also als

$$C = P_{\text{opt}} - P_{\text{sum}} \quad \text{mit} \quad P_{\text{sum}} = \sum_{j=1}^5 (P_j) \quad (7.5)$$

schreiben, wobei P_j die Leistung der Anlage j ist.

Ein weiteres Problem ist die Berechnung der Gradienten in der Kostenfunktion. Nach Gl. 2.93 werden die partiellen Ableitungen der Kostenfunktion nach den Ausgängen des neuronalen Netzes,

in dem Fall also den fünf Induktionsfaktoren, benötigt. Wird bspw. betrachtet, wie das Geschwindigkeitsdefizit Δu_{ij} vom Induktionsfaktor a abhängt (vgl. Gl. 4.14), kann eine Verkettung der Funktionen festgestellt werden. Außerdem werden im digitalen Zwilling über `for`-Schleifen Arrays gefüllt, auf die später wieder zugegriffen wird. Dies erschwert das händische Bilden der partiellen Ableitungen enorm und erhöht die Fehleranfälligkeit. Abhilfe schafft hier die open source Programmbibliothek `jax` [29]. `jax` ermöglicht das Differenzieren von Python-Funktionen, insbesondere solcher, die Schleifen, Abfragen oder Rekursion beinhalten.

Bevor mit Hilfe von `jax` der Gradient der Kostenfunktion berechnet werden kann, bedarf es allerdings noch weitere Änderungen. Das hat zwei Gründe: Zum einen beträgt die Berechnungszeit des Gradienten $\nabla_a C$, wobei C der komplexe digitale Zwilling ist, für einen Datensatz ca. 170s. In Anbetracht, dass später mehrere Tausend Datensätze durchgerechnet werden sollen, ist dies nicht akzeptabel. Zum anderen sind manche Funktionen, die in der Standard Mathe-Bibliothek `numpy` verwendet werden, nicht direkt auf `jax` übertragbar. Diese Änderungen sind im Folgenden aufgeführt.

- Mit Hilfe des in der verwendeten Python-IDE integrierten Profilers konnte herausgefunden werden, dass die Schleife über das Array `dp`, also über alle Datenpunkte, die meiste Zeit in Anspruch nimmt. Aus diesem Grund wird diese Schleife vektorisiert. Das ist möglich, da die Berechnung für jeden Datenpunkt exakt gleich ist und nicht vom jeweils vorherigen Datenpunkt abhängt. Vektorisierung bedeutet konkret, dass jeder Durchlauf der ursprünglichen Schleife quasi ein Vektoreintrag ist. Durch die auf Vektoren ausgelegte Bibliothek `jax` kann so die Rechenzeit enorm verkürzt werden.

Es sei erwähnt, dass die i - und j -Schleife nicht parallelisiert werden können, da hier der nächste Durchlauf vom vorherigen abhängt.

- Da aus vielen skalaren Werten Vektoren werden, können keine `if`-Abfragen mehr verwendet werden. Diese werden durch die `jax`-interne Funktion `jax.numpy.where()` und `Booleans` ersetzt. Hierdurch wird das Ergebnis nicht verändert.
- `jax` ermöglicht das Vorkompilieren von Funktionen beim ersten Aufruf, um diese für weitere Aufrufe gespeichert zu haben. Dies erfolgt durch sog. „jitten“ (`jit` = „just in time“) und wird programmtechnisch durch `@jit` vor einer Funktion gekennzeichnet. Da nur deterministische Abfolgen von Berechnungen gejittet werden können, wird ein Teil der Schleife über alle Anlagern p (Korrekturtermberechnung) ausgelagert und in eine eigene Funktion `p_loop()` geschrieben, die dann genau wie deterministische Teile der bereits bekannten Funktion `calculate_delta_sigma_u()` vorkompiliert werden kann.

Nach Implementierung aller genannten Änderungen konnte die Berechnungszeit eines Datensatzes von ca. 170s auf ca. 0,3s reduziert werden.

Damit das Netzwerk lernen kann, bedarf es zwei weitere Modifizierungen des digitalen Zwillings. Die Induktionsfaktoren werden später vom neuronalen Netz an den digitalen Zwilling übergeben. Dabei sollte aber der erste Induktionsfaktor der stromaufwärts ersten Anlage zugeordnet werden und nicht der WEA 1. Da sich abhängig der Windrichtung die Reihenfolge der Anlagen ändert,

werden nach Transformation des Windparks die Induktionsfaktoren der Reihe nach neu zugeordnet. Außerdem darf der digitale Zwilling nicht das Array \mathbf{P} ausgeben, sondern $P_{\text{opt}} - P_{\text{sum}}$. Der modifizierte digitale Zwilling ist unter dem Namen `digital_twin_opt.py` auf dem Datenträger zu finden. Dank der `jax`-internen Funktion `value_and_grad()` kann mit einem Aufruf nun in kürzester Zeit sowohl die Leistungsdifferenz, also die Cost, auch als der Gradient $\nabla_a C$ ausgegeben werden, der für die Fehlerrückführung benötigt wird.

7.4 Implementierung in Python

Die Implementierung des neuronalen Netzes erfolgt auf Grundlage von Nielsen [47, S. 27 ff.], in dem der Autor den Code für ein grundlegendes ANN in Python bereitstellt. Zwecks Übersichtlichkeit ist der gesamte Code ebenfalls als `ann.py` auf dem angehängten Datenträger zu finden. Da der Code zu größten Teilen bereits existiert und nur benutzt wird, soll dieser hier nicht detailliert erläutert werden. Dennoch wird im Folgenden die grobe Struktur und Wirkungsweise dargestellt.

Das neuronale Netz wird in Form einer Klasse namens `Network` implementiert. In dessen Konstruktor erfolgt die Initialisierung der Gewichtungen und Schwellenwerte des gesamten Netzwerks. Diese werden über die Funktion `numpy.random.randn()` erzeugt und haben damit einen zufälligen Wert sowie einen Mittelwert von 0 und eine Varianz von 1. Die Werte werden in Matrizen bzw. Vektoren gespeichert. Es folgen klasseninterne Funktionen.

Mit der Funktion `scale()` wird die in Abschnitt 7.1 bereits angesprochene Skalierung der Eingangsparameter durchgeführt. Die Funktion `custom_activation()` beinhaltet die in Gl. 7.3 hergeleitete Aktivierungsfunktion und `custom_activation_prime()` dessen Ableitung

$$\frac{da}{dz} = \frac{1}{\cosh(z)^2} \cdot \frac{0,5 - 0,1}{2}. \quad (7.6)$$

In der Hauptfunktion `SGD.py` (stochastic gradient descent) erfolgt das eigentliche Training des Netzes. Dieses wird über sog. Epochen durchgeführt, wobei eine Epoche das Durchrechnen aller Datensätze kennzeichnet und sich der Vorgang dann wiederholt.

In einer Schleife wird daher über alle Epochen iteriert. Die vorhandenen Trainingsdaten werden gemischt und entsprechend einer vorgegebenen Batch-Größe in die zufälligen Mengen \mathbf{X}_i (vgl. Gl. 2.87) eingeteilt. Für jede zufällig generierte Menge wird dann die Funktion `update_mini_batch()` aufgerufen, die die Gewichtungen und Schwellenwerte aktualisiert.

In der Funktion wird durch jeden Datensatz $\mathbf{X}_1, \dots, \mathbf{X}_m$ iteriert. Mit Hilfe der noch folgenden Funktion `backprop()` werden die Gradienten $\frac{\partial C_{\mathbf{X}_i}}{\partial \mathbf{w}}$ und $\frac{\partial C_{\mathbf{X}_i}}{\partial \mathbf{b}}$ bestimmt und schließlich über Gl. 2.89 und Gl. 2.90 die Gewichtungen und Schwellenwerte des Netzwerkes modifiziert.

In der eben erwähnten Funktion `backprop()` findet die Berechnung des Ausgangs des Netzes sowie die Fehlerrückführung statt. Der Trainingseingang \mathbf{X}_i wird vorwärts durch das Netzwerk gespeist. Dies geschieht durch eine Schleife über alle Gewichtungen und Schwellenwerte des Netzwerkes

sowie der bereits vorgestellten Aktivierungsfunktion. Danach folgt die Fehlerrückführung exakt nach Abschnitt 2.7.3. Wie bereits beschrieben, liefert die Funktion `value_and_grad()` sowohl die Cost, also die Leistungsdifferenz, als auch die partiellen Ableitungen der Kostenfunktion nach den Induktionsfaktoren. Beachtet werden sollte allerdings, dass die Eingangsparameter zwar für das Durchlaufen des Netzes mit `scale()` skaliert werden, jedoch unskaliert in die Kostenfunktion gegeben werden (müssen), da die Werte physikalisch sinnvoll sind. Der ausgegebene Gradient wird ebenfalls skaliert, in dem das Ergebnis durch P_{opt} dividiert wird. Somit nimmt der Gradient nur Werte zwischen 0 und 1 an. Würde die Skalierung nicht stattfinden, wären die Werte des Gradienten sehr viel größer als die Gewichtungen und Schwellenwerte, was sich negativ auf den Lernprozess auswirken kann.

Nach Einlesen der Daten in Form einer Excel-Tabelle werden diese mit Hilfe der Funktion `train_test_split()` der Programmbibliothek `sklearn` [39] in einen Trainingsdatensatz und einen Testdatensatz geteilt. Dies geschieht im Verhältnis 80%/20%. Nach Berechnung der Mittelwerte μ und Standardabweichungen σ aller Eingangsparameter (diese werden später für die Skalierung benötigt) wird ein Objekt, das neuronale Netz, erzeugt:

```
net = Network([3, 16, 5]).
```

Das Array `[3, 16, 5]` gibt hierbei die Anzahl der Neuronen auf den Schichten an. Die Eingabeschicht enthält drei Neuronen ($\alpha, u_{\infty}, \sigma_{u,0}$), die versteckte Schicht enthält hier bspw. 16 Neuronen und die Ausgabeschicht fünf Neuronen (a_1, a_2, a_3, a_4, a_5). Der Aufruf der Hauptfunktion `SGD` erfolgt schließlich in der Form

```
net.SGD(Daten, Epochen, Batch-Größe, Lernrate)
```

und das neuronale Netz beginnt zu lernen.

Der Code wird um eine Funktion `evaluate()` erweitert. In dieser wird nach jeder Epoche der Testdatensatz durch das Netz gespeist und ebenfalls die Cost und der Gradient der Kostenfunktion berechnet. Ebenfalls gespeichert wird die durchschnittliche Cost des Testdatensatzes nach jeder Epoche. Diese sollte über die Epochen hinweg genau wie beim Trainingsdatensatz fallen. Die Gewichtungen \mathbf{w} und Schwellenwerte \mathbf{b} werden abgespeichert, um später zur Validierung ein Netz mit derselben Struktur und den gespeicherten Werten zu entwerfen und neue, dem Netz unbekannte, Datensätze vorwärts durch das Netz zu speisen.

8 Validierung des neuronalen Netzes

In diesem Kapitel folgen die Ergebnisse des Lernprozesses sowie der Vergleich der Windparkleistung mit optimalen vom neuronalen Netz berechneten Induktionsfaktoren und dem Fall, dass alle Anlagen bei $a = \frac{1}{3}$ betrieben werden. Das Skript der Validierung ist unter dem Namen `ann_valid.py` zu finden.

Validiert wird das neuronale Netz an einem noch unbekanntem Datensatz aus dem Jahr 2020. Analog zu den Daten aus 2019 sind die 17794 Datensätze aus 2020 in `2020_mod.xlsx` zu finden. Es macht Sinn, die Validierung für einen Fall durchzuführen, in dem Anlagen von anderen abgeschattet werden. Aus diesem Grund werden die Filtereinstellungen aus Fall 1 (vgl. Abschnitt 6.1.1) auch im Jahr 2020 angewandt. Dies führt zu 452 Datensätzen (Datei: `2020_fall1.xlsx`).

Werden die Gewichtungen und Schwellenwerte des Netzes zufällig initialisiert (mit Mittelwert = 0 und Varianz = 1) und das Netz nicht trainiert, kann der unbekanntem Validierungsdatsatz vorwärts durch das untrainierte Netzwerk gespeist werden. Später trainierte Netze sollten dann auf jeden Fall bessere Ergebnisse als das untrainierte Netz aufweisen. Die nachfolgende Abbildung zeigt die Gesamtleistung des Windparks, einmal erzeugt durch die Induktionsfaktoren, die das untrainierte Netz ausgibt (blau, gekennzeichnet durch P_{opt}) und einmal erzeugt durch Induktionsfaktoren, die alle $a = \frac{1}{3}$ betragen (orange, gekennzeichnet durch $P_{1/3}$).

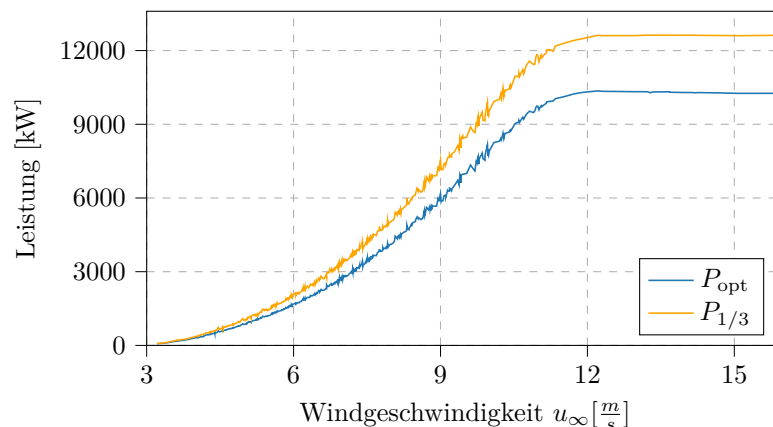


Abbildung 8.1: Vergleich der Gesamtleistung des Windparks mit vom Netz erzeugten Induktionsfaktoren sowie $\mathbf{a} = [\frac{1}{3}, \frac{1}{3}, \frac{1}{3}, \frac{1}{3}]$ mit zufälligen Netzparametern

Geplottet werden die Leistungen über die Umgebungswindgeschwindigkeit, weil so visuell die Abweichungen am besten sichtbar sind, da die Werte nicht stark fluktuieren.

Wie zu erwarten, ist die Leistung P_{opt} weitaus geringer als die Leistung $P_{1/3}$, da das neuronale Netz nicht trainiert ist und quasi zufällige Induktionsfaktoren $0, 1 \leq a \leq 0, 5$ generiert. Die relative gemittelte Abweichung der optimalen vom Netz berechneten Lösung (in dem Fall natürlich nicht optimal, sondern zufällig) im Vergleich zu der Lösung, bei der alle Induktionsfaktoren $a = \frac{1}{3}$

8 Validierung des neuronalen Netzes

betragen, berechnet sich nach

$$\bar{\Delta}_{\text{rel,ann}} = \frac{P_{\text{opt}} - P_{1/3}}{P_{1/3}} \quad (8.1)$$

und beträgt $\bar{\Delta}_{\text{rel,ann}} = -14,64\%$. Auch hier ist die gemittelte relative Abweichung über die 452 Datensätze gleichzusetzen mit der Abweichung der Erträge für diesen Zeitraum (vgl. Abschnitt 6.2.1).

Als erster Ansatz wird die Anzahl der versteckten Schichten auf 1 und die Anzahl der darin befindlichen Neuronen auf 16 gesetzt. Aus der Excel-Liste des Jahres 2019 werden vorerst zufällige 2000 Datensätze ausgewählt (Datei: `random2000.xlsx`). Das bedeutet, dass pro Epoche $2000 \cdot 0,8 = 1600$ Datensätze zum Lernen und 400 Datensätze zum Testen zur Verfügung stehen. Trainiert wird das Netz über 30 Epochen, einer Batch-Größe von 10 Datensätzen sowie einer Lernrate von $\eta = 0,05$. Die nachfolgende Abbildung zeigt die über jede Epoche gemittelte Trainingsset-Cost sowie Testset-Cost für alle 30 Epochen.

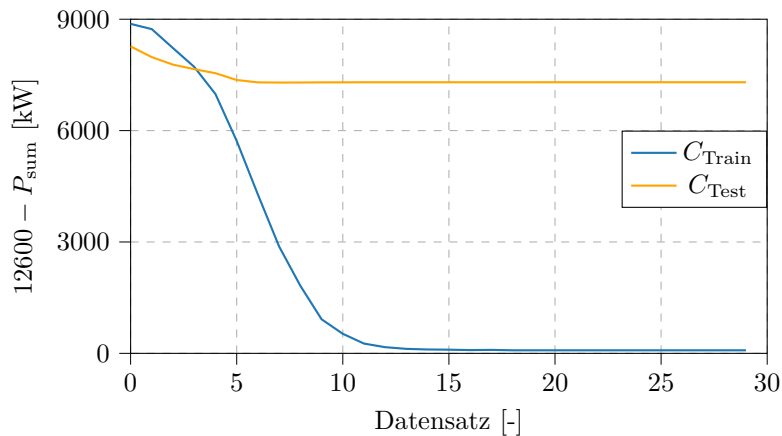


Abbildung 8.2: Trainingsset- und Testset-Cost (`random2000.xlsx`, $\eta = 0,05$, 1 versteckte Schicht mit 16 Neuronen, Batchgröße = 10, 30 Epochen)

Die Cost für die Trainingsdaten fällt durchgängig und konvergiert bereits nach ca. zwölf Epochen gegen 0 kW. Die Cost des Testsets fällt zwar auch, konvergiert allerdings bereits nach ca. 5 Epochen gegen ca. 7500 kW. Dies kann allerdings daran liegen, dass für dieses spezifische Testset, sprich für die Umgebungsparameter, kein besseres Ergebnis möglich ist. Die folgende Abbildung zeigt die Leistung, erzeugt durch die von diesem Netz ausgegebenen Induktionsfaktoren im Vergleich zu $\mathbf{a} = \left[\frac{1}{3}, \frac{1}{3}, \frac{1}{3}, \frac{1}{3}, \frac{1}{3} \right]$.

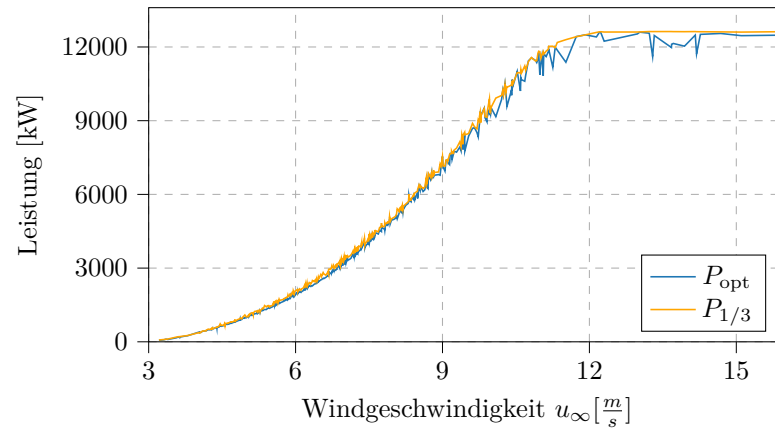


Abbildung 8.3: Vergleich der Gesamtleistung des Windparks mit vom Netz erzeugten Induktionsfaktoren sowie $\mathbf{a} = [\frac{1}{3}, \frac{1}{3}, \frac{1}{3}, \frac{1}{3}, \frac{1}{3}]$ (random2000.xlsx, $\eta = 0,05$, 1 versteckte Schicht mit 16 Neuronen, Batchgröße = 10, 30 Epochen)

Ein Vergleich des Ergebnisses mit dem des untrainierten Netzes (vgl. Abb. 8.1) zeigt, dass das Netz zwar durchaus gelernt hat, jedoch nicht optimal trainiert wurde, da immernoch gilt $P_{\text{opt}} < P_{1/3}$. Die gemittelte relative Abweichung beträgt $\bar{\Delta}_{\text{rel,ann}} = -3,72\%$. Dass eine Erhöhung der Leistung überhaupt möglich ist, zeigt der Maximalwert der Abweichungen. Dieser liegt bei $5,36\%$, d.h. die Leistung konnte für die Umgebungsparameter dieser spezifischen 10 min um $5,36\%$ durch vom Netz berechnete optimale Induktionsfaktoren gesteigert werden.

Da die Cost schon nach ca. 15 Epochen konvergiert, wird im Folgenden nur 20 statt 30 Epochen lang trainiert. Die Anzahl der Neuronen auf der versteckten Schicht wird auf 32 erhöht. Die nachfolgende Abbildungen zeigen analog zum vorherigen Fall die Trainingsset- sowie Testset-Cost über die 20 Epochen und den Vergleich der Gesamtleistungen am unbekanntem Datensatz aus dem Jahr 2020.

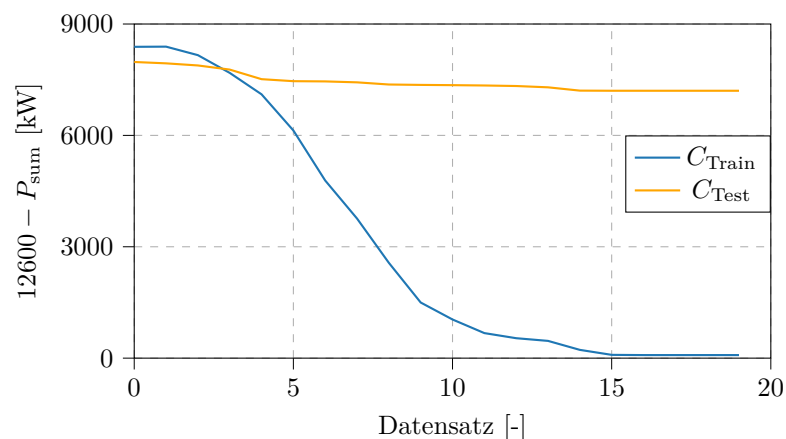


Abbildung 8.4: Trainingsset- und Testset-Cost (random2000.xlsx, $\eta = 0,05$, 1 versteckte Schicht mit 32 Neuronen, Batchgröße = 10, 20 Epochen)

8 Validierung des neuronalen Netzes

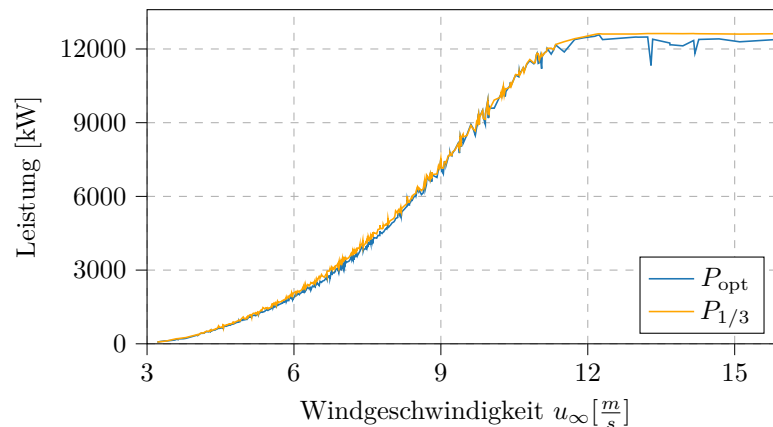


Abbildung 8.5: Vergleich der Gesamtleistung des Windparks mit vom Netz erzeugten Induktionsfaktoren sowie $\mathbf{a} = [\frac{1}{3}, \frac{1}{3}, \frac{1}{3}, \frac{1}{3}]$ (`random2000.xlsx`, $\eta = 0,05$, 1 versteckte Schicht mit 32 Neuronen, Batchgröße = 10, 20 Epochen)

Die gemittelte relative Abweichung beträgt hier $\bar{\Delta}_{rel,ann} = -3,95\%$. Die maximale Abweichung beträgt hier 11,78%. Im Bereich und überhalb der Nennwindgeschwindigkeit ($10 \frac{m}{s} \leq u_{\infty} \leq 15 \frac{m}{s}$) liefert dieses Netz bessere Ergebnisse als das mit 16 Neuronen. Unterhalb dieser Geschwindigkeit ($5 \frac{m}{s} \leq u_{\infty} \leq 9 \frac{m}{s}$) ist dies allerdings nicht der Fall.

Probeweise wird die Anzahl der Neuronen in der versteckten Schicht erneut verdoppelt (64). Die Verläufe der Cost sowie der Gesamtleistung sehen qualitativ ähnlich aus. Die Abweichung liegt jedoch bei $\bar{\Delta}_{rel,ann} = -8,51\%$.

Ein bekanntes Problem im Bereich der neuronalen Netze ist das sog. *overfitting*. Dieses beschreibt eine Überanpassung der Netz-Parameter auf den Trainings-Datensatz, was dazu führt, dass die Trainingsset-Cost weiter fällt, während die Testset-Cost stagniert. Das Netz lernt somit die Trainingsdaten auswendig, kann aber schlecht generalisieren, sprich auf unbekannte Eingänge keine optimalen Ausgänge liefern. [47, S. 73 ff.]

Da genau das in den bisherigen Verläufen zu erkennen ist und vorerst auch nur mit 2000 Datensätzen trainiert wurde, muss die Anzahl der Trainingsdatensätze erhöht werden. Nach dem selben Prinzip wie bisher werden zufällige 8000 Datensätze aus dem Jahr 2019 extrahiert (Datei: `random8000.xlsx`). Es sind also 6400 Datensätze zum Lernen und 1600 Datensätze zum Testen verfügbar. Die Anzahl der Neuronen der versteckten Schicht wird auf 32 gesetzt und trainiert wird über 15 Epochen, da auch trotz der Verbesserung des Codes durch `jax` das Durchrechnen von 8000 Datensätzen über 15 Epochen enorm lange dauert. Alle anderen Parameter werden nicht verändert. Die Ergebnisse sind im Folgenden dargestellt.

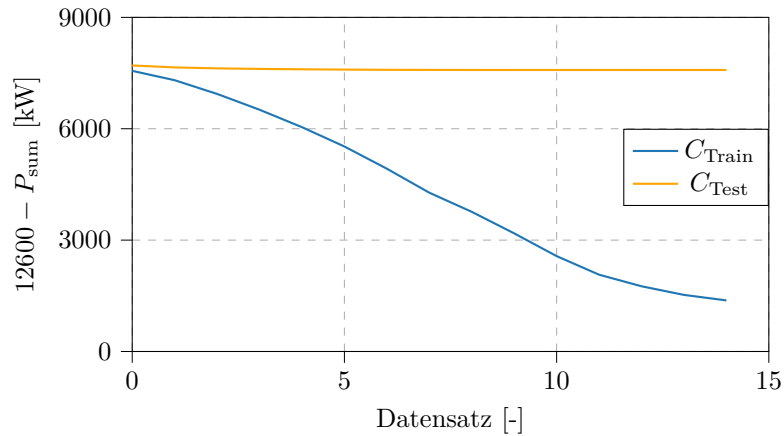


Abbildung 8.6: Trainingsset- und Testset-Cost (random8000.xlsx, $\eta = 0,05$, 1 versteckte Schicht mit 32 Neuronen, Batchgröße = 10, 15 Epochen)

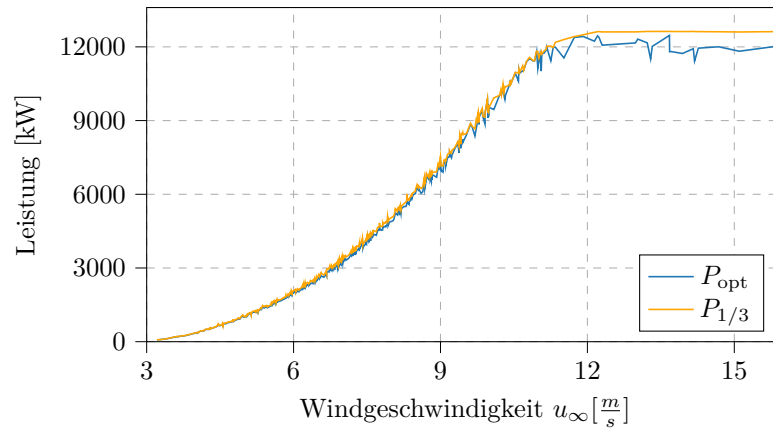


Abbildung 8.7: Vergleich der Gesamtleistung des Windparks mit vom Netz erzeugten Induktionsfaktoren sowie $\mathbf{a} = [\frac{1}{3}, \frac{1}{3}, \frac{1}{3}, \frac{1}{3}, \frac{1}{3}]$ (random8000.xlsx, $\eta = 0,05$, 1 versteckte Schicht mit 32 Neuronen, Batchgröße = 10, 15 Epochen)

Wie in Abb. 8.6 zu sehen ist, fällt die Testset-Cost hier nur in den ersten Epochen minimal und konvergiert schnell. Dies kann wie zuvor auch daran liegen, dass im Mittel für den Test-Datensatz kein besseres Ergebnis möglich ist. Die Trainingsset-Cost fällt auf ca. 1500 kW, konvergiert aber noch nicht vollständig. Vermutlich wären mehr Epochen notwendig, um bessere Aussagen treffen zu können. Die gemittelte Abweichung der Gesamtleistung beträgt allerdings $\bar{\Delta}_{\text{rel,ann}} = -2,81\%$, den bisher höchsten Wert. Auch mit diesem Netz ist die maximale Abweichung mit 4,63% positiv.

Die Ergebnisse zeigen, dass die Entwicklung und das Training der neuronalen Netze zwar erfolgreich war, allerdings noch nicht optimal ist. Die Implementierung des vorher entwickelten Zwillings als Kostenfunktion funktioniert. Die Netze liefern bessere Ergebnisse als ein untrainiertes Netz, welches zufällige Induktionsfaktoren $0,1 \leq a \leq 0,5$ ausgibt. Die gemittelten relativen Abweichungen der Leistung zu jener, die erzeugt wird, wenn alle Anlagen bei $a = \frac{1}{3}$ betrieben werden, konnte im Vergleich zum untrainierten Netz, also bei zufälligen Induktionsfaktoren

8 Validierung des neuronalen Netzes

$0,1 \leq a \leq 0,5$, um 11,83% erhöht werden. Die Abweichungen liegen im Mittel aber immer noch im negativen Bereich, d.h. es wird im Schnitt für den unbekanntem Datensatz mit den vom Netz berechneten Induktionsfaktoren nicht mehr Leistung erzeugt als für den Fall, dass alle Anlagen bei $a = \frac{1}{3}$ betrieben werden. Dass das Netz überhaupt im Stande ist, Induktionsfaktoren zu berechnen, die in einer höheren Leistung resultieren, zeigen die Maximalwerte der jeweiligen reaktiven Abweichungen. In Anbetracht der Zeit, die das neuronale Netz zum Lernen benötigt (bei 8000 Datensätzen und 15 Epochen ca. 30 h Rechenzeit) sowie der Tatsache, dass alle Parameter des neuronalen Netzes miteinander wechselwirken, würde eine systematische Optimierung der Parameter über den Rahmen dieser Arbeit hinausgehen.

9 Zusammenfassung

Ziel dieser Arbeit war es, einen digitalen Zwilling des Windparks, mit besonderem Fokus auf der Modellierung von gegenseitiger Abschattung der Anlagen, zu entwickeln. Mit Hilfe von diesem sowie Methoden des maschinellen Lernens, speziell künstlichen neuronalen Netzen, sollte geprüft werden, inwiefern eine Leistungsmaximierung des Parks möglich ist. Grundlage waren reale Daten aus dem Windpark Curslack aus den Jahren 2019 und 2020.

Die Arbeit begann mit kurzer allgemeiner Einleitung in das Thema in Kapitel 1. Speziell in Abschnitt 1.3 erfolgte die Vorstellung des Windparks Curslack. In Kapitel 2 wurden alle für die Arbeit notwendigen theoretischen Grundlagen hergeleitet bzw. erläutert. Dazu zählen allgemeine physikalische Grundlagen zur Windenergie (Abschnitt 2.1) und speziell die analytische Modellierung von einfachen (Abschnitt 2.4) und mehrfachen (Abschnitt 2.5) Nachlaufströmungen, Grundlagen zur Koordinatentransformation (Abschnitt 2.6) und Grundlagen zu künstlichen neuronalen Netzen (Abschnitt 2.7).

Die Analyse von relevanten Publikationen (Kapitel 3) zeigte, dass sowohl über Variation der Induktionsfaktoren als auch über das Ablenken der Wakes durch Drehen der Anlagen um die Turmachse die Gesamtleistung in einem Windpark erhöht werden kann. Je nach Methodik der Optimierung schwanken die Ergebnisse der Maximierung stark. Der Einsatz von maschinellem Lernen in diesem Themengebiet findet durchaus statt, speziell der Einsatz von neuronalen Netzen zu Optimierungszwecken konnte aber als unkonventionell eingestuft werden.

Die Entwicklung des digitalen Zwillings (Kapitel 4) erfolgte unter einigen Annahmen. Dazu zählen unter anderem eine konstante Dichte der Luft, die Gleichheit aller Sensoren am Messmast bezogen auf deren Ausrichtung und die Tatsache, dass die Umgebungswindgeschwindigkeit vom Messmast entnommen und für jede erste Anlage in Windrichtung, unabhängig von der Entfernung der Anlage zum Mast, gilt. Das Qian-Ishihara-Modell aus Abschnitt 2.5.3 wurde in `Python` implementiert.

Um den digitalen Zwilling zu validieren, bedurfte es reale Windparkdaten, die in Kapitel 5 gefiltert wurden. Daten, die in einem Betriebszustand, der nicht dem Produktionsbetrieb gleicht, entstanden sind, wurden aussortiert. Außerdem wurden nur Tagesdaten (von Sonnenaufgang bis Sonnenuntergang) verwendet und auch Windrichtungen, in denen der Windpark den Messmast abschattet, wurden verworfen.

Die Validierung (Kapitel 6) zeigte, dass die berechneten Leistungen und die realen Leistungen in akzeptabler Übereinstimmung sind. Ein Vergleich der berechneten und realen Leistungskurven für zwei Fälle, in denen viele Anlagen des Parks im Nachlauf von anderen stehen (Abschnitt 6.1) zeigte, dass die errechneten Leistungen für eine gewisse Windgeschwindigkeit näherungsweise den Mittelwert des für diese Windgeschwindigkeit vorliegenden Spektrums an real gemessenen Leistungen abbilden. Durch einen direkten Vergleich der Leistungen für ausgewählte Datensätze konnte diese qualitative Übereinstimmung bestätigt werden. Bezüglich Quantität wurden die relativen Abweichungen für die zwei definierten Fälle sowie für den gesamten Datensatz aus 2019 betrachtet. Sehr kleine reale Leistungen wurden hier, aufgrund des geringen Einflusses auf z.B. einen Tagesertrag, nicht mit einbezogen. Für Fall 1 wurden je nach Anlage gemittelte relative

Abweichungen zwischen 14,00 % und 41,14 % festgestellt. In Fall 2 schwanken diese zwischen 10,65 % und 25,11 %.

Bezogen auf den gesamten Datensatz aus 2019 beträgt die größte relative Abweichung 35,71 % und ist an Anlage 5 bei einer Windrichtung von $157^\circ - 197^\circ$ vorzufinden. Das liegt zum einen daran, dass Anlage 5 hier im Nachlauf von teils Anlage 2 und teils Anlage 3 steht und zum anderen, dass Anlage 2 und 3 weit vom Messmast entfernt sind und für die Umgebungswindgeschwindigkeit vor den beiden Anlagen die vom Messmast benutzt wurde. Dazu kommen allgemeine Schwankungen der 10 min-Mittelwerte der realen Leistungen. Die gemittelte relative Abweichung über den gesamten Datensatz aus dem Jahr 2019 beträgt 24,05 %. Die Betrachtung der Gesamtleistung des Windparks statt den Einzelleistungen, die einer Schwankung unterliegen, zeigte eine noch geringere Abweichung. Für den gesamten Datensatz aus dem Jahr 2019 lag die gemittelte relative Abweichung bei 17,27 %.

Mit den genannten Ergebnissen und unter Berücksichtigung der Gründe für die Abweichungen wurde die Validierung des digitalen Zwilling als erfolgreich eingestuft.

Außerdem wurde der Einfluss der Varianz der Dichte (Abschnitt 6.5) sowie der Einfluss des Höhenwindgradienten (Abschnitt 6.6) auf die Umgebungswindgeschwindigkeit untersucht. Beide Einflüsse wurden als nicht signifikant eingestuft und daher nicht implementiert.

In Kapitel 7 folgte schließlich die Entwicklung eines künstlichen neuronalen Netzes. Dieses beinhaltet den digitalen Zwilling als Kostenfunktion. Durch Optimierung und Umschreibung des digitalen Zwilling konnte die Zeit, die benötigt wird, um die Kostenfunktion und dessen Gradienten zu berechnen, von ca. 170 s auf 0,3 s erfolgreich reduziert werden.

Die Ergebnisse des Trainings des neuronalen Netzes wurden in Kapitel 8 präsentiert. Es konnte gezeigt werden, dass ein trainiertes neuronales Netz bessere Ergebnisse liefert, als ein nicht trainiertes Netz mit zufällig generierten Parametern. Mit den Induktionsfaktoren des untrainierten Netzes wurde im Mittel 14,64 % weniger Leistung erzeugt, als wenn alle Anlagen bei $a = \frac{1}{3}$ betrieben werden. Nach Training des Netzes waren es im besten Fall 2,81 %. Dieses Netz wurde mit 6400 Datensätzen über 15 Epochen trainiert, besitzt eine mit 32 Neuronen besetzte versteckte Schicht, eine Lernrate $\eta = 0,05$ sowie eine Batch-Größe von 10.

Eine Erhöhung der Leistung im Vergleich zum Fall, wenn alle Anlagen bei $a = \frac{1}{3}$ betrieben werden, konnte vereinzelt für eher kleine Windgeschwindigkeiten (maximal 4,61 % mit dem genannten Netz), aber noch nicht im Mittel erzielt werden. Dies bedarf weiterer Optimierung der Netz-Parameter. Nichtsdestotrotz kann festgehalten werden, dass eine Leistungsmaximierung durch Training eines neuronalen Netzes, welches einen digitalen Zwilling des Windparks als Kostenfunktion enthält, mit realen Windparkdaten möglich ist. Die Grundlage dazu, der digitale Zwilling sowie das neuronale Netz, wurde erfolgreich entwickelt und validiert.

10 Kritische Reflexion und Ausblick

Die Ergebnisse der vorliegenden Arbeit sind selbstverständlich kritisch zu betrachten. Zudem können aufbauend auf dieser Arbeit weitere wissenschaftliche Arbeiten verfasst werden, da an einigen Stellen weitere Untersuchungen oder Verbesserungen möglich sind.

Ein erster zu betrachtender Aspekt sind die realen Daten aus dem Windpark. Wie bereits in Kapitel 6 festgehalten wurde, sind die Umgebungsparameter, die dem Messmast entnommen wurden, für gewisse Windrichtungen nur grob gültig für die Parameter vor der stromaufwärts ersten Anlage. Je weiter diese vom Messmast entfernt ist, desto gröber ist die Annahme der Parameter vom Messmast.

Des Weiteren wurde die Windgeschwindigkeit formalerweise von Sensor V3 ($z = 115$ m) entnommen (da dort auch die Windrichtung von Sensor D3 benutzt wurde), obwohl diese auch bei Sensor V1 ($z = 120,5$ m) abgreifbar ist. Streng genommen ist es also genauer, die Windrichtung von Sensor D3 und Windgeschwindigkeit von Sensor V1 zu benutzen, auch wenn diese dann nicht auf der selben Höhe zu finden sind. Eine grobe Implementierung des Höhenwindgradienten erfolgte zwar in Abschnitt 6.6, optimal wäre allerdings, die reale Windgeschwindigkeit in verschiedenen Höhen über den Rotor verteilt messen zu können, um so keine Näherungs-Gleichung verwenden zu müssen. Parallel zum Höhenwindgradienten könnte auch die Varianz der Dichte komplexer implementiert werden. Neben den realen Leistungskurven, die bereits im digitalen Zwilling implementiert sind, sind auch noch Leistungskurven für verschiedene Dichten vorhanden. In einer zukünftigen Arbeit könnten diese ebenfalls implementiert werden. Erwähnt werden muss jedoch, dass auch die Dichte aufgrund der Lage der Sensoren am Messmast nur an bestimmten Punkten berechnet werden kann, was zu Ungenauigkeiten führt.

Außerdem nicht implementiert ist aktuell der Nachtbetrieb. Eine optimale Regelung sollte später auch nachts stattfinden, um die Leistung zu maximieren. Der digitale Zwilling könnte in Zukunft um diese reduzierten Leistungskurven erweitert werden.

Alle angesprochenen möglichen Verbesserungen des digitalen Zwillings resultieren vermutlich in einer noch höheren Komplexität, welche vor allem für die Berechnung der Gradienten im Lernprozess des neuronalen Netzes zu einer größeren Rechenzeit führt. Eine zukünftige Arbeit könnte sich daher damit auseinandersetzen, den digitalen Zwilling im Hinblick auf Rechenzeit noch stärker zu optimieren. Dazu könnte auch zählen, ein vielleicht simpleres Wake-Modell zu testen, welches vielleicht ebenfalls hinreichend genaue Ergebnisse liefert.

Wie bereits in Kapitel 8 erwähnt, ist die Optimierung des neuronalen Netzes ein Thema für eine weiterführende Arbeit. Neben der Anzahl der Trainings-Datensätze sowie der Anzahl der Neuronen in der versteckten Schicht könnte auch die Anzahl der versteckten Schichten selbst variiert werden. Außerdem wurde die Lernrate $\eta = 0,05$ sowie die Batch-Größe von 10 nie verändert. Statt der tanh-Funktion könnte eine andere Aktivierungsfunktion verwendet und auch das Verhältnis von Testdatensätzen zu Trainingsdatensätzen anders gewählt werden.

Ein weiterer in dieser Arbeit nicht angesprochener Aspekt ist die später verwendete Regelung und das damit einhergehende Zeitintervall. Werden wie in dieser Arbeit die 10 min-Werte für die Berechnung der optimalen Induktionsfaktoren genutzt, so gelten diese für die vergangenen 10 Minuten, während zu der Zeit schon leicht andere Bedingungen herrschen. Ob diese Abweichung vertretbar ist, muss untersucht werden.

Wie bereits in Kapitel 3 untersucht wurde, ist der Einsatz von neuronalen Netzen als Optimierung eher unkonventionell, was vermutlich daran liegt, dass die Effizienz eines Optimierungsalgorithmus für das vorliegende Problem bezogen auf die optimale Lösung weitaus höher ist. Der Vorteil eines neuronalen Netzes gegenüber dem klassischen Optimierungsalgorithmus kann später aber die Rechenzeit sein. Eine mögliche Alternative zur jetzigen Methodik ist daher, für jeden vorliegenden Datensatz mit einem Optimierungsalgorithmus ein optimales Set von Induktionsfaktoren zu finden und zu speichern. Das neuronale Netz könnte dann ganz standardmäßig anhand der Umgebungsparameter als Eingang und der korrespondierenden optimalen Induktionsfaktoren als gewünschten Ausgang lernen und so später in einem Bruchteil der Zeit, die der Optimierungsalgorithmus bräuchte, ein Ergebnis liefern.

Literaturverzeichnis

- [1] A. Crespo, J. H.: “Turbulence characteristics in wind-turbine wakes.” (1995).
- [2] A. Marvuglia, A. M.: “Monitoring of wind farms’ power curves using machine learning techniques.” (2012).
- [3] A. Menon, J. S. B.: “Collaborative extremum seeking for welfare optimization.” (2014).
- [4] Annoni, J. u. a.: “Analysis of axial-induction-based wind plant control using an engineering and a high-order wind plant model.” (2015).
- [5] Annoni, J. u. a.: “Evaluating wake models for wind farm control.” (2014).
- [6] Barthelmie, R. J. u. a.: “Modelling and measuring flow and wind turbine wakes in large windfarms offshore.” (2009).
- [7] Burton, T. u. a.: “Wind Energy Handbook.” (2001).
- [8] C. N. Nguyen, O. Z.: “Machine Learning - Kurz und gut.” (2018).
- [9] CC4E: “Windpark Curslack - Daten und Fakten.” (2019). URL: https://www.haw-hamburg.de/fileadmin/CC4E/PDF/CC4E_Brosch%C3%BCren_und_Flyer/Factsheet_Windpark_Curslack_2019.pdf (besucht am 23.09.2022).
- [10] CC4E: “Windpark Curslack.” (2022). URL: <https://www.haw-hamburg.de/cc4e/infrastruktur-und-technische-forschungsausstattung/windpark-curslack/> (besucht am 26.06.2022).
- [11] Dalhoff, P.: “Windenergieanlagen 2. Vorlesung - Physik der Windenergienutzung.” (2021).
- [12] Dhiman, H. S. u. a.: “Bilateral gaussian wake model formulation for wind farms: A forecasting based approach.” (2020).
- [13] E. Bitar, P. S.: “Coordinated control of a wind turbine array for power maximization.” (2013).
- [14] E. Ebegbulem, M. G.: “Power maximization of wind farms using discrete-time distributed extremum seeking control.” (2018).
- [15] Fell, H. J. und Traber: T.: “Sektorale Treibhausgasemissionen weltweit. Fossile Energieträger für deutlich mehr als die Hälfte der weltweiten Treibhausgase verantwortlich - Erdgas wegen steigender Methanemissionen ähnlich klimaschädlich wie Kohle und Erdöl.” (2019).
- [16] Frandsen, S. u. a.: “Analytical modelling of wind speed deficit in large offshore wind farms.” (2005).
- [17] Frandsen, S. u. a.: “Measurements on and modelling of offshore wind farms.” (1996).
- [18] G.-W. Qian, T. I.: “Wind farm power maximization through wake steering with a new multiple wake model for prediction of turbulence intensity.” (2020).
- [19] Giebel, G. u. a.: “State of the art and research gaps in wind farm control - Results of a recent workshop.” (2019).

- [20] H. Sun, H. Y.: “Study on an innovative three-dimensional wind turbine wake model.” (2018).
- [21] Hau, E.: “Windkraftanlagen: Grundlagen, Technik, Einsatz, Wirtschaftlichkeit. 5. Auflage.” (2014).
- [22] Hoek, D. v. d. u. a.: “Effects of axial induction control on wind farm energy production - A field test.” (2019).
- [23] Howland, M. F. u. a.: “Wind farm power optimization through wake steering.” (2019).
- [24] IEC: “IEC 61400-1 Windenergieanlagen - Teil 1: Auslegungsanforderungen.” (2019).
- [25] IEC: “IEC 61400-12-1 Windenergieanlagen - Teil 12-1: Messung des Leistungsverhaltens von Windenergieanlagen.” (2017).
- [26] J. Lee, F. Z.: “GWEC | GLOBAL WIND REPORT 2022” (2022).
- [27] J. Park, K. H. L.: “Bayesian Ascent: A Data-Driven Optimization Scheme for Real-Time Control With Application to Wind Farm Power Maximization.” (2016).
- [28] J. Park, K. H. L.: “Cooperative wind turbine control for maximizing wind farm power using sequential convex programming.” (2015).
- [29] JAX: “JAX Quickstart.” (o.J.). URL: <https://jax.readthedocs.io/en/latest/notebooks/quickstart.html> (besucht am 18.09.2022).
- [30] Jensen, N. O.: “A note on wind generator interaction.” (1983).
- [31] K. E. Johnson, G. F.: “Assessment of extremum seeking control for wind farm energy production.” (2012).
- [32] K. E. Johnson, N. T.: “Wind farm control: Addressing the aerodynamic interaction among wind turbines.” (2009).
- [33] Kaldellis, J. K. u. a.: “Critical evaluation of wind turbines’ analytical wake models.” (2021).
- [34] Karami, F. u. a.: “Real-time wind direction estimation using machine learning on operational wind farm data.” (2021).
- [35] Keck, R.-E. u. a.: “A consistent turbulence formulation for the dynamic wake meandering model in the atmospheric boundary layer.” (2013).
- [36] Khosravi, A. u. a.: “Time-series prediction of wind speed using machine learning algorithms: A case study Osorio wind farm, Brazil.” (2018).
- [37] L. E. Andersson, L. I.: “Real-time optimization of wind farms using modifier adaptation and machine learning.” (2020).
- [38] L. P. Chamorro, F. P.-A.: “A wind-tunnel investigation of wind-turbine wakes: Boundary-layer turbulence effects.” (2009).
- [39] learn, scikit: “sklearn.model selection.train test split.” (o.J.). URL: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html (besucht am 19.09.2022).

- [40] learn, scikit: “sklearn.preprocessing.StandardScaler.” (o.J.). URL: <https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html> (besucht am 09.09.2022).
- [41] Lundquist, J. K. u. a.: “Costs and consequences of wind turbine wake effects arising from uncoordinated wind energy development.” (2018).
- [42] M. Bastankhah, F. P.-A.: “A new analytical model for wind-turbine wakes.” (2014).
- [43] M. F. Howland, J. O. D.: “Wind farm modeling with interpretable physics-informed machine learning.” (2019).
- [44] Marden, J. R. u. a.: “A model-free approach to wind farm control using game theoretic methods.” (2013).
- [45] Marsland, S.: “Machine learning - An algorithmic perspective - Second Edition.” (2015).
- [46] Morice, C. P. u. a.: “An updated assessment of near-surface temperature change from 1850: the HadCRUT5 dataset.” *Journal of Manufacturing Processes, Journal of Geophysical Research (Atmospheres)* (o.J.).
- [47] Nielsen, M.: “Neural Networks and Deep Learning.” (2019).
- [48] Nordex: “Schallemissionen, Leistungskurven, Schubbeiwerte, Nordex N117 2400 (Revision 00).” (2013).
- [49] Nordex: “Schallemissionen, Leistungskurven, Schubbeiwerte, Nordex N117 3000 (Revision 04).” (2019).
- [50] Park, J. u. a.: “Wind farm power maximization based on a cooperative static game approach.” (2013).
- [51] Porté-Agel, F. u. a.: “A numerical study of the effects of wind direction on turbine wakes and power losses in a large wind farm.” (2013).
- [52] R. Braunbehrens, A. S.: “A statistical model for wake meandering behind wind turbines.” (2019).
- [53] Raschka, S.: “Machine Learning mit Phyton. Das Praxis-Handbuch für Data Science, Predictive Analytics und Deep Learning.” (2017).
- [54] Reinwardt, I.: “Validation and improvement of wake models for site-specific load and power calculations in wind farms.” (2021).
- [55] Salamah, Y. B.: “Distributed extremum-seeking for wind farm power maximization using sliding mode control.” (2021).
- [56] Siciliano, B. u. a.: “Springer Handbook of Robotics.” (2008).
- [57] T. Ishihara, G.-W. Q.: “A new gaussian-based analytical wake model for wind turbines considering ambient turbulence intensities and thrust coefficient effects.” (2018).
- [58] Tian, L. u. a.: “Development and validation of a new two-dimensional wake model for wind turbine wakes.” (2015).

Literaturverzeichnis

- [59] Tian, L. u. a.: “Prediction of multi-wake problems using an improved Jensen wake model.” (2016).
- [60] Xiaoxia, G. u. a.: “Optimization of wind turbine layout position in a wind farm using a newly-developed two-dimensional wake model.” (2016).
- [61] Zhan, L. u. a.: “Optimal tuning of engineering wake models through LiDAR measurements.” (2020).

A Datenträger

Auf dem Datenträger sind folgende Dateien zu finden.

- Diese Arbeit als .pdf-Datei.
- Diese Arbeit als .zip-Datei (Rohdateien aus L^AT_EX).
- Alle verwendeten Quellen im Ordner „Quellen“.
- Alle verwendeten Excel-Dateien im Ordner „Excel“:
 - 2019_mod.xlsx, 2020_mod.xlsx
 - 2019_fall1.xlsx
 - 2020_fall1.xlsx
 - 2019_fall1_ohne_0_datensatz.xlsx
 - 2019_fall2.xlsx
 - dichte_untersuchung.xlsx
 - gradient_untersuchung.xlsx
 - Nordex2400.xlsx, Nordex3000.xlsx
 - random2000.xlsx, random8000.xlsx
 - S1.xlsx, S2.xlsx, S3.xlsx, S4.xlsx, S5.xlsx, S6.xlsx, S7.xlsx
- Alle Python-Dateien im Ordner „Python“:
 - digital_twin.py
 - digital_twin_opt.py
 - ann.py
 - digital_twin_valid.py
 - ann_valid.py



Erklärung zur selbstständigen Bearbeitung einer Abschlussarbeit

Gemäß der Allgemeinen Prüfungs- und Studienordnung ist zusammen mit der Abschlussarbeit eine schriftliche Erklärung abzugeben, in der der Studierende bestätigt, dass die Abschlussarbeit – bei einer Gruppenarbeit die entsprechend gekennzeichneten Teile der Arbeit [(§ 18 Abs. 1 APSO-TI-BM bzw. § 21 Abs. 1 APSO-INGI)] – ohne fremde Hilfe selbstständig verfasst und nur die angegebenen Quellen und Hilfsmittel benutzt wurden. Wörtlich oder dem Sinn nach aus anderen Werken entnommene Stellen sind unter Angabe der Quellen kenntlich zu machen.

Quelle: § 16 Abs. 5 APSO-TI-BM bzw. § 15 Abs. 6 APSO-INGI

Dieses Blatt, mit der folgenden Erklärung, ist nach Fertigstellung der Abschlussarbeit durch den Studierenden auszufüllen und jeweils mit Originalunterschrift als letztes Blatt in das Prüfungsexemplar der Abschlussarbeit einzubinden. Eine unrichtig abgegebene Erklärung kann -auch nachträglich- zur Ungültigkeit des Studienabschlusses führen.

Erklärung zur selbstständigen Bearbeitung der Arbeit

Hiermit versichere ich, Frederic Hoops, dass ich die vorliegende Bachelorthesis mit dem Thema:

Leistungsmaximierung eines Windparks mit Hilfe eines digitalen Zwillinges sowie Methoden des maschinellen Lernens

ohne fremde Hilfe selbstständig verfasst und nur die angegebenen Quellen und Hilfsmittel benutzt habe. Wörtlich oder dem Sinn nach aus anderen Werken entnommene Stellen sind unter Angabe der Quellen kenntlich gemacht.

Ort

Datum

Unterschrift im Original