



Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Masterarbeit

Sohail Karimi

Entwicklung eines Berechnungsmodells zur Abschätzung des Energiebedarfs von eVTOLs und Konzeption eines wirtschaftlich optimierten eVTOL-Gesamtkonzepts

*Fakultät Technik und Informatik
Department Fahrzeugtechnik und Flugzeugbau*

*Faculty of Engineering and Computer Science
Department of Automotive and
Aeronautical Engineering*

Sohail Karimi

**Entwicklung eines Berechnungsmodells
zur Abschätzung des Energiebedarfs
von eVTOLs und Konzeption eines wirt-
schaftlich optimierten eVTOL-Gesamt-
konzepts**

Masterarbeit eingereicht im Rahmen der Masterprüfung

im Studiengang Flugzeugbau
am Department Fahrzeugtechnik und Flugzeugbau
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Erstprüfer/in: Prof. Dr.-Ing. Alexander Frenkel
Zweitprüfer/in: Prof. Dr.-Ing. Benedikt Plaumann

Abgabedatum: 20.05.2025

Zusammenfassung

Sohail Karimi

Thema der Masterarbeit

Entwicklung eines Berechnungsmodells zur Abschätzung des Energiebedarfs von eVTOLs und Konzeption eines wirtschaftlich optimierten eVTOL-Gesamtkonzepts

Stichworte

eVTOL, Urban Air Mobility, elektrische Luftfahrt, elektrisch angetriebenes Fluggerät, Energiebedarf, Pro-Kopf-Verbrauch, Reichweite, Schwebeflug, Reiseflug, Nutzlast, Energiedichte, Batteriekapazität, spezifischer Energieverbrauch, Python-Modellierung, wirtschaftliche Tragfähigkeit, Lebenszykluskosten, Betriebskosten, Amortisationsdauer, Kapitalwert, Luftfahrtökonomie, Elektromobilität in der Luftfahrt

Kurzzusammenfassung

Diese Masterarbeit befasst sich mit der Entwicklung eines Berechnungsmodells zur realitätsnahen Abschätzung des Energiebedarfs elektrisch betriebener, senkrecht startender und landender Fluggeräte (eVTOLs). Aufbauend auf einem existierenden Berechnungsverfahren wurde ein erweitertes Modell in Python implementiert, das eine detaillierte Analyse technischer Parameter sowie eine automatisierte Optimierung hinsichtlich Reichweite und Energieeffizienz ermöglicht. Ergänzend wurde ein Kostenmodell integriert, das die wirtschaftliche Tragfähigkeit verschiedener Auslegungsvarianten auf Basis realitätsnaher Betriebsparameter untersucht. Mit dem technischen Konzept wird ein Pro-Kopf-Verbrauch von 8,01 kWh/100km erreicht. Bei einem autonomen Flugbetrieb und einem Passagierpreis von 1,5 \$/km wird eine Amortisierung nach vier Jahren erzielt.

Sohail Karimi

Title of the paper

Development of a Model to Estimate the Energy Demand of eVTOLs and Design of an Economically Optimized Overall eVTOL Concept

Keywords

eVTOL, urban air mobility, electric aviation, electrically powered aircraft, energy demand, per-passenger energy consumption, range, hovering flight, cruising flight, payload, energy density, battery capacity, specific energy consumption, Python modelling, economic viability, life cycle costs, operating costs, payback period, net present value, aviation economics, electromobility in aviation

Abstract

This master's thesis addresses the development of a model for estimating the energy demand of electrically powered vertical take-off and landing aircraft (eVTOLs). Building on an existing calculation method, an extended Python-based model was implemented to enable a detailed analysis of technical parameters as well as automated optimization with regard to range and energy efficiency. In addition, a cost model was integrated to assess the economic viability of different design variants based on realistic operational parameters. The technical concept achieves a per-passenger energy consumption of 8.01 kWh/100 km. With autonomous operation and a passenger price of \$1.50/km, amortization is achieved after four years.

Inhaltsverzeichnis

Abbildungsverzeichnis	V
Tabellenverzeichnis	VI
Abkürzungsverzeichnis	VII
Symbolverzeichnis	X
1. Einleitung	1
1.1 Hintergrund und Motivation	1
1.2 Aufbau der Arbeit	2
1.3 Zielsetzung der Arbeit	3
2. Grundlagen	4
2.1 Einführung eVTOLs.....	4
2.1.1 Designparameter.....	4
2.2 Grundlagen der Flugmechanik	6
2.2.1 Grundlagen der Leistungsberechnung	6
2.2.2 Einfluss von Bodeneffekten	9
2.3 Wirtschaftlichkeitsbetrachtung.....	10
2.3.1 Kostenbetrachtung aus der Perspektive des Flugzeugherstellers.....	11
2.3.2 Kostenbetrachtung aus der Perspektive des Flugzeugbetreibers	12
2.3.3 Wartungskosten nach DOC-Methoden	14
2.3.4 Definition von Direct Operating Cost (DOC)	15
2.3.5 Kostenmodell zur Betrachtung der wirtschaftlichen Tragfähigkeit von eVTOLs ..	18
.....	18
3. Methodik und Modellentwicklung	26
3.1 Anforderungen an das Modell	26
3.2 Analyse des bestehenden Berechnungsverfahrens.....	28
3.2.1 Identifikation von Schwächen und Verbesserungspotenzialen	36
3.2.2 Erweiterung und Anpassung für das eigene Modell.....	37
3.3 Implementierung des eigenen Berechnungsmodells in Python.....	38
3.3.1 Aufbau des Berechnungsmodells	40
3.3.2 Berechnung der Reichweite und des Energiebedarfs	41
3.3.3 Automatisierte Optimierung der Reichweite und des Energiebedarfs	42
3.3.4 Berechnung der wirtschaftlichen Tragfähigkeit	47
4. Analyse bestehender eVTOL-Konzepte	55
4.1 Dokumentation aktueller eVTOL-Konzepte	55

4.1.1	Joby Aviation S4	55
4.1.2	Archer Aviation Midnight	58
4.1.3	Lilium Jet.....	60
4.1.4	Vertical Aerospace VX4	62
4.2	Anwendung des Berechnungsmodells auf die eVTOL-Konzepte.....	66
4.2.1	Abschätzung des Energiebedarfs des Joby Aviation S4.....	66
4.2.2	Abschätzung des Energiebedarfs des Archer Aviation Midnight.....	66
4.2.3	Abschätzung des Energiebedarfs des Lilium Jets	67
4.2.4	Abschätzung des Energiebedarfs des Vertical Aerospace VX4.....	68
4.2.5	Vergleich der Energieeffizienz der verschiedenen Konzepte	68
4.2.6	Ableitung der Erkenntnisse zur Identifikation von Optimierungspotenzial	70
5.	Entwicklung eines optimierten eVTOL-Gesamtkonzepts	72
5.1	Ableitung eines technischen Konzepts unter Berücksichtigung relevanter Designparameter	72
5.2	Entwicklung eines wirtschaftlich tragfähigen Geschäftsmodells.....	74
6.	Diskussion	82
6.1	Bewertung der Ergebnisse	82
6.1.1	Technische Bewertung des eVTOLs-Konzepts	82
6.1.2	Diskussion des wirtschaftlichen Konzepts	84
6.1.3	Erreichte Ziele und Grenzen des Modells.....	86
7.	Fazit und Ausblick.....	88
7.1	Erforderliche Rahmenbedingungen für wirtschaftlich tragfähige eVTOLs	88
7.2	Weiterentwicklung von eVTOLs und Modellverbesserungen	89
8.	Literaturverzeichnis	90
9.	Anhang.....	93
9.1	Anhang 1: Ermittlung der Gesamtrotorfläche einzelner eVTOL-Konzepte	93
9.1.1	Joby Aviation S4	93
9.1.2	Archer Aviation Midnight	94
9.1.3	Vertical Aerospace VX4	95
9.2	Anhang 2: Python-Code des Berechnungsmodells	96
9.2.1	Berechnungsmodell.py.....	96
9.2.2	flugzeug.py.....	96
9.2.3	optimierer.py	99
9.2.4	flet_gui.py.....	102
9.2.5	kostenmodell.py	127
9.2.6	flugzeugManager.py.....	132

10. Selbstständigkeitserklärung..... 134

Abbildungsverzeichnis

Abbildung 1-1: Schematische Darstellung des Aufbaus der Arbeit	3
Abbildung 2-1: Erklärung zum Verhältnis von Auftrieb zu Widerstand nach [8].....	5
Abbildung 2-2: Der Stromfaden nach Bernoulli [13]	6
Abbildung 2-3: Bodeneffekt bei einem Helikopter [13]	10
Abbildung 2-4: Kostenverursacher im Luftverkehr nach [15].....	11
Abbildung 2-5: Übersicht der Bewertungsmethoden im Flugzeugbau nach [16]	14
Abbildung 2-6: Schematische Darstellung der Teilkosten und des Barwerts nach [21]	20
Abbildung 3-1: Schematische Darstellung des 3P-Modells	28
Abbildung 3-2: Variationsrechnungen am Beispiel des Lilium-Jets [4]	36
Abbildung 3-3: Programmierung in VS Code	39
Abbildung 3-4: Programmierung in VS Code mit Hervorhebung der Versionsverwaltung	40
Abbildung 3-5: Startseite des Berechnungsmodells.....	41
Abbildung 3-6: Eingabemaske für die Berechnung im Modell	42
Abbildung 3-7: Ausschnitt des Grid-Search Algorithmus.....	46
Abbildung 3-8: Eingabemaske für die Optimierung im Modell (Ausschnitt)	47
Abbildung 3-9: Programmcode zur Berechnung der Amortisationsdauer (Ausschnitt)	48
Abbildung 3-10: Ausschnitt aus dem Programmcode zur Berechnung des Barwerts (NPV)	48
Abbildung 3-11: Programmcode zur Berechnung der jährlichen Kosten (Ausschnitt).....	49
Abbildung 3-12: Programmcode zur Berechnung der nivellierten Kosten (Ausschnitt).....	49
Abbildung 3-13: Programmcode zur Berechnung der jährlichen Einnahmen (Ausschnitt).....	50
Abbildung 3-14: Eingabemaske für das Kostenmodell.....	53
Abbildung 3-15: Ausgabe des Kostenmodells	54
Abbildung 4-1: Joby Aviation S4 [Joby]	55
Abbildung 4-2: Mögliches Einsatzgebiet für den Joby Aviation S4 [30].....	57
Abbildung 4-3: Archer Aviation Midnight [Archer Aviation]	58
Abbildung 4-4: Mögliches Einsatzgebiet für den Archer Aviation Midnight [37]	60
Abbildung 4-5: Lilium Jet [Lilium]	60
Abbildung 4-6: Mögliches Einsatzgebiet für den Lilium Jet [26]	62
Abbildung 4-7: Vertical Aerospace VX4 [43]	62
Abbildung 4-8: Mögliches Einsatzgebiet für den VX4 (Battersea – LHR) [44]	64
Abbildung 4-9: Mögliches Einsatzgebiet für den VX4 (MIA - Fort Lauderdale) [44].....	64
Abbildung 4-10: Mögliches Einsatzgebiet für den VX4 (Yumeshima Port - KIX) [44]	65
Abbildung 4-11: Modellausgabe für Reichweite und Energiebedarf des Joby Aviation S4	66
Abbildung 4-12: Modellausgabe für Reichweite und Energiebedarf des Archer Midnight.....	67
Abbildung 4-13: Modellausgabe für Reichweite und Energiebedarf des Lilium Jets.....	67
Abbildung 4-14: Modellausgabe für Reichweite und Energiebedarf des Vertical VX4.....	68
Abbildung 5-1: Modellausgabe der verworfenen Kombinationen	74

Tabellenverzeichnis

Tabelle 1: Übersicht der berücksichtigten Kostenelemente nach DOC-Methode nach [16]...	16
Tabelle 2: Anforderungen an das Berechnungsmodell	26
Tabelle 3: Auswirkungen einzelner Parameter auf die Reichweite.....	32
Tabelle 4: Auswirkungen einzelner Parameter auf den Pro-Kopf-Verbrauch	33
Tabelle 5: Übersicht der fixen Parameter für die Optimierung.....	43
Tabelle 6: Übersicht der freien Parameter für die Optimierung	44
Tabelle 7: Randbedingungen für die Optimierung.....	45
Tabelle 8: Parameter für die Kostenberechnung nach Kostenszenario.....	51
Tabelle 9: Parameter für den eVTOL Betrieb.....	52
Tabelle 10: Weitere Rahmenbedingungen zur Kostenrechnung	53
Tabelle 11: Eingabeparameter Joby Aviation S4	56
Tabelle 12: Eingabeparameter Archer Aviation.....	59
Tabelle 13: Eingabeparameter Lilium Jet.....	61
Tabelle 14: Eingabeparameter Vertical Aerospace VX4	63
Tabelle 15: Energiebedarf und Reichweite des Joby Aviation S4	66
Tabelle 16: Energiebedarf und Reichweite des Archer Aviation Midnight	66
Tabelle 17: Energiebedarf und Reichweite des Lilium Jets.....	67
Tabelle 18: Energiebedarf und Reichweite des Vertical Aerospace VX4	68
Tabelle 19: Vergleich des Energiebedarfs mit Überblick auf die Designparameter	69
Tabelle 20: Übersicht der Designparameter und der Ergebnisse des technischen Konzepts	72
Tabelle 21: Übersicht der Gesamteinnahmen nach Passagierpreis und Flugbetrieb	78
Tabelle 22: Übersicht der nivellierten Kosten (LC) nach Kostenszenario und Flugbetrieb	81

Abkürzungsverzeichnis

A

AEA · *Association of European Airlines*

ATA · *Air Transport Association*

B

BC · *Base Costs*

C

CAA · *Civil Aviation Authority*

CFK · *Kohlenstofffaserverstärkter Kunststoff*

COC · *Cash Operating Costs*

COO · *Cost of Ownership*

D

DOC · *Direct Operating Costs*

E

eVTOL · *electric Vertical Take-Off and Landing*

EWB · *Newark Liberty International Airport*

F

FAA · *Federal Aviation Administration*

G

GUI · *Graphical User Interface*

H

HC · *High Costs*

I

IOC · *Indirect Operating Costs*

J

JFK · *John F. Kennedy International Airport*

K

KIX · *Kansai International Airport*

L

LC · *Low Costs*

LCC · *Life Cycle Costs*

LHR · *London Heathrow Airport*

M

MIA · *Miami International Airport*

N

NASA · *National Aeronautics and Space Administration*

NRC · *non recurring costs*

O

OOP · *Objektorientierte Programmierung*

P

PT · *Powertrain*

R

RC · *recurring costs*

S

SV · *Salvage Value*

T

TFI · *Taxes, Fees, Insurance*

TOC · *Total Operating Costs*

U

UAD · *Unmanned Autonomous Driving*

V

VBM · *VTOL Body Manufacturing*

VS Code · *Visual Studio Code*

VTOL · *Vertical Take-Off and Landing*

Symbolverzeichnis

Symbol	Bezeichnung
β	Anteil an Flügen mit extremen Wetterbedingungen
η_R	Gesamtwirkungsgrad im Reiseflug
η_S	Gesamtwirkungsgrad in der Schwebeflugphase
μ_A	Batteriemassenverhältnis
ρ	Dichte (der Luft)
AM	Gesamtmasse der Batterieeinheit
A_r	Gesamtrotorkreisfläche der Triebwerke
AR	Flügelstreckung
b	Spannweite
c	Mittlere Flügeltiefe
$c_{A,R}$	Auftriebskoeffizient im Reiseflug
c_{D0}	Beiwerte weiterer parasitärer Widerstände
$c_{D0,LW}$	Reibung des Leitwerks
$c_{D0,OR}$	Oberflächenreibung des Flügels
$c_{D0,R}$	Reibung und Formwiderstand des Rumpfes
$c_{D0,T}$	Reibung der Triebwerksgehäuse
$c_{Di,R}$	Auftriebsinduzierter Widerstandsbeiwert
$C_{a/c,a}$	Jährliche Betriebskosten des Flugzeugs
$C_{a/c,m}$	Flugkosten-pro-Kilometer
$C_{a/c,t}$	Flugmissionskosten
$C_{s,m}$	Flugkosten-pro-Passagier-Kilometer
C_C	Kosten für Flugpersonal (Gesamt)
$C_{C,CA}$	Kosten für Flugpersonal (Kabine)
$C_{C,CO}$	Kosten für Flugpersonal (Cockpit)
C_{DEP}	Kosten für Abschreibung
C_{DOC}	Betriebskosten
C_{E_t}	Energiekosten im Jahr t
C_{INT}	Kosten für Zinsen
C_{INS}	Kosten für Versicherung
C_{I_t}	Kosten für die Infrastruktur im Jahr t
C_F	Kosten für Kraftstoff
$C_{FEE,LD}$	Kosten für Landegebühren

$C_{FEE,NAV}$	Kosten der Gebühren für die Flugsicherung
$C_{FEE,GND}$	Kosten für Abfertigungsgebühren
C_{jt}	Teilkosten im Jahr t
C_M	Kosten für Wartung (Gesamt)
$C_{M,AF}$	Kosten für die Wartung der Flugzeugzelle
$C_{M,PP}$	Kosten für die Wartung der Triebwerke
C_{M_t}	Wartungskosten im Jahr t
C_{PT_t}	Kosten für die Herstellung des Antriebsstrangs
C_{RFC_0}	Baukosten für die Energieversorgungseinrichtung
C_{RFO_t}	Betriebskosten der Energieversorgungseinrichtung
$C_{SV_{VL}}$	Restwert im letzten Jahr der Lebensdauer VL des eVTOLs
C_{TFI_t}	Kosten für Steuern, Gebühren und Versicherungen im Jahr t
C_{UAD_1}	Kosten für das autonome System
C_{VBM_1}	Kosten für die Herstellung der eVTOL-Zelle im ersten Jahr
C_{VBM-F}	Fixen Herstellungskosten der eVTOL-Zelle
C_{VBM-V}	Variable Herstellungskosten der eVTOL-Zelle
C_{VPC_0}	Baukosten für den Vertiport
C_{VPO_t}	Betriebskosten des Vertiports
D	Widerstand
DR	Zinsrate
e	Oswald-Faktor
e_A	Energiedichte der Batterie
E_A	Die von der Batterie bereitgestellte Energie
E_c	Energieverbrauch
E_R	Energiebedarf für den Reiseflug
E_S	Energiebedarf in der Schwebeflugphase
F	Rotorkreisfläche
F_2	Rotorkreisfläche ohne Eigenbewegung des Rotors
g	Erdbeschleunigung
G	Gewichtskraft
GZ_{max}	Maximale Gleitzahl
GZ_R	Gleitzahl im Reiseflug
H	Höhe
IRC	Amortisationsdauer (Investment Return Cycle)
\dot{m}	Massenstrom

$m_{0_{VBM}}$	eVTOL Strukturmasse
MAM / m_{TO}	Maximale Abflugmasse
LC	Nivellierte Kosten
n_D	Anzahl Werktage pro Jahr
$n_M / n_{t,a}$	Anzahl Flugmissionen pro Tag / pro Jahr
n_T	Anzahl der Triebwerke
N_R	Rotorleistung
$N_{Ro,0}$	Im Schwebeflug von Rotoren erforderliche Antriebsleistung
NPV	Kapitalwert (Net Present Value)
p_0	Umgebungsdruck
p_1	Druck oberhalb der Rotorebene
p'_1	Druck unterhalb der Rotorebene
p_E	Stromkosten
p_P	Flugpreis pro Passagier
$P_{charging}$	Ladeleistung
P_{DS}	Leistung des Antriebssystems
$P_{R,A}$	Erforderliche Batterieleistung für den Reiseflug
$P_{R,erf}$	Erforderliche Antriebsnettleistung für den Reiseflug
$P_{S,A}$	Erforderliche Batterieleistung in der Schwebeflugphase
$P_{S,erf}$	Erforderliche Antriebsnettleistung in der Schwebeflugphase
PKV	Pro-Kopf-Verbrauch
PAX/n_P	Passagieranzahl
PYC_t	Abgezinst jährliche Kosten im Jahr t
PYR_t	Abgezinst jährliche Einnahmen im Jahr t
Q	Batteriekapazität
R	Reichweite
$Ratio_{DE}$	Anteil der Strecke für leere Rückflüge
$Ratio_M$	Anteil der fixen Wartungskosten an den Herstellungskosten für die eVTOL-Zelle
$Ratio_P$	Nutzladefaktor
$Ratio_{SV}$	Anteil des Restwerts an den Herstellungskosten für die eVTOL-Zelle
$Ratio_{TFI}$	Anteil der Kosten für Steuern, Gebühren und Versicherungen von den direkten Betriebskosten
R_d	Rotordurchmesser
RFL	Lebensdauer der Energieversorgungseinrichtung
r_N	Nabenradius
r_P	Propellerradius

S	Schub
S_w	Flügelfläche
$t_{boarding}$	Zeit für das Einsteigen der Passagiere
t_{climb}	Steigflugdauer
$t_{descent}$	Sinkflugdauer
$t_{disembarking}$	Zeit für das Aussteigen der Passagiere
$t_{mission}$	Dauer einer Flugmission
t_r	Anzahl der Jahre, bis Amortisierung erreicht wird
t_R	Gesamtdauer des Reiseflugs
t_S / t_{hover}	Gesamtdauer der Schwebeflugphase
$t_{supplement}$	Zeit zum Aufladen oder Austauschen der Batterie
$t_{swapping}$	Zeit zum Austauschen der Batterie
$t_{transport}$	Flugzeit, zusammengesetzt aus Reise-, Schweb-, Steig- und Sinkflug
T_R	Erforderlicher Schub für den Reiseflug
T_S	Erforderlicher vertikaler Schub in der Schwebeflugphase
UC_{DS}	Kosten für das Antriebssystem
UC_{ES}	Kosten für die Batterie
UC_{VM}	Variable Wartungskosten
v_i	Induzierte Geschwindigkeit
v_R / V	Reisefluggeschwindigkeit
v_0	Strömungsgeschwindigkeit in der Ebene 0
v_1	Strömungsgeschwindigkeit in der Ebene 1
v_2	Strömungsgeschwindigkeit in der Ebene 2
VL	Lebensdauer eVTOL
VPL	Lebensdauer des Vertiports
YT	Jährliche Flugzeit

1. Einleitung

Die rasante Urbanisierung und das starke Bevölkerungswachstum in Ballungszentren auf der gesamten Welt stellen bereits bestehende Verkehrssysteme und Mobilitätsträger vor große Herausforderungen. Neben Umweltbelastungen und Überlastung der Infrastruktur erfordern begrenzte Flächenressourcen innovative Mobilitätslösungen, um den urbanen Verkehr effizienter und nachhaltiger zu gestalten. In Anbetracht dieses Kontexts gewinnen Luftfahrzeuge, insbesondere elektrisch betriebene, senkrecht startende und landende Flugzeuge (eVTOLs) zunehmend an Bedeutung. Unternehmen und Investoren versprechen sich damit den urbanen Luftraum für den Personentransport zu erschließen, aber auch andere Einsatzszenarien effizienter zu gestalten. Obwohl zahlreiche technische Untersuchungen zu eVTOLs vorliegen, ist die wirtschaftliche Tragfähigkeit dieser Konzepte nach wie vor umstritten.

1.1 Hintergrund und Motivation

Der globale Markt für eVTOLs befindet sich in einer Wachstumsphase. Der Marktwert von 1,35 Milliarden USD in 2023 soll Schätzungen zufolge auf 17,7 Milliarden US Dollar bis zum Jahr 2030 steigen [1]. Nicht nur in den Vereinigten Staaten und Europa investieren Unternehmen und Regierungen in die Entwicklung und Implementierung von eVTOLs. In China plant die Regierung bis zum Jahr 2030 in etwa 100.000 eVTOLs in die Luft zu bringen, um den urbanen Verkehr zu entlasten [2]. Trotz des dynamischen Wachstums der eVTOL-Technologien und zahlreichen Investitionen stehen eVTOLs vor großen Herausforderungen, insbesondere im Hinblick auf die Zertifizierung und damit das Einhalten von gesetzlich-regulativen Rahmenbedingungen.

Bisherige Forschungen und Untersuchungen haben das Verständnis für die technologische Machbarkeit von eVTOLs verbessert. Die Verknüpfung zwischen der technischen Auslegung und deren Auswirkungen auf die wirtschaftliche Tragfähigkeit bleibt jedoch bislang weitgehend unzureichend erforscht. Die Integration technischer und ökonomischer Betrachtungen ist jedoch essenziell, um die praktische Umsetzbarkeit von zukünftigen eVTOL-Konzepten bewerten zu können. Bei der Untersuchung der Wirtschaftlichkeit wird der Energiebedarf dabei oft als zentraler Einflussfaktor betrachtet.

Vor diesem Hintergrund besteht die Motivation dieser Arbeit darin, ein Berechnungsmodell zu entwickeln, das eine realitätsnahe Bestimmung des Energiebedarfs von eVTOLs ermöglicht. Das Modell basiert dabei auf einem in der Fachliteratur [3] beschriebenen Berechnungsverfahren, welches auf Schwächen und auf Verbesserungspotenzial untersucht, angepasst und erweitert wird. Darüber hinaus wird ein Ansatz integriert, der eine Optimierung des Energiebedarfs ermöglicht, indem geeignete Designparameter des eVTOLs automatisiert ausgegeben und angepasst werden. Aufbauend auf dieser Berechnung wird innerhalb desselben Modells

Einleitung

ein Kostenmodell integriert, das eine Abschätzung der wirtschaftlichen Tragfähigkeit verschiedener Auslegungskonzepte erlaubt. Besonderer Fokus liegt dabei auf der Einbeziehung betrieblicher Rahmenbedingungen.

1.2 Aufbau der Arbeit

Nach der Einleitung wird in Kapitel 2 dieser Arbeit der theoretische Rahmen vorgestellt, der das Verständnis des Berechnungsverfahrens und der nachfolgenden Ergebnisse erleichtern soll. Es werden grundlegende Aspekte der Flugmechanik, zentrale Designparameter von eVTOLs sowie die Basis für die wirtschaftliche Bewertung erläutert.

In Kapitel 3 werden zunächst die Anforderungen für das zu entwickelnde Berechnungsmodell definiert. Darüber hinaus erfolgt eine Analyse des bestehenden Berechnungsansatzes, auf dem die zentralen Modellberechnungen beruhen, mit Blick auf Korrektheit, Erweiterbarkeit und mögliche Verbesserungen. Als Resultat darauf wird dann ein eigenes Berechnungsmodell entwickelt und vorgestellt, das den vorher definierten Anforderungen entspricht.

Daraufhin werden in Kapitel 4 bereits bestehende eVTOL-Konzepte analysiert. Zunächst werden die wichtigsten Daten zu den Konzepten dokumentiert. Mit diesen Daten wird dann das vorher entwickelte Berechnungsmodell angewendet, um den Energieverbrauch dieser eVTOL-Konzepte abzuschätzen und die Konzepte untereinander zu vergleichen.

In Kapitel 5 wird aus den Erkenntnissen aus den vorherigen Kapiteln ein eigenes, optimiertes eVTOL-Gesamtkonzept erarbeitet. Dieses beinhaltet neben technischen Parametern auch ein wirtschaftliches Konzept.

Anschließend werden die Ergebnisse in Kapitel 6 diskutiert. Dabei erfolgt eine Bewertung des technischen und wirtschaftlichen Konzepts. Des Weiteren wird untersucht, inwieweit die Anforderungen an das Modell erfüllt wurden und wo die Grenzen und Schwächen des Modells liegen.

Im letzten Abschnitt der Arbeit wird zusammengefasst, welche technischen und wirtschaftlichen Randbedingungen erfüllt werden müssen, um eVTOLs als wirtschaftlich tragfähige Systeme klassifizieren zu können. Schlussendlich wird ein Ausblick auf die Weiterentwicklung von eVTOLs und Modellverbesserung gegeben. Eine schematische Darstellung des Aufbaus dieser Arbeit ist in Abbildung 1-1 zu sehen.

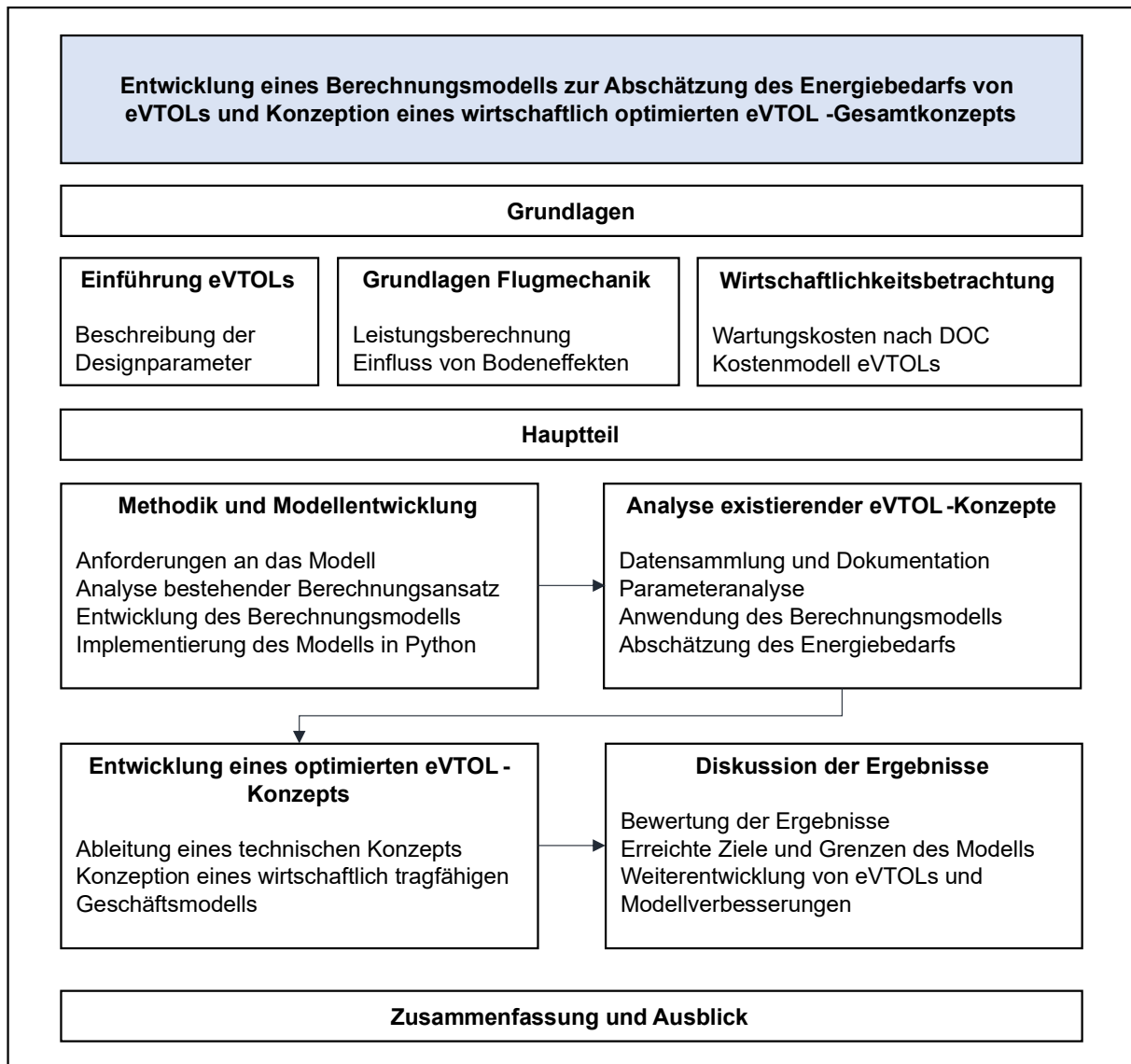


Abbildung 1-1: Schematische Darstellung des Aufbaus der Arbeit

1.3 Zielsetzung der Arbeit

Das Hauptziel dieser Masterarbeit besteht darin, ein Berechnungsmodell zur Abschätzung des Energiebedarfs von eVTOLs zu entwickeln. Grundlage hierfür bildet eine Konzeptberechnung aus der Fachliteratur [4]. Darüber hinaus soll eine Optimierung implementiert werden, die es ermöglicht, relevante Designparameter automatisiert so auszugeben, dass der Energiebedarf minimiert wird. Ergänzend dazu wird ein Kostenmodell integriert, das die wirtschaftliche Bewertung verschiedener eVTOL- und Betriebskonzepte erlaubt. Ziel ist es schlussendlich, mithilfe dieser Analysen ein technisch und wirtschaftlich optimiertes eVTOL-Konzept zu entwickeln.

2. Grundlagen

Um ein besseres Verständnis für die Ergebnisse dieser Arbeit zu schaffen, werden im folgenden Kapitel grundlegende Aspekte der Flugmechanik, zentrale Designparameter von eVTOLs sowie Methoden für die wirtschaftliche Bewertung in der Luftfahrt erläutert.

2.1 Einführung eVTOLs

Das senkrechte Starten und Landen, welches im englischen auch als Vertical Take-Off and Landing bezeichnet wird [5], erfordert eine Antriebskraft, die mindestens dem Abfluggewicht des Fluggeräts entspricht. Eine erfolgreiche Umsetzung dieser Herausforderung schafft der Helikopter. Durch die besonders große Kreisfläche des Rotors und der daraus resultierenden geringen Kreisflächenbelastung kann der Hubschrauber den benötigten Schub für den Senkrechtstart mit relativ geringer Triebwerksleistung aufbringen. Allerdings ist die Effizienz eines Hubschraubers im Reiseflug begrenzt, da der Vortrieb hier durch das Neigen des Rotors entsteht. Dies führt insbesondere bei hohen Geschwindigkeiten zu aerodynamischen Nachteilen, welche Hubschrauber in ihrer Reichweite und in ihrer Fähigkeit, hohe Reisefluggeschwindigkeiten zu erreichen limitiert. Im Hinblick auf die Wirtschaftlichkeit im Reiseflug sind Hubschrauber aus diesem Grund konventionellen Flugzeugen unterlegen [6].

Um die Vorteile des senkrechten Starten und Landens mit der Effizienz eines konventionellen Reiseflugs zu kombinieren, sind alternative Lösungen erforderlich. An dieser Stelle gewinnen VTOLs bzw. eVTOLs an Bedeutung. Als VTOLs werden demnach Fluggeräte bezeichnet, die über so ein Start- und Landeverhalten verfügen, aber auch die konventionellen Reiseflugleistungen erbringen können. Die Erweiterung um eVTOL bedeutet dabei, dass das VTOL elektrisch betrieben wird.

2.1.1 Designparameter

In diesem Abschnitt werden einige der grundlegenden Designparameter erläutert, die später als Eingabewerte im Berechnungsmodell verwendet werden. Basierend auf der Recherche der Fachliteratur werden dabei Wertebereiche definiert, die diese Parameter in der Praxis annehmen können. Dies bildet eine Grundlage für die Optimierungsfunktion des Berechnungsmodells und trägt zugleich dazu bei, ein vertieftes Verständnis der Parameter zu schaffen, das für die nachfolgenden Analysen von Bedeutung ist.

Maximale Abflugmasse

Die maximale Abflugmasse ist die Gesamtmasse eines Luftfahrzeugs beim Start und umfasst die strukturelle Masse, das Antriebssystem, die Energiespeicher sowie die transportierte Nutzlast [7].

Gleitzahl

Ein Flugzeug, das mit konstanter Geschwindigkeit fliegt, wie in Abbildung 2-1 dargestellt, erfährt die aerodynamische Kraft R . Diese setzt sich aus zwei Komponenten zusammen: einer Kraft in Richtung der Anströmung sowie einer senkrecht dazu wirkenden Kraft. Die Komponente in Anströmungsrichtung wird als Widerstand W bezeichnet, während die senkrecht dazu stehende Kraft den Auftrieb A darstellt [8]. Das Verhältnis von Auftrieb zu Widerstand ist entscheidend für die aerodynamische Effizienz eines Flugzeugs und wird als Gleitzahl (GZ) definiert [8], [9]. Diese hängt maßgeblich von der Gestaltung der Tragflächen ab. Eine höhere Gleitzahl gilt als flugmechanisch vorteilhaft. Entsprechend kann die Gleitzahl als eine Gütezahl für die aerodynamische Leistungsfähigkeit eines Flugzeugs betrachtet werden [8].

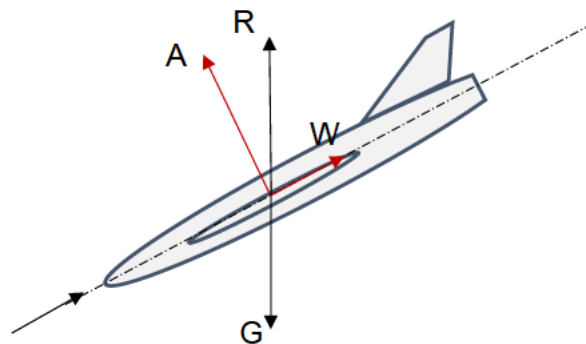


Abbildung 2-1: Erklärung zum Verhältnis von Auftrieb zu Widerstand nach [8]

Tragflügel zeichnen sich im Vergleich zu anderen Körpergeometrien durch ein besonders hohes Verhältnis von Auftrieb zu Widerstand aus. Für eine inkompressible Strömung ergeben sich bei einer rechteckigen Platte mit einem Seitenverhältnis von $\frac{b}{l} = 6$ Werte für die Gleitzahl von $GZ_{max} = 6$ bis 8. Deutlich mehr Auftrieb kann man erreichen, wenn die Platte eine Wölbung aufweisen kann. Hier sind Werte für $GZ_{max} = 10$ bis 12 zu erreichen. Höhere Werte erreicht man mit profilierten Tragflächen. Bei einer besonders guten Abrundung an der Vorderkante und einer scharfauslaufenden Hinterkante können Werte von $GZ_{max} = 25$ und höher erzielt werden [8]. Eine Methode zur Berechnung der Gleitzahl wird in Kapitel 3.2 erläutert. In die Berechnung der Gleitzahl fließt durch die Auftriebsbeiwerte der Oswald-Faktor mit ein. Empirische Daten liegen nahe, dass der Oswald-Faktor Werte von 0,5 bis 0,8 einnehmen kann, wenn der Rotor auch als Tragfläche operiert [10].

Wirkungsgrade

Der Wirkungsgrad einer Maschine beschreibt in der Technik das Verhältnis von Nutzen zu Aufwand [11]. Bei elektrischen Antrieben können Wirkungsgrade bis zu 0,8 erreicht werden [12]. Aus der Literaturrecherche geht hervor, dass der Wirkungsgrad von Rotoren Spitzenwerte von 0,6 bis 0,85 annehmen kann [10].

2.2 Grundlagen der Flugmechanik

Für eine fundierte Analyse des Berechnungsverfahrens sowie der zugrunde liegenden Annahmen ist ein solides Verständnis der flugmechanischen Prinzipien unerlässlich. Dieses Kapitel zielt darauf ab, dieses Verständnis zu etablieren, indem es sich auf die Darstellung grundlegender Berechnungen in der Flugmechanik konzentriert und eine detaillierte Erläuterung der relevanten aerodynamischen und flugmechanischen Effekte bietet und flugmechanischen Effekten behandelt werden.

2.2.1 Grundlagen der Leistungsberechnung

In diesem Kapitel werden wichtige Zusammenhänge erläutert, die zum Verständnis der Analyse des Berechnungsverfahrens in Kapitel 3.2 beitragen sollen. Herleitungen zu Gleichungen für die spätere Berechnung, die aus Kapitel 3.2 nicht ersichtlich werden, werden im folgenden Kapitel behandelt.

Strahltheorie nach Bernoulli

Für die Strahltheorie nach Bernoulli wird ein „idealer“ Rotor angenommen. Dieser wird als kreisförmiges Kontinuum definiert, welches die Fähigkeit besitzt, einen gerichteten Luftstrom nach unten zu erzeugen [13]. Der sogenannte allgemeine Stromfaden ist in Abbildung 2-2 zu sehen.

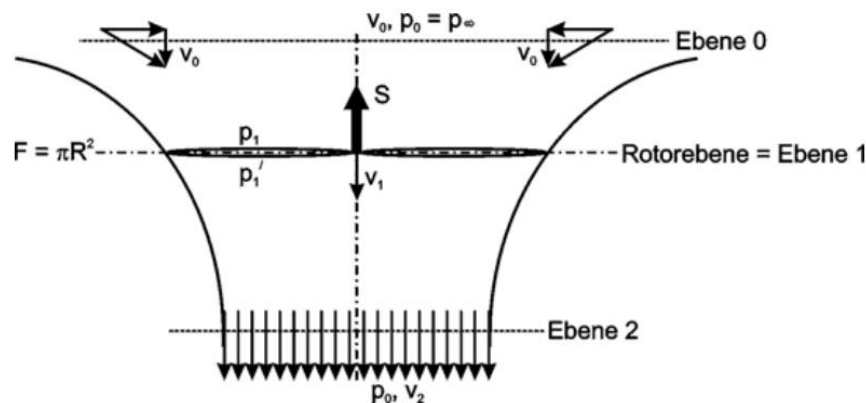


Abbildung 2-2: Der Stromfaden nach Bernoulli [13]

Für den idealisierten Fall werden hierbei folgende realen Effekte vernachlässigt [13]:

- Der Profilwiderstand der Rotorblätter, der insbesondere unerwünschte Sekundärströmungen im Nachstrom bewirkt.
- Der Drall im abgehenden Luftstrom, der die zentrale Sekundärströmung darstellt.
- Die ungleichmäßige Druckverteilung über der Rotorebene, die zu Scherströmungen führt.
- Der inhomogene Durchfluss durch die Rotorebene, der die Strömungscharakteristik

beeinflusst.

- Der zentrale Störkörper, der durch Verdrängung, Verwirbelung und zusätzlichen Strömungswiderstand die Effizienz des Systems beeinträchtigt.
- Die Blattspitzenverluste, die durch den Druckunterschied zwischen der Ober- und Unterseite der Rotorblätter entstehen.
- Der nicht profilierte Blatthals (Cutout), der keinen Beitrag zur Strömungsbeschleunigung leistet.
- Die endliche Anzahl an Rotorblättern, wodurch lokale Unterschiede in der Strömungsbeschleunigung entstehen.
- Die Kompressibilität der Luft, die in diesem Fall noch eine untergeordnete Rolle spielt.
- Der Einfluss von Machzahleffekten, die hier nur in geringem Maße auftreten.
- Die Reynolds-Zahl, deren Einfluss insbesondere den Übergang von laminarer zu turbulenter Strömung innerhalb der Strömungsgrenzschicht bestimmt.

Durch das Beschleunigen des Massenstroms (\dot{m}) entsteht der Schub (S). Des Weiteren erzeugt der Rotor unmittelbar oberhalb der Rotorkreisfläche einen Unterdruck (p_1) und unmittelbar unterhalb einen Überdruck (p'_1) [13]:

$$p_1 < p_0 < p'_1 \quad (1)$$

Der Druckunterschied von Ober- und Unterseite wirkt auf die Rotorkreisfläche (F) und erzeugt damit den Schub [13]:

$$S = (p'_1 - p_1) \cdot F \quad (2)$$

Da die Drücke für die Leistungsberechnung rechnerisch nicht verfügbar sind, müssen andere Parameter verwendet werden. Hierzu bieten sich zum einen die Strömungsgeschwindigkeit v_0 in der Ebene (0) und v_2 in der Ebene (2), in der der Druck bereits wieder den Umgebungsdruck angenommen hat. Zum anderen kann der Massenstrom verwendet werden, da er nach Definition des Stromfadens überall gleich sein muss. Zuletzt kann auch die Luftdichte verwendet werden, da sie in der Ebene (2) wie der Druck auch den Umgebungswert angenommen hat [13].

Demnach gelten folgende Zusammenhänge [13]:

$$\text{Massenfluss: } \dot{m} = \rho \cdot F \cdot v_1 = \rho \cdot F_2 \cdot v_2 \quad (3)$$

$$\text{Impulssatz: } S = \dot{m} \cdot (v_2 - v_0) \quad (4)$$

$$\text{Energiesatz: } N_R = \frac{1}{2} \cdot \dot{m} \cdot (v_2^2 - v_1^2) = \quad (5)$$

$$= \frac{1}{2} \cdot \dot{m} \cdot (v_2 + v_0) \cdot (v_2 - v_0) \quad (6)$$

In der Ebene (1) arbeitet der Rotor unter den dort geltenden Bedingungen und benötigt entsprechend die erforderliche Rotorleistung (N_R) [13]:

$$N_R = S \cdot v_1 \quad (7)$$

Die Geschwindigkeit v_1 , mit der der Rotor durchströmt wird ergibt sich hier aus der Summe der Anströmgeschwindigkeit in der Ebene (0) und der induzierten Geschwindigkeit v_i , die vom Rotor auf die Strömung übertragen wird [13]:

$$v_1 = v_0 + v_i \quad (8)$$

Nach Gleichung (8) wird aus Gleichung (7):

$$N_R = S \cdot (v_0 + v_i) \quad (9)$$

Unabhängig davon ob die Berechnung nach dem Impulssatz (4) oder dem Energiesatz (6) erfolgt, muss das Ergebnis für den Rotorleistungsbedarf derselbe sein [13]:

$$N_R = \dot{m} \cdot v_1 \cdot (v_2 - v_0) = \frac{1}{2} \cdot \dot{m} \cdot (v_2 + v_0) \cdot (v_2 - v_0) \quad (10)$$

Woraus sich für die Geschwindigkeiten ergibt:

$$v_1 = \frac{v_2 + v_0}{2}, v_2 = 2 \cdot v_1 - v_0 \quad (11)$$

Ohne Eigenbewegung des Rotors, also bei der Geschwindigkeit $v_0 = 0$ ergibt sich mit Gleichung (3) und Gleichung (8) für die Rotorkreisfläche F_2 [13]:

$$F_2 = \frac{F}{2} \quad (12)$$

Das Einsetzen von Gleichung (8) und (10) in Gleichung (4) liefert:

$$S = 2 \cdot \dot{m} \cdot v_i = 2 \cdot \rho \cdot F \cdot v_1 \cdot v_i = \quad (13)$$

$$= 2 \cdot \rho \cdot F \cdot (v_0 + v_i) \cdot v_i \quad (14)$$

Der stationäre Schwebeflug

Aus der Strahltheorie nach Bernoulli ergeben sich wesentliche Zusammenhänge für den stationären Schwebeflug. In diesem Flugzustand liegt keine Anströmgeschwindigkeit v_0 in der Ebene (0) weit oberhalb der Rotoren vor. Die Strömung durch die Rotorebene erfolgt ausschließlich durch die induzierte Geschwindigkeit v_{i0} , die für den Schwebeflug charakteristisch ist. Für die Geschwindigkeit an der Ebene (2) (vgl. Abbildung 2-2) ergibt sich aus den Gleichungen (8) und (10) für den Schwebeflug [13]:

$$v_2 = 2 \cdot v_{i0} \quad (15)$$

Der Schub, der von den Rotoren erzeugt wird, gleicht die Gewichtskraft, also das Abfluggewicht des Fluggeräts aus. Daraus folgt, dass Schub (S) und Gewichtskraft (G) die gleichen Beträge annehmen. Aus Gleichung (14) der Strahltheorie wird somit mit Gleichung (15) [13]:

$$|G| = |S| = 2 \cdot \rho \cdot F \cdot v_{i0}^2 \quad (16)$$

Demnach wird:

$$v_{i0} = \sqrt{\frac{G}{2 \cdot \rho \cdot F}} \quad (17)$$

Und aus Gleichung (7) und (8) ergibt sich:

$$N_{Ro,0} = G \cdot \sqrt{\frac{G}{2 \cdot \rho \cdot F}} \quad (18)$$

Damit ist die im Schwebeflug von den Rotoren erforderliche Antriebsleistung erläutert, welche in Kapitel 3.2 erneut aufgegriffen wird. Für einige dieser Größen werden in Kapitel 3.2 jedoch andere Symbole verwendet, welche vorher benannt werden.

2.2.2 Einfluss von Bodeneffekten

Die kritischsten Flugphasen bilden der Start und die Landung. Hier befindet sich ein Fluggerät unmittelbar in Bodennähe. Der bei der Auftriebserzeugung entstehende Abwind wird in Bodennähe eingeschränkt, wodurch sich die Strömungsverhältnisse um das Flugzeug deutlich verändern. In diesem Fall tritt der sogenannte Bodeneffekt ein [14].

Der Bodeneffekt ist ein physikalisches Phänomen, bei dem ein umströmender Körper in Bodennähe zusätzlichen dynamischen Auftrieb und verminderten Strömungswiderstand erfährt. Bei der Anströmung eines Flügelprofils von vorne bildet sich an der Unterseite des Profils ein Überdruck sowie ein Unterdruck an der Oberseite. Der Auftrieb ergibt sich dabei aus der

Druckdifferenz. Nähert sich ein Flugzeug dem Boden, so staut sich die Luft unter den Tragflügeln. Daraus resultiert, dass sich der Überdruck an der Unterseite vergrößert und somit auch die Auftriebskraft [14]. Der Bodeneffekt tritt ab folgender Höhe auf:

$$H \leq 2 \cdot R_d \quad (19)$$

Dabei ist R_d der Rotordurchmesser [13]. Laut anderer Definitionen tritt der Bodeneffekt bei Höhen unterhalb der Flügelspannweite ein. Abbildung 2-3 veranschaulicht die Auswirkung des Bodeneffekts am Beispiel eines Helikopters. Durch den Stauwirkung verringert sich die induzierte Geschwindigkeit im Vergleich zum freien Flug. Infolgedessen benötigt der Rotor bei gleichem Schub weniger Antriebsleistung. Gleichzeitig bedeutet dies auch, dass der Rotor bei voller Leistung mit Bodeneffekt mehr Gewicht heben kann [13].

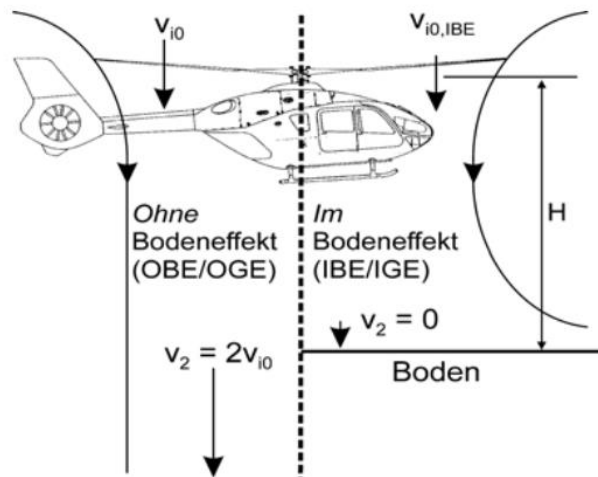


Abbildung 2-3: Bodeneffekt bei einem Helikopter [13]

Der Bodeneffekt gilt als schwer zu modellierender Einfluss und wird oftmals in der Leistungsberechnung vernachlässigt [14].

2.3 Wirtschaftlichkeitsbetrachtung

Die Entwicklung und die Einführung von eVTOLs wird maßgeblich von wirtschaftlichen Faktoren beeinflusst. Neben den bekannten technischen Herausforderungen wie der Energieeffizienz, der Batterietechnologie und der benötigten Infrastruktur sind insbesondere die Betriebskosten entscheidend für die Marktfähigkeit der eVTOLs.

Dieses Kapitel soll eine Einführung in die Kostenbetrachtung von Flugzeugen geben. Hierfür werden zunächst die Definition und Bestandteile der Betriebskosten erläutert. Im Anschluss werden verschiedene Berechnungsmethoden vorgestellt um abschließend ihre Eignung für eVTOLs zu einzuordnen.

Im Luftverkehr gibt es zahlreiche Kostenverursacher, wie Abbildung 2-4 verdeutlicht. Im Rahmen dieser Arbeit liegt der zentrale Fokus jedoch auf den Betriebskosten.

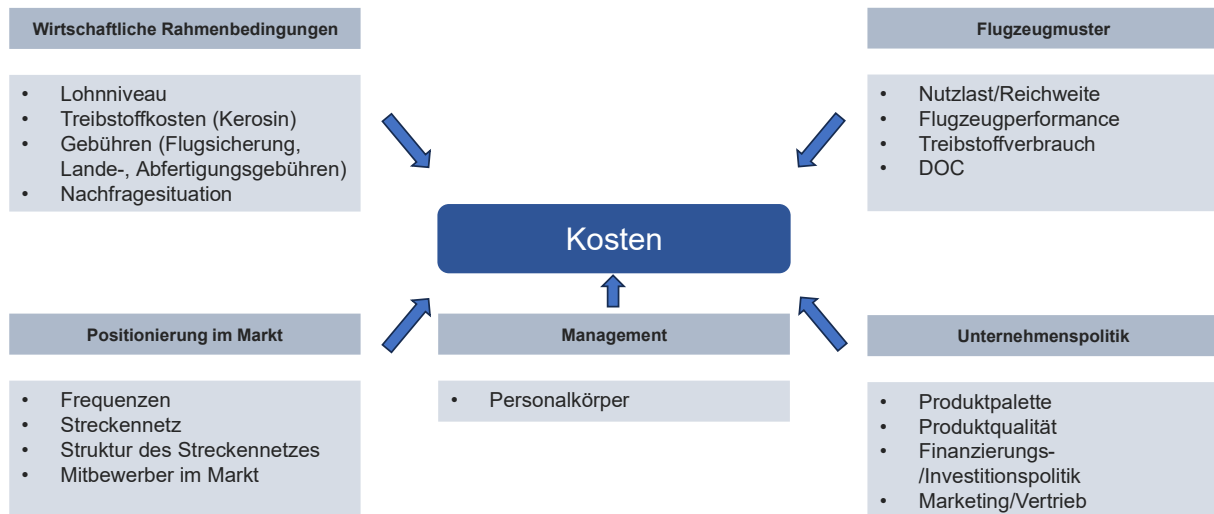


Abbildung 2-4: Kostenverursacher im Luftverkehr nach [15]

2.3.1 Kostenbetrachtung aus der Perspektive des Flugzeugherstellers

Die Kostenbetrachtung aus Sicht des Flugzeugherstellers grenzt sich eindeutig von der des Flugzeugbetreibers ab. Der Flugzeughersteller unterteilt die Kosten als Produktionsbetrieb in fixe Kosten, auch als non recurring costs (NRC) bekannt und variable Kosten, den sogenannten recurring costs (RC) [16].

Die fixen Kosten (NRC) setzen sich hauptsächlich aus den Kosten zusammen, die während der Entwurfs- und Entwicklungsphase entstehen. Darin sind enthalten [16]:

- **Entwicklungs- und Konstruktionskosten:** Aufwendungen für Ingenieurleistungen (intern und extern) sowie für die technische Ausstattung der Entwicklungs- und Konstruktionsabteilungen, einschließlich IT-Infrastruktur und Simulationssoftware.
- **Kosten für experimentelle Abteilungen:** Personalaufwand für Ingenieurleistungen sowie Kosten für die Errichtung und den Betrieb von Versuchsaufbauten, wie beispielsweise Windkanalmodelle und Windkanaltests.
- **Ausgaben für die Zusammenarbeit mit Zulieferern:** Kosten, die durch externe Beschaffung, Abstimmungsprozesse und Lieferkettenmanagement entstehen.
- **Kosten für Analysen in den Bereichen Zuverlässigkeit, Wartung und Zertifizierungsprozesse:** Aufwand für die Untersuchung und Optimierung der Betriebs- und Wartungssicherheit sowie die Einhaltung regulatorischer Anforderungen.
- **Kosten für die Evaluierung und Optimierung von Fertigungsmethoden:** Analyse und Verbesserung der Produktionsprozesse zur Effizienzsteigerung.
- **Aufwendungen für den Vorrichtungsbau:** Kosten für die Entwicklung und Herstellung

von Produktionsvorrichtungen und Fertigungshilfsmitteln.

- **Prototypenbaukosten:** Falls erforderlich, Kosten für die Herstellung und Erprobung eines funktionsfähigen Prototyps.
- **Kosten für Flugversuche und Zertifizierung:** Finanzielle Aufwendungen für Testflüge, Validierungsprozesse und behördliche Zulassungsverfahren.
- **Finanzierungskosten:** Zinsaufwendungen und sonstige Kosten im Zusammenhang mit der Kapitalbeschaffung für die Entwicklung und Produktion.

Die variablen Kosten (RC) hingegen beinhalten Kosten, die durch die Fertigung und dem Product Support aufkommen [16]:

- **Kosten für die Fertigungsbetreuung:** Aufwendungen für die Fehlerbehebung in der Produktion, die Umsetzung kundenspezifischer Anpassungen sowie die Erstellung und Pflege technischer Dokumentationen und Handbücher.
- **Produktionskosten:** Umfassen Lohnkosten, Materialkosten, Werkzeugkosten sowie Aufwendungen für die Qualitätssicherung und die Beschaffung von Kaufteilen
- **Kosten für Flugversuche mit Serienmaschinen und Kundenbetreuung:** Finanzielle Aufwendungen für die Erprobung der Serienflugzeuge sowie für die technische Unterstützung und Betreuung von Kunden während der Inbetriebnahme.
- **Finanzierungskosten:** Zinsaufwendungen und weitere Kosten im Zusammenhang mit der Kapitalbeschaffung für die Serienproduktion und Markteinführung.

Für die wirtschaftliche Bewertung eines Flugzeugentwurfs ist die Betrachtung aus Sicht des Flugzeugherstellers unzureichend. In Modellen zur Kostenbetrachtung aus dieser Perspektive sind keine Flugzeugentwurfparameter enthalten, wie die Auflistung der fixen und variablen Kosten verdeutlicht [16]. Daher ermöglicht diese Betrachtung keine Aussage über die Auswirkungen einer Variation der Entwurfparameter auf die Kosten – ein Aspekt, der für diese Arbeit von zentraler Bedeutung ist. Aus diesem Grund ist die Betrachtung aus der Perspektive des Flugzeugbetreibers erforderlich, die im nachfolgenden Kapitel erläutert wird.

2.3.2 Kostenbetrachtung aus der Perspektive des Flugzeugbetreibers

Die Wirtschaftlichkeit eines Verkehrsflugzeugs ist rein ökonomischer Natur. Das Flugzeug muss ausreichend Einnahmen generieren, die über die Betriebskosten hinausgehen, um profitabel zu werden [10]. Aus der Sicht des Betreibers gibt es zahlreiche Modelle zur Analyse der Kosten. Diese Modelle werden häufig durch Abkürzungen wie LCC, IOC, DOC, COC, COO und TOC klassifiziert [16]. In diesem Kapitel sollen diese Modelle kurz charakterisiert werden.

Cost of Ownership (COO)

Die COO sind die Kosten, die allein durch den Besitz des Flugzeugs anfallen [16]. Hierbei werden die Beschaffung und fixe Betriebskosten betrachtet [17]. Andere lassen hier auch noch variable Betriebskosten einfließen [18].

Life Cycle Costs (LCC)

Dieses Modell findet Anwendung im militärischen Bereich. In den LCC sind die gesamten Kosten während des Produktlebenszyklus enthalten. Aus Sicht des Betreibers sind diese Kosten nebensächlich, da beispielsweise die Kosten für die Entwicklung und Fertigung des Flugzeugs bereits in den Anschaffungskosten stecken. Die Kosten für die Entsorgung des Flugzeugs zum Ende des Lebenszyklus sind für den Betreiber ebenfalls uninteressant, da das Flugzeug zum Ende seiner Nutzung verkauft wird [16].

Direct Operating Costs (DOC)

Die Direct Operating Costs (DOC) umfassen die gesamten Betriebskosten eines Flugzeugs und werden maßgebend von den technischen Daten eines Flugzeugs bestimmt [11]. In der Luftfahrtindustrie haben sich DOC-Methoden gegenüber anderen Ansätzen als am weitesten verbreitet etabliert. Dabei gibt es verschiedene Methoden zur Berechnung der DOC. Diese wurden unter anderem von Flugzeugherstellern, Flugzeugbetreibern und Vereinigungen wie der Air Transport Association of America (ATA) erstellt. Die Direct Operating Costs (DOC) erfassen ausschließlich die flugzeugbezogenen Kosten. Daher dienen sie Flugzeugbetreibern zur Auswahl eines für ihren Einsatzzweck optimalen Luftfahrzeugs und Flugzeugherstellern zur Bewertung neuer Konzepte [16].

Cash Operating Costs (COC)

Die COC werden als DOC ohne Abschreibung definiert. Sie sind auch bekannt als Cash DOC und relevant für Flugzeugbetreiber, die das Fluggerät nicht erwerben, sondern leasen [16].

Indirect Operating Costs (IOC)

Die IOC sollen die DOC ergänzen, indem neben den flugzeugabhängigen Kosten der DOC hier auch passagierabhängige Kosten mit den IOC dazukommen. Diese sind abhängig von dem Management der Fluggesellschaft. Die Bewertung eines Flugzeugentwurfs kann auf die DOC limitiert werden, da die IOC laut ihrer Definition nicht vom Flugzeug abhängig sind, keiner standardisierten statistischen Analyse unterliegen und nur geringfügig mit der Konstruktion des Flugzeugs korrelieren [10], [16]. Die IOC sind ohnehin schon schwer zu quantifizieren, da sie maßgeblich von der Art und Weise, wie die Fluggesellschaft ihre betrieblichen Abläufe

gestaltet, abhängen. Während einige Fluggesellschaften dafür bekannt sind, die IOC auf ein absolutes Minimum zu reduzieren, verfolgen andere Fluggesellschaften bewusst ein „Full-Service“-Modell. In der Regel belaufen sich die jährlichen IOC in einem Bereich von etwa einem Drittel bis zu einem Äquivalent der DOC. Für die IOC kommen weitere Ausgaben wie für Gebäude, Ausrüstung, Entgelte, Kosten für das Kabinenpersonal, Passagierservice, Ticketvertrieb, Marketing und Gepäck- und Frachtabfertigung hinzu [16]. Aufgrund der hohen Variabilität dieser Kosten ist eine zuverlässige Ermittlung der IOC für wirtschaftliche Analysen nur durch direkte Daten von den jeweiligen Fluggesellschaften möglich [10].

Total Operating Costs (TOC)

Die TOC bilden die Summe aus den DOC und den IOC [16].

2.3.3 Wartungskosten nach DOC-Methoden

Wie in den vorherigen Abschnitt beschrieben, können die Kosten aus Sicht des Flugzeugherstellers und des Flugzeugbetreibers betrachtet werden. Aus diesem Grund existieren Berechnungsmethoden für beide Perspektiven. Aus Betreibersicht gibt es wie in Kapitel 2.3.2 zu sehen verschiedene Kostenmodelle. Im Rahmen dieser Arbeit liegt der Fokus dabei auf den DOC. Dabei gibt es verschiedene Methoden zur Berechnung der DOC, welche unter anderem von Flugzeugherstellern, Fluggesellschaften oder Vereinigungen definiert wurden [16]. Abbildung 2-5 zeigt zunächst eine Übersicht für Bewertungsmethoden zu Kosten im Flugzeugbau. Da im Folgenden die DOC, insbesondere die nach der Methode der AEA, betrachtet werden, wurden diese in Abbildung 2-5 hervorgehoben.

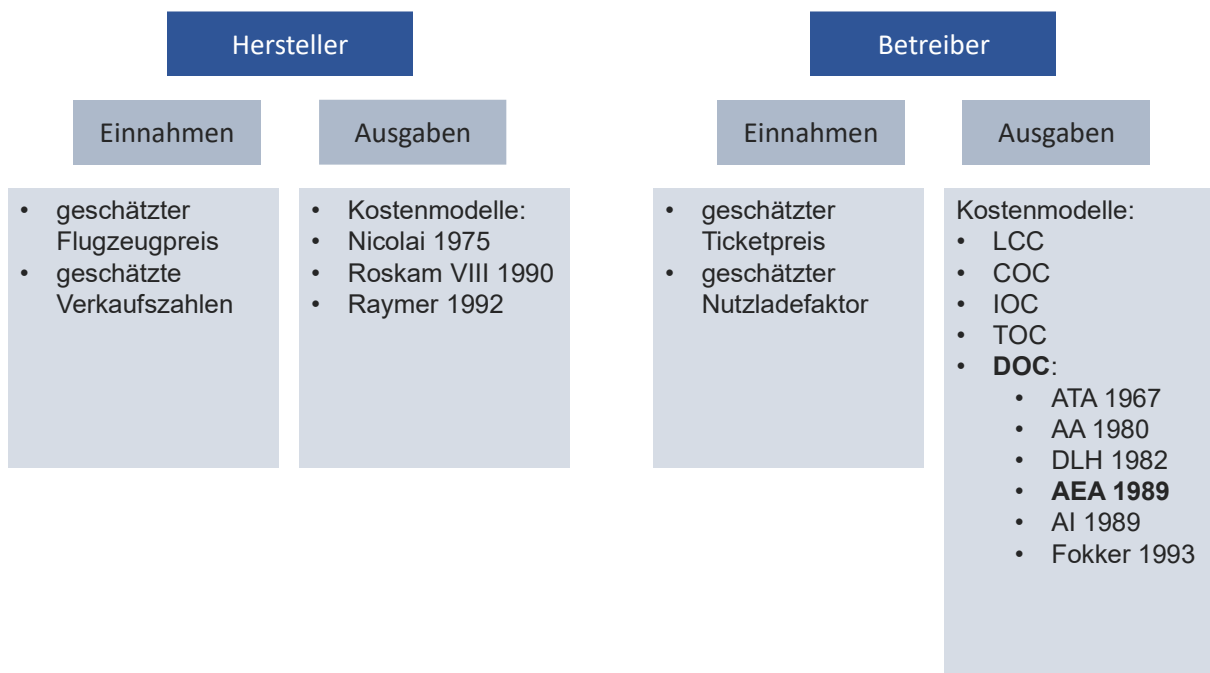


Abbildung 2-5: Übersicht der Bewertungsmethoden im Flugzeugbau nach [16]

2.3.4 Definition von Direct Operating Cost (DOC)

Als Direct Operating Cost werden wie in 2.3.2 beschrieben die direkten Betriebskosten verstanden. Dabei existieren verschiedene DOC-Methoden. Für die Betrachtung dieser Arbeit wird die DOC-Methode nach Association of European Airlines (AEA) genauer beschrieben [19].

Die direkten Betriebskosten eines Flugzeugs setzen sich zusammen aus den Kosten (costs) C [16]:

- Abschreibung (depreciation) C_{DEP}
- Zinsen (interest) C_{INT}
- Versicherung (insurance) C_{INS}
- Kraftstoff (fuel) C_F
- Wartung (maintenance) C_M zusammengesetzt aus
 - Wartung an der Flugzeugzelle (airframe maintenance) $C_{M,AF}$
 - Wartung an den Triebwerken (powerplant maintenance) $C_{M,PP}$
- Flugpersonal (crew) C_C zusammengesetzt aus
 - Flugpersonal im Cockpit (cockpit crew) $C_{C,CO}$
 - Flugpersonal in der Kabine (cabin crew) $C_{C,CA}$
- Gebühren (fees and charges) C_{FEE} zusammengesetzt aus
 - Landegebühren (landing fees) $C_{FEE,LD}$
 - Gebühren für die Flugsicherung (navigation charges) $C_{FEE,NAV}$
 - Abfertigungsgebühren (ground handling charges) $C_{FEE,GND}$

Damit ergeben sich die DOC aus der Summe aller dieser Kosten [16]:

$$C_{DOC} = C_{DEP} + C_{INT} + C_{INS} + C_F + C_M + C_C + C_{FEE} \quad (20)$$

Grundlagen

Für die DOC gibt es wie in Abbildung 2-5 bereits zu sehen verschiedene Methoden. Nach der Beschreibung der einzelnen Kostenelemente bietet Tabelle 1 eine Übersicht, welche Kostenelemente je nach DOC-Methode berücksichtigt werden.

Tabelle 1: Übersicht der berücksichtigten Kostenelemente nach DOC-Methode nach [16]

	ATA 67	AA 1980	DLH 1982	AEA 1989	AI 1989	Fokker 1993
C_{DEP}						
auf Flugzeugpreis	x	x	x	x	x	x
auf Ersatzteile	x		x	x	x	x
C_{INT}			x	x	x	x
C_{INS}	x	x	x	x	x	x
C_F	x	x	x	x	x	x
$C_{M,AF}$	x	x	x	x	x	x
$C_{M,PP}$	x	x	x	x	x	x
$C_{C,CO}$	x	x	x	x	x	x
$C_{C,CA}$		x	x	x	x	x
$C_{FEE,LD}$		x	x	x	x	x
$C_{FEE,NAV}$		x	x	x	x	x
$C_{FEE,GND}$		x	x	x	x	x

Die Darstellung der DOC kann auf verschiedene Weisen erfolgen. So können DOC absolut verstanden werden oder bezogen auf Parameter wie Passagieranzahl, Nutzlast, Flugstrecke bzw. Reichweite und Flugzeit [16].

Grundlagen

Absolut gesehen wird von den jährlichen (annual costs, Index a) Kosten eines Flugzeugs (aircraft, Index a/c) gesprochen. Hierfür gilt [16]:

$$C_{DOC} = C_{a/c,a} \quad (21)$$

Da die DOC hier jährlich betrachtet werden, müssen die einzelnen Kostenelemente C_{DEP} , C_{INT} , C_{INS} , C_F , C_M , C_C , C_{FEE} aufs Jahr gerechnet werden [16].

Bezieht man die DOC auf eine Flugmission (trip, Index t), die durch eine bestimmte Flugstrecke, eine bestimmte Flugzeit und eine bestimmte Blockdauer gekennzeichnet ist, erhält man die Flugmissionskosten (aircraft trip costs) [16]:

$$C_{a/c,t} = \frac{C_{a/c,a}}{n_{t,a}} \quad (22)$$

Hierbei stellt $n_{t,a}$ die Anzahl an Flügen dar, die das Flugzeug bei einer so definierten Flugmission jährlich maximal durchführen kann. Eine weitere Möglichkeit ist es, die DOC in Abhängigkeit von der zurückgelegten Flugstrecke (range, R) darzustellen. Je nachdem, welche Einheit hier für die Flugstrecke verwendet wird, erhält man die Flugkosten-pro-Kilometer (aircraft-mile costs) [16]:

$$C_{a/c,m} = \frac{C_{a/c,t}}{R} = \frac{C_{a/c,a}}{n_{t,a} \cdot R} \quad (23)$$

Des Weiteren können die DOC auf die geflogene Flugstrecke und auf die maximale Anzahl der Passagiere (n_{pax}) bzw. die Anzahl der Sitze (n_s) bezogen werden. Hieraus ergibt sich je nach verwendeter Einheit die Kosten-pro-Passagier-km (seat-mile costs) [16]:

$$C_{s,m} = \frac{C_{a/c,t}}{n_{pax} \cdot R} = \frac{C_{a/c,a}}{n_s \cdot n_{t,a} \cdot R} \quad (24)$$

Bei der Berechnung der Kosten-pro-Passagier-km handelt es sich immer noch um eine Kostenbetrachtung. Da sich die Kosten-pro-Passagier-km jedoch mit einem anderen Flugzeugentwurf, der beispielsweise mehr Sitzplätze aufweist verringern, zeigt diese Kenngröße gewissermaßen auch ein Einnahmepotenzial [16]. Für wirtschaftliche Analysen erfolgt die Darstellung der DOC als seat-mile costs. Diese Darstellung wird häufig zum Vergleich von Flugzeugen herangezogen und dient als Bewertungsgröße für Design- und Konzeptstudien [20]. Dadurch kann mit dieser Kenngröße auch die wirtschaftliche Tragfähigkeit abgeschätzt werden, was Ziel dieser Arbeit ist. Für die Betrachtung der Wirtschaftlichkeit der eVTOLs werden aus diesem Grund in späteren Kapiteln die DOC auf die Flugstrecke und die Passagieranzahl bezogen, um die Kosten-pro-Passagier-km zu erhalten.

2.3.5 Kostenmodell zur Betrachtung der wirtschaftlichen Tragfähigkeit von eVTOLs

Berechnungsmethoden wie jene aus ATA 67 finden typischerweise Anwendung bei konventionellen Verkehrsflugzeugen. Für die wirtschaftliche Bewertung von eVTOLs ist jedoch ein alternativer Ansatz erforderlich, da sich diese hinsichtlich der Infrastruktur und des Betriebsablaufs deutlich unterscheiden. Diese Unterschiede machen die Verwendung eines angepassten Kostenmodells notwendig.

Zur Bewertung der wirtschaftlichen Tragfähigkeit von eVTOLs wird im Rahmen dieser Arbeit ein Kostenmodell herangezogen, das auf dem Ansatz aus [21] basiert. Dabei wird die Berechnung der nivellierten Betriebskosten (Levelized Costs, LC) in Kombination mit dem Kapitalwert oder auch Barwert (Net Present Value, NPV) und der Amortisationsdauer (Investment Return Cycle, IRC) betrachtet.

Levelized Costs (LC)

Zur Berechnung der nivellierten Kosten wird ein Cost of Ownership (COO) Ansatz gewählt, welcher bereits in Abschnitt 2.3.3 erläutert wurde. Hier werden Kosten für die Anschaffung sowie fixe und variable Betriebskosten berücksichtigt. Die Levelized Costs stellen die durchschnittlichen Betriebskosten pro Passagierkilometer über die gesamte Lebensdauer des eVTOLs dar. Dabei werden alle über die Nutzungsdauer anfallenden Kosten auf den heutigen Wert abgezinst. Sie werden berechnet mit [21]:

$$LC = \frac{\sum_{t=1}^{VL} PYC_t}{n_p \cdot Ratio_p \cdot VL \cdot n_D \cdot n_M \cdot R} \quad (25)$$

Dabei bezeichnen [21]:

- LC die nivellierten Betriebskosten in \$/Passagierkilometer
- PYC_t die abgezinsten jährlichen Kosten im Jahr t in \$
- n_p die Anzahl der Passagiere
- $Ratio_p$ den Ladefaktor oder Nutzladefaktor
- VL die Lebensdauer des eVTOLs in Jahren
- n_D die Anzahl der Werktage pro Jahr
- n_M die Anzahl der Flugmissionen pro Jahr
- R die vorhergesehene Reichweite in km

Die Nachfrage der Passagiere mit der Anzahl der verfügbaren Sitze gleichzustellen ist aus wirtschaftlicher Sicht falsch, da es unrealistisch ist, dass Flugzeuge zu jedem Zeitpunkt

vollausgelastet werden. Mit dem Ladefaktor ($Ratio_p$), oder auch Nutzladefaktor wird für dieses Problem eine Korrektur vorgenommen. Der Nutzladefaktor ergibt sich aus dem Verhältnis der nachgefragten – und somit verkauften – zur angebotenen Transportleistung [16].

Net Present Value (NPV)

Der Barwert oder Kapitalwert (NPV) ergibt sich aus der Differenz zwischen den diskontierten Einnahmen (PYR_t) und den diskontierten Kosten (PYC_t) über die gesamte Lebensdauer des eVTOLs und gibt Auskunft über den Profit einer Investition [21]:

$$NPV = \sum_{t=1}^{VL} (PYR_t - PYC_t) \quad (26)$$

Für ein wirtschaftliches Konzept muss dieser Wert positiv ausfallen. Ein negativer Wert weist auf Defizite in der technischen Auslegung oder im Geschäftsmodell hin. Je geringer der Barwert ausfällt, desto besser wird der Flugzeugentwurf bzw. das Geschäftsmodell bewertet [16].

Investment Return Cycle (IRC)

Die Amortisationsdauer ist die Zeitspanne, in der sich die Investition aus den Einnahmen bezahlt, und somit amortisiert hat und ist damit eine wichtige Kenngröße für die Rentabilität einer Investition. Der Investment Return Cycle beschreibt damit die Anzahl an Jahren (t_r), bis die kumulierten Einnahmen die Investitions- und Betriebskosten decken. Er wird berechnet mit [21]:

$$IRC = t_r, \left[\sum_{t=1}^{t_r} (PYR_t - PYC_t) = 0 \right] \quad (27)$$

Je kürzer die Amortisationsdauer, desto geringer ist das wirtschaftliche Risiko, womit die Investorenattraktivität gesteigert wird.

Berechnung der Kosten

Die jährlichen Gesamtkosten im Jahr t ergeben sich aus der Summe aller Teilkosten und werden mit einer bestimmten Zinsrate (DR) diskontiert [21]:

$$PYC_t = \frac{\sum C_{jt}}{(1 + DR)^t} \quad (28)$$

Die Teilkosten aus dem Kostenmodell können Abbildung 2-6 entnommen werden. Es verdeutlicht zudem die Zusammensetzung des Barwerts.

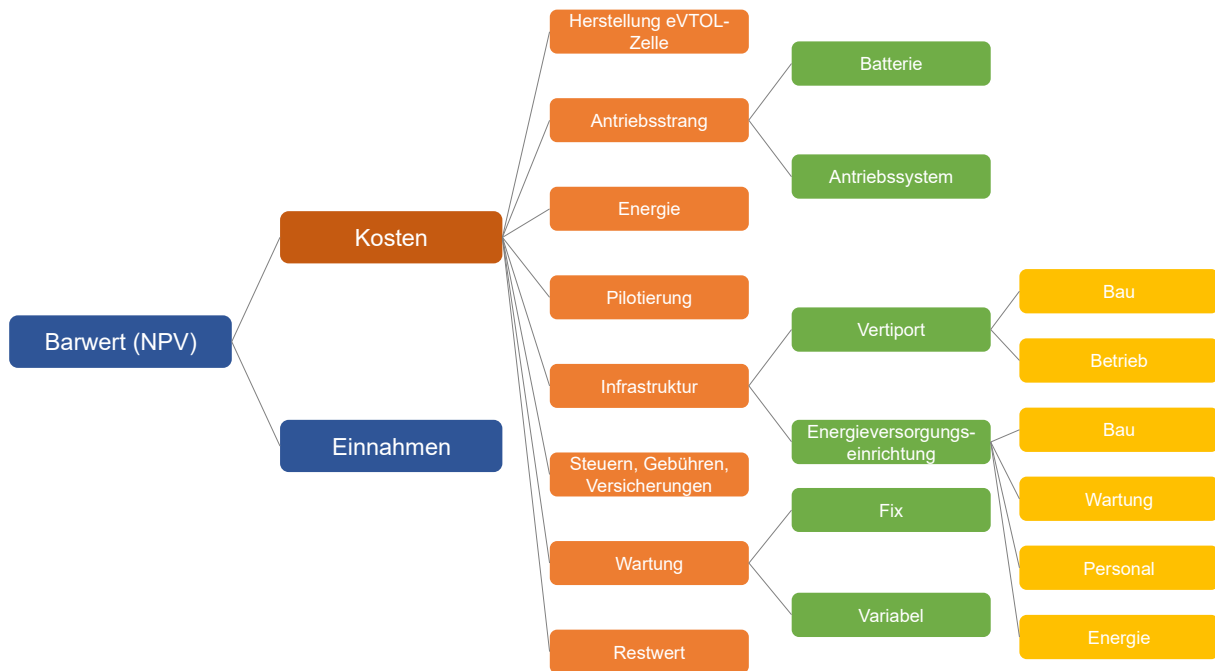


Abbildung 2-6: Schematische Darstellung der Teilkosten und des Barwerts nach [21]

Die Teilkosten setzen sich, wie in Abbildung 2-6 dargestellt, aus den Aufwendungen für die Herstellung der eVTOL-Zelle, den Antriebsstrang, die Energieversorgung, die Pilotierung, die erforderliche Infrastruktur sowie aus Steuern, Gebühren, Versicherungen und der Wartung zusammen. Der Restwert stellt den zu erwartenden Rückfluss am Ende der Lebensdauer der eVTOL-Zelle dar und wird als negativer Kostenfaktor berücksichtigt [21]. Da in diesem Modell der Hersteller der eVTOL-Zelle und des Antriebsstrangs mit als Teil des eVTOL Betreibers betrachtet wird, ergeben sich die Kosten für die eVTOL-Zelle und dem Antriebsstrang unmittelbar aus der Herstellung dieser Komponenten ohne Gemeinkosten von Herstellern zu berücksichtigen.

Die Kosten für die Herstellung der eVTOL-Zelle (VTOL Body Manufacturing, VBM) fallen nur im ersten Jahr an und ergeben sich aus Gleichung (29) [21]:

$$C_{VBM_1} = C_{VBM-F} + C_{VBM-V} \cdot (m_{TO} - m_{0VBM}) \quad (29)$$

Dabei sind:

- C_{VBM_1} die Kosten für die Herstellung der eVTOL-Zelle im ersten Jahr in \$
- C_{VBM-F} die fixen Herstellungskosten der eVTOL-Zelle in \$
- C_{VBM-V} die variablen Herstellungskosten der eVTOL-Zelle in \$
- m_{TO} die maximale Abflugmasse in kg, alternativ in dieser Arbeit auch *MAM* genannt
- m_{0VBM} die eVTOL Strukturmasse in kg

Grundlagen

Die Kosten für den Antriebsstrang (Powertrain, PT) fallen vollständig im Jahr der Herstellung des eVTOLs an, wobei die Kosten für den Energiespeicher im weiteren Betrieb ausschließlich dann erneut anfallen, wenn dessen Lebensdauer erreicht ist und ein Austausch erforderlich wird. Sie werden berechnet mit Gleichung (30) [21]:

$$C_{PT_t} = \sum_k (Q \cdot UC_{ES}) + P_{DS} \cdot UC_{DS} \quad (30)$$

Hierbei sind:

- C_{PT_t} die Kosten für die Herstellung des Antriebsstrangs im Jahr t in \$
- Q die Batteriekapazität in kWh
- UC_{ES} die Kosten für die Batterie in \$/kg oder \$/kWh
- P_{DS} die Leistung des Antriebssystems in kW
- UC_{DS} die Kosten für das Antriebssystem in \$/kW

Die Energiekosten ergeben sich aus dem Strompreis und des jährlichen Energieverbrauchs, welcher sich aus dem Energieverbrauch pro Mission und der Anzahl an Missionen pro Jahr ergibt. Die Energiekosten werden mit Gleichung (31) berechnet [21]:

$$C_{E_t} = p_E \cdot \frac{E_c}{R} \cdot V \cdot YT \cdot (1 + Ratio_{DE}) \quad (31)$$

Dabei bezeichnen:

- C_{E_t} die Energiekosten im Jahr t in \$
- p_E die Stromkosten in \$/kWh
- E_c den Energieverbrauch kWh
- R die Reichweite der Flugmission in km
- V die Reisefluggeschwindigkeit in km/h, alternativ auch v_R in dieser Arbeit genannt
- YT die jährliche Flugzeit in h
- $Ratio_{DE}$ der Anteil der Strecke für leere Rückflüge

Bei den Kosten für den Piloten werden zwei Fälle unterschieden, je nachdem, ob autonom geflogen wird oder nicht. Bei einem nicht autonomen Flug ergeben sich für die Kosten aus einem fixen Pilotengehalt. Für autonomes Fliegen (Unmanned Autonomous Driving, UAD) berechnen sich die Kosten aus den Herstellungskosten für das autonome System im ersten Jahr und dem Gehalt von Piloten bei extremen Wetterbedingungen, bei denen autonomes Fliegen nicht möglich ist. Die Berechnung wird in Gleichung (32) beschrieben [21]:

Grundlagen

$$C_{Plt_t} = \begin{cases} \text{Pilotengehalt, (nicht autonomes Fliegen)} \\ C_{UAD_1} + \beta \cdot \text{Pilotengehalt, (autonomes Fliegen)} \end{cases} \quad (32)$$

Dabei sind:

- C_{Plt_t} die Kosten für Pilotierung im Jahr t in \$
- C_{UAD_1} die Kosten für das autonome System (UAD) in \$
- β der Anteil an Flügen mit extremen Wetterbedingungen

Für den Betrieb der eVTOLs muss die benötigte Infrastruktur geschaffen werden. Diese besteht aus den sogenannten Vertiports, die die Start- und Landefläche der eVTOLs darstellen und den Energieversorgungseinrichtungen, in denen die Batterien aufgeladen oder ausgetauscht werden. Die Kosten für die Infrastruktur setzen sich aus dem Bau der Vertiports und der Energieversorgungseinrichtungen sowie den Betriebskosten für diese Einrichtungen zusammen und werden mit Gleichung (33) berechnet. Die Betriebskosten für die Energieversorgungseinrichtungen ergeben sich aus den Kosten für Wartung, Personal und Energiekosten. Die Kosten für den Bau der Einrichtungen werden über ihre jeweilige Lebensdauer amortisiert [21].

$$C_{I_t} = \frac{C_{VPC_0}}{VPL} + C_{VPO_t} + \frac{C_{RFC_0}}{RFL} + C_{RFO_t} \quad (33)$$

Hier sind:

- C_{I_t} die Kosten für die Infrastruktur im Jahr t in \$
- C_{VPC_0} die Baukosten für den Vertiport in \$
- VPL die Lebensdauer des Vertiports in Jahren
- C_{VPO_t} die Betriebskosten des Vertiports im Jahr t in \$
- C_{RFC_0} die Baukosten für die Energieversorgungseinrichtung in \$
- RFL die Lebensdauer der Energieversorgungseinrichtung in Jahren
- C_{RFO_t} die Betriebskosten der Energieversorgungseinrichtung im Jahr t in \$

Kosten für Steuern, Gebühren und Versicherungen (Taxes, Fees, Insurance, TFI) wurden zusammengefasst und ergeben sich unter der Annahme einer Proportionalität zu den direkten Betriebskosten aus einem festen Anteil der Kosten für den Antriebstrang, der Pilotierung und der Infrastruktur. Sie werden mit Gleichung (34) berechnet [21]:

$$C_{TFI_t} = \text{Ratio}_{TFI} \cdot (C_{E_t} + C_{Plt_t} + C_{I_t}) \quad (34)$$

Grundlagen

Hier bezeichnen:

- C_{TFI_t} die Kosten für Steuern, Gebühren und Versicherungen im Jahr t in \$
- $Ratio_{TFI}$ der Anteil der Kosten für Steuern, Gebühren und Versicherungen von den direkten Betriebskosten

Auch die Kosten für die Wartung (Maintenance, M) wurden aufgeteilt. Zum einen gibt es fixe Wartungskosten, welche proportional zu den Kosten für die Herstellung der eVTOL-Zelle sind und variable Wartungskosten, die mit der Flugzeit zusammenhängen. Berechnet werden die Wartungskosten mit Gleichung (35) [21]:

$$C_{M_t} = Ratio_M \cdot C_{VBM_1} + YT \cdot UC_{VM} \quad (35)$$

Dabei sind:

- C_{M_t} die Wartungskosten im Jahr t in \$
- $Ratio_M$ der Anteil der fixen Wartungskosten an den Herstellungskosten für die eVTOL-Zelle
- UC_{VM} die variablen Wartungskosten in \$/h

Der Restwert (Salvage Value, SV) bezeichnet den geschätzten Wert eines Objekts zum Ende seiner Nutzungsdauer. Dieser wird ebenfalls proportional zu den Herstellungskosten für die eVTOL-Zelle angenommen und fällt im letzten Jahr der Lebensdauer des eVTOLs an, wird als negativer Kostenwert berücksichtigt und wird mit Gleichung (36) berechnet [21] :

$$C_{SV_{VL}} = -Ratio_{SV} \cdot C_{VBM_1} \quad (36)$$

Hier beschreiben:

- $C_{SV_{VL}}$ den Restwert im letzten Jahr der Lebensdauer VL des eVTOLs in \$
- $Ratio_{SV}$ der Anteil des Restwerts an den Herstellungskosten für die eVTOL-Zelle

Berechnung der Einnahmen

Die Einnahmen aus dem eVTOL-Betrieb ergeben sich aus dem Flugpreis sowie der zurückgelegten Strecke in einem Jahr. Dabei wird auch hier für jedes Betriebsjahr über die Lebensdauer des eVTOLs der Jahreswert abgezinst. Die jährlichen Einnahmen im Jahr t werden berechnet mit Gleichung (37) [21]:

$$PYR_t = \frac{p_P \cdot n_P \cdot Ratio_P \cdot R \cdot n_M \cdot n_D}{(1 + DR)^t} \quad (37)$$

Grundlagen

Hier sind:

- PYR_t die abgezinste jährlichen Einnahmen im Jahr t in \$
- p_p der Flugpreis pro Passagier in \$/km
- $Ratio_p$ der Ladefaktor

Charakterisierung des Betriebs

Für die Bewertung der wirtschaftlichen Tragfähigkeit ist eine präzise Abbildung des betrieblichen Ablaufs unerlässlich. Zur Quantifizierung des Gesamtbedarfs einer Flugmission werden nachfolgende Gleichungen genutzt.

Die Gesamtdauer einer Flugmission ergibt sich aus den Zeiten für Boarding und Deboarding der Passagiere, der eigentlichen Flugzeit und der Zeit für die Energieversorgung und wird mit Gleichung (38) berechnet [21]:

$$t_{mission} = t_{boarding} + t_{transport} + t_{disembarking} + t_{supplement} \quad (38)$$

Dabei sind:

- $t_{mission}$ die Dauer einer Flugmission in min
- $t_{boarding}$ die Zeit für das Einsteigen der Passagiere in min
- $t_{disembarking}$ die Zeit für das Aussteigen der Passagiere in min
- $t_{transport}$ die eigentliche Flugzeit in min, zusammengesetzt aus Reise-, Schweb-, Steig- und Sinkflug
- $t_{supplement}$ die Zeit zum Aufladen oder Austauschen der Batterie in min

Die Flugzeit $t_{transport}$ wird dabei mit Gleichung (39) berechnet und ist die Summe der Reiseflugzeit inklusive dem Anteil für leere Rückflüge sowie der Schweb-, Steig- und Sinkflugzeit [21]:

$$t_{transport} = \frac{R + (1 + Ratio_{DE})}{V} + t_{hover} + t_{climb} + t_{descent} \quad (39)$$

Hier bezeichnen:

- t_{hover} die Schwebeflugdauer in min
- t_{climb} die Steigflugdauer in min
- $t_{descent}$ die Sinkflugdauer in min

Grundlagen

Für die Energieversorgung werden zwei Szenarien unterschieden. Zum einen gibt es den Fall, bei dem die Batterie ausgetauscht wird und zum anderen den üblichen Fall, bei dem die Batterie aufgeladen wird. Demnach wird $t_{supplement}$ berechnet mit [21]:

$$t_{supplement} = \begin{cases} t_{swapping}, & (\text{Austausch der Batterie}) \\ \frac{Q}{P_{charging}} \cdot 60, & (\text{Aufladen der Batterie}) \end{cases} \quad (40)$$

Hier beschreiben:

- $t_{swapping}$ die Zeit zum Austauschen der Batterie in min
- Q die Batteriekapazität in kWh
- $P_{charging}$ die Ladeleistung in kW

Die Anzahl möglicher Flugmissionen, die maßgeblich von den maximal verfügbaren jährlichen Flugstunden (YT) abhängt, erlaubt Rückschlüsse auf die operationelle und damit auch wirtschaftliche Effizienz eines eVTOLs und wird gemäß Gleichung (41) berechnet [21]:

$$n_M = \frac{YT}{t_{transport} \cdot n_D} \quad (41)$$

3. Methodik und Modellentwicklung

Im folgenden Kapitel werden die Anforderungen an das Berechnungsmodell definiert und das bereits existierende Berechnungsverfahren analysiert. Hierbei werden Schwächen und Erweiterungspotenziale identifiziert, um damit das eigene Berechnungsmodell aufzubauen.

3.1 Anforderungen an das Modell

Um zu überprüfen, ob die Anforderungen durch das entwickelte Produkt, in diesem Fall das Berechnungsmodell, erfüllt werden, müssen diese vorher präzise und technisch formuliert werden [22]. Die Anforderungen können Tabelle 2 entnommen werden.

Tabelle 2: Anforderungen an das Berechnungsmodell

ID	Priorität	Anforderungsbeschreibung	Klasse	Status
B.1	hoch	Das Modell muss den Energiebedarf von eVTOLs basierend auf Designparametern, die vorher eingegeben werden, berechnen können.	Berechnung	
B.2	hoch	Das Modell muss die Reichweite von eVTOLs basierend auf Designparametern, die vorher eingegeben werden, berechnen können.	Berechnung	
B.3	hoch	Alle Designparameter müssen in ihrem gültigen Bereich flexibel anpassbar sein	Berechnung	
B.4	hoch	Es soll möglich sein, Reichweite und Energieverbrauch für mehrere eVTOLs zu berechnen und diese miteinander zu vergleichen.	Berechnung	
B.5	hoch	Das Modell muss die Gleitzahl anhand der Geometrieparameter berechnen können.	Berechnung	
O.1	hoch	Die vorab definierter Parametergrenzen und Intervalle müssen bei der Optimierung eingehalten werden.	Optimierung	
O.2	hoch	Kombinationen aus Parametern, die nicht im vorgegeben Bereich liegen, müssen bei der Optimierung verworfen werden.	Optimierung	
O.3	hoch	Die harten Randbedingungen müssen bei der Optimierung eingehalten werden.	Optimierung	
O.4	hoch	Die Optimierung muss unter Einhaltung technischer Randbedingungen erfolgen.	Optimierung	
O.5	hoch	Die Optimierung soll die beste Konfiguration für Reichweite und Energieverbrauch getrennt ermitteln können.	Optimierung	
O.6	niedrig	Die Anzahl der verworfenen Kombinationen soll am Ende angezeigt werden.	Optimierung	

Methodik und Modellentwicklung

O.7	hoch	Im Modell soll es die Möglichkeit geben, einen Ausgabeparameter durch maximieren oder minimieren zu verbessern. Hierzu soll es eine Option in der GUI geben. Die Parameter müssen dabei in einem gültigen Bereich angepasst werden.	Optimierung	
O.8	hoch	Die Optimierung soll als Grid Search mit konfigurierbaren Parameterbereichen durchgeführt werden.	Optimierung	
O.9	hoch	Es sollen Mindest- und Maximalgrenzen für die Optimierungsparameter definiert werden können.	Optimierung	
G.1	hoch	Das Modell muss über eine grafische Benutzeroberfläche (GUI) verfügen.	GUI	
G.2	hoch	Die Benutzeroberfläche soll über Eingabefelder und Regler (optional) verfügen	GUI	
G.3	mittel	Das Berechnungsmodell soll über Hinweise zur Bedienung der Benutzeroberfläche verfügen.	GUI	
G.4	niedrig	Es soll möglich sein, der Berechnung einen Namen zu geben (z.B. Namen des eVTOLs).	GUI	
G.9	hoch	Bei keiner oder falscher Eingabe der Input-Parameter muss eine Fehlermeldung erscheinen.	GUI	
G.11	mittel	Die GUI muss beim Start ein Hauptmenü mit Auswahloptionen für Berechnung und Optimierung anzeigen.	GUI	
G.15	hoch	Die GUI muss die optimierten Ergebnisse anzeigen (beste Konfiguration).	GUI	
G.16	hoch	In der GUI sollen Eingabefehler (z. B. fehlende Felder, falsche Formate) erkannt und gemeldet werden.	GUI	
G.17	hoch	Es soll möglich sein, benutzerdefinierte Optimierungsbereiche in der GUI einzugeben.	GUI	
G.19	niedrig	Die Eingabefelder bei den Parametergrenzen und Intervallen sollen mit Standardwerten vorbelegt sein (optional).	GUI	
G.18	mittel	Nach der Optimierung soll ein Zurücknavigieren ins Hauptmenü möglich sein.	GUI	
S.2	hoch	Das Programm soll ohne Installation von zusätzlicher Entwicklungsumgebung lauffähig sein.	System	
S.1	mittel	Das Modell soll die Möglichkeit haben, die Ergebnisse zu exportieren (z.B. als PDF, xls, csv).	System	
K.1	hoch	Das Modell muss die wirtschaftliche Tragfähigkeit auf Basis der Kenngrößen Kapitalwert (NPV), Amortisationsdauer (IRC) und nivellierte Kosten (LC) berechnen können.	Kostenmodell	

K.2	mittel	Die Ergebnisse des Kostenmodells sollen exportierbar sein (z. B. als Excel-Datei) und eine jährliche Aufschlüsselung der einzelnen Kostenarten enthalten.	Kostenmodell	
K.3	hoch	Das Modell muss die Auswirkungen technischer Entwurfparameter (z. B. Abflugmasse, Batteriegröße, Reichweite) auf die wirtschaftlichen Kennzahlen abbilden können.	Kostenmodell	
K.4	hoch	Die verwendeten Kostendaten (z. B. Stromkosten, Herstellungskosten) sollen für verschiedene Szenarien konfigurierbar sein.	Kostenmodell	
K.5	hoch	Es sollen unterschiedliche Betriebsparameter einstellbar sein, um deren Einfluss auf die Wirtschaftlichkeit zu analysieren.	Kostenmodell	

3.2 Analyse des bestehenden Berechnungsverfahrens

In dem Berechnungsansatz aus dem Artikel „Die Flugleistung des Lilium-„Jets“ und eVTOL’s im Allgemeinen – eine Konzeptberechnung“ ohne bekannten Verfasser wird der Ansatz genutzt um speziell die Kenngrößen wie Reichweite und Energieverbrauch des Lilium-Jets zu berechnen. Die qualitativen Schlussfolgerungen seien aber unabhängig vom Lilium-Jet für alle eVTOLs allgemein gültig.

Für den Berechnungsansatz wird die Flugmission in die drei Phasen Schweben, Reiseflug und erneut Schweben unterteilt. Dies umfasst den vertikalen Start, den Reiseflug und die vertikale Landung. Der Verfasser bezeichnet diese Unterteilung als das 3P-Modell, welches schematisch in Abbildung 3-1 dargestellt ist. Auf Beschleunigungs- und Verzögerungsphasen sowie Steig- und Sinkflugphasen wird verzichtet. Dies wird damit begründet, dass diese zum einen nur einen geringen Teil der Gesamtmissionszeit ausmachen und zum anderen, dass der erhöhte Energieverbrauch in der Steigflugphase durch den verringerten Energieverbrauch in der Sinkflugphase kompensiert wird. Für die Energiebilanz in der späteren Berechnung werden neben den Triebwerken keine weiteren elektrischen Verbraucher berücksichtigt [4].

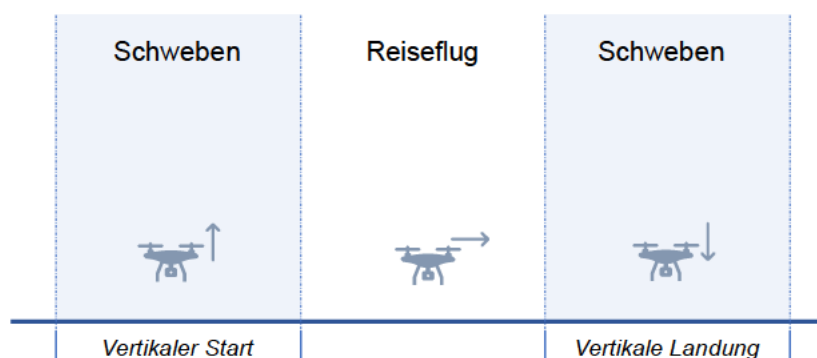


Abbildung 3-1: Schematische Darstellung des 3P-Modells

Der Energiebedarf wird im 3P-Modell in zwei Schritten berechnet. Zunächst wird der Energiebedarf für die Schwebeflugphasen berechnet. Dieser ergibt sich aus der erforderlichen Batterieleistung in der Schwebeflugphase ($P_{S,A}$) multipliziert mit der Gesamtdauer des Schwebeflugs (t_S) [4]:

$$E_S = P_{S,A} \cdot t_S \quad (42)$$

Für die erforderlichen Batterieleistung muss jedoch noch der Gesamtwirkungsgrad des Antriebsstrangs im Schwebeflug berücksichtigt werden [4]:

$$P_{S,A} = P_{S,erf} \cdot \eta_S^{-1} \quad (43)$$

Bis hier hin wurden grundlegende physikalische Zusammenhänge verwendet. Für die Berechnung der erforderlichen Antriebsnettoleistung zur Bereitstellung des Schubs in der Schwebephase wird eine Formel aus der Fachliteratur [13] entnommen. Diese wird insbesondere bei der Berechnung mit Hubschrauberrotoren verwendet [4]:

$$P_{S,erf} = T_S \cdot \sqrt{\frac{T_S}{2 \cdot \rho \cdot A_r}} \quad (44)$$

Die Herleitung der Gleichung (44) wurde bereits im theoretischen Rahmen der Arbeit erläutert. Da der erforderliche Schub die Gewichtskraft des eVTOLs ausgleicht, sind die Beiträge dieser beiden Größen gleich. Der Schub ergibt sich somit aus der maximalen Abflugmasse und der Erdbeschleunigung [4]:

$$T_S = MAM \cdot g \quad (45)$$

Für die Berechnung der Gesamtrorfläche werden neben der Anzahl der Triebwerke (n_T) die Propellerradien (r_P) und Nabenradien (r_N) verwendet [4]:

$$A_r = n_T \cdot \pi \cdot (r_P^2 - r_N^2) \quad (46)$$

In der Leistungsberechnung des 3P-Modells wird der im Kapitel 2.2.2 erläuterte Bodeneffekt nicht berücksichtigt. Dies wird damit begründet, dass die Schwebephasen nicht nur den Start und die Landung beinhalten, sondern auch die Be- und Entschleunigungsphasen, für die der Bodeneffekt nicht vorausgesetzt werden sollte. Bei Einbindung des Bodeneffektes würde sich die erforderliche Leistung im Schwebeflug um 30% verringern [4].

Im zweiten Schritt wird der Energiebedarf im Reiseflug berechnet. Hier werden für den erforderlichen Energiebedarf im Reiseflug und der erforderlichen Batterieleistung im Reiseflug auch wieder die Gleichungen (42) und (43) mit kleiner Anpassung der Indizes und Formelzeichen verwendet [4]:

$$E_R = P_{R,A} \cdot t_R \quad (47)$$

$$P_{R,A} = P_{R,erf} \cdot \eta_R^{-1} \quad (48)$$

Durch die Gleitleistung im Reiseflug sinkt der erforderliche Schub in dieser Flugphase. Der erforderliche Schub in dieser Phase ist somit deutlich geringer als in der Schwebeflugphase. Die erforderliche Leistung wird dabei mit Gleichung (49) berechnet [4]:

$$P_{R,erf} = T_R \cdot v_R = MAM \cdot g \cdot GZ_R^{-1} \cdot v_R \quad (49)$$

Gleichung (49) wird aus der Berechnung der Leistung für einen ebenen, nicht beschleunigten Flug aus der Fachliteratur hergeleitet [23]. Aus dem Zusammenhang

$$Leistung(P) = \frac{Arbeit}{Zeit} = \frac{Kraft \cdot Strecke}{Zeit} = Kraft (F) \cdot \frac{Strecke (d)}{Zeit (t)} \quad (50)$$

ergeben sich Gleichung (51) und somit Gleichung (52):

$$P = F \cdot \left(\frac{d}{t_2 - t_1} \right) = F \cdot v \quad (51)$$

$$P_{R,erf} = T_R \cdot v_R \quad (52)$$

Betrachtet man nun ein Flugzeug im Reiseflug, so besteht zwischen erforderlichem Schub T_R und dem Widerstand D folgende Beziehung, um die Reisefluggeschwindigkeit zu halten:

$$T_R = D \quad (53)$$

Dabei kann der Widerstand bei ebenem Flug substituiert werden durch:

$$D = \frac{W}{GZ_R} \quad (54)$$

Nach Einsetzen in Gleichung (53), bzw. (52) und der Ausführung von W mit $W = MAM \cdot g$ erhält man Gleichung (55) [4]:

$$P_{R,erf} = \frac{W}{GZ_R} \cdot v_R = MAM \cdot g \cdot GZ_R^{-1} \cdot v_R \quad (55)$$

Die Energie, die von der Batterie bereitgestellt wird (E_A), wird in Gleichung (56) definiert. Dabei ist e_A die Energiedichte und AM die Gesamtmasse der Batterieeinheit [4].

$$E_A = e_A \cdot AM \quad (56)$$

Bilanziert man nun die verfügbare Energie und den Energiebedarf in den verschiedenen Flugphasen, so erhält man die Energiegleichung [4]:

$$R = v_R \cdot t_R = GZ_R \cdot \eta_R \cdot \left(\frac{\mu_A \cdot e_A}{g} - \frac{t_S}{\eta_S} \cdot \sqrt{\frac{MAM \cdot g}{2 \cdot \rho \cdot A_r}} \right) \quad (57)$$

Sofern es keine Angaben zur Gleitzahl gibt, kann diese auch berechnet werden. Die Gleitzahl im Reiseflug wurde im Berechnungsansatz aus der Fachliteratur wie folgt berechnet [4]:

$$GZ_R = \frac{c_{A,R}^2}{c_{D0} + c_{Di,R}} \quad (58)$$

Dabei ist $c_{A,R}$ der Auftriebskoeffizient bei Reisefluggeschwindigkeit und $c_{Di,R}$ der Koeffizient für den auftriebsinduzierten Widerstand. Dieser wird mit c_{D0} um die Beiwerte weiterer parasitärer Widerstände erweitert [4].

$$c_{A,R} = \frac{2 \cdot MAM \cdot g}{\rho \cdot S_w \cdot v_R^2} \quad (59)$$

Der Auftriebskoeffizient im Reiseflug ($c_{A,R}$) wird mit der maximalen Abflugmasse (MAM), der Erdbeschleunigung (g), der Luftdichte (ρ), der Flügelfläche (S_w) und der Reisefluggeschwindigkeit (v_R) berechnet. Die Flügelfläche ergibt sich dabei aus der mittleren Flügeltiefe (c) und der Spannweite (b) [4]:

$$S_w = c \cdot b \quad (60)$$

Neben dem Oswald-Faktor (e) und der Flügelstreckung (AR) fließt auch der Auftriebskoeffizient im Reiseflug in die Berechnung des auftriebsinduzierten Widerstandsbeiwert ein [4]:

$$c_{Di,R} = \frac{C_{A,R}^2}{\pi \cdot e \cdot AR} \quad (61)$$

Für die Flügelstreckung gilt dabei [4]:

$$AR = \frac{b^2}{S_w} \quad (62)$$

Der Beiwert (c_{D0}) ergibt sich aus der Summe der Oberflächenreibung des Flügels ($c_{D0,OR}$), der Reibung und dem Formwiderstand des Rumpfes ($c_{D0,R}$), der Reibung der Triebwerksgehäuse ($c_{D0,T}$) und der Reibung des Leitwerks ($c_{D0,LW}$) [4]:

$$c_{D0} = c_{D0,OR} + c_{D0,R} + c_{D0,T} + c_{D0,LW} \quad (63)$$

Um eine quantifizierbare Kenngröße zu erhalten, die den Vergleich der Transportleistung ermöglicht, kann der Pro-Kopf-Verbrauch herangezogen werden. Hierfür muss lediglich die mit Gleichung (57) berechnete Reichweite gegenüber der bereitgestellten Energie und der Passagieranzahl bilanziert werden [4]:

$$PKV = \frac{e_A \cdot AM}{PAX \cdot R} \quad (64)$$

Aus den Gleichungen für die Reichweite sowie Pro-Kopf-Verbrauch lassen sich einige qualitative Aussagen ableiten, die für die spätere Optimierung hilfreich sind.

Ein Blick auf die Berechnung der Reichweite zeigt, dass sie linear ansteigt, wenn die Gleitzahl oder der Wirkungsgrad im Reiseflug linear gesteigert werden. Wird die Energiedichte, der Wirkungsgrad im Schwebeflug oder das Batteriemassenverhältnis linear gesteigert, steigt die Reichweite linear proportional [4].

Der zweite Term im Klammerausdruck der Reichweitengleichung beschreibt den Einfluss der Schwebephase auf die Reichweite. Durch lineare Steigerung des Rotor-Außendurchmessers und folglich quadratischer Steigerung der Gesamttrotorfläche, nimmt auch die Reichweite zu. Wird die Dauer der Schwebeflugphase erhöht, so sinkt die Reichweite. Zudem sinkt sie auch, wenn die maximale Abflugmasse erhöht wird, ohne dabei auch die Batteriemasse zu erhöhen [4].

Schafft man es die Gleitzeit und den Wirkungsgrad in der Reiseflugphase bei verschiedenen Reisefluggeschwindigkeiten konstant zu halten, so ist die Reichweite unabhängig von der Reisefluggeschwindigkeit. Eine Erhöhung oder Verringerung dieser würde somit nur die Reisezeit beeinflussen [4]. Die Auswirkungen der Parameter werden in Tabelle 3 dargestellt.

Tabelle 3: Auswirkungen einzelner Parameter auf die Reichweite

Parameter	Veränderung des Parameters	Auswirkung auf die Reichweite	Bedingung
Gleitzahl (GZ_R)	↑	↑	-
Wirkungsgrad im Reiseflug (η_R)	↑	↑	-
Wirkungsgrad im Schwebeflug (η_S)	↑	↑	

Maximale Abflugmasse (MAM)	↑	↓	Masse Batterie konstant
Schwebeflugdauer (t_S)	↑	↓	Masse Batterie konstant
Gesamtrotorfläche (A_r)	↑	↑	-
Batteriemassenverhältnis (μ_A)	↑	↑	-
Energiedichte der Batterie (e_A)	↑	↑	-
Reisefluggeschwindigkeit (v_R)	↑/↓	=	GZ_R und η_R konstant

Bei Betrachtung von Gleichung (64) für den Pro-Kopf-Verbrauch wird ersichtlich, dass dieser sinkt, wenn die Gleitzahl, der Wirkungsgrad im Reiseflug oder das Passagiermassenverhältnis erhöht werden. Da der Pro-Kopf-Verbrauch nicht auf null reduziert werden kann, nähert er sich asymptotisch einem Grenzwert an. Zukünftige Reduktionen des Verbrauchs beziehen sich daher auf dieses Verhalten. Der Pro-Kopf-Verbrauch wird auch reduziert, wenn das Batteriemassenverhältnis oder die Energiedichte der Batterie bei konstant bleibender, maximaler Abflugmasse erhöht wird [4].

Erhöht wird der Pro-Kopf-Verbrauch durch eine Erhöhung der maximalen Abflugmasse oder Verringerung der Gesamtrotorfläche oder des Wirkungsgrads im Schwebeflug bei konstanter Gleitzahl [4]. In Tabelle 4 werden die Zusammenhänge der Parameter auch für den Pro-Kopf-Verbrauch dargestellt.

Tabelle 4: Auswirkungen einzelner Parameter auf den Pro-Kopf-Verbrauch

Parameter	Veränderung des Parameters	Auswirkung auf den Pro-Kopf-Verbrauch	Bedingung
Gleitzahl (GZ_R)	↑	↓	-
Wirkungsgrad im Reiseflug (η_R)	↑	↓	-

Wirkungsgrad im Schwebeflug (η_s)	↓	↑	GZ_R konstant
Maximale Abflugmasse (MAM)	↑	↑	GZ_R konstant
Gesamtrotorfläche (A_r)	↓	↑	GZ_R konstant
Passagiermassenverhältnis (μ_P)	↑	↓	-
Batteriemassenverhältnis (μ_A)	↑	↓	-
Energiedichte der Batterie (e_A)	↑	↓	-

Die wichtigste Erkenntnis aus dieser Betrachtung ist jedoch, dass Reichweite und Energieverbrauch miteinander konkurrieren. Wenn die Schwebeflugphase bei konstanter Gleitzahl, Energiedichte und Wirkungsgrad im Reiseflug asymptotisch gegen null gehen würde, geht hervor dass die Reichweite maßgebend von dem Batteriemassenverhältnis vorangetrieben wird, während der wesentliche Treiber für den Energieverbrauch das Passagiermassenverhältnis wäre. Unter Voraussetzung einer gleichbleibenden Restmasse des Flugzeugs stehen diese beiden also in direkter Wechselwirkung zueinander. Eine erhöhte Reichweite muss durch einen ineffizienteren Verbrauch „erkauft“ werden [4].

Um einen Ausblick über die Sensitivitäten der Parameter und Optimierungspotenzial im Allgemeinen zu erhalten, wurden vom Verfasser Variationsrechnungen durchgeführt, welche in Abbildung 3-2 zu sehen sind. Zu erkennen sind neben den Eingangsparametern einige berechnete Ausgangsgrößen, darunter auch die Reichweite und der Pro-Kopf-Verbrauch, welche für diese Arbeit maßgebend sind. Die Variation 0 gibt die Ausgangslage wieder. Sie basiert auf Annahmen und Angaben zum Lilium Jet, die öffentlich zugänglich sind oder abgeschätzt wurden. Auf diese Parameter bauen alle weiteren Variation auf [4]. Es muss jedoch erwähnt werden, dass der in Kapitel 4 betrachtete Lilium Jet sich auf das aktuellste Konzept des Unternehmens bezieht, während hier ein älterer Prototyp analysiert wurde.

In den Variationen 1 bis 7 wurde jeweils nur ein einzelner Parameter verändert, um den Einfluss auf die Ausgangsgrößen zu analysieren. Im Rahmen dieser Arbeit spielen jedoch die Reichweite und der Energiebedarf, bzw. der Pro-Kopf-Verbrauch eine übergeordnete Rolle.

Die maximale Schwebeflughöhe hat ebenfalls einen Einfluss auf die Gesamtbetrachtung, spielt aber im Vergleich zur Reichweite und des Pro-Kopf-Verbrauchs eine untergeordnete Rolle. In den ersten beiden Variationen wurde zunächst das Batteriemassenverhältnis und dann das Passagiermassenverhältnis um denselben Betrag erhöht. Folglich hat sich bei Variation 1 die Reichweite mehr als verdoppelt, während der Pro-Kopf-Verbrauch halbiert wurde. In Variation 2 wurde hingegen das Passagiermassenverhältnis erhöht. Die Reichweite blieb unverändert und der Pro-Kopf-Verbrauch ist nur um etwa 15% des Ausgangswertes gesunken. Die Analyse der ersten beiden Variationen zeigt, dass zur Optimierung von Reichweite und Pro-Kopf-Verbrauch bevorzugt mehr Masse für die Batterie vorgesehen werden sollte, sofern weitere Masse durch Leichtbau reduziert werden kann [4].

Bei der Variation 3 wurde die Gesamtrotorfläche verdoppelt. Am Beispiel des Lilium-Jets geschah dies durch eine Vergrößerung des Außendurchmessers um 4cm. Daraus resultierte eine Verdreifachung der Reichweite bei in etwa nur einem Viertel an Pro-Kopf-Verbrauch. Des Weiteren erhöht sich dadurch die maximale Schwebedauer um etwa 50% der Ausgangssituation [4].

Variation 4 verdeutlicht, dass eine Erhöhung der Gleitzahl um 50 % zu einer proportionalen Steigerung der Reichweite um 50 % führt, während der Pro-Kopf-Verbrauch um 30 % reduziert wird [4].

In den Variationen 5 und 6 liegt das Augenmerk auf die Wirkungsgrade. Dabei wird in Variation 5 der Wirkungsgrad im Reiseflug und in Variation 6 der Wirkungsgrad im Schwebeflug um jeweils 10% erhöht. Die resultierende Veränderung der Ausgangsgrößen verdeutlicht eindrucksvoll, dass eine Erhöhung des Wirkungsgrades im Schwebeflug einen wesentlich stärkeren Einfluss auf die Verbesserung der Reichweite und des Pro-Kopf-Verbrauchs hat als eine Steigerung des Wirkungsgrades im Reiseflug [4].

Variation 7 basiert auf der Annahme, dass das Triebwerk nicht mehr explizit auf die Reisefluggeschwindigkeit optimiert ist. Stattdessen liegt die Auslegungsgeschwindigkeit nun zwischen null und der Reisefluggeschwindigkeit, sodass sich annähernd gleiche Wirkungsgrade für den Schwebeflug und den Reiseflug ergeben. Infolgedessen steigt die Reichweite auf das Dreifache, während der Pro-Kopf-Verbrauch auf etwa ein Viertel des ursprünglichen Werts sinkt. Variation 7 soll aufzeigen, dass eine Entwicklung in diese Richtung mit den errechneten Erwartungen gerechtfertigt werden könnte [4].

Variation 8 beinhaltet alle Veränderungen aus den bisherigen Variationen und zeigt somit die bestmöglichen Ergebnisse für Reichweite und Pro-Kopf-Verbrauch mit dem Konzept des Lilium Jets. Um den Einfluss der Schwebedauer zu analysieren, wurde diese bei Variation 9 um

eine Minute verlängert. Im direkten Vergleich zu Variation 8 verringert sich die Reichweite um etwa ein Drittel, während der Pro-Kopf-Verbrauch um rund 50 % ansteigt. Diese Variation spielt hinsichtlich des Optimierungsbedarfs eine eher untergeordnete Rolle. Vielmehr soll sie veranschaulichen, wie sich Reichweite und Energiebedarf verändern, wenn die Schwebedauer verlängert wird – eine Anforderung, die seitens der Luftfahrtbehörden bei einer Schwebedauer von nur einer Minute durchaus gestellt werden könnte [4].

Variation 10 stellt die Ausgangsgrößen für den Fall einer Verdopplung der Energiedichte der Batterie dar. Zum Zeitpunkt der Veröffentlichung dieses Papers waren dem Autor Batterien, die diesem Wert nahe kommen, nicht bekannt [4].

Für Variation 11 wurde die maximale Abflugmasse verdoppelt. Die restlichen Parameter sind identisch mit denen der Variation 8. Im Vergleich zur Variation 8 nimmt die Reichweite um etwa 15% ab und der Pro-Kopf-Verbrauch erhöht sich leicht. Hiermit soll verdeutlicht werden, dass eVTOLs im Allgemeinen von einer geringeren Masse profitieren. Variation 12 konzentriert sich hauptsächlich auf die Konzeptkritik am Lilium Jet und spielt für die Identifikation von Optimierungspotenzial eine nebensächliche Rolle. Ziel dieser Variation war es, aufzuzeigen, welche Annahmen erforderlich wären, um die von Lilium angegebenen Leistungswerte zu erreichen [4].

Var-#	t.S [s]	e.A [Wh/kg]	MAM [kg]	$\mu.A$ [-]	$\mu.P$ [-]	A.r [m ²]	GZ.R [-]	$\eta.R$ [-]	$\eta.S$ [-]	t.Smax [s]	R [km]	t.R [min]	PKV [kWh/100km]	RM [kg]
0	60	220	1500	0,33	0,33	1	10	0,6	0,2	67,7	18,4	3,7	119,7	320
1	60	220	1500	0,4	0,33	1	10	0,6	0,2	81,3	50,6	10,1	52,2	220
2	60	220	1500	0,33	0,4	1	10	0,6	0,2	67,7	18,4	3,7	99,8	220
3	60	220	1500	0,33	0,33	2	10	0,6	0,2	99,1	63,7	12,7	34,5	320
4	60	220	1500	0,33	0,33	1	15	0,6	0,2	67,7	27,6	5,5	79,8	320
5	60	220	1500	0,33	0,33	1	10	0,7	0,2	67,7	21,4	4,3	102,6	320
6	60	220	1500	0,33	0,33	1	10	0,6	0,3	101,6	65,9	13,2	33,4	320
7	60	220	1500	0,33	0,33	1	10	0,4	0,4	135,5	59,8	12,0	36,8	320
8	60	220	1500	0,4	0,33	2	15	0,4	0,4	237,9	144,5	28,9	18,3	220
9	120	220	1500	0,4	0,33	2	15	0,4	0,4	237,9	95,8	19,2	27,6	220
10	120	440	1500	0,4	0,33	2	15	0,4	0,4	475,9	289,0	57,8	18,3	220
11	60	220	3000	0,4	0,33	2	15	0,4	0,4	168,2	124,5	24,9	21,2	620
12	60	220	400	0,4	0,25	16	15	0,7	0,4	1290,9	322,7	64,5	10,9	180

Abbildung 3-2: Variationsrechnungen am Beispiel des Lilium-Jets [4]

3.2.1 Identifikation von Schwächen und Verbesserungspotenzialen

In Kapitel 3.2 wurde der Berechnungsansatz analysiert. Das grundlegende Verfahren zur Berechnung der Reichweite und des Energiebedarfs wurde dort erläutert und unzureichend erklärte Gleichungen wurden mit dem theoretischen Rahmen dieser Arbeit ergänzt, wodurch sich der Berechnungsansatz als akzeptabel erwiesen hat. Der Ansatz hat jedoch einige Einschränkungen, die im Folgenden behandelt werden.

Eine Einschränkung der Berechnungsmethode aus der Fachliteratur ist die Annahme eines konstanten Luftwiderstands. Obwohl diese Vereinfachung nur eine geringe Abweichung verursacht, da viele eVTOLs in ähnlichen Flughöhen operieren, kann sie durch eine differenzierte Berücksichtigung der Flughöhe leicht korrigiert werden. Gleichzeitig muss erwähnt werden, dass die meisten eVTOLs in ähnlichen Flughöhen operieren. Für den Vergleich ausgewählter Konzepte dürfte dieser Aspekt keine signifikante Einflussnahme auf die Ergebnisse haben. Dennoch könnte sich die Flughöhe mit der Weiterentwicklung der eVTOL-Konzepte zunehmend differenzieren. Eine Berücksichtigung dieser Variabilität würde das Modell in diesem Kontext zukunftssicherer und anpassungsfähiger gestalten.

Die Vernachlässigung des Bodeneffekts ist nach Recherche der Fachliteratur und wie aus Kapitel 2.2.2 ersichtlich zulässig, da im 3P-Modell wie bereits erwähnt bei den Schwebeflugphasen nicht nur Start und Landung dargestellt werden, sondern implizit auch Beschleunigungs- und Verzögerungsphasen, für die kein Bodeneffekt gelten darf [4].

Die im eigenen Modell getroffenen Annahmen, etwa hinsichtlich des Oswald-Faktors, weichen zum Teil von denen im Ansatz aus Abschnitt 3.2 ab. Der Oswald-Faktor fließt in die Berechnung der Gleitzahl ein und kann je nach herangezogener Fachliteratur unterschiedliche Werte annehmen. Die Verwendung eines Wertes von 0,85 anstelle von 0,95 führt jedoch nur zu geringen Abweichungen im Gesamtergebnis.

3.2.2 Erweiterung und Anpassung für das eigene Modell

Aufbauend aus der Analyse des vorhandenen Berechnungsansatzes soll ein eigenes Modell zur Berechnung des Energiebedarfs entwickelt werden. Da es sich bei dem Ansatz aus der Fachliteratur nur um ein Berechnungsprinzip handelt und kein Modell, müssen einige Erweiterungen vorgenommen werden. Die vorabdefinierten Anforderungen an das Modell gehen aus Tabelle 2 hervor.

Die erste grundlegende Anpassung ist die Implementierung einer Benutzeroberfläche. Sie ermöglicht eine intuitive, interaktive und visuelle Bedienung, um über wenige Eingaben den Energiebedarf und die Reichweite zu berechnen. Des Weiteren werden die Eingaben durch Warnungen bei falschen Eingaben direkt validiert. Über die Benutzeroberfläche soll das Durchspielen verschiedener Konstruktionsszenarien ermöglicht werden, um verschiedene Designs und Konzepte miteinander zu vergleichen. Der Einsatz einer Benutzeroberfläche minimiert zudem die Wahrscheinlichkeit von menschlichen Fehlern im Vergleich zur Kommandozeileingabe, wie man sie von einigen anderen Modellen kennt.

Das Modell soll dazu in der Lage sein, verschiedene Konfigurationen zu berechnen und die Ergebnisse dieser zu speichern, um spätere Vergleiche zu ermöglichen. Zudem soll die

Berechnung um eine detaillierte Ermittlung des Luftwiderstands erweitert werden. Während dieser im bisherigen Berechnungsansatz aus der Fachliteratur als konstanter Wert angenommen wird, könnte eine differenzierte Betrachtung zu präziseren Ergebnissen führen. Da sich die verschiedenen eVTOL-Konzepte in ihrer vorgesehenen Flughöhe unterscheiden, soll der Luftwiderstand künftig in Abhängigkeit von der Flughöhe bestimmt werden.

Die größte Erweiterung des Modells ist jedoch der Optimierungsansatz. Neben der grundlegenden Berechnung von Reichweite und Energiebedarf auf Basis eingegebener Parameter soll das Modell in der Lage sein, diese Größen durch eine automatisierte Parametereingabe zu optimieren. Dazu werden die Ausgangsgrößen Reichweite sowie Energiebedarf bzw. Pro-Kopf-Verbrauch als Zielfunktionen formuliert, die entweder maximiert oder minimiert werden sollen. Dabei folgt die Optimierung den Kausalitäten der Parameter, wie beispielsweise die definierten Eingabebereiche der Parameter oder ihre Beziehungen untereinander, die auf fachlichen Grundlagen beruhen.

Neben der Berechnung und der Optimierung des Energiebedarfs steht auch die wirtschaftliche Tragfähigkeit der eVTOLs im Vordergrund. Die wirtschaftliche Tragfähigkeit wird dabei wie im theoretischen Rahmen erläutert unter anderem anhand der direkten Betriebskosten (DOC) gemessen. Die Darstellung dieser erfolgt anhand eines Preises-pro-Passagier-Kilometer. Hierfür wird in das Kostenmodell aus Abschnitt 2.3.5. integriert, mit dem Kenngrößen wie die Amortisationsdauer, die nivellierten Kosten und der Kapitalwert ausgegeben werden, über welche dann Aussagen über die wirtschaftliche Tragfähigkeit getroffen werden können.

Für die Optimierung werden auf Basis der Literaturrecherche feste Bereiche und Intervalle für die Parameterausgabe definiert. Der Nutzer soll jedoch die Möglichkeit haben, diese Grenzen anzupassen, um das Modell auch bei technologischen Fortschritten weiterhin verwenden zu können.

3.3 Implementierung des eigenen Berechnungsmodells in Python

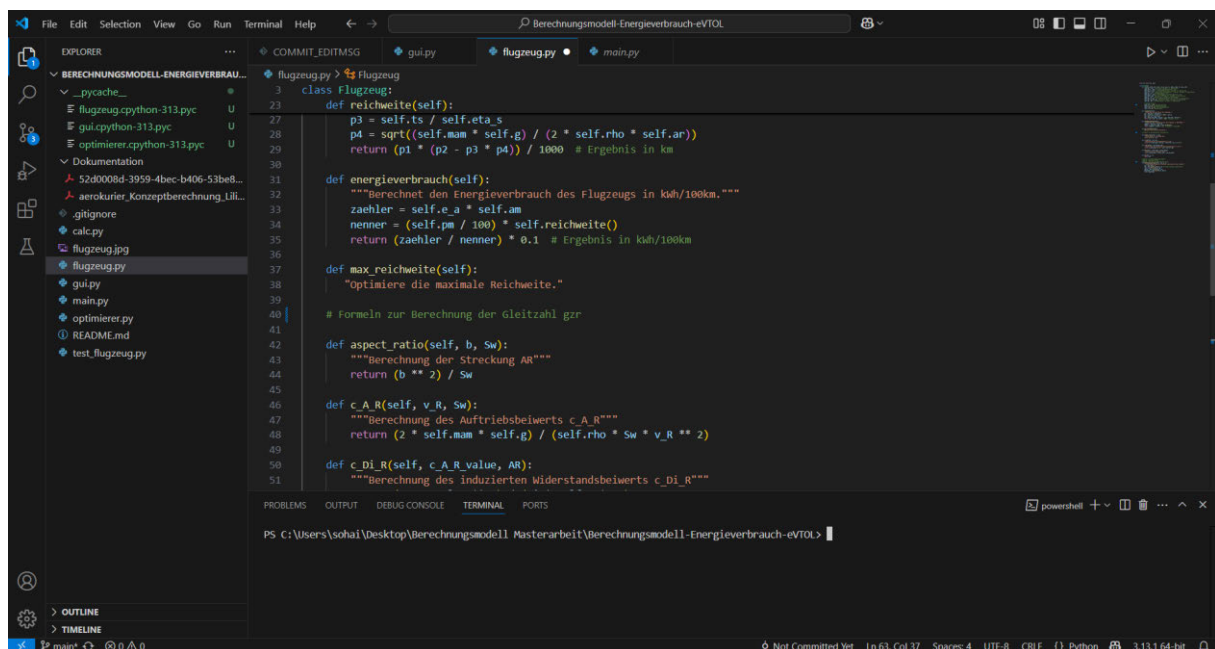
Für die Umsetzung des Berechnungsmodells wurde Python als Programmiersprache ausgewählt. Die Entscheidung hierfür beruht dabei auf allgemeinen Vorteilen dieser Programmiersprache, aber auch welchen, die im spezifischen Kontext dieser Masterarbeit relevant sind.

Python ist open-source, plattformunabhängig und zudem noch kostenlos verfügbar, womit sich das Tool ideal für wissenschaftliche Arbeiten eignet. Dadurch werden keine kommerziellen Softwarelizenzen vorausgesetzt und die Weitergabe des Projekts ermöglicht. Durch die klare Strukturierung in Python eignet es sich gut für wissenschaftliche und technische Vorhaben. Ein großer Vorteil von Python bei der Entwicklung eines Berechnungsmodells wie in dieser Arbeit ist die Verfügbarkeit umfangreicher Bibliotheken für wissenschaftliche Anwendungen,

Methodik und Modellentwicklung

die auch zahlreich genutzt wurden. Mit vergleichsweise geringem Aufwand können damit beispielsweise Benutzeroberflächen und Algorithmen in das Modell implementiert werden. Durch die objektorientierte Programmierung, die nachfolgend noch genauer erläutert wird, erlaubt die Umsetzung in Python eine hohe Flexibilität. Funktionen des Modells können dadurch strukturiert gekapselt und modular aufgebaut werden, was das Modell für zusätzliche Anforderungen und alternativen Methoden leicht erweiterbar macht. Ein weiterer Vorteil dieses Tools ist die breite Akzeptanz in der Forschung und Industrie. Python hat eine sehr aktive Entwickler-Community, wodurch bei Problemen schnell Hilfe oder vergleichbare Projekte gefunden werden können.

Die Programmierung erfolgt in einem Editor. Für diese Arbeit wurde der Editor „Visual Studio Code“ (VS Code) genutzt, wie in Abbildung 3-3 zu sehen ist.



```
class Flugzeug:
    def reichweite(self):
        p3 = self.ts / self.eta_s
        p4 = sqrt((self.mam * self.g) / (2 * self.rho * self.ar))
        return (p1 * (p2 - p3 * p4)) / 1000 # Ergebnis in km

    def energieverbrauch(self):
        """berechnet den Energieverbrauch des Flugzeugs in kWh/100km."""
        zaehler = self.e_a * self.am
        nenner = (self.pm / 100) * self.reichweite()
        return (zaehler / nenner) * 0.1 # Ergebnis in kWh/100km

    def max_reichweite(self):
        """Optimiere die maximale Reichweite."

        # Formeln zur Berechnung der Gleitzahl gzc

    def aspect_ratio(self, b, Sw):
        """Berechnung der Streckung AR"""
        return (b ** 2) / Sw

    def c_A_R(self, v_R, Sw):
        """Berechnung des Auftriebsbeiwerts c_A_R"""
        return (2 * self.mam * self.g) / (self.rho * Sw * v_R ** 2)

    def c_Di_R(self, c_A_R_value, AR):
        """Berechnung des induzierten Widerstandsbeiwerts c_Di_R"""
```

Abbildung 3-3: Programmierung in VS Code

Der grundlegende Vorteil bei der Nutzung von einem modernen Editor ist die Fehlererkennung, die Warnungen bei beispielsweise Tippfehlern oder Logikproblemen gibt. Dadurch, dass sich die Python Skripte direkt im Editor ausführen lassen, ist kein Wechsel zu externen Terminals erforderlich. VS Code verfügt zudem noch über eine vollständige Versionsverwaltung mit Git, was die Nachverfolgbarkeit von Änderungen ermöglicht (siehe Abbildung 3-4).

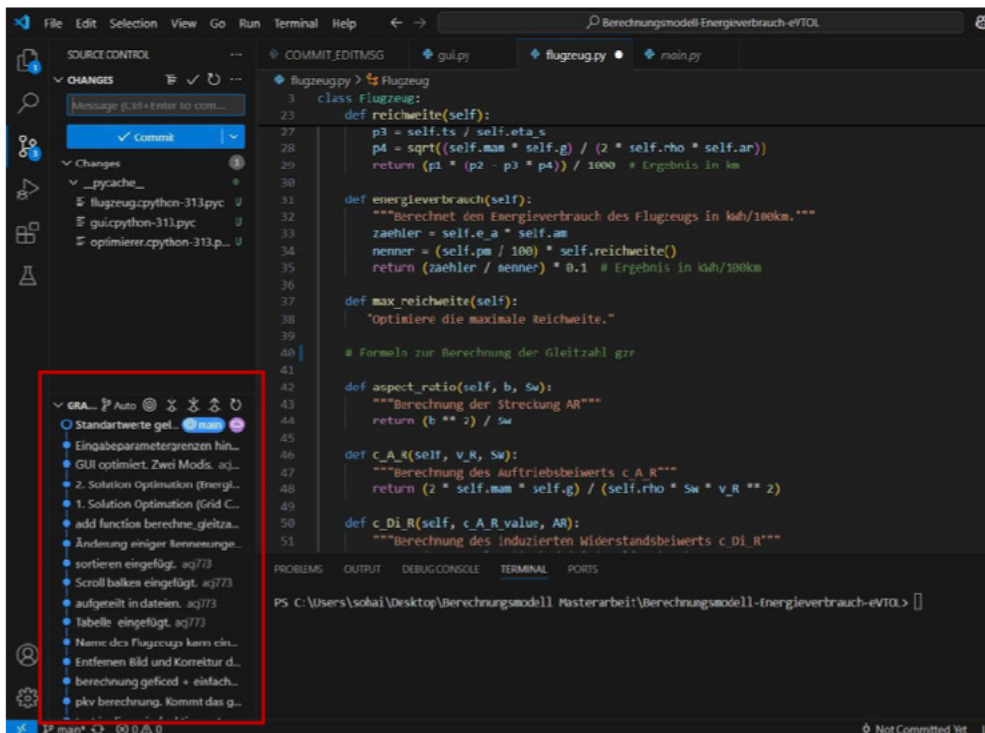


Abbildung 3-4: Programmierung in VS Code mit Hervorhebung der Versionsverwaltung

3.3.1 Aufbau des Berechnungsmodells

Um eine übersichtliche Struktur zu gewährleisten, wird in Python mit Klassen gearbeitet. Neben der imperativen skriptbasierten Programmierung bietet Python auch die Möglichkeit einer objektorientierten Programmierung (OOP) in Form von Klassen. Der zentrale Bestandteil der OOP ist es, reale Objekte in Software zu modellieren. Dabei werden die Daten eines Objekts als Eigenschaften und dessen Methoden als Funktionen in einer Klasse gebündelt [24]. Ziel ist ein strukturiertes Programm, das leicht wiederverwendbar ist und eine gute Skalierung für zukünftige Erweiterungen ermöglicht. Das Verwalten und Erweitern des Codes wird vereinfacht, da zusammengehörige Funktionen in einer Klasse organisiert werden.

Für das Berechnungsmodell wurden sechs Klassen definiert. Die Klasse *flet_gui.py* beinhaltet alle Funktionen für das Graphical User Interface (GUI). Sie beinhaltet alle Funktionen für die Benutzeroberfläche, um visuell mit dem Modell zu interagieren, statt textbasierte Befehle zu nutzen. Eine weitere Klasse, *flugzeug.py*, enthält alle Funktionen zur grundlegenden Berechnung der Reichweite und des Energiebedarfs samt aller Informationen zu den Interaktionen zwischen den Eingabeparametern. Die dritte Klasse, *optimierer.py* bündelt alle Funktionen zur Optimierung der zentralen Ausgangsgrößen. Während sich *flugzeug.py* nur der grundlegenden Berechnung dient, wird *optimierer.py* zur Optimierung des Energiebedarfs und der Reichweite genutzt. Hier werden alle Randbedingungen für die Optimierung beschrieben. Die Klasse *flugzeugManager.py* wurde für den Export der verschiedenen eVTOL-Konzepte erstellt. Die Klasse *kostenmodell.py* beinhaltet alle Funktionen zur Berechnung der wirtschaftlichen

Größen. Schlussendlich wird mit *Berechnungsmodell.py* die Hauptfunktion und somit das Berechnungsmodell ausgeführt bzw. gestartet. Das Modell verfügt über eine Startseite mit Hinweisen für den Nutzer, wie in Abbildung 3-5 zu sehen:

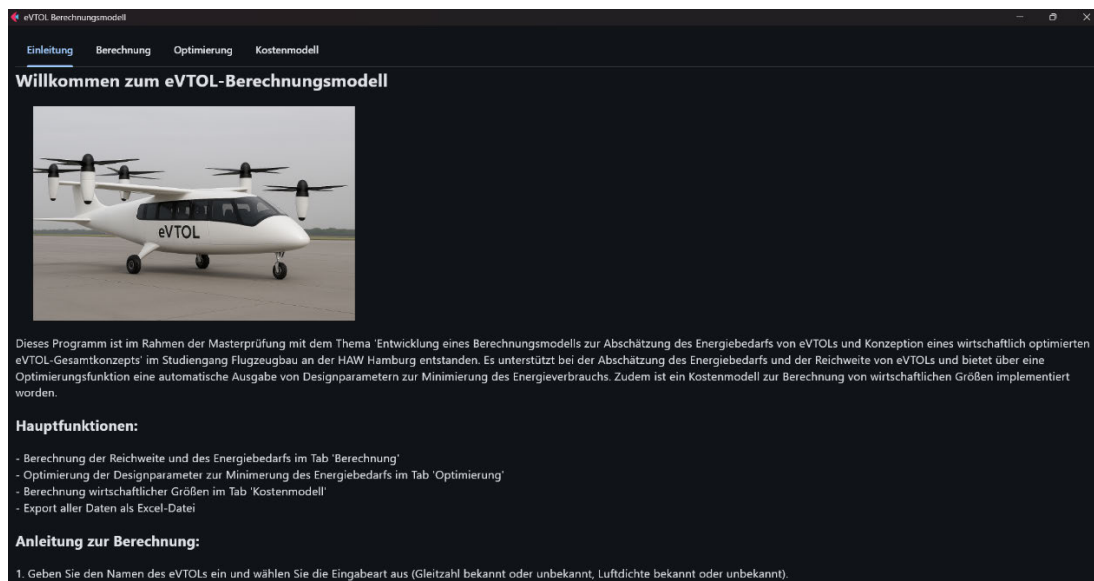


Abbildung 3-5: Startseite des Berechnungsmodells

3.3.2 Berechnung der Reichweite und des Energiebedarfs

Die Gleichungen für die Berechnung der Reichweite und des Energiebedarfs gehen aus Kapitel 3.2 hervor. Die Reichweite ergibt sich aus:

$$R = v_R \cdot t_R = GZ_R \cdot \eta_R \cdot \left(\frac{\mu_A \cdot e_A}{g} - \frac{t_S}{\eta_S} \cdot \sqrt{\frac{MAM \cdot g}{2 \cdot \rho \cdot A_r}} \right) \quad (57)$$

Der Energiebedarf wird wie folgt berechnet:

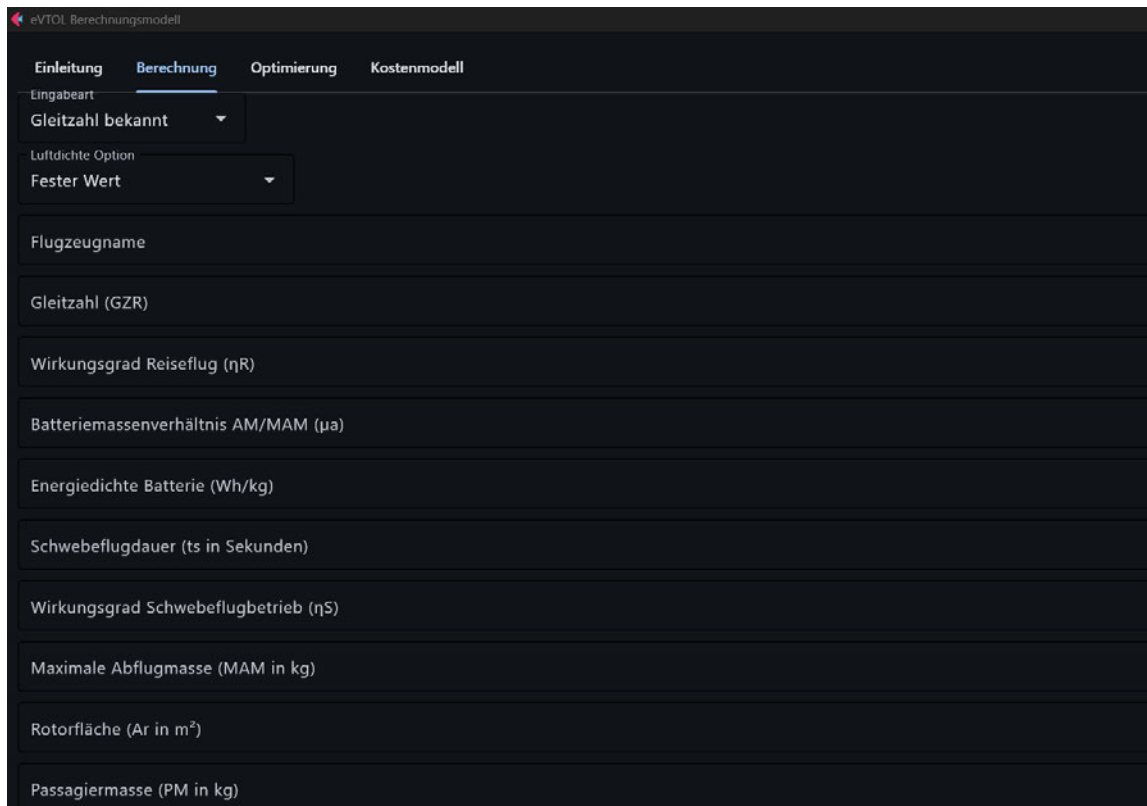
$$PKV = \frac{e_A \cdot AM}{PAX \cdot R} \quad (64)$$

Die Berechnung dieser Ausgangsgrößen erfolgt über vorherige Eingabe der Parameter durch den Nutzer.

Eingabe im Berechnungsmodell

Die Eingabe in das Berechnungsmodell erfolgt auf Grundlage der zuvor definierten Designparameter. Zusätzlich stehen verschiedene Eingabeoptionen zur Verfügung, wobei insbesondere die Behandlung der Gleitzahl eine Unterscheidung erfordert. So kann entweder mit einer bekannten Gleitzahl direkt gearbeitet werden oder alternativ eine Berechnung der Gleitzahl auf Basis der Flügelspannweite, der mittleren Flügeltiefe sowie der Reisefluggeschwindigkeit

erfolgen. Ähnlich ist es mit der Luftdichte. Hier kann der Wert ebenfalls direkt eingegeben werden oder über Eingabe der Flughöhe automatisch ermittelt werden. Abbildung 3-6 zeigt die Eingabemaske:



The image shows a software interface titled "eVTOL Berechnungsmodell" with four tabs: "Einleitung", "Berechnung", "Optimierung", and "Kostenmodell". The "Berechnung" tab is active. Below the tabs, there are several input fields and dropdown menus:

- "Eingabeart" with a dropdown menu set to "Gleitzahl bekannt".
- "Luftdichte Option" with a dropdown menu set to "Fester Wert".
- "Flugzeugname" (text input).
- "Gleitzahl (GZR)" (text input).
- "Wirkungsgrad Reiseflug (η_R)" (text input).
- "Batteriemassenverhältnis AM/MAM (μ_a)" (text input).
- "Energiedichte Batterie (Wh/kg)" (text input).
- "Schwebeflugdauer (t_s in Sekunden)" (text input).
- "Wirkungsgrad Schwebeflugbetrieb (η_S)" (text input).
- "Maximale Abflugmasse (MAM in kg)" (text input).
- "Rotorfläche (A_r in m^2)" (text input).
- "Passagiermasse (PM in kg)" (text input).

Abbildung 3-6: Eingabemaske für die Berechnung im Modell

Die Ergebnisse der einzelnen eVTOLs können zudem noch als Excel-Datei exportiert werden. Wird ein eVTOL-Konzept aus der Liste im Modell gelöscht, so wird diese Änderung auch in der Excel-Datei übernommen. Diese muss dann jedoch erneut geöffnet werden.

3.3.3 Automatisierte Optimierung der Reichweite und des Energiebedarfs

Neben der grundlegenden Berechnung der Reichweite und des Energiebedarfs soll das Modell auch über eine automatisierte Optimierung für diese Zielgrößen verfügen. Ziel der Optimierung ist es, je nach Größe die Zielfunktion zu maximieren oder zu minimieren und dabei die physikalischen und aerodynamischen Randbedingungen einzuhalten. Als Ergebnis soll die Wahl der Parameter für ein optimiertes Ergebnis ausgegeben werden, die eine realitätsnahe Auslegung eines eVTOLs darstellt.

Definition der Zielfunktionen

Für die Optimierung der Reichweite wird Zielfunktion (57) maximiert, welche zuerst in Kapitel 3.2 erschienen ist:

$$R = v_R \cdot t_R = GZ_R \cdot \eta_R \cdot \left(\frac{\mu_A \cdot e_A}{g} - \frac{t_S}{\eta_S} \cdot \sqrt{\frac{MAM \cdot g}{2 \cdot \rho \cdot A_r}} \right) \quad (57)$$

Die Optimierung des Energiebedarfs wird dabei über die Minimierung der Zielfunktion (64) erreicht:

$$PKV = \frac{e_A \cdot AM}{PAX \cdot R} \quad (64)$$

Definition der Randbedingungen

Zuvor müssen jedoch die Randbedingungen für die Optimierung festgelegt werden. Als erstes erfolgt eine Differenzierung zwischen fixen und freien Parametern. Die fixen Parameter bekommen einen eindeutigen Wert zugeteilt, während die variablen Parameter Intervalle und Intervallgrenzen zugeteilt bekommen. Diese beruhen auf vorheriger Recherche und den Informationen aus dem theoretischen Rahmen dieser Arbeit. Tabelle 5 zeigt die fixen Parameter:

Tabelle 5: Übersicht der fixen Parameter für die Optimierung

Parameter	Wert	Beschreibung	Referenz
g	$9,81 \frac{m}{s^2}$	Erdbeschleunigung	-
ρ	$1,190 \frac{kg}{m^3}$	Luftdichte auf Flughöhe (Beispiel)	-
η_R	0,8	Wirkungsgrad im Reiseflug	[12]
η_S	0,8	Wirkungsgrad im Schwebeflug	[12]
t_S	60 s	Schwebeflugdauer	[25]
c_{D0}	0,0317	Beiwerte weiterer Widerstände	[4]
e	0,85	Oswald-Faktor	[10]

Für die Werte der Erdbeschleunigung und der Dichte der Luft bedarf es keiner weiteren Erklärung. Die Luftdichte wird vorher über die Abfrage der Flughöhe ermittelt, kann jedoch trotzdem als fixer Parameter angenommen werden, da diese nicht an die Designparameter des Fluggeräts gekoppelt ist. Bei elektrischen Antrieben können Wirkungsgrade von bis zu 0,8 erreicht werden [12]. Eine detaillierte Betrachtung der technischen Umsetzung dieser Antriebe wird in dieser Arbeit nicht vorgenommen, da der Fokus auf der Optimierung liegt. Die Schwebeflugdauer wurde zum einen festgelegt, um eine Vergleichbarkeit der bereits bestehenden eVTOL-Konzepte in Kapitel 4 zu gewährleisten und zum anderen, da hier noch keine konkreten gesetzlich-regulativen Vorgaben vorhanden sind. Die Werte für den Oswald-Faktor und den Beiwerten weiterer Widerstände, die für die Ermittlung der Gleitzahl verwendet werden, wurden ebenfalls aus der Fachliteratur entnommen.

Bei den variablen Designparametern wird nach dem gleichen Prinzip vorgegangen. Tabelle 6 gibt eine Übersicht:

Tabelle 6: Übersicht der freien Parameter für die Optimierung

Parameter	Beschreibung	Min	Max	Intervall	Referenz
$\mu_A [-]$	Batteriemassenverhältnis	0,3	0,7	0,1	siehe Tabelle 7
$\mu_P [-]$	Personenmassenverhältnis	0,3	0,7	0,1	siehe Tabelle 7
$MAM [kg]$	Maximale Abflugmasse	1500	3500	100	angenommen
$A_r [m^2]$	Gesamtrotorfläche	1	50	5	angenommen
$e_A [\frac{Wh}{kg}]$	Energiedichte der Batterie	100	330	23	[26]
$v_R [\frac{km}{h}]$	Reisefluggeschwindigkeit	200	350	30	[25]
$GZ_R [-]$	Gleitzahl	1	25	1	[8]
$b [m]$	Spannweite	5	20	1	angenommen
$c [m]$	Mittlere Flügeltiefe	10% von b			angenommen

Die Parameter die angenommen wurden, orientieren sich hierbei an bereits bestehende Konzepte und Erfahrungen aus der Praxis. Um das Modell für zukünftige Betrachtungen im Hinblick auf technologische Fortschritte sicher zu gestalten, können diese Werte auch vom Nutzer angepasst werden.

Des Weiteren werden noch einige weitere „harte“ Randbedingungen erfüllt. Kombinationen der Parameter, die diese Randbedingungen nicht erfüllen, werden verworfen. Diese Randbedingungen können Tabelle 7 entnommen werden:

Tabelle 7: Randbedingungen für die Optimierung

Nr.	Bedingung	Bedeutung	Referenz
1.	$\mu_A + \mu_P \leq 0,7$	Begrenzung von Batterie- und Passagiermasseanteil	[4]
2.	$P_M = \mu_P \cdot MAM$	Passagiermasse berechnet sich aus μ_P	-
3.	$0,35 \leq c_{A,R} \leq 0,4$	Auftriebsbeiwert muss einen aerodynamisch realistischen Wert einhalten	[4]

Die erste Bedingung resultiert aus der Notwendigkeit, dass die Summe der Passagier- und Batteriemassenverhältnisse nicht größer als 1 sein darf, da sonst impliziert würde, dass das restliche Flugzeug masselos sind. Eine Leermasse, die 30% der maximalen Abflugmasse entspricht, stellt konstruktiv eine Herausforderung dar, ist jedoch für einen eVTOL durchaus plausibel [4].

Die zweite Bedingung gilt mehr der Berechnung der resultierenden Passagiermasse für die Ausgabe, koppelt jedoch somit die Passagiermasse an das Passagiermassenverhältnis, wie auch vorgesehen.

Die letzte Bedingung dient der Berechnung einer sinnvollen Gleitzahl. Da eine automatisierte Ausgabe für den Wert der Gleitzahl keine direkte Auskunft über das Design des eVTOLs gibt, wurde für die Gleitzahl eine Berechnung vorgeschaltet. Eine kurze Wiederholung der Gleichungen für die Berechnung der Gleitzahl aus Kapitel 3.2 verdeutlichen die Zusammenhänge:

$$GZ_R = \frac{c_{A,R}^2}{c_{D0} + c_{Di,R}} \quad (58)$$

$$c_{A,R} = \frac{2 \cdot MAM \cdot g}{\rho \cdot S_w \cdot v_R^2} \quad (59)$$

$$S_w = c \cdot b \quad (60)$$

Aus dieser Berechnung resultieren nach der Optimierung automatisch ausgegebene Werte für die Flügelspannweite (b), der mittleren Flügelstreckung (c) und der Reisefluggeschwindigkeit (v_R). In die Berechnung der Gleitzahl fließt auch der Auftriebsbeiwert $c_{A,R}$ mit ein. Dieser nimmt bei Reisefluggeschwindigkeit übliche Werte von 0,35 – 0,40 ein [4]. Demnach werden mit dieser Bedingung Kombinationen aus den Parametern Flügelspannweite, mittlere Flügeltiefe, Reisefluggeschwindigkeit etc. herausgefiltert, aus denen keine sinnvollen Auftriebsbeiwerte resultieren.

Ausgewählter Algorithmus für die Optimierung

Für die Optimierung wurde ein vollständiger „Grid-Search“ Algorithmus implementiert. Dabei wurden alle möglichen Kombinationen der definierten Parameterbereiche systematisch und lückenlos durchlaufen. Da die Rechenzeit sich als überschaubar erwiesen hat, war die Verwendung von Methoden wie beispielsweise „Random Sampling“ nicht erforderlich. Durch diese Vorgehensweise ist sichergestellt, dass das globale Maximum oder Minimum der Zielfunktion innerhalb des gewählten Parameterraums gefunden wird.

In einem verschachtelten Schleifenprozess, wie in dem Ausschnitt in Abbildung 3-7 zu erkennen, wird eine vollständige Grid-Search-Optimierung durchgeführt, bei der alle Kombinationen von den vorgegebenen Designparametern systematisch durchlaufen werden. Kombinationen, die vordefinierte Randbedingungen verletzen, werden dabei ausgeschlossen. Für jede gültige Kombination werden die Gleitzahl, Reichweite und der Energieverbrauch berechnet, wobei die jeweils beste Lösung im Hinblick auf maximale Reichweite und minimaler Energiebedarf gespeichert wird.

```
56         # Optimierung
57         for mu_a in mu_a_range:
58             for e_a in e_a_range:
59                 for mam in mam_range:
60                     for ar in ar_range:
61                         for mu_p in mu_p_range:
62                             for v_R in v_R_range:
63                                 for b in b_range:
64                                     c = 0.09 * b
65                                     self.counter += 1
```

Abbildung 3-7: Ausschnitt des Grid-Search Algorithmus

Eingabe im Berechnungsmodell

Zunächst muss der richtige Tab gewählt werden, um zur Optimierungsfunktion zu gelangen. Dort kann die Optimierung direkt gestartet werden, da die Standardwerte bereits verwendet werden. Die Standardwerte basieren auf der Literaturrecherche und können Tabelle 6 entnommen werden. Um die Funktion jedoch auch für andere Randbedingungen und Parameterbereiche und Intervalle verwenden zu können, können diese vom Nutzer frei gewählt werden.

Dabei muss die Eingabe nach dem Prinzip Anfangswert, Endwert, Intervall erfolgen, wie in Abbildung 3-8 zu sehen ist:

Parameter	Standardwert	Intervall
Batteriemassenverhältnis	0.3 - 0.8	0.3, 0.7, 0.1
spezifische Energiedichte	100 - 330 Wh/kg	100, 330, 23
maximale Abflugmasse	1500 - 3500 kg	1500, 3500, 100
Rotorfläche	1 - 50 m ²	1, 50, 5
Passagiermassenverhältnis	0.3 - 0.8	0.3, 0.7, 0.1
Reisefluggeschwindigkeit	55 - 100 m/s	55, 100, 5
Spannweite	5 - 20 m	5, 20, 1

Abbildung 3-8: Eingabemaske für die Optimierung im Modell (Ausschnitt)

3.3.4 Berechnung der wirtschaftlichen Tragfähigkeit

Ein wesentlicher Schwerpunkt dieser Arbeit liegt – neben der technischen Auslegung der eVTOLs – auf der Untersuchung ihrer ökonomischen Tragfähigkeit. Die Grundlage für die ökonomischen Berechnungen wurde im theoretischen Rahmen dieser Arbeit in Kapitel 2.3, insbesondere in Abschnitt 2.3.5 geschaffen.

Zur besseren Übersicht werden im Folgenden die zentralen Gleichungen für die ökonomischen Berechnung aus den theoretischen Grundlagen zusammengefasst und die Implementierung in Python aufgezeigt. Zur Bewertung der wirtschaftlichen Tragfähigkeit wird grundsätzlich überprüft, ob und in welchem Zeitraum eine Amortisierung der anfänglichen Investitionen und der Betriebskosten möglich ist:

$$IRC = t_r, \left[\sum_{t=1}^{t_r} (PYR_t - PYC_t) = 0 \right] \quad (27)$$

Die Umsetzung in Python erfolgt über eine Schleife, die die Differenz aus den jährlichen Einnahmen und Kosten ($PYR_t - PYC_t$) aufsummiert. Sobald die hieraus kumulierte Summe positiv oder null ist, wird der entsprechende Zeitpunkt als der IRC, bzw. die Amortisationsdauer zurückgegeben. Falls eine Amortisierung innerhalb der Lebensdauer des eVTOLs nicht möglich

ist, liefert die Funktion *none* zurück. Abbildung 3-9 zeigt den hier beschriebenen Abschnitt im Code.

```

23 | # Berechnet Investment Return Cycle (IRC, Amortisationsdauer)
24 | def calculate_irc(self):
25 |     cumulative_net = 0
26 |     for t in range(len(self.pyr)):
27 |         cumulative_net += self.pyr[t] - self.pyc[t]
28 |         if cumulative_net >= 0:
29 |             return t + 1 # 1-basiert (Jahr)
30 |     return None # Kein Break-even während der Lebensdauer möglich
    
```

Abbildung 3-9: Programmcode zur Berechnung der Amortisationsdauer (Ausschnitt)

Zuvor wird jedoch der jährliche Barwert berechnet, welcher sich aus der Differenz der jährlichen Einnahmen und Kosten ergibt:

$$NPV = \sum_{t=1}^{VL} (PYR_t - PYC_t) \quad (26)$$

Für die Implementierung in Python wird der NPV ermittelt, indem die Differenz der aufsummierten Einnahmen und Kosten berechnet wird. Der Code ist dementsprechend direkt und selbsterklärend umgesetzt und in Abbildung 3-10 zu sehen.

```

206 | Berechnet den Barwert (NPV) ueber die gesamte Lebensdauer.\n\n
207 |
208 | Rueckgabewert: Barwert (NPV) in $.\n
209 | """
210 | pyc_sum = sum(self.calculate_pyc())
211 | pyr_sum = sum(self.calculate_pyr())
212 | return pyr_sum - pyc_sum
    
```

Abbildung 3-10: Ausschnitt aus dem Programmcode zur Berechnung des Barwerts (NPV)

Zur Berechnung der jährlichen Kosten werden alle Teilkosten (Gleichung (29) bis (36)), die in Abschnitt 2.3.5 erläutert werden aufsummiert und auf das entsprechende Jahr abgezinst:

$$PYC_t = \frac{\sum C_{jt}}{(1 + DR)^t} \quad (28)$$

Da sich die Zusammensetzung der Teilkosten je nach Jahr unterscheidet, ist eine jahresabhängige Differenzierung der Kostenbestandteile erforderlich. Beispielsweise fallen die Kosten zur Herstellung der eVTOL-Zelle oder des autonomen Flugsystems nur im ersten Jahr an. Kosten für den Austausch der Batterie fallen nur dann an, wenn ein Austausch erforderlich ist und somit die maximale Anzahl an Ladezyklen erreicht ist, welches jedoch bereits bei der Berechnung dieses Kostenbestandteils berücksichtigt wird. Der Restwert des eVTOLs, der im

Kostenmodell negativ berücksichtigt wird, fällt nur im letzten Jahr der Nutzungsdauer an. In Abbildung 3-11 ist der Ausschnitt zur Berechnung von PYC_t zu sehen.

```

51     self.pyc = [0]
52     for t in range(1, self.VL + 1):
53         if t == 1:
54             self.Cj.append(self.Cvbm1() + self.Cpt(t) + self.Cplt(t) + self.Ce() + self.Ci() + self.Ctfti(t) + self.Cm())
55         elif t < self.VL:
56             self.Cj.append(self.Cpt(t) + self.Cplt(t) + self.Ce() + self.Ci() + self.Ctfti(t) + self.Cm())
57         elif t == self.VL:
58             self.Cj.append(self.Cpt(t) + self.Cplt(t) + self.Ce() + self.Ci() + self.Ctfti(t) + self.Cm() + self.Csv())
59         else:
60             raise ValueError("Wert t darf nicht negativ sein.")
61
62         nenner = (1 + self.DR) ** t
63         self.pyc.append(self.Cj[t] / nenner)
64     return self.pyc

```

Abbildung 3-11: Programmcode zur Berechnung der jährlichen Kosten (Ausschnitt)

Dementsprechend wurden unterschiedliche Summen bei der Implementierung in Python definiert (siehe Abbildung 3-11):

- **Im ersten Jahr ($t == 1$)** werden zusätzlich einmalige Anschaffungskosten berücksichtigt, wie z. B. für die eVTOL-Zelle, das autonome Flugsystem und weitere initiale Investitionen.
- **Für alle Jahre $t < VL$ (Lebensdauer)** werden laufende Betriebskosten, Wartungskosten, Energiekosten sowie ggf. anteilige Austauschkosten (z. B. Batterie) berücksichtigt.
- **Im letzten Jahr ($t == VL$)** wird zusätzlich der Restwert des eVTOLs ($C_{SV_{VL}}$) mit negativem Vorzeichen in das Kostenmodell aufgenommen, da dieser einen Rückfluss darstellt.

Die nivellierten Kosten über die gesamte Lebensdauer des eVTOLs (Gleichung (25)) sowie die jährlichen Einnahmen aus dem eVTOL Betrieb (Gleichung (37)) hängen beide maßgeblich von der transportierten Leistung ab. Wie in Abbildung 3-12 zu sehen, erfolgt die Berechnung der nivellierten Kosten in Python direkt durch Division der aufsummierten Kosten durch das Produkt der Einflussgrößen.

$$LC = \frac{\sum_{t=1}^{VL} PYC_t}{n_p \cdot Ratio_p \cdot VL \cdot n_D \cdot n_M \cdot R} \quad (25)$$

```

192     def LC(self):
193         """
194         Berechnet die nivellierten Betriebskosten in $/Passagierkilometer.\n
195
196         """
197
198         pyc = self.calculate_pyc()
199         zaehler = sum(pyc)
200         nenner = self.nP * self.ratioP * self.VL * self.nD * self.nM * self.R
201
202         return zaehler/nenner

```

Abbildung 3-12: Programmcode zur Berechnung der nivellierten Kosten (Ausschnitt)

$$PYR_t = \frac{p_P \cdot n_P \cdot Ratio_P \cdot R \cdot n_M \cdot n_D}{(1 + DR)^t} \quad (37)$$

Die jährlichen Einnahmen PYR_t ergeben sich aus der transportierten Passagierleistung multipliziert mit dem Flugpreis. Der Nenner berücksichtigt den Barwertfaktor, um zukünftige Einnahmen auf das jeweilige Jahr t abzuzinsen. In Python wird die Berechnung wie in Abbildung 3-13 dargestellt mit einer Schleife realisiert, wobei die Einnahmen für jedes Jahr einzeln berechnet und in einer Liste `self.pyr` gespeichert werden.

```

32     def calculate_pyr(self):
33         """
34         Berechnet die Einnahmen fuer ein bestimmtes Jahr t und fuegt es in die Liste ein.\n\n
35         Parameter:\n
36         t: Jahr fuer das die Einnahmen berechnet werden sollen.\n\n
37         Rueckgabewert: Eine Liste mit den Einnahmen für alle Jahre.\n
38         """
39         self.pyr = [0]
40         for t in range(1, self.VL + 1):
41             zaehler = self.pP * self.nP * self.ratioP * self.R * self.nM * self.nD
42             nenner = (1 + self.DR) ** t
43             self.pyr.append(zaehler / nenner)
44         return self.pyr

```

Abbildung 3-13: Programmcode zur Berechnung der jährlichen Einnahmen (Ausschnitt)

Definition der Rahmenbedingungen und Parameter

Aufbauend auf dem in Abschnitt 2.3.5 vorgestellten Kostenmodell in den theoretischen Grundlagen werden im Folgenden die zur Berechnung notwendigen Parameter konkret definiert und durch weitere Rahmenbedingungen ergänzt. Diese umfassen technische und betriebswirtschaftliche Größen, die für Anwendung des Modells erforderlich sind. Ziel hiermit ist es, die Eingangsgrößen nachvollziehbar darzulegen und die Voraussetzungen für die Berechnung transparent zu machen.

Tabelle 8 gibt eine Übersicht über die zentralen Eingangsparameter, die zur Wirtschaftlichkeitsberechnung verwendet werden. Dabei werden die Parameter, sofern verfügbar, mit Quellenangaben versehen und dienen als Grundlage zur Berechnung der wirtschaftlichen Kennzahlen LC , NPV und IRC gemäß dem in Abschnitt 2.3.5 beschriebenen Kostenmodell. Für einzelne Kostenpositionen werden unterschiedliche Werte angesetzt, abhängig davon, ob ein hohes, mittleres oder niedriges Kostenniveau (HC, BC, LC) angenommen wird.

Tabelle 8: Parameter für die Kostenberechnung nach Kostenszenario

Parameter	Symbol	Einheit	Wert			Referenz
			HC	BC	LC	
Kosten eVTOL-Zelle (fix)	C_{VBM-F}	\$	600.000	400.000	200.000	[27]
Kosten eVTOL-Zelle (variabel)	C_{VBM-V}	\$/kg	800	550	270	[21]
Kosten Antriebssystem	UC_{DS}	\$/kW	25			[21]
Kosten Batterie	UC_{ES}	\$/kWh	110	100	90	[28]
Stromkosten	p_E	\$/kWh	0,13	0,09	0,05	
Kosten Pilotengehalt	-	\$/Jahr	75.000			[25]
Kosten autonomes System	C_{UAD_1}	\$	60.000			[25]
Baukosten Vertiport	C_{VPC_0}	\$	1.450.000			[25]
Betriebskosten Vertiport	C_{VPO_t}	\$/Jahr	80.000			[25]
Baukosten Energieversorgungseinrichtung (Austausch)	C_{RFC_0}	\$	600.000			[21]
Betriebskosten Energieversorgungseinrichtung (Austausch)	C_{RFO_t}	\$/Jahr	82.000			[21]
Baukosten Energieversorgungseinrichtung (Aufladen)	C_{RFC_0}	\$	200.000			[25]
Betriebskosten Energieversorgungseinrichtung (Aufladen)	C_{RFO_t}	\$/Jahr	44.000			[21]
Fixe Wartungskosten	$Ratio_M$	\$	7,5% von C_{VBM_1}			[27]
Variable Wartungskosten	UC_{VM}	\$/h	20			[25], [27]

Methodik und Modellentwicklung

Bei den Kosten für die Energieversorgungseinrichtung werden zwei Fälle unterschieden, die auch nochmal bei der Beschreibung des eVTOL-Betriebs aufgegriffen werden. Das eine Betriebsszenario sieht einen Austausch der Batterie vor, während beim anderen Betriebsszenario vom üblichen Aufladen ausgegangen wird.

Die ökonomische Bewertung von eVTOLs erfordert eine präzise Modellierung des operativen Einsatzprofils. Tabelle 9 fasst die relevanten Informationen diesbezüglich zusammen.

Tabelle 9: Parameter für den eVTOL Betrieb

Parameter	Symbol	Einheit	Wert	Referenz
Jährlich maximale Flugzeit	YT	h	2400	[25]
Tägliche Flugzeit	-	h	8	angenommen
Anzahl an Werktagen im Jahr	n_D	Tag	300	angenommen
eVTOL Lebensdauer	VTL	Jahr	13	[25]
Batterie Lebensdauer	-	Zyklus	2000	[21]
Ladeleistung	$P_{charging}$	kW	200	[21]
Reichweite einer Flugmission	R	km	100	angenommen
Anteil leerer Rückflüge	$Ratio_{DE}$	-	0,2	[25]
Anteil an Flügen mit extremen Wetterbedingungen	β	-	0,2	angenommen
Passagierboardingzeit	$t_{boarding}$	min	15	[21]
Passagierdeboardingzeit	$t_{disembarking}$	min	15	[21]
Schwebeflugdauer	t_{hover}	min	1	[25]
Steigflugdauer	t_{climb}	min	1	[21]
Sinkflugdauer	$t_{descent}$	min	1	[21]
Zeit zum Austauschen der Batterie	$t_{swapping}$	min	5	[21]
Flugpreis	p_P	\$/km	1-1,5\$	angenommen
Ladefaktor	$Ratio_P$	-	0,5	[21]
Diskontierungsrate	DR	-	7%	[21]

In der Berechnung nach dem Kostenmodell aus 2.3.5 werden weitere Rahmenbedingungen festgelegt, welche Tabelle 10 entnommen werden können.

Tabelle 10: Weitere Rahmenbedingungen zur Kostenrechnung

Parameter	Symbol	Einheit	Wert	Referenz
Anteil Strukturmasse	-	-	$0,3 \cdot m_{TO}$	[25]
Anteil des Restwerts an den Herstellungskosten für die eVTOL-Zelle	<i>Ratio_{SV}</i>	-	0,1	angenommen

Eingabe im Berechnungsmodell

Die Berechnung der wirtschaftlichen Kenngrößen erfolgt im Berechnungsmodell ebenfalls mit den vorab definierten Parametern im Tab Kostenmodell. Hier können die Parameterwerte jedoch auch beliebig vom Nutzer verändert werden, um andere Konzepte zu bewerten oder Veränderungen zu untersuchen. Abbildung 3-14 zeigt einen Ausschnitt einiger dieser Parameter:

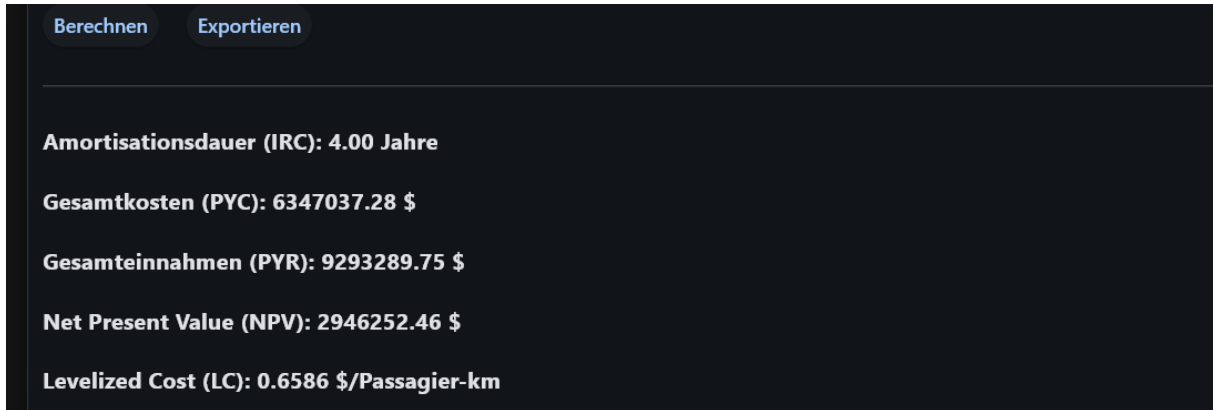
The screenshot shows the 'eVTOL Berechnungsmodell' interface with the 'Kostenmodell' tab selected. It is divided into two main sections: 'Betriebsparameter' and 'Herstellungskosten eVTOL-Zelle (Cvbm1)'. Each section contains several input fields with numerical values.

Parameter	Wert
Flugpreis pro Passagier (\$/km)	1.5
Anzahl der Passagiere	7
Ladefaktor	0.5
Reichweite pro Flugmission in km	100
Anzahl der Flugmissionen pro Tag	7.06
Anzahl der Werktage im Jahr	300
Diskontierungsrate	0.07
Lebensdauer des eVTOLs	13
Jährliche Flugzeit in h	2400
Cvbm_f (fixe Herstellungskosten)	600000
Cvbm_v (variable Herstellungskosten)	800

Abbildung 3-14: Eingabemaske für das Kostenmodell

Methodik und Modellentwicklung

Im Rahmen der Berechnung werden die Amortisationsdauer, der Kapitalwert, die Gesamteinnahmen, die Gesamtkosten sowie die nivellierten Kosten, wie in Abbildung 3-15 dargestellt. Dabei werden standardisiert die Ergebnisse für den autonomen Flugbetrieb ausgegeben. Alle anderen Ergebnisse lassen sich zudem als Excel-Datei exportieren, in der eine detaillierte Jahresübersicht für beide Flugbetriebe bereitgestellt wird.



Berechnen	Exportieren
<hr/>	
Amortisationsdauer (IRC):	4.00 Jahre
Gesamtkosten (PYC):	6347037.28 \$
Gesamteinnahmen (PYR):	9293289.75 \$
Net Present Value (NPV):	2946252.46 \$
Levelized Cost (LC):	0.6586 \$/Passagier-km

Abbildung 3-15: Ausgabe des Kostenmodells

4. Analyse bestehender eVTOL-Konzepte

In diesem Kapitel wird das Berechnungsmodell auf bereits bestehende eVTOL-Konzepte angewendet. Zuvor folgt eine Dokumentation ausgewählter eVTOL-Konzepte samt einer Kurzbeschreibung, der Spezifikationen und des Einsatzszenarios. Im Anschluss erfolgt die Berechnung des Energiebedarfs und der Betriebskosten.

4.1 Dokumentation aktueller eVTOL-Konzepte

Im folgenden Kapitel werden ausgewählte eVTOL-Konzepte dokumentiert. Dazu gehört eine kurze Beschreibung des Unternehmens, die Spezifikationen des eVTOLs und ein Ausblick auf das vom Unternehmen vorgesehene Einsatzszenario und Geschäftsmodell. Des Weiteren werden die Eingabeparameter für die nachfolgende Berechnung für jedes Konzept definiert.

4.1.1 Joby Aviation S4



Abbildung 4-1: Joby Aviation S4 [Joby]

Unternehmen und Entwicklung

Das in Abbildung 4-1 dargestellte Joby Aviation S4 des US-amerikanischen Unternehmens Joby Aviation gilt als eines der vielversprechendsten eVTOL-Konzepte. Mit Sitz in Kalifornien verfolgt das Unternehmen das Ziel, eVTOLs zu entwickeln und zu produzieren, um in den Markt für Lufttaxis einzusteigen. Das im Jahr 2009 von CEO Joe-Ben Bevirt gegründete Startup wird durch Investitionen von Unternehmen wie unter anderem Toyota, Uber und Baillie Gifford finanziert [29].

Im Jahr 2012 begann das Unternehmen in Zusammenarbeit mit der NASA, verschiedene Forschungsprojekte im Bereich des elektrischen Fliegens durchzuführen, darunter die Programme X-57 und LEAPTech. Nach Jahren des Testens flog 2015 ihr erster Subscale

Demonstrator mit der heutigen Konfiguration, bis sie 2017 zum ersten Mal mit dem fullsize Demonstrator flogen. Das Flugtestprogramm ihres Pre-Production Prototyps begann 2019. Kurz darauf beteiligte sich Toyota als Investor und unterstützte Joby Aviation mit ihrer Expertise in der Fertigungsprozessentwicklung, Produktionsplanung und Massenproduktion. Als erstes eVTOL-Unternehmen erlangte Joby Aviation 2020 die Lufttüchtigkeitszulassung von der U.S Air Force [30].

Spezifikationen

Das Flugzeug bietet neben einem Platz für einen Piloten Sitzraum für vier weitere Passagiere inklusive Gepäck. Das Unternehmen gibt eine Reichweite von 161 km und eine Reisefluggeschwindigkeit von 322 km/h an, wobei Energiereserven inkludiert sind. Mit einer Leermasse von 1950 kg und einer maximalen Nutzlast von 453 kg kommt der Flieger auf eine maximale Abflugmasse von 2403 kg. Der Rumpf besteht dabei aus CFK. Angetrieben wird das eVTOL von insgesamt sechs Schwenkpropellern, bzw. sechs Hochleistungs-Elektromotoren mit je 236 kW. Die Batterie des Fliegers verfügt über eine Energiedichte von 235 Wh/kg, mit dem über 10.000 Flugzyklen möglich sein sollen [29].

Die wichtigsten Designparameter, welche zur Eingabe in das Berechnungsmodell genutzt werden, sind nachfolgend in Tabelle 11 dargestellt. Werte, die direkt vom Hersteller oder aus der Fachliteratur stammen, sowie selbst ermittelte Parameter sind mit einer Referenz versehen. Letztere können dem jeweiligen Anhang entnommen werden. Für alle eVTOL-Konzepte werden die Wirkungsgrade im Reise- und Schwebeflug, sowie die Schwebeflugdauer auf Grundlage wohlwollender Literaturwerte angesetzt, da vonseiten der Hersteller hierzu keine konkreten Angaben vorliegen. Diese Parameter sind in den Tabellen entsprechend farblich gekennzeichnet. Die Gleitzahl wird entsprechend der Gleichungen aus 3.2 vom Berechnungsmodell berechnet. Die Massenverhältnisse werden aus der Nutzlast und der Batteriemasse abgeleitet.

Tabelle 11 enthält eine Übersicht weiterer relevanter Spezifikationen und Eingabegrößen für das Berechnungsmodell.

Tabelle 11: Eingabeparameter Joby Aviation S4

Parameter	Wert	Einheit	Referenz
Gleitzahl (GZ_R)	4,49	-	berechnet
Spannweite (b)	11,6	m	[31]
Wirkungsgrad im Reiseflug (η_R)	0,8	-	angenommen
Wirkungsgrad im Schwebeflug (η_S)	0,8	-	angenommen
Maximale Abflugmasse (MAM)	2403	kg	[29]
Gesamtrotorfläche (A_r)	42,02	m^2	Anhang 9.1.1

Passagiermassenverhältnis (μ_P)	0,19	-	[29]
Batteriemassenverhältnis (μ_A)	0,37	-	[32]
Energiedichte der Batterie (e_A)	235	Wh/kg	[29]
Reisefluggeschwindigkeit (v_R)	322	km/h	[29]
Schwebeflugdauer (t_s)	60	s	[25]

Einsatzgebiet und Geschäftsmodell

Das hauptsächliche Einsatzgebiet des Joby S4 soll der urbane Raum in Städten mit hoher Verkehrsbelastung sein. Als besonderes Einsatzszenario präsentiert das Unternehmen die Strecke von Downtown in New York City zum JFK Flughafen, wie in Abbildung 4-2 zu sehen. Mit dem Auto werden für die Strecke 49 Minuten benötigt, während der Joby Aviation S4 die Strecke in sieben Minuten bewältigen kann. Bekräftigt wird die überwiegende Nutzung eines solchen Einsatzszenarios durch die Partnerschaft mit Delta Air Lines, welche mit Joby Aviation einen home-to-airport Taxi Service anbieten möchten, um das Kundenerlebnis zu verbessern. Des Weiteren soll das Lufttaxi in die Uber-App integriert werden, um ein Ridesharing Service für Lufttaxis anzubieten. Hierfür wurde die Partnerschaft mit Uber intensiviert, um ihre Software-Tools und Expertise für den kommerziellen Markteintritt ins eigene Unternehmen zu implementieren [30].

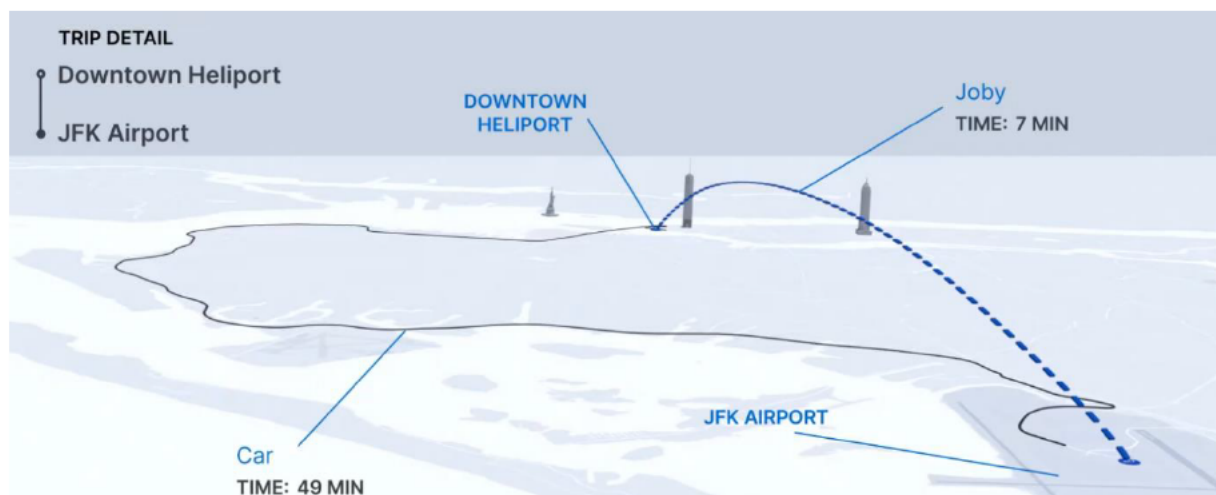


Abbildung 4-2: Mögliches Einsatzgebiet für den Joby Aviation S4 [30]

4.1.2 Archer Aviation Midnight



Abbildung 4-3: Archer Aviation Midnight [Archer Aviation]

Unternehmen und Entwicklung

Ein weiteres eVTOL-Konzept eines in Kalifornien sitzenden Unternehmens ist der Midnight von Archer Aviation. Gegründet von Brett Adcock und Adam Goldstein im Jahr 2018 befindet sich Archer Aviation auf dem Weg zur Kommerzialisierung ihres eVTOLs. Dazu begannen sie eine Partnerschaft mit dem Automobilhersteller Stellantis, um fortschrittliche Fertigungstechniken in ihre eigene Produktion zu etablieren [33].

Die ersten Flugtests mit ihrem ersten fullsize eVTOL-Konzept haben von 2018-2021 stattgefunden. 2023 fand der erste Testflug mit ihrem heutigem Konzept, dem Midnight, statt. 2024 hat Archer Aviation das Part 134 und Part 135 Zertifikat von der FAA erhalten. Im selben Jahr hat die FAA Archer Aviation die endgültigen Lufttüchtigkeitskriterien für den Midnight mitgeteilt. Mit den Vereinigten Arabischen Emiraten und Indien kündigte das Unternehmen die ersten geplanten internationalen Markteintrittsstandorte an [34].

Spezifikationen

Der Midnight bietet Platz für vier Passagiere und einen Piloten. In Zukunft soll der Midnight jedoch autonom fliegen. Bei einer Reisegeschwindigkeit von 241 km/h soll der Flieger von Archer Aviation eine Reichweite von 100 km erreichen. Dabei verfügt der Midnight über eine maximale Nutzlast von 456 kg und kommt damit auf eine maximale Abflugmasse von 3175 kg. Angetrieben wird das eVTOL von 12 schwenkbaren Propellern mit 12 elektrischen Motoren. Die Energie liefern sechs unabhängige Batterieeinheiten. Archer Aviation gibt einen Lärmpegel

Analyse bestehender eVTOL-Konzepte

von 45 dB für den Reiseflug an, welches ihren eVTOL geeignet für den urbanen Raum machen soll [35], [33]. Die vollständigen Spezifikationen und Eingabeparameter des Berechnungsmodells sind in Tabelle 12 aufgeführt.

Tabelle 12: Eingabeparameter Archer Aviation

Parameter	Wert	Einheit	Referenz
Gleitzahl (GZ_R)	8,39	-	berechnet
Spannweite (b)	14,6	m	[36]
Wirkungsgrad im Reiseflug (η_R)	0,8	-	angenommen
Wirkungsgrad im Schwebeflug (η_S)	0,8	-	angenommen
Maximale Abflugmasse (MAM)	3175	kg	[33]
Gesamtrotorfläche (A_r)	46,55	m^2	Anhang 0
Passagiermassenverhältnis (μ_P)	0,14	-	[33]
Batteriemassenverhältnis (μ_A)	0,3	-	angenommen
Energiedichte der Batterie (e_A)	260	Wh/kg	[33]
Reisefluggeschwindigkeit (v_R)	241	km/h	[33]
Schwebeflugdauer (t_s)	60	s	[25]

Einsatzgebiet und Geschäftsmodell

Archer Aviation möchte Urban Air Mobility für die Massen zugänglich machen. Hierfür versuchen sie die Wartung, die Wartungskosten und die Energiekosten zu minimieren. Die Nutzung von einem Ridesharing Service soll die Produktivität des eVTOLs steigern [33]. Archer Aviation nimmt gezielt 20-50 Meilen lange Strecken in Angriff, für die man mit den üblichen Verkehrsmitteln über eine Stunde Fahrzeit hätte. Als Beispiel führt das Unternehmen auf der eigenen Website die Route von Downtown New York City zum Newark EWR Flughafen auf, wie in Abbildung 4-4 dargestellt. Während die Fahrt mit dem Auto über 50 Minuten und mit dem Zug sogar mehr als eine Stunde dauert, soll der Midnight die Strecke in nur 9 Minuten zurücklegen [37].



Abbildung 4-4: Mögliches Einsatzgebiet für den Archer Aviation Midnight [37]

4.1.3 Lilium Jet



Abbildung 4-5: Lilium Jet [Lilium]

Unternehmen und Entwicklung

Das eVTOL des deutschen Unternehmens Lilium steht seit jeher im besonderen Fokus der Luftfahrtbranche. Gegründet wurde das Unternehmen 2015 von Daniel Wiegand, Sebastian Born, Patrick Nathen und Matthias Meiner [38]. Das erste eVTOL Konzept entstand allerdings bereits 2013 durch Wiegand. Darauf folgten vier Prototypen namens Gleiter, Hexa, Dragon und Falcon. 2017 erfolgten unbemannte Testflüge mit dem Zweisitzer Konzept „Eagle“, dessen Erfolg zu der Entwicklung ihres ersten fullsize Prototypen mit vier Sitzen geführt hat. Im Jahr 2019 wurde bekannt gegeben, dass Lilium an einem eVTOL mit Platz für bis zu sieben Personen arbeitet [39]. Aufgrund fehlender Gelder zum Aufbau der Produktion meldete Lilium im Herbst

2024 Insolvenz an. Nach gescheiterter zweiter Rettung meldete das Unternehmen im Frühjahr 2025 erneut die Insolvenz an [40].

Spezifikationen

Der Lilium Jet verfügt über 36 elektrische Motoren und Mantelpropellern, welche je nach Flugphase umschwenken. Laut Angaben des Unternehmens hat der Lilium Jet das geringste „lift-to-drag“ Verhältnis in der eVTOL-Branche und ist somit in diesem Aspekt vergleichbar mit Verkehrsflugzeugen, die in 35.000 Fuß Höhe fliegen. Mit einer Energiedichte von 330 Wh/kg der Batterie soll der Lilium Jet bei einer Reisefluggeschwindigkeit von 248 km/h eine maximale Reichweite von 175 km erreichen. Lilium hebt hervor, dass die 3 Meter lange Kabine ihres eVTOLs die längste in der eVTOL-Branche ist. Dabei ermöglicht sie unterschiedliche Konfigurationen für verschiedene Einsatzszenarien, jedoch eine maximale Passagieranzahl von sechs Personen exklusive Pilot [26]. Weitere Spezifikationen und Eingabeparameter für das Berechnungsmodell können Tabelle 13 entnommen werden.

Tabelle 13: Eingabeparameter Lilium Jet

Parameter	Wert	Einheit	Referenz
Gleitzahl (GZ_R)	8,90	-	berechnet
Spannweite (b)	13,9	m	[41]
Wirkungsgrad im Reiseflug (η_R)	0,8	-	angenommen
Wirkungsgrad im Schwebeflug (η_S)	0,8	-	angenommen
Maximale Abflugmasse (MAM)	3175	kg	[39]
Gesamtrotorfläche (A_r)	2,02	m^2	[42]
Passagiermassenverhältnis (μ_P)	0,22	-	[41]
Batteriemassenverhältnis (μ_A)	0,30	-	[41]
Energiedichte der Batterie (e_A)	330	Wh/kg	[26]
Reisefluggeschwindigkeit (v_R)	248	km/h	[26]
Schwebeflugdauer (t_s)	60	s	[25]

Einsatzszenario und Geschäftsmodell

Anders als andere Unternehmen in ihrer Branche, soll der Lilium Jet längere Intercity und regionale Reisen ersetzen. Das Unternehmen konzentriert sich gezielt auf die Effizienz von größeren Distanzen. Eine mögliche Strecke ist in Abbildung 4-6 zu sehen [26].



Abbildung 4-6: Mögliches Einsatzgebiet für den Lilium Jet [26]

4.1.4 Vertical Aerospace VX4

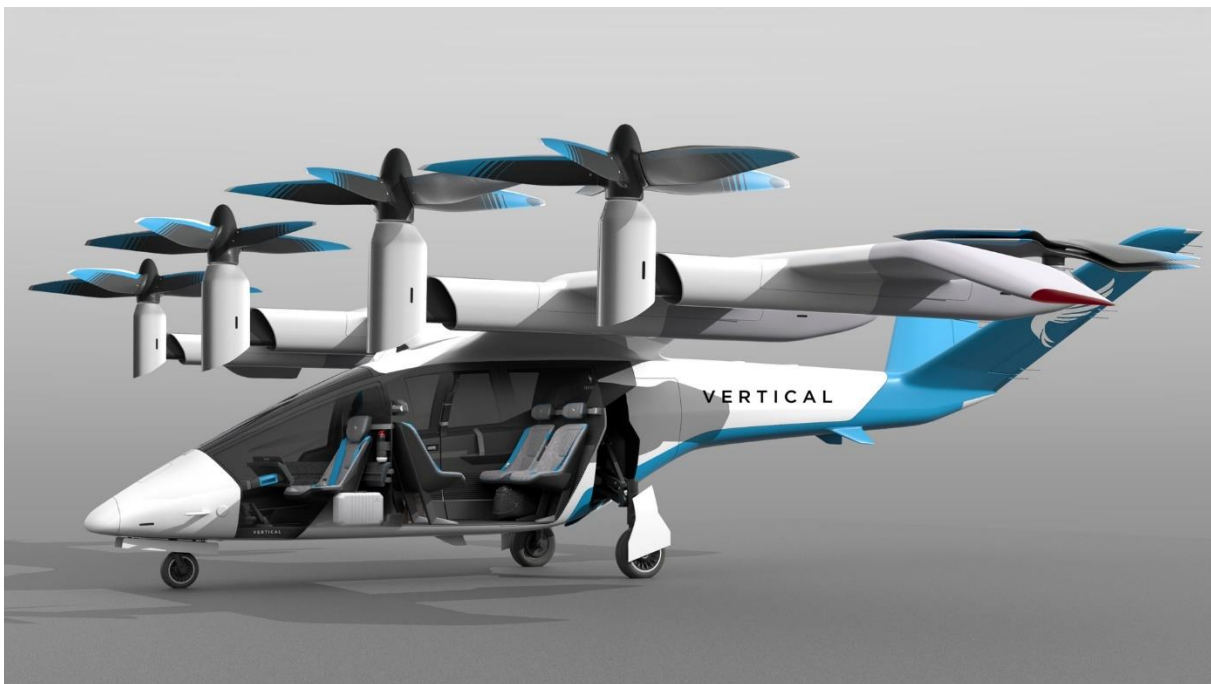


Abbildung 4-7: Vertical Aerospace VX4 [43]

Unternehmen und Entwicklung

Mit der Vision der Dekarbonisierung des Luftverkehrs gründete Steven Fitzpatrick, zugleich CEO von OVO Energy, im Jahr 2016 Vertical Aerospace. Im September 2018 gab das Startup bekannt, den ersten fullscale eVTOL im Vereinigten Königreich gebaut und erfolgreich im Freien geflogen zu haben. Etwa ein Jahr später, im Oktober 2019, gaben sie bekannt, das weltweit erste Lufttaxi mit einer Nutzlast von 250 kg gebaut und geflogen zu haben. 2020

veröffentlichte das Unternehmen einen Teaser und stellte kurz später ihren neuen eVTOL, den VX4 (siehe Abbildung 4-7) vor. Im September 2022 erhielt das Unternehmen von der britischen Luftfahrtbehörde CAA die Genehmigung, mit der Flugtestphase für den VX4 zu beginnen [43].

Spezifikationen

Die Konfiguration des VX4 soll eine maximale Reichweite von 161 km bei einer Reisegeschwindigkeit von 241 km/h erreichen. Dabei soll er unter einem Lärmpegel von 50 dB bleiben. Insgesamt bietet der VX4 Platz für fünf Personen, darunter ein Pilot und vier Passagiere. Damit kommt er auf eine maximale Nutzlast von 450 kg. Neben einer Spannweite von 15 m und einer Höhe von 4 m verfügt der VX4 über insgesamt acht Propeller und acht elektrische Motoren [43]. Eine detaillierte Auflistung zusätzlicher Eingabeparameter und Spezifikationen findet sich in Tabelle 14. Da es zum VX4 keine offiziellen Angaben zur maximalen Abflugmasse gibt, wurde in dieser Arbeit auf die Werte des Lilium Jet zurückgegriffen, da dessen Konfiguration der des VX4 in wesentlichen Aspekten ähnelt.

Tabelle 14: Eingabeparameter Vertical Aerospace VX4

Parameter	Wert	Einheit	Referenz
Gleitzahl (GZ_R)	7,75	-	berechnet
Spannweite (b)	15	m	[43]
Wirkungsgrad im Reiseflug (η_R)	0,8	-	angenommen
Wirkungsgrad im Schwebeflug (η_S)	0,8	-	angenommen
Maximale Abflugmasse (MAM)	3175	kg	angenommen
Gesamtrotorfläche (A_r)	65,14	m^2	Anhang 0
Passagiermassenverhältnis (μ_P)	0,14	-	[44]
Batteriemassenverhältnis (μ_A)	0,3	-	angenommen
Energiedichte der Batterie (e_A)	220	Wh/kg	[45]
Reisefluggeschwindigkeit (v_R)	241	km/h	[43]
Schwebeflugdauer (t_s)	60	s	[25]

Einsatzszenario und Geschäftsmodell

Mit einer maximalen Reichweite von 100 km ist der VX4 für den urbanen Raum konzipiert. Vertical Aerospace konzentriert sich dabei insbesondere auf Kurzstrecken, die aufgrund hoher Verkehrsbelastung mit herkömmlichen Verkehrsmitteln viel Zeit in Anspruch nehmen und durch urbanes Fliegen deutlich schneller zurückgelegt werden können. Die in Abbildung 4-8 bis Abbildung 4-10 dargestellten Beispiele zeigen Strecken zwischen 15 und 20 Meilen, die

Analyse bestehender eVTOL-Konzepte

mit dem VX4 in einem Bruchteil der Zeit im Vergleich zu herkömmlichen Verkehrsmitteln zurückgelegt werden sollen [44].



Abbildung 4-8: Mögliches Einsatzgebiet für den VX4 (Battersea – LHR) [44]

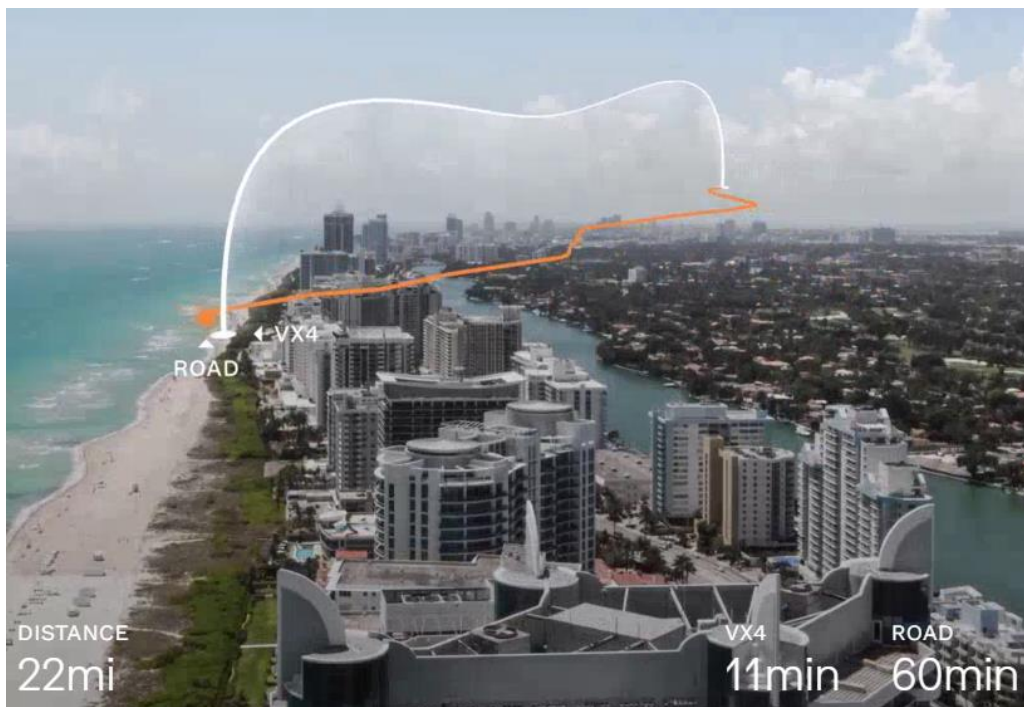


Abbildung 4-9: Mögliches Einsatzgebiet für den VX4 (MIA - Fort Lauderdale) [44]

Analyse bestehender eVTOL-Konzepte



Abbildung 4-10: Mögliches Einsatzgebiet für den VX4 (Yumeshima Port - KIX) [44]

4.2 Anwendung des Berechnungsmodells auf die eVTOL-Konzepte

Im Folgendem wird der Energiebedarf für die in Kapitel 4.1 vorgestellten eVTOL-Konzepte berechnet. Die relevanten Designparameter werden dabei von der Recherche entnommen und durch eigene Ermittlungen ergänzt.

4.2.1 Abschätzung des Energiebedarfs des Joby Aviation S4

Die Ergebnisse für die Abschätzung des Energiebedarfs und der Reichweite des Joby Aviation S4 können Tabelle 15 und der direkt aus dem Modell aus Abbildung 4-11 entnommen werden.

Tabelle 15: Energiebedarf und Reichweite des Joby Aviation S4

Parameter	Einheit	Wert
Energiebedarf pro Kopf (<i>PKV</i>)	$\frac{kWh}{100 km}$	16,84
Energiebedarf	$\frac{kWh}{100 km}$	84,20
Maximale Reichweite (<i>R</i>)	<i>km</i>	273,86



Abbildung 4-11: Modellausgabe für Reichweite und Energiebedarf des Joby Aviation S4

4.2.2 Abschätzung des Energiebedarfs des Archer Aviation Midnight

Tabelle 16 und Abbildung 4-12 zeigen die modellbasierten Ergebnisse zur Abschätzung des Energiebedarfs und der Reichweite des Archer Aviation Midnight.

Tabelle 16: Energiebedarf und Reichweite des Archer Aviation Midnight

Parameter	Einheit	Wert
Energiebedarf pro Kopf (<i>PKV</i>)	$\frac{kWh}{100 km}$	17,47

Energiebedarf	$\frac{kWh}{100 km}$	87,35
Maximale Reichweite (R)	km	310,87



Abbildung 4-12: Modellausgabe für Reichweite und Energiebedarf des Archer Midnight

4.2.3 Abschätzung des Energiebedarfs des Lilium Jets

Die ermittelten Werte für Energiebedarf und Reichweite des Lilium Jets lassen sich Tabelle 17 sowie der in Abbildung 4-13 gezeigten Modellausgabe entnehmen.

Tabelle 17: Energiebedarf und Reichweite des Lilium Jets

Parameter	Einheit	Wert
Energiebedarf pro Kopf (PKV)	$\frac{kWh}{100 km}$	12,79
Energiebedarf	$\frac{kWh}{100 km}$	89,53
Maximale Reichweite (R)	km	351,11



Abbildung 4-13: Modellausgabe für Reichweite und Energiebedarf des Lilium Jets

4.2.4 Abschätzung des Energiebedarfs des Vertical Aerospace VX4

Die Ergebnisse zur Abschätzung von Energiebedarf und Reichweite des Vertical Aerospace VX4 sind in Tabelle 18 sowie direkt in der Modellausgabe in Abbildung 4-14 dargestellt.

Tabelle 18: Energiebedarf und Reichweite des Vertical Aerospace VX4

Parameter	Einheit	Wert
Energiebedarf pro Kopf (<i>PKV</i>)	$\frac{kWh}{100 km}$	18,19
Energiebedarf	$\frac{kWh}{100 km}$	90,95
Maximale Reichweite (<i>R</i>)	<i>km</i>	256,07

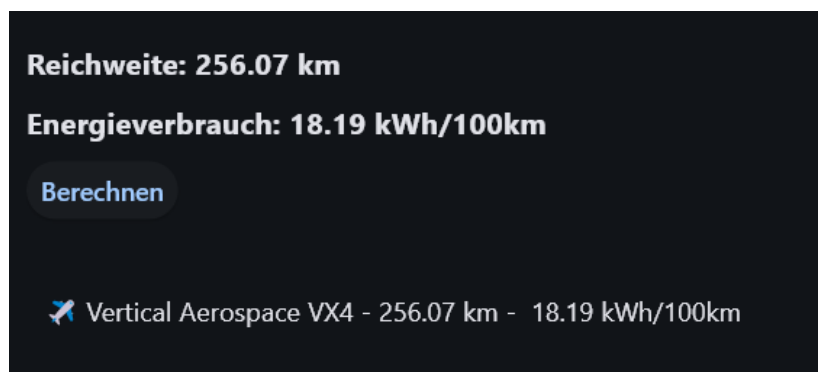


Abbildung 4-14: Modellausgabe für Reichweite und Energiebedarf des Vertical VX4

4.2.5 Vergleich der Energieeffizienz der verschiedenen Konzepte

Zur besseren Vergleichbarkeit der Designparameter, des berechneten Energiebedarfs und der Reichweite der einzelnen eVTOL-Konzepte sind sämtliche Eingaben und Ergebnisse in Tabelle 19 zusammengefasst. Dabei wird neben dem Bedarf pro Kopf auch der Gesamtbedarf pro Strecke dargestellt.

Analyse bestehender eVTOL-Konzepte

Tabelle 19: Vergleich des Energiebedarfs mit Überblick auf die Designparameter

Parameter	eVTOL-Konzept			
	Joby S4	Midnight	Lilium	VX4
Gleitzahl (GZ_R) [-]				
Spannweite (b) [m]	11,6	14,6	13,9	15
Wirkungsgrad im Reiseflug (η_R) [-]	0,8	0,8	0,8	0,8
Wirkungsgrad im Schwebeflug (η_S) [-]	0,8	0,8	0,8	0,8
Maximale Abflugmasse (MAM) [kg]	2403	3175	3175	3175
Gesamtrotorfläche (A_r) [m ²]	42,02	46,55	2,02	65,14
Passagiermassenverhältnis (μ_P) [-]	0,19	0,14	0,22	0,14
Batteriemassenverhältnis (μ_A) [-]	0,37	0,3	0,3	0,3
Energiedichte der Batterie (e_A) [$\frac{Wh}{kg}$]	235	260	330	220
Reisefluggeschwindigkeit (v_R) [$\frac{km}{h}$]	322	241	248	241
Schwebeflugdauer (t_s) [s]	60	60	60	60
Maximale Reichweite (R) [km]	273,86	310,87	351,11	256,07
Energiebedarf pro Kopf (PKV) [$\frac{kWh}{100 km}$]	16,84	17,47	12,79	18,19
Energiebedarf [$\frac{kWh}{100 km}$]	84,20	87,35	89,53	90,95

Die Darstellung des Energiebedarfs pro Kopf ermöglicht eine Vergleichbarkeit bei unterschiedlichen Passagieranzahlen und ist aus Sicht der Wirtschaftlichkeit und Umwelteinwirkung relevanter und wird daher bevorzugt. Für eine technisch neutrale Perspektive wird zudem auch der Gesamtenergiebedarf aufgezeigt.

Die Ergebnisse zeigen, dass der Lilium Jet mit $12,79 \frac{kWh}{100 km}$ den geringsten Energiebedarf pro Kopf erzielt, während der Vertical Aerospace VX4 mit $18,19 \frac{kWh}{100 km}$ am schlechtesten

abschneidet. Dieselbe Beobachtung kann auch bei der berechneten maximalen Reichweite gemacht werden. Hier verfügt der Lilium Jet über 40% mehr Reichweite als der Vertical Aerospace VX4. Betrachtet man den Gesamtenergiebedarf, so bewegen sich die untersuchten Konzepte auf einem vergleichbaren Niveau. Nach dem Lilium Jet erzielt der Joby Aviation S4 den geringsten Pro-Kopf Energiebedarf und den besten Gesamtenergiebedarf mit $84,20 \frac{kWh}{100 km}$. Vergleicht man den Lilium Jet mit dem Archer Aviation Midnight, fällt auf, dass die Designparameter – bis auf die Gesamtrotorfläche – sehr ähnlich sind. Der verbesserte Energiebedarf pro Kopf liegt jedoch an dem höheren Passagiermassenverhältnis vom Lilium Jet und somit mit der höheren Passagier-, bzw. Insassenanzahl.

Die errechneten Reichweiten berücksichtigen keine Reserven, die die Hersteller gegebenenfalls vorgesehen haben. Zudem werden durch wohlwollende Annahmen wie die Wirkungsgrade im Reise- und Schwebeflug bessere Ergebnisse erzielt, als die Angaben der Hersteller.

4.2.6 Ableitung der Erkenntnisse zur Identifikation von Optimierungspotenzial

Angesichts fehlender Validierung einiger Designparameter durch den Hersteller ist die Aussagekraft der darauf basierenden Ergebnisse nur eingeschränkt belastbar. Dennoch können über die bisherige Betrachtung einige qualitative Aussagen getroffen werden.

Bereits die Struktur der Zielfunktionen für Reichweite und Pro-Kopf-Verbrauch erlaubt Rückschlüsse darauf, wie sich einzelne Parameter – etwa durch Erhöhung oder Verringerung – auf das Ergebnis auswirken. Diese Zusammenhänge wurden in Abschnitt 3.2 anhand Tabelle 3 und Tabelle 4 veranschaulicht. Die hier erzielten Ergebnisse spiegeln diese Auswirkungen wider. Aus diesem Grund werden diese Zusammenhänge hier nicht erneut wiederholt.

Interessant ist die deutlich geringere Gesamtrotorfläche des Lilium Jets im Vergleich zu den anderen eVTOL-Konzepten, die durch Nutzung von vielen kleinen Mantelpropellern zustande kommt und den Lilium Jet damit in seiner Antriebsart von den anderen eVTOLs unterscheidet. Allgemein gilt, wie auch in Kapitel 1 beschrieben, die Flächenbelastung möglichst gering zu halten. Dies kann durch eine Erhöhung der Rotorfläche und einer Verringerung der maximalen Abflugmasse erzielt werden. Auch im Hinblick auf die Wirtschaftlichkeit ist es sinnvoll, die maximale Abflugmasse bei eVTOLs gering zu halten, da somit die Kosten zur Herstellung des eVTOLs verringert werden. Die maximale Abflugmasse ist ein entscheidender Parameter aus weiteren Gründen, welche in der Diskussion genauer behandelt werden.

Zusammenfassend gilt, die Parameter, die sich positiv auf den Energiebedarf auswirken, in einem gültigen Bereich zu optimieren, um damit den Energiebedarf zu minimieren. Gleichzeitig müssen ökonomische Aspekte nicht außer Acht gelassen werden, da ein Zusammenspiel aus

Analyse bestehender eVTOL-Konzepte

ökonomischen und technischen Auslegungen letztendlich die wirtschaftliche Tragfähigkeit bestimmt.

5. Entwicklung eines optimierten eVTOL-Gesamtkonzepts

In diesem Kapitel wird auf Basis der bisherigen Erkenntnisse und Ergebnisse ein optimiertes Gesamtkonzept für ein eVTOL entwickelt. Dabei erfolgt die Definition eines technischen Konzepts unter Angabe aller relevanten Designparameter sowie die Ausarbeitung eines Betriebs- bzw. Geschäftsmodells, um die wirtschaftliche Tragfähigkeit des entworfenen technischen und operationellen Ansatzes zu bewerten.

5.1 Ableitung eines technischen Konzepts unter Berücksichtigung relevanter Designparameter

Für die Ableitung des technisch optimierten Konzepts wird die im Berechnungsmodell implementierte Optimierungsfunktion herangezogen. Dabei erfolgt gezielt eine Anpassung der Designparameter mit dem Ziel, den Energieverbrauch des eVTOLs zu minimieren.

Die Erkenntnisse aus Kapitel 3.2 zeigen, dass zur Minimierung des Pro-Kopf-Verbrauchs das Passagiermassenverhältnis priorisiert wird. Da eine höhere Passagierkapazität ebenfalls positive Effekte auf die Profitabilität hat, wird die Reduzierung des Energieverbrauchs aus technischer wie auch aus ökonomischer Perspektive als zielführend angesehen.

Für die Bestimmung der Designparameter wurden neben der fixen Parameter aus Tabelle 5 die Rahmenbedingungen aus Tabelle 6 und Tabelle 7 verwendet. Die automatisiert ausgegebenen Designparameter können der Übersicht aus Tabelle 20 entnommen werden. Dabei wurden die vorher fest definierten Parameter hinzugefügt und farblich gekennzeichnet. Zusätzlich zu der Gleitzahl werden zudem noch Werte für die Spannweite, der mittleren Flügeltiefe und der Reisefluggeschwindigkeit ausgegeben, aus denen die Gleitzahl berechnet wurde.

Tabelle 20: Übersicht der Designparameter und der Ergebnisse des technischen Konzepts

Parameter	Einheit	Wert
Gleitzahl (GZ_R)	-	10,92
Spannweite (b)	m	14
Mittlere Flügeltiefe (c)	m	1,26
Reisefluggeschwindigkeit (v_R)	$\frac{km}{h}$	234
Maximale Abflugmasse (MAM)	kg	1800
Gesamtrotorfläche (A_T)	m^2	46
Passagiermassenverhältnis (μ_P)	-	0,40
Batteriemassenverhältnis (μ_A)	-	0,30

Energiedichte der Batterie (e_A)	$\frac{Wh}{kg}$	330
Wirkungsgrad Reiseflug (η_R)	-	0,8
Wirkungsgrad Schwebeflug (η_S)	-	0,8
Schwebeflugdauer (t_S)	s	60
Maximale Reichweite (R)	km	308,92
Energiebedarf pro Kopf (PKV)	$\frac{kWh}{100km}$	8,01
Energiebedarf	$\frac{kWh}{100km}$	57,66

Bei einer maximalen Abflugmasse von 1800 kg und einem Passagierverhältnis von 0,4 ergibt sich eine maximale Nutzlast von 720 kg. Dies entspricht unter der Annahme von 100 kg pro Passagier mit Gepäck einer Passagieranzahl von sieben für einen autonomen Flug. Wird mit Pilot geflogen, beläuft sich die Passagieranzahl entsprechend auf sechs. Für ein Batteriemassenverhältnis von 0,3 ergibt sich für die Batterie eine Masse von 540 kg. Bei einer Energiedichte von 330 Wh/kg verfügt das eVTOL-Konzept über eine Batteriekapazität von rund 178,2 kWh. Für die Wirkungsgrade im Reise- und im Schwebeflug wurde wie bereits in Abschnitt 3.3.3 erläutert ein Literaturwert von 0,8 angesetzt. Das aus den Designparametern resultierende Konzept weist einen spezifischen Energieverbrauch von $8,01 \frac{kWh}{100 km}$ pro Passagier und einen Gesamtenergiebedarf von $57,66 \frac{kWh}{100 km}$ auf und liegt damit im Bereich aktueller Designziele für eVTOLs sowie unterhalb der Ergebnisse der bereits bestehenden eVTOL-Konzepte. Mit einer Reichweite von über 308 km ist das Konzept im oberen Bereich vergleichbarer eVTOL-Systeme einzuordnen. Damit erfüllt das entwickelte Konzept zentrale Anforderungen an Energieeffizienz und Reichweitenfähigkeit.

Im Rahmen der Optimierung wurden – wie in Abbildung 5-1 dargestellt – insgesamt 13.305.600 Kombinationen getestet. Aufgrund der Randbedingungen aus Tabelle 6 schieden 12.196.800 Kombinationen aufgrund der Massenverhältnisse sowie 1.049.400 wegen eines aerodynamisch unzulässigen Auftriebsbeiwerts aus. Das optimierte Konzept wurde somit aus 2.158.200 gültigen Kombinationen ausgewählt.

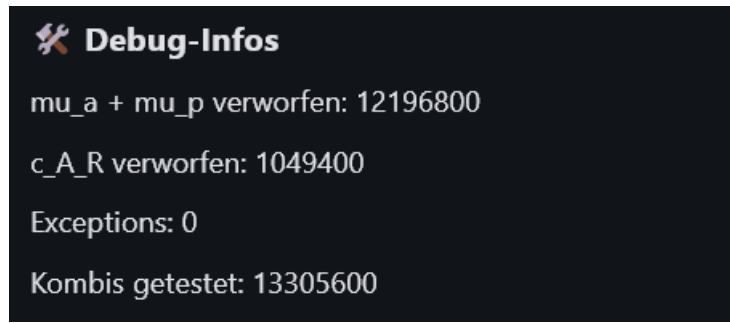


Abbildung 5-1: Modellausgabe der verworfenen Kombinationen

5.2 Entwicklung eines wirtschaftlich tragfähigen Geschäftsmodells

Aufbauend auf dem im theoretischen Teil dieser Arbeit vorgestellten Kostenmodell sowie den in Kapitel 3.3.4 definierten Parametern und Randbedingungen werden in diesem Abschnitt die zentralen wirtschaftlichen Kennzahlen für das entwickelte eVTOL-Konzept berechnet und dargestellt. Dabei werden unterschiedliche Flugpreis- und Betriebskonzepte einbezogen, um die wirtschaftliche Tragfähigkeit unter variierenden Rahmenbedingungen zu analysieren.

Diagramm 1 zeigt die jährlichen Gesamtkosten (*PYC*) über die Lebensdauer von 13 Jahren für ein Szenario mit hohen Kosten und stellt dabei sowohl die Kosten bei einem autonomen (links) als auch bei einem nicht-autonomen Flugbetrieb (rechts) dar. Hierbei werden auch die Anteile der Teilkosten visualisiert:

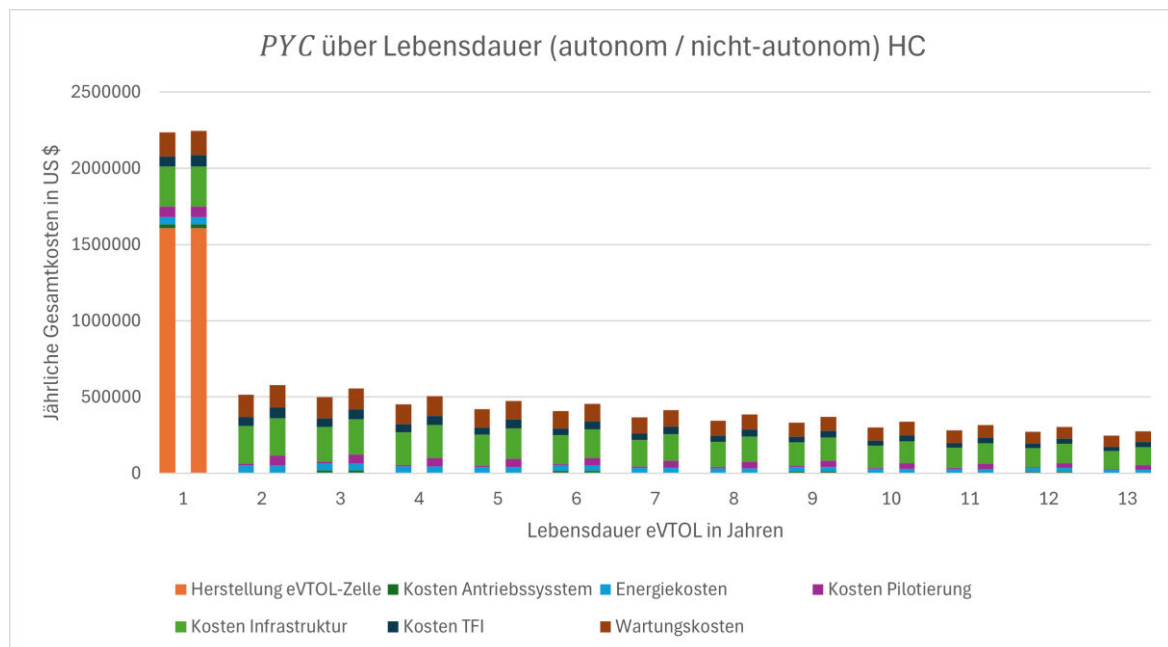


Diagramm 1: Jährliche Gesamtkosten über die Lebensdauer für HC Szenario

Die Unterscheidung eines *High Cost* (HC), *Base Cost* (BC) und *Low Cost* (LC) Szenarios haben dabei wie bereits aus

Entwicklung eines optimierten eVTOL-Gesamtkonzepts

Tabelle 8 in Abschnitt 3.3.4 ersichtlich nicht auf alle Teilkosten einen Einfluss. Lediglich die Kosten für die Herstellung der eVTOL-Zelle, die Energiekosten und die Kosten für das Antriebssystem berücksichtigen hierbei eine Variation, da es gerade hier großes Optimierungspotenzial und Schwankungen geben kann.

Die Kosten für die Herstellung der eVTOL-Zelle und die Herstellung des Antriebssystems fallen nur im ersten Jahr an. Die darauffolgenden Kosten für das Antriebssystem ergeben sich aus den laufenden Kosten für den Antriebsstrang sowie Kosten für den Austausch der Batterie, wenn das Ende der Lebensdauer der Batterie erreicht ist. Bei 2000 möglichen Ladezyklen ist für das technische Konzept in etwa alle drei Jahre ein Austausch fällig und damit ca. vier über die gesamte Lebensdauer des eVTOLs. Im ersten Jahr sind die Kosten für die Pilotierung für einen autonomen und einen nicht-autonomen Flugbetrieb identisch, da im ersten Jahr auch die Kosten zur Herstellung des autonomen Flugsystems anfallen. Über die restlichen Jahre fallen für die Pilotierungskosten für einen nicht-autonomen Flugbetrieb 75.000 US-Dollar an, während sich bei einem autonomen Flugbetrieb 15.000 US-Dollar ergeben. Letztere berücksichtigen die Kosten für einen Piloten für Tage mit extremen Wetterbedingungen, an denen ein autonomer Betrieb eine zu hohe Gefahr darstellen könnte. Die Baukosten für die Infrastruktur werden über die Lebensdauer der entsprechenden Einrichtungen amortisiert, welche über der Lebensdauer des eVTOLs liegen. Dementsprechend sind diese Kosten zum Ende der Lebensdauer des eVTOLs noch nicht amortisiert, welches jedoch auch im Kostenmodell so vorgesehen war. Mit über 2 Mio. US-Dollar liegen die Gesamtkosten im ersten Jahr weit über den Gesamtkosten in den Folgejahren, welche sich im Bereich von 200.000 bis 600.000 US-Dollar bewegen. Hier ist die Herstellung der eVTOL-Zelle der größte Kostenverursacher, welche über 75% der Kosten ausmacht. Da auch die jährlichen Gesamtkosten diskontiert werden, nehmen diese für die Berechnung des heutigen Barwerts mit der Lebensdauer ab.

Diagramm 2 bis Diagramm 5 stellen die Einnahmen (*PYR*) über die Lebensdauer des eVTOLs für unterschiedliche Passagierpreise und Flugbetriebe dar. Dabei wurde hier nur eine Unterscheidung zwischen einem Passagierpreis von 1,00 und 1,50 \$/Passagier-km gemacht.

Entwicklung eines optimierten eVTOL-Gesamtkonzepts

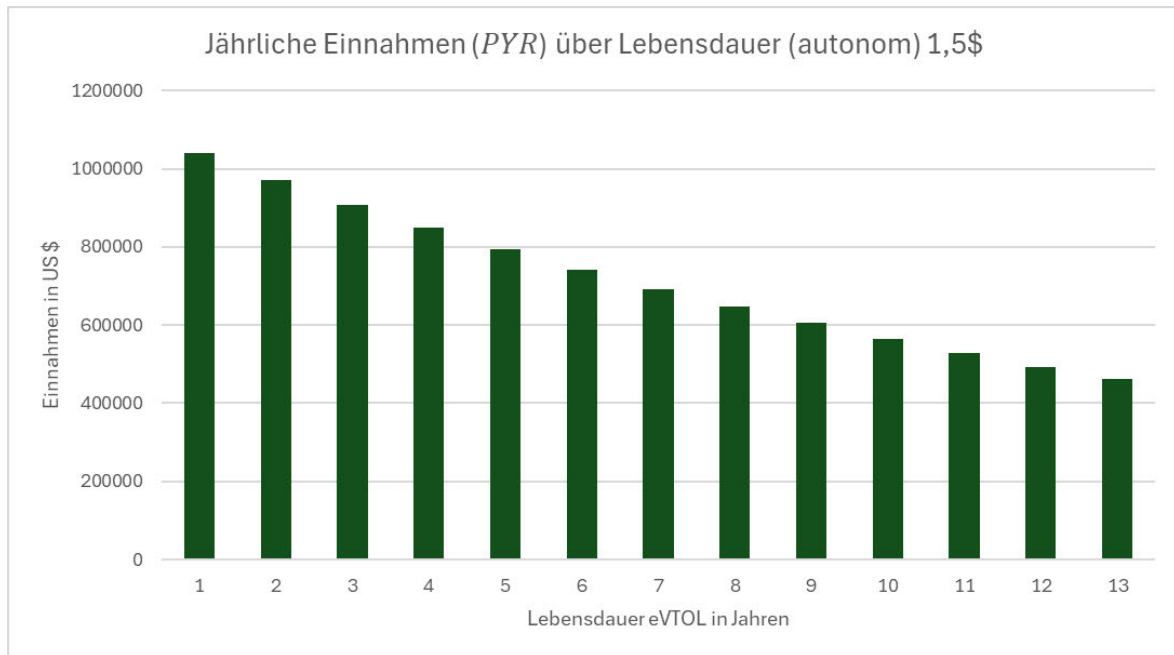


Diagramm 2: Jährliche Einnahmen beim autonomen Flugbetrieb bei einem Preis von 1,5 \$/Passagier-km

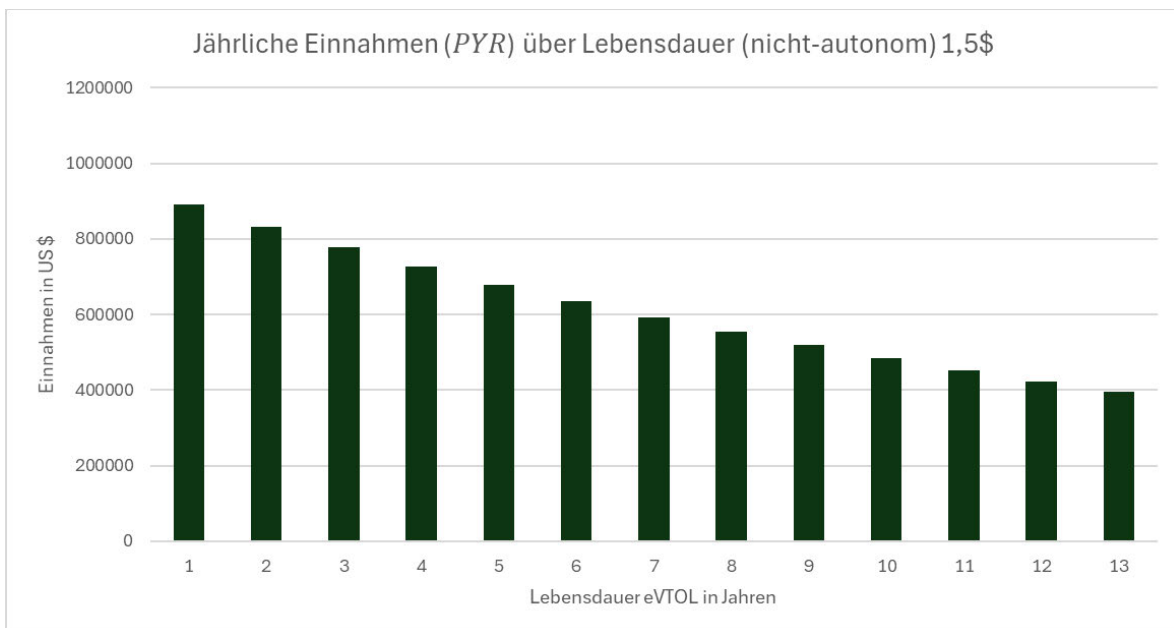


Diagramm 3: Jährliche Einnahmen beim nicht-autonomen Flugbetrieb bei einem Preis von 1,5 \$/Passagier-km

Entwicklung eines optimierten eVTOL-Gesamtkonzepts

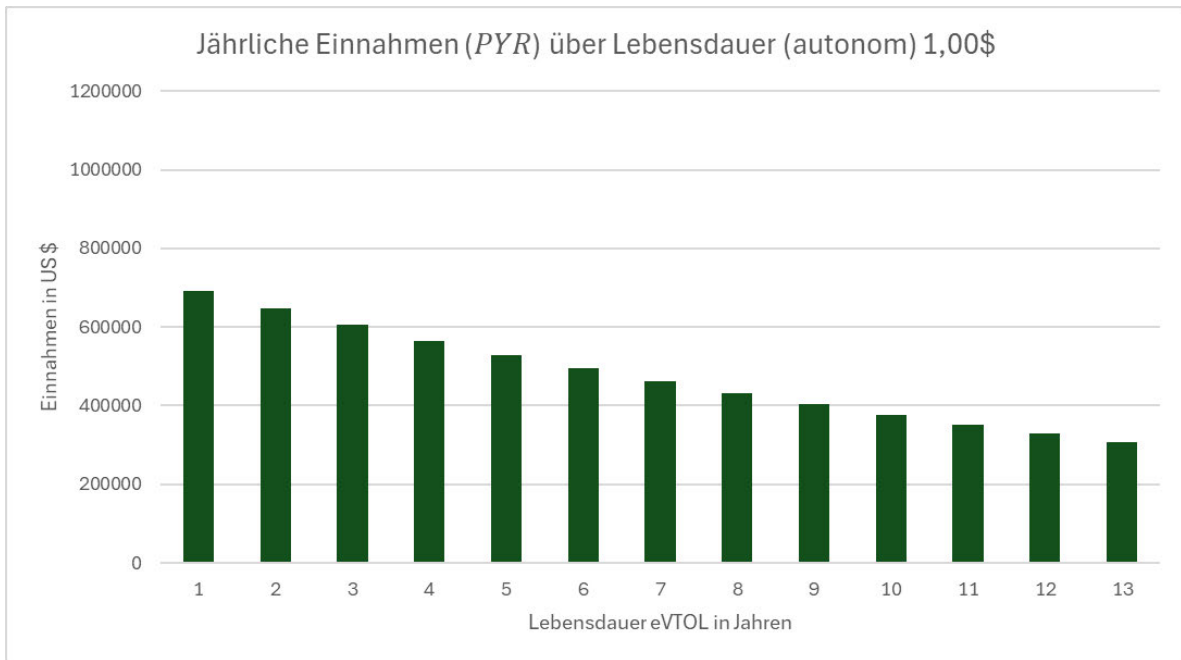


Diagramm 4: Jährliche Einnahmen beim autonomen Flugbetrieb bei einem Preis von 1,00 \$/Passagier-km

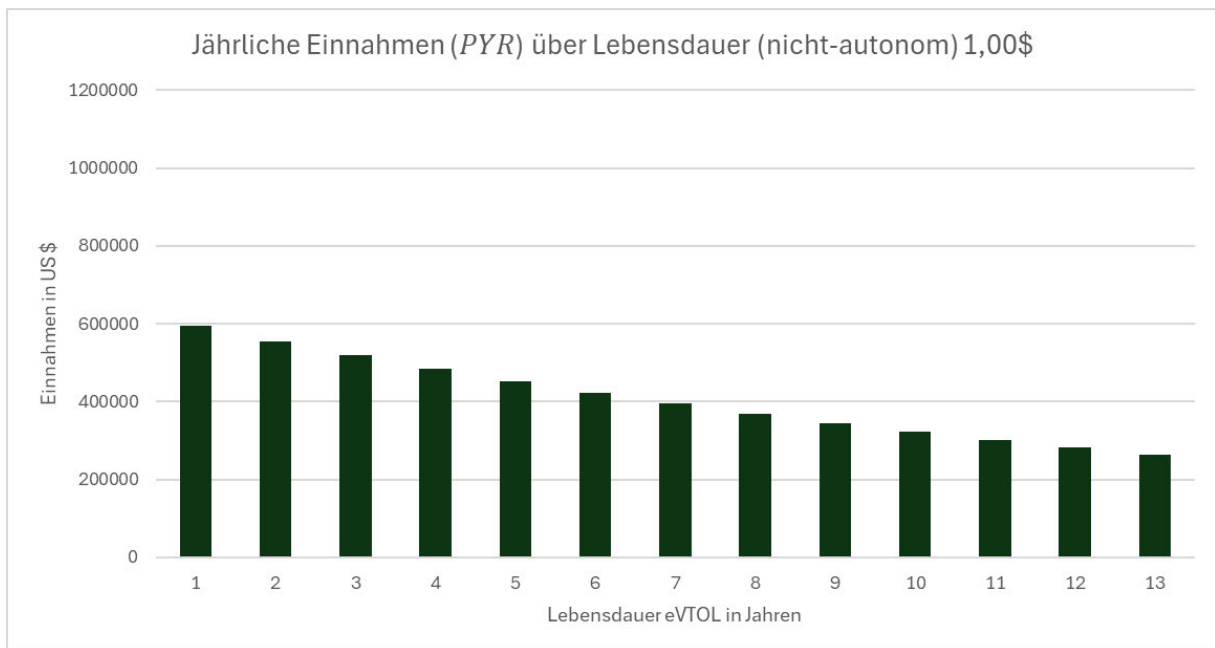


Diagramm 5: Jährliche Einnahmen beim nicht-autonomen Flugbetrieb bei einem Preis von 1,00 \$/Passagier-km

Mit der Diskontierung betragen die Einnahmen im ersten Jahr bei einem autonomen Flugbetrieb und einem Passagierpreis von 1,50 \$/Passagier-km über 1 Mio. US-Dollar, während sie im letzten Jahr für denselben Anwendungsfall nur noch weniger als die Hälfte betragen.

Tabelle 21 fasst die Summe der Einnahmen über die Lebensdauer des eVTOLs je nach Flugbetrieb und Passagierpreis zusammen und verdeutlicht den starken Einfluss dieser beiden Parameter auf die Einnahmen.

Entwicklung eines optimierten eVTOL-Gesamtkonzepts

Tabelle 21: Übersicht der Gesamteinnahmen nach Passagierpreis und Flugbetrieb

PYR über gesamte Lebensdauer	1,00 \$	1,50 \$
autonom	6.195.526,497\$	9.293.389,745\$
nicht-autonom	5.310.451,283	7.965.676,925\$
Δ	885.075,214\$	1.327.712,82\$

Bei einem autonomen Flugbetrieb sind durch das Hinzufügen eines weiteren zahlenden Passagiers für einen Passagierpreis von 1,00 \$/Passagier-km rund 885.000 US-Dollar und für 1,50 \$/Passagier-km über 1.3 Mio. US-Dollar höhere Einnahmen zu erzielen.

Anhand Diagramm 6 und Diagramm 7 wird die wirtschaftliche Tragfähigkeit des technischen und wirtschaftlichen Konzepts aufgezeigt. Dabei werden – sofern vorhanden – die Gewinne auf Basis der Kosten und Einnahmen mithilfe des Kapital- oder Barwerts (*NPV*) sowie der daraus abgeleiteten Amortisationsdauer (*IRC*) je nach Kostenszenario und Passagierpreis dargestellt. Diagramm 6 behandelt dabei den Fall eines autonomen Flugbetriebs.

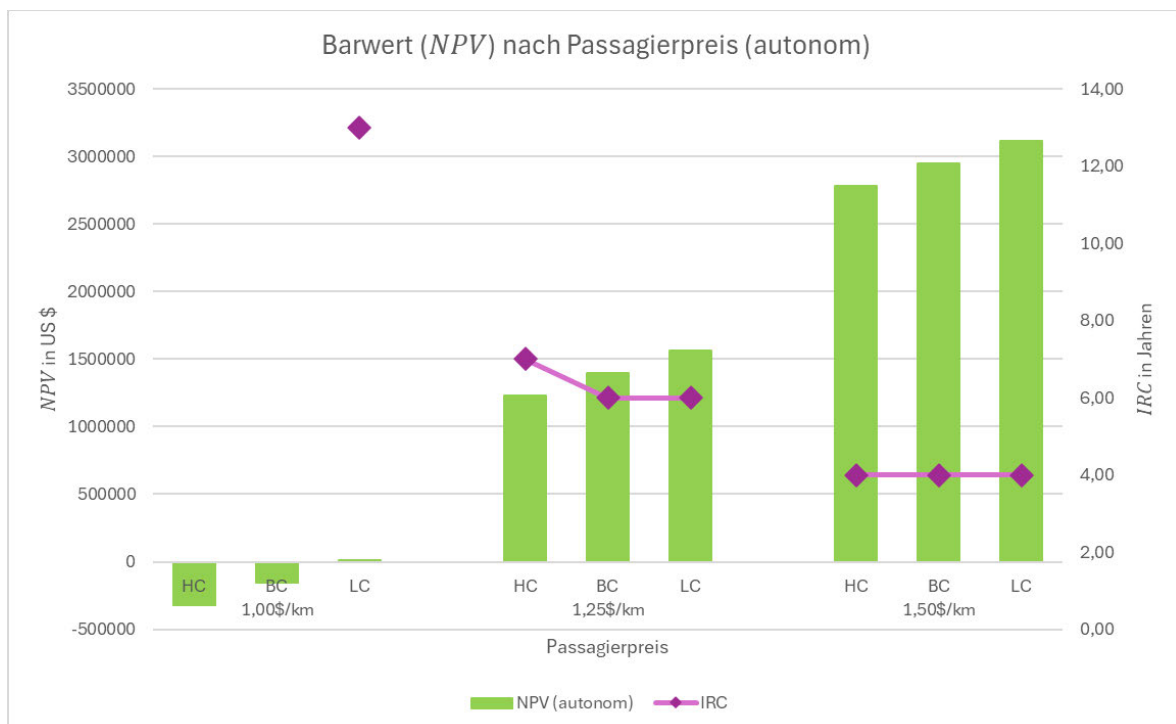


Diagramm 6: Barwert (NPV) und Amortisationsdauer (IRC) nach Passagierpreis beim autonomen Flugbetrieb

Eine Amortisierung muss hierbei nach Definition des Kostenmodells in Abschnitt 2.3.5 innerhalb der Lebensdauer des eVTOLs und somit nach spätestens 13 Jahren möglich sein. Dementsprechend ist eine Amortisierung gemäß des Kostenmodells nicht möglich, wenn dieser

Entwicklung eines optimierten eVTOL-Gesamtkonzepts

Zeitraum überschritten wird oder kein Gewinn erzielt wird. Es wird deutlich, dass bei einem autonomem Flugbetrieb und bei einem Passagierpreis von 1,00 \$/Passagier-km eine Amortisierung erst nach 13 Jahren und nur für das Szenario mit geringen Kosten möglich ist. Für die beiden anderen Szenarien ergeben sich negative Kapitalwerte. Bei einer Erhöhung des Passagierpreises auf 1,25 \$/Passagier-km kann für alle Kostenszenarien eine Amortisierung innerhalb der Lebensdauer erreicht werden. Bei einem Szenario mit hohen Kosten wird diese nach sieben Jahren erreicht, bei den beiden anderen Szenarien hingegen bereits nach sechs Jahren. Wird der Passagierpreis auf 1,50 \$/Passagier-km festgesetzt, beträgt die Amortisationsdauer bei allen Kostenszenarien vier Jahre.

Diagramm 7 zeigt die Gewinnentwicklung für einen nicht-autonomen Flugbetrieb.

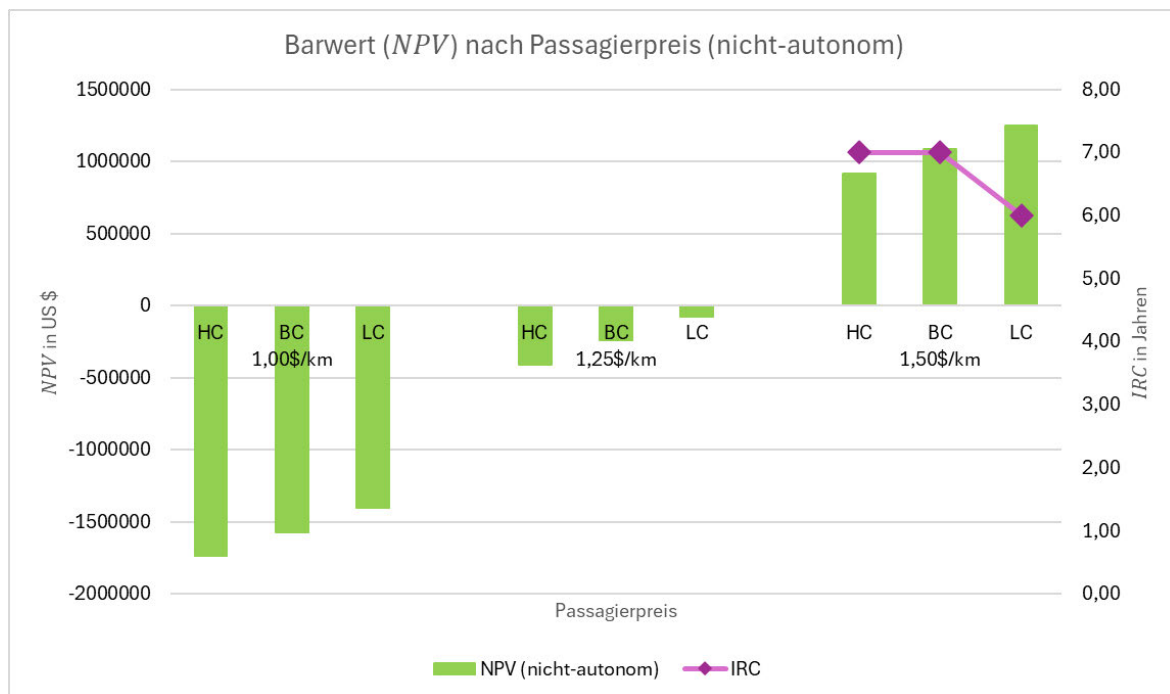


Diagramm 7: Barwert (NPV) und Amortisationsdauer (IRC) nach Passagierpreis beim nicht-autonomen Flugbetrieb

Sowohl bei einem Passagierpreis von 1,00 als auch 1,25 \$/Passagier-km ergeben sich negative Kapitalwerte für alle Kostenszenarien. Somit ist für keine dieser Konstellationen eine Amortisierung innerhalb der Lebensdauer des eVTOLs erzielbar. Erst ab einem Passagierpreis von 1,5\$/Passagier-km ist eine Amortisierung innerhalb des gültigen Zeitbereichs für diesen Fall möglich. Dabei beträgt die Amortisationsdauer für die ersten beiden Kostenszenarien sieben Jahre und im Falle von geringen Kosten sechs Jahre.

In Diagramm 8 und Diagramm 9 sind die nivellierten Kosten nach Kostenszenario und Wahl des Flugbetriebs zu sehen. Die nivellierten Kosten beschreiben die durchschnittlichen Kosten pro Leistungseinheit und Passagier – gemessen an der insgesamt zurückgelegten Strecke – unter Einbeziehung aller relevanten Kosten und zeitlicher Wertverluste über die gesamte

Entwicklung eines optimierten eVTOL-Gesamtkonzepts

Lebensdauer des eVTOLs. Sie ermöglichen eine quantitative Grundlage zur Bewertung der Wirtschaftlichkeit verschiedener Technologien.

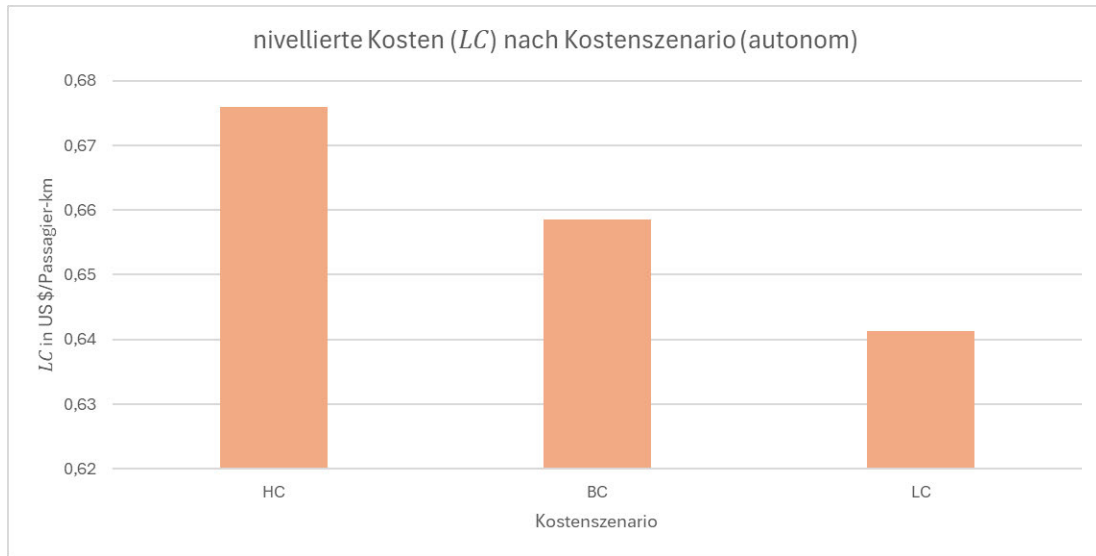


Diagramm 8: Nivellierte Kosten (LC) nach Kostenszenario beim autonomen Flugbetrieb

Die nivellierten Kosten bei einem autonomen Flugbetrieb bewegen sich je nach Kostenszenario zwischen 0,641 bis 0,676 \$/Passagier-km. Beim nicht-autonomen Flugbetrieb belaufen sich die nivellierten Kosten hingegen in einem Bereich zwischen 0,813 bis 0,853 \$/Passagier-km.

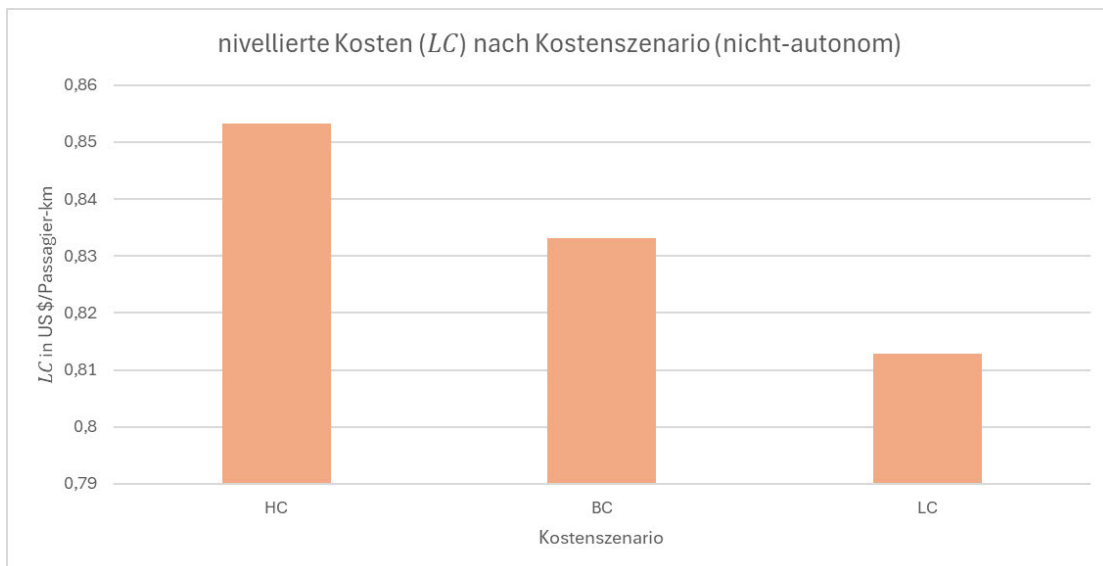


Diagramm 9: Nivellierte Kosten (LC) nach Kostenszenario beim nicht-autonomen Flugbetrieb

Entwicklung eines optimierten eVTOL-Gesamtkonzepts

Tabelle 22 gibt eine Übersicht der nivellierten Kosten für diese beiden Flugbetriebe:

Tabelle 22: Übersicht der nivellierten Kosten (LC) nach Kostenszenario und Flugbetrieb

Nivellierte Kosten (LC) [\$/Passagier-km]	HC	BC	LC
autonom	0,676	0,659	0,641
nicht-autonom	0,853	0,833	0,813

6. Diskussion

Im folgenden Kapitel werden die Plausibilität und Relevanz der Ergebnisse diskutiert. Dabei erfolgt eine Bewertung des technischen und wirtschaftlichen Konzepts, einschließlich eines Vergleichs mit anderen Mobilitätsformen. Abschließend werden die Grenzen und Einschränkungen des zugrunde liegenden Berechnungsmodells erläutert. Besonderes Augenmerk liegt auf den modellbedingten Annahmen und deren Einfluss auf die Belastbarkeit der Resultate.

6.1 Bewertung der Ergebnisse

Ausgehend von den in den vorangegangenen Kapiteln dargestellten Modellierungen erfolgt im Folgenden eine differenzierte Bewertung der technischen und wirtschaftlichen Ergebnisse.

6.1.1 Technische Bewertung des eVTOLs-Konzepts

Das entwickelte technische Konzept basiert auf einer gezielten Optimierung der Energieeffizienz unter Berücksichtigung der Designparameter und der hierfür vorgesehenen Wertebereiche und Annahmen.

Im direkten Vergleich zu den vorgestellten Konzepten verfügt das eigene Konzept bei der geringsten maximalen Abflugmasse über die höchste Nutzlast. Dies liegt vor allem an den Annahmen für die Massenverhältnisse. Die Annahme, dass die Strukturmasse lediglich 30% der Gesamtmasse beträgt, stellt eine große Herausforderung dar, kann aber mit Optimierung der Leichtbauprinzipien erreicht werden. Allerdings ist zu beachten, dass die Umsetzung solcher Leichtbauziele stark von der Materialwahl, der Strukturarchitektur und der Fertigungstechnologie abhängt. Insbesondere der Einsatz von Faserverbundwerkstoffen kann die Zielwerte ermöglichen, ist jedoch mit hohen Material- und Fertigungskosten sowie komplexeren Reparaturprozessen verbunden. Dementsprechend muss auch die Kabinenausstattung im Vergleich zu anderen Konzepten sparsamer ausfallen. Das im eigenen Konzept verwendete Batteriemassenverhältnis berücksichtigt auch die Masse der Batterieschutzstruktur, während bei den betrachteten Konzepten nur Angaben zu der Batteriemasse allein zu finden waren. Die fehlende Kopplung zwischen Gesamttrotorfläche und Abflugmasse im Optimierungsmodell wirft Zweifel an der Umsetzbarkeit des berechneten Werts von 1800 kg für die maximale Abflugmasse auf.

Aus den Ergebnissen der Optimierung für das technische Konzept sowie aus den Untersuchungen aus Kapitel 3.2 geht hervor, dass eine geringere maximale Abflugmasse bei eVTOLs bevorzugt wird. Eine geringere Abflugmasse wird sowohl aus technischen als auch ökonomischen Gründen präferiert. Je höher die Abflugmasse, desto mehr Leistung wird zum Starten benötigt. Dementsprechend steigt der Energiebedarf und es muss mehr Masse für den Antriebsstrang verwendet werden. Dies führt wiederum zu einer erhöhten Lärmemission, was

Diskussion

insbesondere mit Blick auf den vorgesehenen Einsatz im urbanen Raum einen limitierenden Faktor darstellt. Da eVTOLs im urbanen Raum agieren, darf eine gewisse Lärmschwelle nicht überschritten werden. Dabei ist zu berücksichtigen, dass die Lärmemission nicht nur von der Gesamtleistung, sondern auch von der Rotoranzahl, -größe und -drehzahl beeinflusst wird [46]. Da im Rahmen des Modells lediglich die Gesamttrotorfläche berücksichtigt wurde, wäre für eine fundiertere Analyse eine detailliertere Betrachtung der einzelnen Rotorparameter wie Anzahl, Durchmesser und Drehzahl erforderlich. Eine höhere Abflugmasse geht zudem noch mit höheren Herstellungskosten der eVTOL-Zelle und mit höheren fixen Wartungskosten einher, da diese an die Masse des eVTOLs gekoppelt sind. Da die Herstellungskosten der eVTOL-Zelle im ersten Jahr über 75% der Gesamtkosten ausmachen, ist eine geringere maximale Abflugmasse auch aus ökonomischer Sicht positiv zu bewerten.

Bei den Ergebnissen zur maximalen Reichweite und zum Energiebedarf ist zu berücksichtigen, dass keine Energiereserven einkalkuliert wurden. In der realen Flugplanung müssen jedoch verpflichtend Energiereserven für Ausweichmanöver oder unvorhergesehene Wetterbedingungen berücksichtigt werden. Die reale Reichweite würde unter Berücksichtigung dieser Faktoren also deutlich unter dem berechneten Maximalwert liegen. Dies erklärt auch die berechneten Reichweiten für die in Kapitel 4 betrachteten eVTOL-Konzepte. Des Weiteren wurden im Berechnungsmodell keine weiteren Energieverbraucher wie beispielsweise die Klimatisierung oder weitere Systeme berücksichtigt.

Zur Gewährleistung der Vergleichbarkeit mit den anderen betrachteten eVTOL-Konzepten wurde im Rahmen der Optimierung eine Energiedichte angesetzt, die dem höchsten Wert der Vergleichsmodelle entspricht, jedoch nicht darüber hinausgeht. Da die Energiedichte ein Schlüsselfaktor zur Kommerzialisierung von eVTOLs ist, gilt es diese in jedem Fall zu erhöhen. Verschiedene Studien setzen hierfür eine Energiedichte von 400 Wh/kg voraus [21], [25]. Für zukünftige Batterien wird erwartet, dass sie auf Basis von Li-Schwefel eine Energiedichte von über 400 Wh/kg erreichen werden [47]. Bei gleichbleibender vorgesehener Kapazität der Batterie und einer höheren Energiedichte muss weniger Masse für die Batterie verwendet werden. In diesem Fall kann entweder die maximale Abflugmasse gesenkt oder zusätzliche Masse für Nutzlast und Struktur vorgesehen werden.

Insgesamt zeigt sich, dass das entwickelte technische Konzept in vielen Aspekten Potenzial bietet, jedoch auch mit signifikanten Unsicherheiten und technischen Herausforderungen behaftet ist. Um das Konzept weiter zu verbessern, ist es notwendig, vereinfachende Annahmen zu reduzieren und die Wechselwirkungen zwischen den Designparametern präziser zu modellieren, um realitätsnähere und aussagekräftige Ergebnisse zu erhalten.

6.1.2 Diskussion des wirtschaftlichen Konzepts

In Abschnitt 5.2 wurden die Ergebnisse für die Kenngrößen aus dem in 2.3.5 beschriebenen Kostenmodell vorgestellt. Die Rahmenbedingungen für die Berechnungen wurden dabei in Abschnitt 3.3.4 definiert.

Um einen realitätsnahen Flugbetrieb abzubilden, wurde bei den Betriebsparametern auf Vergleichswerte aus der Literaturrecherche zurückgegriffen. Hierbei wurde ein Passagierpreis im Bereich von 1 bis 1,5 \$/km gewählt, da dieser mit bestehenden Mobilitätslösungen vergleichbar ist und somit eine potenzielle Konkurrenzfähigkeit im Hinblick auf das Kostenniveau ermöglicht. Die Darstellung des berechneten Kapitalwertes und der resultierenden Amortisationsdauer zeigen, dass ein Passagierpreis von 1 \$/km, wie er beispielsweise bei Taxis vorzufinden ist, nicht wirtschaftlich tragfähig ist. Für einen nicht-autonomen Flugbetrieb ergeben sich je nach Kostenszenario negative Kapitalwerte von über 1,5 Mio. \$. Ähnlich ist es bei einem Passagierpreis von 1,25 \$/km. Auch hier ist bei einem nicht-autonomen Flugbetrieb für kein Kostenszenario eine Amortisierung innerhalb der Lebensdauer des eVTOLs möglich, welche mit 13 Jahren realitätsnah ist [25]. Erst ab einem Preis von 1,5 \$/km kann bei einem nicht-autonomen Flugbetrieb eine Amortisierung innerhalb der Lebensdauer erzielt werden, welche sich bei einem Szenario mit hohen Kosten auf sieben Jahre beläuft und bei geringen Kosten auf sechs. Eine Amortisierung, die erst nach der Hälfte der gesamten Nutzungsdauer eintritt, birgt ein hohes wirtschaftliches Risiko und senkt damit die Investorenattraktivität.

Auch ohne vorherige Untersuchungen war absehbar, dass ein autonomer Flugbetrieb bevorzugt wird und langfristig notwendig ist. Ziel dieser Unterscheidung war es jedoch vielmehr aufzuzeigen, ob das beschriebene eVTOL-Konzept unter den ausgewählten Rahmenbedingungen auch bei einem nicht autonomen Flugbetrieb wirtschaftlich tragfähig sein kann. Dies ist gerade unter dem Hintergrund wichtig, da ein autonomer Betrieb zu Betriebsbeginn aufgrund fehlender Zulassungen nicht möglich ist. Die Inbetriebnahme erfolgt mit Piloten und eine Entwicklung zu einem vollautonomen System erfolgt in Etappen. Der Anfang erfolgt mit Unterstützungssystemen, bei denen der Pilot die automatische Steuerung manuell übersteuern kann. Diese Übergangsformen senken die Zertifizierungshürden. Ein voll autonomer Flugbetrieb kann jedoch erst etabliert werden, wenn die gesetzlich-regulatorischen Anforderungen erfüllt sind, die bei eVTOLs bislang noch nicht klar definiert wurden. Des Weiteren bedarf es hier auch auf Akzeptanz und Vertrauen beim Konsumenten [25].

Bei einem autonomen Flugbetrieb und einem Passagierpreis von 1,5 \$ kann eine Amortisierung unter den gegebenen Rahmenbedingungen bereits nach vier Jahren erreicht werden. Dies verdeutlicht, dass eVTOLs langfristig autonom fliegen müssen, um konkurrenzfähige Passagierpreise anbieten zu können. Selbstverständlich kann der Passagierpreis deutlich

Diskussion

höher angesetzt werden. Dabei ist jedoch zu beachten, dass mit steigenden Preisen auch das Risiko sinkender Nachfrage einhergeht, da nicht alle potenziellen Konsumenten bereit sein werden, einen entsprechend hohen Preis zu zahlen. Dies könnte die Auslastung und damit die Wirtschaftlichkeit des Betriebs erheblich beeinträchtigen. Nach Schätzungen von Joby Aviation geht man im Jahr 2026 von einem Passagierpreis von 3 \$/Meile auf Basis einer Strecke von 24 Meilen aus. Mit geschätzten Kosten pro Passagier-Meile von 0,86 \$ seitens Joby Aviation kommt dieser Wert sehr nahe an die berechneten nivellierten Kosten aus dieser Arbeit [48]. Diese belaufen sich, wie Tabelle 22 zu entnehmen ist, auf 0,853 \$ für einen nicht-autonomen Flugbetrieb und einem Szenario mit hohen Kosten.

Für die Kostenrechnung wurde ein Flugstrecke von 100 km pro Mission angenommen. Dieser Wert wurde so hoch angesetzt, da Erfahrungen aus der Praxis zeigen, dass bei größeren Strecken ein höherer Nutzladefaktor erreicht werden kann [16]. Diese Erfahrungen basieren jedoch auf Beobachtungen bei konventionellen Flugzeugen und sind nicht unbedingt gleichwertig auf eVTOLs zu übertragen. Andere Untersuchungen gehen für eVTOLs von einem Nutzladefaktor von 67% bei bereits viel kürzeren Strecken aus [25]. Hierbei muss jedoch beachtet werden, dass mit kürzeren Strecken zwar die Anzahl an möglichen Missionen pro Tag steigt, jedoch gleichzeitig auch die Zeit am Boden für das Boarding und Deboarding, bei denen keine Einnahmen generiert werden.

Die Stromkosten für die Berechnung der Energiekosten orientieren sich im hier verwendeten Kostenmodell an den Preisen aus dem US-Amerikanischen Raum. Hier lagen die Kosten für ein Szenario mit hohen Kosten bei 0,13 \$/kWh und bei geringen Kosten bei 0,05 \$/kWh. Diese können je nach Einsatzort höher ausfallen, haben jedoch aufgrund des relativ geringen Anteils der Energiekosten an den Gesamtkosten nur einen begrenzten Einfluss, wie in Diagramm 1 ersichtlich ist.

Einen signifikanten Einfluss auf die Gesamtkosten haben die Herstellungskosten der eVTOL-Zelle, wie in Abschnitt 5.2 zu sehen. Um die Herstellungskosten zu senken, muss das Produktionsvolumen gesteigert werden. Geringe Stückzahlen sind in der Luftfahrt üblich. Obwohl kleine Flugzeuge und Helikopter ähnlich in Größe, Masse und Komplexität zu einem Auto sind, kosten sie etwa 20-mal so viel. Ziel ist es somit auch bei eVTOLs das Produktionsvolumen mit der Zeit zu steigern. Untersuchungen von Uber zeigen, dass die Herstellungskosten von eVTOLs bei einer Stückzahl von 500 pro Jahr im Vergleich zu 100 pro Jahr um bis zu 50% gesenkt werden können. Langfristig wird bei Stückzahlen von 5000 pro Jahr sogar eine Reduktion von über 80% prognostiziert [25].

Abschließend ist festzuhalten, dass einige Annahmen des Kostenmodells mit Unsicherheiten behaftet sind. Da es sich bei eVTOLs um neue Systeme handelt, liegen bislang kaum

belastbare Daten über die Entwicklung der Wartungskosten im Lebenszyklus vor. Diese könnten je nach Materialwahl, Modularität und Komplexität variieren. Auch Infrastrukturkosten wie für Vertiports und Ladeeinrichtungen könnten durch Skalierungseffekte sinken, sofern ein dichteres Netzwerk aufgebaut wird. Derzeit fehlen jedoch belastbare Daten über Skalierungspotenziale dieser neuen Infrastrukturform.

Insgesamt zeigt die wirtschaftliche Betrachtung, dass das untersuchte eVTOL-Konzept bei dem angenommenen Preisbereich nur unter bestimmten Rahmenbedingungen wirtschaftlich tragfähig ist, welche im eigenen Konzept einen autonomen Betrieb bei einem Passagierpreis von 1,5 \$/km voraussetzen. Insbesondere die Autonomie des Flugbetriebs, ein ausreichend hoher Auslastungsgrad sowie Skaleneffekte in der Produktion und Infrastruktur wirken sich maßgeblich auf die Wirtschaftlichkeit aus. Vor dem Hintergrund der regulatorischen Unsicherheiten und der fehlenden Erfahrungswerte bleibt die Rentabilität stark vom weiteren technologischen und marktwirtschaftlichen Fortschritt abhängig. Diese Aspekte sind bei der Bewertung des Gesamtkonzepts unbedingt zu berücksichtigen

6.1.3 Erreichte Ziele und Grenzen des Modells

Die in Tabelle 2 vorab definierten Anforderungen an das Berechnungsmodell wurden alle ausnahmslos erfüllt. Die Berechnung des Energiebedarfs und der Reichweite wird unter den beschriebenen Annahmen und Vereinfachungen erfüllt. Es können jedoch präzisere Ergebnisse erreicht werden, indem Vereinfachungen reduziert werden. Besonders hervorzuheben ist jedoch, dass das Modell zur Berechnung akzeptabler Werte abhängig von validierten Parameterwerten ist. Die Abschätzung des Energiebedarfs der in dieser Arbeit ausgewählten eVTOL-Konzepte beruht auf teilweise nicht vom Hersteller validierten Parameterwerten und ist somit mit Unsicherheiten behaftet. Auch die Berechnung der Gleitzahl beruht teilweise auf Annahmen bezüglich von Widerstandsbeiwerten, die nicht präzise auf das zu untersuchende Konzept zugeschnitten sind.

Die Optimierung basiert auf einem Grid-Search-Ansatz, bei dem über 13 Millionen Kombinationen getestet wurden. Dies stellt zwar eine umfassende Parametervariation dar, jedoch wurden die Bewertungskriterien neben den festgelegten Randbedingungen auf nur eine Zielfunktion reduziert: dem Energieverbrauch bzw. der Energieeffizienz. Weitere wichtige Aspekte wie die Veränderung von aerodynamischen Effekten, die Lärmentwicklung und weitere reale Wechselwirkungen zwischen den einzelnen Parametern wurden nicht berücksichtigt. Beispielsweise können bei einer höheren Streckung durch eine höhere Gleitzahl Veränderungen im Flugverhalten auftreten und die Steuerbarkeit verschlechtern oder erschweren. Die technische Bewertung ist daher als vorläufig zu verstehen. Die Gesamttrotorfläche wurde im Rahmen der Optimierung nicht dynamisch an die Abflugmasse gekoppelt, sondern unabhängig davon

Diskussion

skaliert. Diese fehlende Kopplung stellt ein wesentliches Defizit im Hinblick auf die Realisierbarkeit der maximalen Abflugmasse dar. Zudem handelt es sich bei einem Grid-Search Algorithmus um einen Brute Force-Ansatz. Dieser Ansatz gewährleistet zwar, dass globale Maxima oder Minima im betrachteten Parameterraum identifiziert werden können, ist jedoch mit steigender Komplexität rechenintensiv und nicht adaptiv. Im Gegensatz dazu bieten lernende Optimierungsverfahren wie genetische Algorithmen oder Verfahren aus dem Bereich des maschinellen Lernens potenziell effizientere und zukunftsorientierende Lösungswege, da sie Zusammenhänge im Parameterraum erkennen und gezielt nutzen können.

Die Ergebnisse liefern wertvolle Hinweise auf Wirkzusammenhänge und Optimierungspotenziale, können aber nicht als belastbare Prognose für eine konkrete technische Umsetzung verstanden werden. Vielmehr ist das Modell ein nützliches Werkzeug zur Abschätzung und Sensitivitätsanalyse – nicht aber zur technisch detaillierten Auslegung eines marktfähigen eVTOLs.

7. Fazit und Ausblick

Zum Abschluss dieser Arbeit werden die zentralen Ergebnisse zusammengefasst, insbesondere im Hinblick auf die technischen und betrieblichen Voraussetzungen, die eVTOLs erfüllen müssen, um als wirtschaftlich tragfähige Mobilitätslösung zur Entlastung urbaner Verkehrssysteme beitragen zu können. Darüber hinaus wird ein Ausblick auf zukünftige Entwicklungspotenziale und Forschungsbedarfe gegeben.

7.1 Erforderliche Rahmenbedingungen für wirtschaftlich tragfähige eVTOLs

Das in dieser Arbeit entwickelte eVTOL-Gesamtkonzept erfordert einen autonomen Flugbetrieb, um bei einem konkurrenzfähigen Preis von 1,5 \$/km pro Passagier wirtschaftlich tragfähig zu sein. Unter diesen Rahmenbedingungen und den Annahmen im Kostenmodell kann eine Amortisierung nach vier Jahren erreicht werden. Für einen nicht-autonomen Flugbetrieb ist unter denselben Bedingungen erst nach sieben Jahren mit einer Amortisierung zu rechnen. Zudem zeigt sich, dass aktuelle eVTOL-Konzepte durch geringere Nutzlasten deutlich höhere Passagierpreise ansetzen müssen, um positive Kapitalwerte und eine Amortisation innerhalb kurzer Zeit zu erzielen. Hersteller kalkulieren daher mit Preisen von über 3 \$/km pro Passagier. Da die Wirtschaftlichkeit nicht nur vom Preisniveau abhängt, sondern auch vom subjektiven Wert, den Kunden dem Reisen mit eVTOLs beimessen, besteht weiterer Bedarf an Marktforschung. Neben dem Preis können auch weitere Faktoren wie Reisezeit, Exklusivität und Umweltbewusstsein die Nachfrage stark beeinflussen können. Die Markteinführung wird zudem stark davon abhängen, wie schnell Vertrauen in die Sicherheit, Zuverlässigkeit und den Nutzen von eVTOLs aufgebaut werden kann.

Zusammenfassend lassen sich zentrale Einflussfaktoren benennen, die die wirtschaftliche Tragfähigkeit von eVTOLs maßgeblich bestimmen. Vorreiter dabei ist die Weiterentwicklung der Batterietechnologie. Es müssen höhere Energiedichten erreicht werden, um die Batteriemasse möglichst gering zu halten und die Effizienz zu erhöhen. Auch wenn ein autonomer Betrieb anfangs nicht realisierbar ist, ist er langfristig erforderlich, um die Produktivität zu steigern. Um die Herstellungskosten zu senken, muss das Produktionsvolumen erhöht werden und eine Annäherung an die Automobilindustrie diesbezüglich angestrebt werden. Da die Gesamtmasse von eVTOLs einen erheblichen Einfluss auf die Betriebseffizienz hat, ist der Leichtbau weiterhin zu verbessern. Außerdem sind Strecken mit ausreichend hoher Nachfrage zu erforschen, um den Nutzladefaktor zu maximieren. Hierbei müssen neben dem Personentransport auch weitere Einsatzmöglichkeiten untersucht werden. Zudem ist zu berücksichtigen, dass eVTOLs nur dann wirtschaftlich tragfähig operieren können, wenn sie in ein ganzheitliches Mobilitätskonzept eingebettet werden. Schnittstellen zu konventionellen Verkehrsmitteln, digitale Buchungsplattformen und sinnvolle Vertiport-Standorte sind entscheidend für eine

hohe Auslastung und Akzeptanz. Insgesamt zeigt sich, dass technologische Fortschritte und wirtschaftliche Effizienz eng miteinander verknüpft sind. Um eVTOLs langfristig als wirtschaftlich tragfähige Verkehrslösung zu etablieren, müssen nicht nur technische, sondern auch gesellschaftliche und strukturelle Rahmenbedingungen geschaffen und kontinuierlich weiterentwickelt werden.

7.2 Weiterentwicklung von eVTOLs und Modellverbesserungen

Die erforderlichen Verbesserungen zur Weiterentwicklung von eVTOLs gehen aus den vorherigen Kapiteln hervor. Eine zentrale Rolle spielt dabei die Batterietechnologie. Neben der Energiedichte sind auch die Lebensdauer sowie die Ladeleistung weiterhin zu verbessern. Fortschritte im Leichtbau sind entscheidend, um weiterhin das Verhältnis von Strukturmasse zur Gesamtmasse zu verbessern. Gleichzeitig darf beim Einsatz von Leichtbaumaterialien und neuen Fertigungstechnologien Wartungs- und Kosteneffizienz nicht aus den Augen verloren werden. Darüber hinaus ist der Übergang zum autonomen Flugbetrieb ein Schlüsselfaktor zur Skalierung von eVTOLs. Die Entwicklung und Zulassung zuverlässiger Autonomiesysteme bleibt daher ein zentrales Zukunftsthema.

Das Berechnungsmodell bietet Potenzial für weiterführende Verbesserungen. Zum einen können die unterschiedlichen Flugphasen noch detaillierter differenziert werden, um den Energieverbrauch noch präziser abbilden zu können. Eine solche Betrachtung könnte die Einbeziehung von Bodeneffekten vereinfachen. Zudem könnten zusätzliche Energieverbraucher wie etwa für Kabinenklimatisierung oder Avionik integriert werden, um eine noch realistischere Energiebedarfsabschätzung zu ermöglichen.

Ein weiteres Ziel zukünftiger Modellverbesserungen ist die realitätsnähere Abbildung der Wechselwirkungen zwischen den Designparametern. Im bisherigen Modell wurden einige Parameter unabhängig voneinander variiert, obwohl sie in realen Systemen in komplexer Abhängigkeit zueinanderstehen. Diese vereinfachte Betrachtung kann dazu führen, dass technisch nicht umsetzbare oder inkonsistente Konfigurationen als optimal erscheinen.

Auch der verwendete Optimierungsansatz auf Basis eines Grid-Search-Ansatzes ließe sich durch lernende Algorithmen wie genetische Optimierungen oder Verfahren des maschinellen Lernens erweitern. Solche Ansätze könnten Zusammenhänge zwischen den Designparametern effizienter erkennen und gezielter zu optimalen Lösungen führen, ohne alle Kombinationen untersuchen zu müssen. Dies wäre insbesondere bei höherdimensionalen Problemstellungen mit wachsender Komplexität von Vorteil.

8. Literaturverzeichnis

- [1] Globe Newswire, „eVTOL Aircrafts Business Research Report 2024: Global Market to Reach \$17.7 Billion by 2030 - Ongoing Development of Regulatory Frameworks Paves Way for Adoption“, GlobeNewswire News Room. Zugegriffen: 21. Dezember 2024. [Online]. Verfügbar unter: <https://www.globenewswire.com/news-release/2024/10/22/2967055/28124/en/eVTOL-Aircrafts-Business-Research-Report-2024-Global-Market-to-Reach-17-7-Billion-by-2030-Ongoing-Development-of-Regulatory-Frameworks-Paves-Way-for-Adoption.html>
- [2] The US Sun, „China plans to have 100k flying cars buzzing around cities in just 6yrs“, The US Sun. Zugegriffen: 22. Dezember 2024. [Online]. Verfügbar unter: <https://www.the-sun.com/motors/12985345/china-plans-flying-cars-taxis/>
- [3] L. Reinhold, „Urban Air Mobility: Liliums Scheinwelt“. Zugegriffen: 23. Dezember 2024. [Online]. Verfügbar unter: <https://www.aerokurier.de/elektroflug/lilium-jet-dossier/>
- [4] ohne Verfasser, „Die Flugleistung des Lilium- ‚Jets‘ und eVTOaL´s im Allgemeinen - eine Konzeptberechnung“.
- [5] N. Klußmann und A. Malik, *Lexikon der Luftfahrt*, 4. Aufl. 2018. Berlin, Heidelberg: Springer Berlin Heidelberg, 2018. doi: 10.1007/978-3-662-54040-4.
- [6] X. Hafer und G. Sachs, *Senkrechtstarttechnik*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1982. doi: 10.1007/978-3-642-81724-3.
- [7] A. Filippone, *Flight performance of fixed and rotary wing aircraft*, 1. ed. in Elsevier aerospace engineering series. Amsterdam Heidelberg: Elsevier Butterworth-Heinemann, 2006.
- [8] H. Schlichting und E. Truckenbrodt, *Aerodynamik des Flugzeuges*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001. doi: 10.1007/978-3-642-56911-1.
- [9] G. Brüning, X. Hafer, und G. Sachs, *Flugleistungen: Grundlagen, Flugzustände, Flugabschnitte*, 3., erg. Aufl., Nachdr. in veränd. Ausstattung. in *Klassiker der Technik*. Berlin Heidelberg: Springer, 2006.
- [10] D. P. Raymer, *Aircraft design: a conceptual approach*, 4th ed. in AIAA education series. Reston, Va: American Institute of Aeronautics and Astronautics, 2006.
- [11] K. Engmann, Hrsg., *Technologie des Flugzeuges*, 7., neu Bearbeitete und Erweiterte Auflage. in *Wissen für Profis*. Würzburg: Vogel Communications Group GmbH & Co. KG, 2019.
- [12] G. Strickert, „Faktencheck Multikopter: Ähnlichkeiten und Unterschiede zu etablierten VTOL-Konfigurationen“, gehalten auf der Deutscher Luft- und Raumfahrtkongress 2016, 2016.
- [13] W. Bittner, *Flugmechanik der Hubschrauber: Technologie, das flugdynamische System Hubschrauber, Flugstabilitäten, Steuerbarkeit*, 3. Aufl. 2009. in *VDI-Buch*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009. doi: 10.1007/978-3-540-88972-4.
- [14] R. Brockhaus, W. Alles, und R. Luckner, *Flugregelung*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011. doi: 10.1007/978-3-642-01443-7.
- [15] H. Mensen, *Handbuch der Luftfahrt*. in *Engineering online library*. Berlin Heidelberg: Springer, 2003.
- [16] Dieter Scholz, *Flugzeugentwurf*.
- [17] Odell, T.T., „Boeing HSCT OpCost Methodology“, Boeing Commercial Airplane Group, Seattle, 6-1442-MES-HSCT-002–93, 1993.
- [18] Carubba, E., „Integrating Life Cycle Cost and Cost of Ownership in the Commercial Sector“, gehalten auf der Proceedings of the Annual Reliability and Maintainability Symposium, IEEE, 1992.
- [19] Association of European Airlines, „Short-Medium Range Aircraft AEA Requirements“, Association of European Airlines (AEA), Brüssel, G(T)5656, 1989.
- [20] D. P. Raymer, *Aircraft design: a conceptual approach*, 4. ed. in AIAA education series. Reston, Va: American Institute of Aeronautics and Astronautics, 2006.

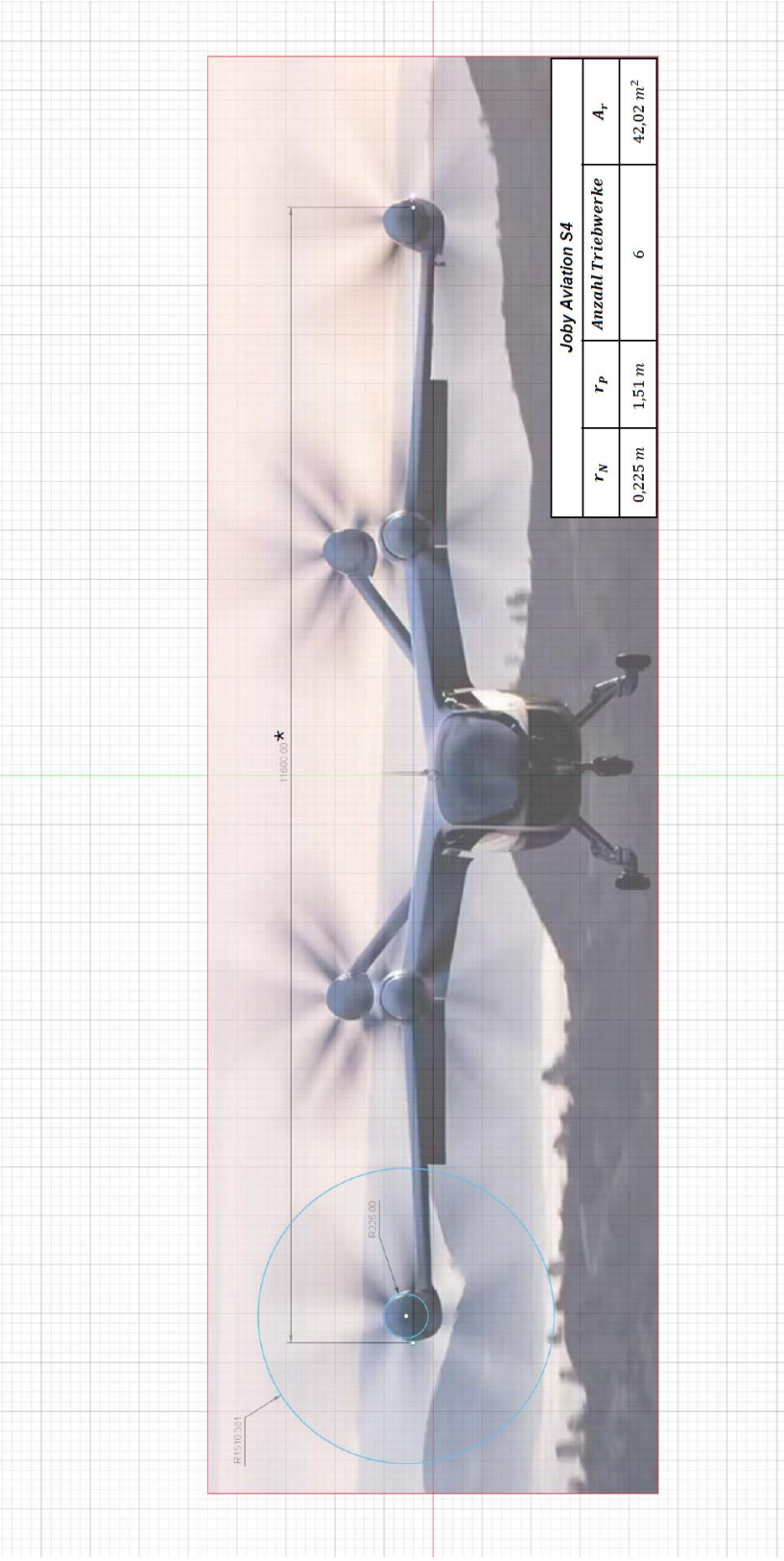
- [21] M. Liu *u. a.*, „Exploring the key technologies needed for the commercialization of electric flying cars: A leveled cost and profitability analysis“, *Energy*, Bd. 303, S. 131991, Sep. 2024, doi: 10.1016/j.energy.2024.131991.
- [22] B. Bender, K. Gericke, G. Pahl, und W. Beitz, Hrsg., *Pahl/Beitz Konstruktionslehre: Methoden und Anwendung erfolgreicher Produktentwicklung*, 9. Auflage. in Springer eBook Collection. Berlin [Heidelberg]: Springer Vieweg, 2021. doi: 10.1007/978-3-662-57303-7.
- [23] J. D. Anderson, *Introduction to flight*, Eighth edition. New York, NY: McGraw-Hill Education, 2016.
- [24] M. Inden, *Einfach Python: gleich richtig programmieren lernen*, 1. Auflage. Heidelberg: dpunkt.verlag, 2022.
- [25] J. Holden und N. Goel, „Fast-Forwarding to a Future of On-Demand Urban Air Transportation“, Uber Technologies Inc., San Francisco, Okt. 2016.
- [26] „Lilium“. Zugegriffen: 7. März 2025. [Online]. Verfügbar unter: https://jet.lilium.com/?_gl=1*7u95z2*_up*MQ..*_ga*NDE3NDc1NDMxLjE3NDEzNTM3MDA.*_ga_9YC7ETNZ98*MTc0MTM1MzcwMC4xLjEuMTc0MTM1MzgyMS4wLjAuMA..
- [27] R. K. Ahluwalia, J.-K. Peng, X. Wang, D. Papadias, und J. Kopasz, „Performance and cost of fuel cells for urban air mobility“, *International Journal of Hydrogen Energy*, Bd. 46, Nr. 36, S. 36917–36929, 2021, doi: 10.1016/j.ijhydene.2021.08.211.
- [28] R. Schmuch, R. Wagner, G. Hörpel, T. Placke, und M. Winter, „Performance and cost of materials for lithium-based rechargeable automotive batteries“, *Nature Energy*, Bd. 3, Nr. 4, S. 267–278, Apr. 2018, doi: 10.1038/s41560-018-0107-2.
- [29] „Joby Aviation S4 (production prototype)“. Zugegriffen: 1. Mai 2025. [Online]. Verfügbar unter: <https://evtol.news/joby-aviation-s4-production-prototype>
- [30] „Our Story | Joby“. Zugegriffen: 1. Mai 2025. [Online]. Verfügbar unter: <https://www.jobyaviation.com/about/>
- [31] „Joby Aviation S4 2.0 (pre-production prototype)“. Zugegriffen: 3. Mai 2025. [Online]. Verfügbar unter: <https://evtol.news/joby-s4>
- [32] P. S. Moller, „Review of Selected Advanced Air Mobility Aircraft“, Dixon, CA, USA, Jan. 2024.
- [33] „Archer Aviation Midnight (production aircraft)“. Zugegriffen: 4. März 2025. [Online]. Verfügbar unter: <https://evtol.news/archer>
- [34] „Archer | Company“. Zugegriffen: 4. März 2025. [Online]. Verfügbar unter: <https://archer.com/company>
- [35] „Archer | Midnight Aircraft“. Zugegriffen: 4. März 2025. [Online]. Verfügbar unter: <https://archer.com/aircraft>
- [36] R. Nay, „Archer Aviation Midnight (Production Model)“, AeroCrunch. Zugegriffen: 3. Mai 2025. [Online]. Verfügbar unter: <https://aerocrunch.com/archer-aviation-midnight-production-model/>
- [37] „Archer | Electric Air Taxis“. Zugegriffen: 4. März 2025. [Online]. Verfügbar unter: <https://archer.com/>
- [38] „Lilium Company - Lilium“. Zugegriffen: 7. März 2025. [Online]. Verfügbar unter: https://lilium.com/company?_gl=1*1xyocno*_up*MQ..*_ga*NDE3NDc1NDMxLjE3NDEzNTM3MDA.*_ga_9YC7ETNZ98*MTc0MTM1MzcwMC4xLjEuMTc0MTM1MzgwNS4wLjAuMA..
- [39] „Lilium Jet (7-seater)“. Zugegriffen: 7. April 2025. [Online]. Verfügbar unter: <https://evtol.news/lilium-gmbh-lilium-jet-7-seater>
- [40] „Elektroflugzeugbauer Lilium meldet zum zweiten Mal Insolvenz an“, BR24. Zugegriffen: 7. April 2025. [Online]. Verfügbar unter: <https://www.br.de/nachrichten/wirtschaft/elektroflugzeugbauer-lilium-meldet-zum-zweiten-mal-insolvenz-an,UdS4Y1x>
- [41] „Lilium Unveils SPAC Deal, Seven-Seat eVTOL | Aviation Week Network“. Zugegriffen: 3. Mai 2025. [Online]. Verfügbar unter: <https://aviationweek.com/air-transport/aircraft-propulsion/lilium-unveils-spac-deal-seven-seat-evtol>
- [42] Lilium GmbH, „Architectural performance assessment of an electric vertical take-off and landing jet“, Lilium, 2021.
- [43] „Vertical Aerospace VX4 (production model)“, Electric VTOL News. Zugegriffen: 29. März 2025. [Online]. Verfügbar unter: <https://evtol.news/vertical-aerospace-va-1x>

- [44] „Vertical Aerospace · Pioneering Electric Aviation“, Vertical Aerospace. Zugegriffen: 1. Mai 2025. [Online]. Verfügbar unter: <https://vertical-aerospace.com/>
- [45] „Vertical Announces the Opening of the Vertical Energy Centre, the UK’s Most Advanced Aerospace Battery Facility“. Zugegriffen: 3. Mai 2025. [Online]. Verfügbar unter: <https://www.businesswire.com/news/home/20230314005332/en/Vertical-Announces-the-Opening-of-the-Vertical-Energy-Centre-the-UKs-Most-Advanced-Aerospace-Battery-Facility>
- [46] B. Smith und F. Gandhi, „High Solidity, Low Tip-Speed Rotors for Reduced eVTOL Tonal Noise“, Fort Worth, Texas, USA: Vertical Flight Society, Mai 2022.
- [47] P.-G. Redaktion, „Energiedichte von Lithium-Ionen-Akkus: Bedeutung“, Profi-Guide. Zugegriffen: 10. Mai 2025. [Online]. Verfügbar unter: <https://www.jh-profishop.de/profi-guide/lithium-ionen-akku-energiedichte/>
- [48] C. Stonor, „Do Flying Taxi Projected Flight Costs Add Up Or Is It All “Pie In The Sky?” - eVTOL Insights“. Zugegriffen: 12. Mai 2025. [Online]. Verfügbar unter: <https://evtolinsights.com/2022/05/do-flying-taxi-projected-flight-costs-add-up-or-is-it-all-pie-in-the-sky/>, <https://evtolinsights.com/2022/05/do-flying-taxi-projected-flight-costs-add-up-or-is-it-all-pie-in-the-sky/>

9. Anhang

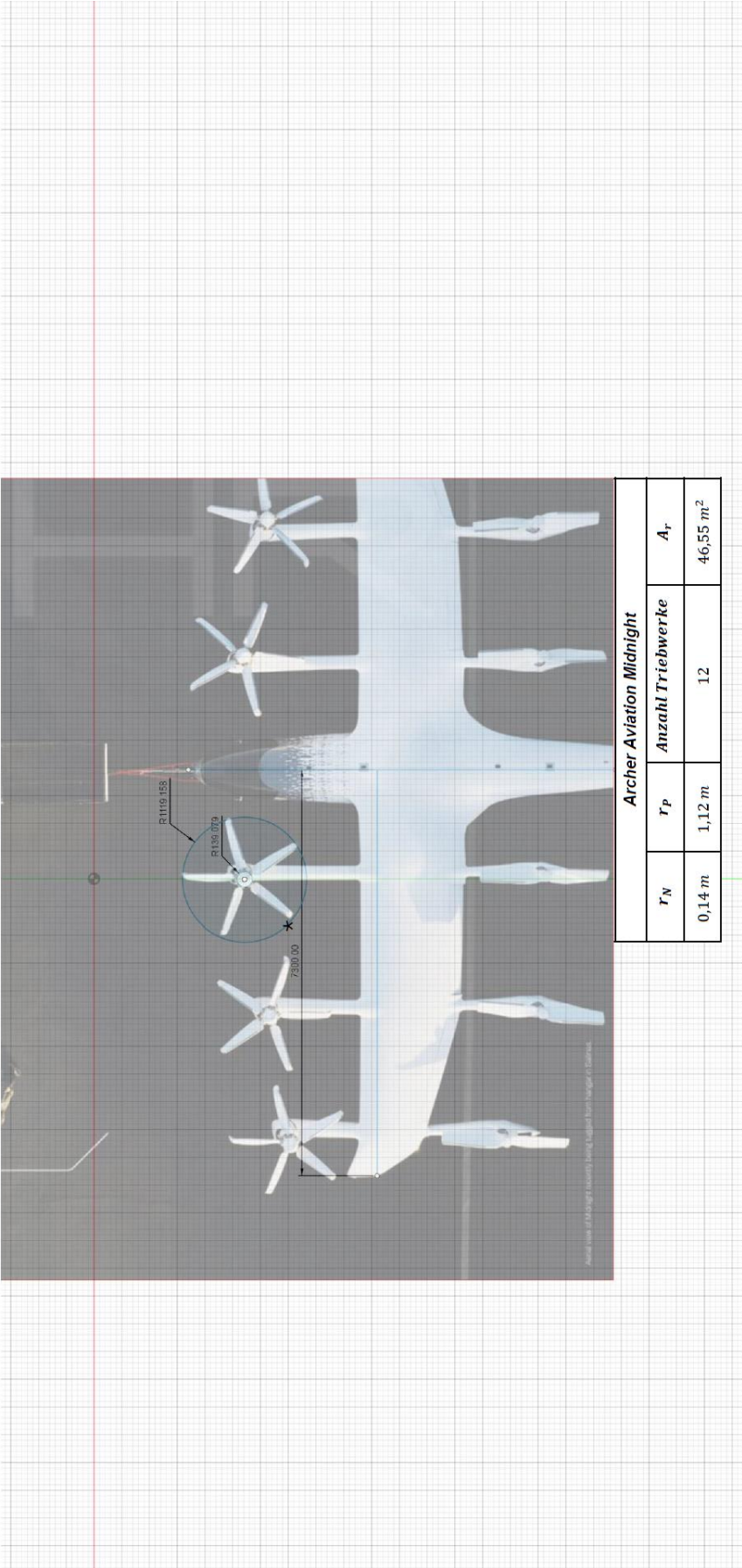
9.1 Anhang 1: Ermittlung der Gesamtrotorfläche einzelner eVTOL-Konzepte

9.1.1 Joby Aviation S4



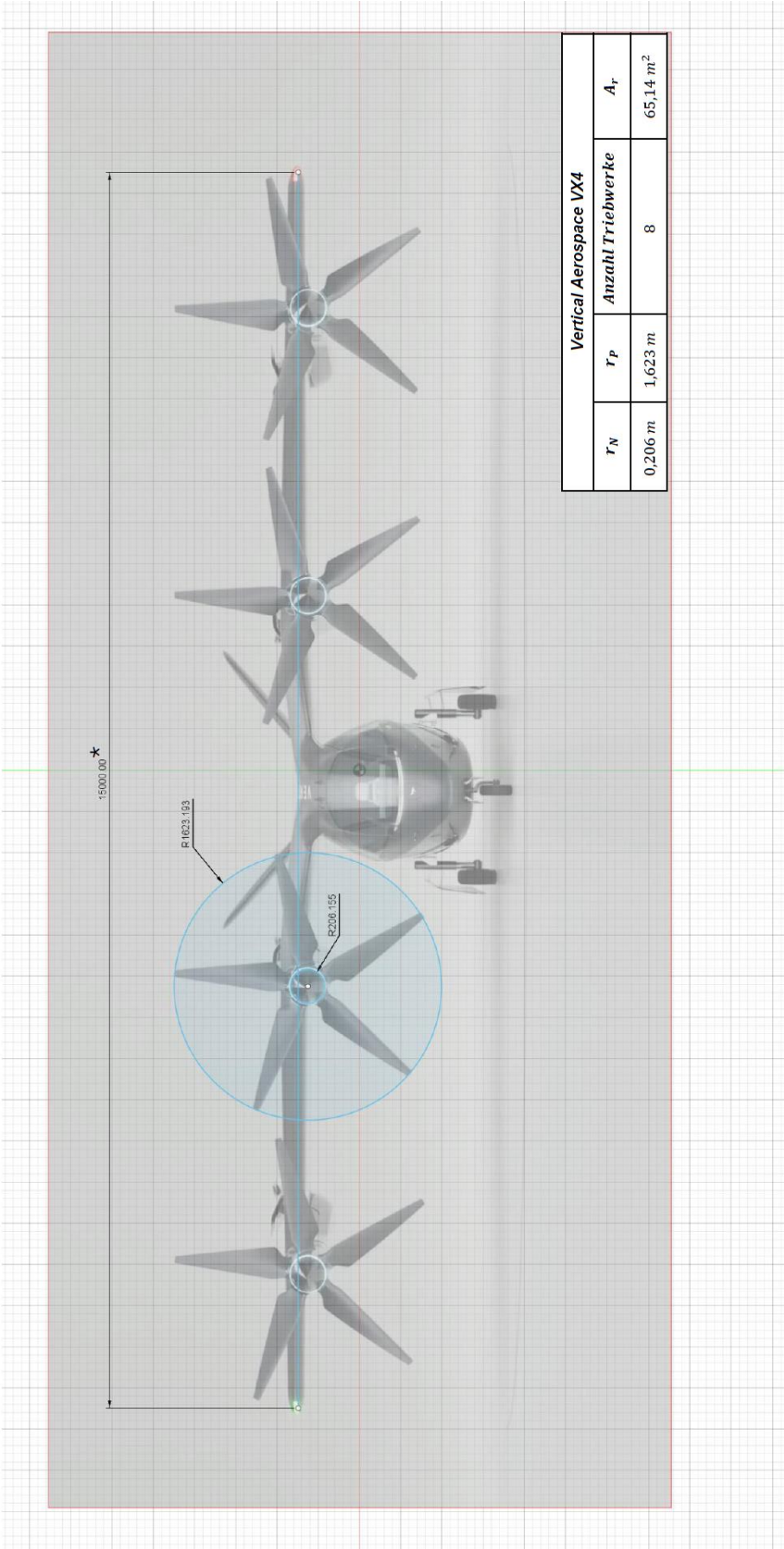
*Spannweite aus [31]

9.1.2 Archer Aviation Midnight



*Spannweite aus [36]

9.1.3 Vertical Aerospace VX4



*Spannweite aus [43]

9.2 Anhang 2: Python-Code des Berechnungsmodells

9.2.1 Berechnungsmodell.py

```
import flet as ft
from flet_gui import FletGUI
from flugzeug import Flugzeug
from kostenmodell import Kostenmodell
from flugzeugManager import FlugzeugManager

def main_gui(page: ft.Page):
    FletGUI(page) # GUI-Funktion

if __name__ == "__main__":
    ft.app(target=main_gui)
```

9.2.2 flugzeug.py

```
from math import pi, sqrt

class Flugzeug:
    @classmethod
    def from_reiseflugdaten(cls, eta_r, mu_a, e_a, ts, eta_s, mam, ar, mu_p,
pm, v_R, b, c, rho=None, altitude=None):
        # Berechne die Luftdichte, wenn altitude gesetzt ist
        if rho is None and altitude is not None:
            rho = cls(0, 0, 0, 0, 0, 0, 0, 0, 0, 0).berechne_luftdichte(alti-
tude)

        dummy = cls(
            gzc=0, # wird gleich berechnet
            eta_r=eta_r,
            mu_a=mu_a,
            e_a=e_a,
            ts=ts,
            eta_s=eta_s,
            mam=mam,
            ar=ar,
            mu_p=mu_p,
            pm=pm,
            rho=rho,
            altitude=altitude
        )
        gzc = dummy.berechne_gleitzahl(v_R, b, c)
        dummy.gzc = gzc
        return dummy

    def __init__(self, gzc, eta_r, mu_a, e_a, ts, eta_s, mam, ar, mu_p, pm,
rho=None, altitude=None):
```

Anhang

```
self.gzr = gzr # Gleitzahl, variabel für Optimierung
self.eta_r = eta_r # Wirkungsgrad Reiseflug, fest
self.mu_a = mu_a # Akkumassenverhältnis, variabel für Optimierung
self.e_a = e_a # spezifische Energiedichte, in Wh/kg, variabel für
Optimierung
self.ts = ts # Schwebeflugdauer, fest
self.eta_s = eta_s # Wirkungsgrad Schwebeflugbetrieb, fest
self.mam = mam # maximale Abflugmasse, variabel für Optimierung
self.ar = ar # Rotorfläche, variabel für Optimierung
self.mu_p = mu_p # Personenmassenverhältnis, variabel für Optimierung
self.am = mu_a * mam # Masse der Akkus, variabel für Optimierung
self.pm = pm # Passagiermasse, variabel für Optimierung

# Konstante Werte
self.g = 9.81 # Erdbeschleunigung
self.c_D0 = 0.0317 # Widerstandsbeiwert
self.e = 0.95 # Oswald-Faktor

# Luftdichte als fixer Wert oder basierend auf Höhe
if altitude is not None:
    self.rho = self.berechne_luftdichte(altitude)
elif rho is not None:
    self.rho = rho
else:
    self.rho = 1.190

def berechne_luftdichte(self, altitude):
    """Berechnet die Luftdichte basierend auf der Höhe (Standardat-
mosphäre)."""
    if altitude < 11000: # Troposphäre
        T = 288.15 - 0.0065 * altitude
        p = 101325 * (T / 288.15) ** 5.2561
    else: # Stratosphäre, einfache Näherung
        T = 216.65
        p = 22632.1 * (2.71828 ** (-0.0001577 * (altitude - 11000)))

    R = 287.05 # J/(kg·K)
    return p / (R * T)

def reichweite(self):
    """Berechnet die Reichweite des Flugzeugs."""
    p1 = self.gzr * self.eta_r
    p2 = (self.mu_a * self.e_a * 3600) / self.g
    p3 = self.ts / self.eta_s
    p4 = sqrt((self.mam * self.g) / (2 * self.rho * self.ar))
    return (p1 * (p2 - p3 * p4)) / 1000 # Ergebnis in km

def energieverbrauch(self):
    """Berechnet den Energieverbrauch des Flugzeugs in kWh/100km."""
    zaehler = self.e_a * self.am
```

Anhang

```
nenner = (self.pm / 100) * self.reichweite()
return (zaehler / nenner) * 0.1 # Ergebnis in kWh/100km

def max_reichweite(self):
    "Optimiere die maximale Reichweite."

# NEUE FUNKTIONEN basierend auf den Formeln

def aspect_ratio(self, b, Sw):
    """Berechnung der Streckung AR"""
    return (b ** 2) / Sw

def c_A_R(self, v_R, Sw):
    """Berechnung des Auftriebsbeiwerts c_A_R"""
    return (2 * self.mam * self.g) / (self.rho * Sw * v_R ** 2)

def c_Di_R(self, c_A_R_value, AR):
    """Berechnung des induzierten Widerstandsbeiwerts c_Di_R"""
    return (c_A_R_value ** 2) / (pi * self.e * AR)

def GZ_R(self, c_A_R_value, c_Di_R_value):
    """Berechnung der Gleitzahl GZ_R"""
    return c_A_R_value / (self.c_D0 + c_Di_R_value)

def Sw(self, b, c):
    return b*c

# @param v_R Reisefluggeschwindigkeit
# @param b Spannweite 5-20
# @param c Mittlere Flügeltiefe 0.45-1.8 (9% von b)
def berechne_gleitzahl(self, v_R, b, c):
    """Komplettberechnung der Gleitzahl basierend auf den Formeln."""
    try:
        # Validierung der Eingabewerte
        if b <= 0 or c <= 0:
            raise ValueError(f"Ungültige Spannweite (b + "") oder Flügeltiefe (c): b={b}, c={c}. Beide müssen positiv sein.\n")

        Sw = self.Sw(b, c)
        AR = self.aspect_ratio(b, Sw) # Flügelstreckung (Aspect Ratio)
        cA_R = self.c_A_R(v_R, Sw)

        # Realistische Grenzen für c_A_R (Auftriebsbeiwert)
        if not (0.1 <= cA_R <= 1.5):
            raise ValueError(f"Ungültiger Auftriebsbeiwert (c_A_R + ""
für Geschwindigkeit {v_R}: {cA_R}. Erwarteter Bereich: 0.1 - 1.5\n")

        cDi_R = self.c_Di_R(cA_R, AR)
        gleitzahl = self.GZ_R(cA_R, cDi_R)
```

```
        # Validierung der Gleitzahl
        if gleitzahl <= 0:
            raise ValueError(f"Ungültige Gleitzahl: {gleitzahl}. Muss po-
sitiv sein.\n" + "")

        return gleitzahl

    except Exception as ex:
        raise ValueError(f"Fehler bei der Gleitzahlberechnung (v_R={v_R},
b={b}, c={c} + ""): {ex}\n")
```

9.2.3 optimierer.py

```
from flugzeug import Flugzeug

class Optimierung:
    def __init__(self):
        self.reset()

    def reset(self):
        self.best_reichweite = -1
        self.best_combination = None
        self.best_energieverbrauch = float('inf')
        self.best_combination_energieverbrauch = None
        self.counter = 0
        self.count_mu_limit = 0
        self.count_cAR_limit = 0
        self.count_exception = 0
        self.energieverbrauch_max_reichweite = -1
        self.reichweite_max_energieverbrauch = -1

    def frange(self, start, stop, step):
        current = start
        while current <= stop:
            yield round(current, 10) # Rundung an dieser Stelle
            current += step

    def grid_search_optimierung(self, param_dict=None, max_combina-
tions=1000000000000000):
        # Reset the state before each run
        self.reset()

        # Bereichsdefinition
        def get_range(key, default):
            if param_dict and key in param_dict:
                start, stop, step = param_dict[key]
                # Generiere Bereich exakt ohne Rundungsfehler
                return [round(x, 10) for x in self.frange(start, stop, step)]
```

```

        return [round(x, 10) for x in self.frange(*default)]

# Parameterbereiche
mu_a_range = get_range("mu_a", (0.3, 0.8, 0.1))
e_a_range = get_range("e_a", (100, 330, 23))
mam_range = get_range("mam", (1500, 3500, 100))
ar_range = get_range("ar", (1, 50, 5))
mu_p_range = get_range("mu_p", (0.3, 0.8, 0.1))
v_R_range = get_range("v_R", (55, 100, 5))
b_range = get_range("b", (5, 20, 1))

# Maximal erlaubte Kombinationen
total_combinations = (
    len(mu_a_range) * len(e_a_range) * len(mam_range) *
    len(ar_range) * len(mu_p_range) * len(v_R_range) * len(b_range)
)
if total_combinations > max_combinations:
    raise ValueError(f"Zu viele Kombinationen: {total_combinations}.
Bitte die Bereiche reduzieren.")

# Optimierung
for mu_a in mu_a_range:
    for e_a in e_a_range:
        for mam in mam_range:
            for ar in ar_range:
                for mu_p in mu_p_range:
                    for v_R in v_R_range:
                        for b in b_range:
                            c = 0.09 * b
                            self.counter += 1

# Bedingung für mu_a + mu_p
if (mu_a + mu_p) > 0.7:
    self.count_mu_limit += 1
    continue

pm = mu_p * mam

flugzeug = Flugzeug(
    gzc=0,
    eta_r=0.8,
    mu_a=mu_a,
    e_a=e_a,
    ts=60,
    eta_s=0.8,
    mam=mam,
    ar=ar,
    mu_p=mu_p,
    pm=pm
)

```

```

try:
    c_A_R = flugzeug.c_A_R(v_R, flug-
zeug.Sw(b, c))

    if (c_A_R < 0.35 or c_A_R > 0.4):
        self.count_cAR_limit += 1
        continue

    gZR = flugzeug.berechne_gleitzahl(v_R,
b, c)

    flugzeug.gZR = gZR

    reichweite = flugzeug.reichweite()
    energieverbrauch = flugzeug.energie-
verbrauch()

    if reichweite > self.best_reichweite:
        self.best_reichweite = reichweite
        self.best_combination = (mu_a,
e_a, mam, ar, mu_p, v_R, b, c, gZR)

        self.energieverbrauch_max_reich-
weite = energieverbrauch

    if energieverbrauch < self.best_ener-
gieverbrauch:

        self.best_energieverbrauch = ener-
gieverbrauch

        self.best_combination_energiever-
brauch = (mu_a, e_a, mam, ar, mu_p, v_R, b, c, gZR)
        self.reichweite_max_energiever-
brauch = reichweite

except Exception as ex:
    self.count_exception += 1
    continue

return {
    "beste_reichweite": self.best_reichweite,
    "werte_reichweite": self.best_combination,
    "werte_energieverbrauch_max_reichweite": self.energiever-
brauch_max_reichweite,
    "bester_verbrauch": self.best_energieverbrauch,
    "werte_verbrauch": self.best_combination_energieverbrauch,
    "werte_reichweite_max_energieverbrauch": self.reichweite_max_ener-
gieverbrauch,
    "debug": {
        "mu_limit": self.count_mu_limit,
        "cAR_limit": self.count_cAR_limit,
        "exceptions": self.count_exception,
        "kombis_getestet": self.counter
    }
}

```

```
    }  
}
```

9.2.4 flet_gui.py

```
import flet as ft  
import pandas as pd  
import os  
from flugzeug import Flugzeug  
from optimierer import Optimierung  
from flugzeugManager import FlugzeugManager  
from kostenmodell import Kostenmodell  
import threading  
  
class FletGUI:  
    def __init__(self, page: ft.Page):  
        self.page = page  
        self.page.title = "eVTOL Berechnungsmodell"  
        self.page.scroll = "auto"  
        self.optimierer = Optimierung()  
        self.manager = FlugzeugManager()  
        self.init_ui()  
  
    def init_ui(self):  
        # Eingabefelder erstellen  
        parameter_section = self.create_input_fields()  
  
        # Ergebnisfelder erstellen  
        result_section = self.create_result_fields()  
  
        # Tabellenansicht hinzufügen  
        self.list_view = self.create_flight_list()  
  
        # Tab-Erstellung  
        self.page.add(self.create_tabs(parameter_section, result_section))  
  
        # Initiale Feldsichtbarkeit anpassen  
        self.update_input_fields(None)  
        self.update_dichte_fields(None)  
  
    def create_berechnung_tab(self):  
        # Flugzeugname  
        self.flugzeug_name = ft.TextField(label="Flugzeugname", expand=True)  
  
        # Eingabeart Dropdown  
        self.eingabe_option = ft.Dropdown(  
            label="Eingabeart",  
            options=[  
                ft.dropdown.Option("Gleitzahl bekannt"),  
                ft.dropdown.Option("Gleitzahl unbekannt"),
```

Anhang

```
    ],
    value="Gleitzahl bekannt",
    on_change=self.update_input_fields
)

# Luftdichte Option
self.dichte_option = ft.Dropdown(
    label="Luftdichte Option",
    options=[
        ft.dropdown.Option("Fester Wert"),
        ft.dropdown.Option("Höhenbasierte Berechnung")
    ],
    value="Fester Wert",
    on_change=self.update_dichte_fields
)

# Parameterfelder
self.inputs = {
    "gZR": ft.TextField(label="Gleitzahl (GZR)", expand=True, visible=True),
    "eta_r": ft.TextField(label="Wirkungsgrad Reiseflug ( $\eta_R$ )", expand=True),
    "mu_a": ft.TextField(label="Verhältnis AM/MAM ( $\mu_a$ )", expand=True),
    "e_a": ft.TextField(label="Energiedichte Batterie (Wh/kg)", expand=True),
    "ts": ft.TextField(label="Schwebeflugdauer (ts in Minuten)", expand=True),
    "eta_s": ft.TextField(label="Wirkungsgrad Schwebeflugbetrieb ( $\eta_S$ )", expand=True),
    "mAM": ft.TextField(label="Maximale Abflugmasse (MAM in kg)", expand=True),
    "ar": ft.TextField(label="Rotorfläche (Ar in m2)", expand=True),
    "pm": ft.TextField(label="Passagiermasse (PM in kg)", expand=True),
    "v_R": ft.TextField(label="Reisefluggeschwindigkeit (v_R in m/s)", expand=True, visible=False),
    "b": ft.TextField(label="Spannweite (b in m)", expand=True, visible=False),
    "c": ft.TextField(label="Flügeltiefe (c in m)", expand=True, visible=False),
    "rho": ft.TextField(label="Luftdichte (kg/m3)", expand=True, visible=True),
    "altitude": ft.TextField(label="Höhe (m)", expand=True, visible=False)
}

# Linke Spalte (Flugzeugname und Dropdowns)
left_column = ft.Column([
    self.flugzeug_name,
    self.eingabe_option,
```

```

        self.dichte_option
    ], spacing=10, expand=True)

# Rechte Spalte (Parameterfelder)
right_column = ft.Column([
    self.inputs["gZR"],
    self.inputs["eta_r"],
    self.inputs["mu_a"],
    self.inputs["e_a"],
    self.inputs["ts"],
    self.inputs["eta_s"],
    self.inputs["mam"],
    self.inputs["ar"],
    self.inputs["pm"],
    self.inputs["v_R"],
    self.inputs["b"],
    self.inputs["c"],
    self.inputs["rho"],
    self.inputs["altitude"]
], spacing=10, expand=True)

# Ergebnisbereich
self.result_reichweite = ft.Text(value="Reichweite: - km", size=16,
weight="bold")
self.result_energie = ft.Text(value="Energieverbrauch pro Kopf: -
kWh/100km", size=16, weight="bold")
self.calculate_button = ft.ElevatedButton("Berechnen",
on_click=self.berechne)

# Zusammenfügen
return ft.Column([
    ft.Text("✈️ Flugzeug Berechnung", size=24, weight="bold"),
    ft.Row([left_column, right_column], spacing=20, expand=True),
    ft.Divider(),
    self.calculate_button,
    self.result_reichweite,
    self.result_energie
], spacing=20, expand=True, scroll="auto")

def berechne(self, e):
    try:
        name_field = self.inputs.get("flugzeug_name", None)
        if not name_field or not name_field.value.strip():
            self.show_popup("Fehler", "Bitte einen Flugzeugnamen einge-
ben.", success=False)
        return

```

Anhang

```
name = name_field.value.strip()

# Parameter sammeln
params = {key: float(field.value) for key, field in self.in-
puts.items() if field.visible and field.value and key != "flugzeug_name"}

# mu_p automatisch setzen. wird in Berechnung nicht gebraucht,
deshalb auf 0 setzen.
mu_p = 0

# Sicherstellen, dass die Gleitzahl vorhanden ist
if self.eingabe_option.value == "Gleitzahl bekannt":
    gzc = params.get("gzc")
    if gzc is None:
        self.show_popup("Fehler", "Bitte die Gleitzahl (GZ) ein-
geben.", success=False)
        return
    else:
        v_R = params.get("v_R")
        b = params.get("b")
        c = params.get("c")
        if not (v_R and b and c):
            self.show_popup("Fehler", "Bitte alle benötigten Parameter
eingeben.", success=False)
            return

# Gleitzahl berechnen
flz = Flugzeug.from_reiseflugdaten(
    eta_r=params["eta_r"],
    mu_a=params["mu_a"],
    e_a=params["e_a"],
    ts=params["ts"],
    eta_s=params["eta_s"],
    mam=params["mam"],
    ar=params["ar"],
    pm=params["pm"],
    mu_p=mu_p,
    v_R=v_R,
    b=b,
    c=c,
    rho=params.get("rho"),
    altitude=params.get("altitude")
)
gzc = flz.gzc
flugzeug = None
if params.get("rho"):
    # Flugzeug erstellen
    flugzeug = Flugzeug(
        gzc=gzc,
```

```

        eta_r=params["eta_r"],
        mu_a=params["mu_a"],
        e_a=params["e_a"],
        ts=params["ts"],
        eta_s=params["eta_s"],
        mam=params["mam"],
        ar=params["ar"],
        pm=params["pm"],
        mu_p=mu_p,
        rho=params["rho"]
    )
elif params.get("altitude"):
    # Flugzeug erstellen
    flugzeug = Flugzeug(
        gzr=gzr,
        eta_r=params["eta_r"],
        mu_a=params["mu_a"],
        e_a=params["e_a"],
        ts=params["ts"],
        eta_s=params["eta_s"],
        mam=params["mam"],
        ar=params["ar"],
        pm=params["pm"],
        mu_p=mu_p,
        altitude=params["altitude"]
    )

    # Ergebnisse anzeigen
    self.result_reichweite.value = f"Reichweite: {flugzeug.reich-
weite():.2f} km"
    self.result_energie.value = f"Energieverbrauch pro Kopf: {flug-
zeug.energieverbrauch():.2f} kWh/100km"
    self.page.update()

#
if flugzeug:
    self.manager.add_flugzeug(
        name,
        flugzeug.gzr,
        flugzeug.eta_r,
        flugzeug.mu_a,
        flugzeug.e_a,
        flugzeug.ts,
        flugzeug.eta_s,
        flugzeug.mam,
        flugzeug.ar,
        mu_p,
        flugzeug.pm,
        flugzeug.rho if hasattr(flugzeug, "rho") else None,

```

Anhang

```
        flugzeug.altitude if hasattr(flugzeug, "altitude") else
None
    )

    self.add_list_item(name, flugzeug.reichweite(), flugzeug.ener-
gieverbrauch())

    # Automatischer Excel-Export
    self.manager.export_flugzeuge_to_excel()

except ValueError as ex:
    self.show_popup("Fehler", f"Ungültiger Wert: {ex}", success=False)
except ZeroDivisionError:
    self.show_popup("Fehler", "Division durch Null. Überprüfe die Ein-
gabewerte.", success=False)
except Exception as ex:
    self.show_popup("Fehler", f"Unerwarteter Fehler: {ex}", suc-
cess=False)

def create_kostenmodell_tab(self):
    # Eingabefelder für Kosten initialisieren
    self.kostenmodell_inputs = {}

    def create_section(title, inputs):
        return ft.Column(
            controls=[ft.Text(title, size=20, weight="bold")] + inputs,
            spacing=10
        )

    # Herstellungskosten eVTOL-Zelle (Cvbm1)
    betriebs_parameter = create_section("Betriebsparameter", [
        self.kostenmodell_inputs.setdefault("pP", ft.TextField(la-
bel="Flugpreis pro Passagier ($/km)", value="1.5", width=250)),
        self.kostenmodell_inputs.setdefault("np", ft.TextField(label="An-
zahl der Passagiere", value="7", width=250)),
        self.kostenmodell_inputs.setdefault("ratioP", ft.TextField(la-
bel="Ladefaktor", value="0.5", width=250)),
        self.kostenmodell_inputs.setdefault("R", ft.TextField(la-
bel="Reichweite pro Flugmission in km", value="100", width=250)),
        self.kostenmodell_inputs.setdefault("nM", ft.TextField(label="An-
zahl der Flugmissionen pro Tag", value="7.06", width=250)),
        self.kostenmodell_inputs.setdefault("nD", ft.TextField(label="An-
zahl der Werktage im Jahr", value="300", width=250)),
        self.kostenmodell_inputs.setdefault("DR", ft.TextField(label="Dis-
kontierungsrate", value="0.07", width=250)),
```

Anhang

```
        self.kostenmodell_inputs.setdefault("VL", ft.TextField(label="Lebensdauer des eVTOLs", value="13", width=250)),
        self.kostenmodell_inputs.setdefault("YT", ft.TextField(label="Jährliche Flugzeit in h", value="2400", width=250)),

    ])

    # Herstellungskosten eVTOL-Zelle (Cvbm1)
    cvbm1_section = create_section("Herstellungskosten eVTOL-Zelle (Cvbm1)", [
        self.kostenmodell_inputs.setdefault("Cvbm_f", ft.TextField(label="Cvbm_f (fixe Herstellungskosten)", value="600000", width=250)),
        self.kostenmodell_inputs.setdefault("Cvbm_v", ft.TextField(label="Cvbm_v (variable Herstellungskosten)", value="800", width=250)),
        self.kostenmodell_inputs.setdefault("mto", ft.TextField(label="mto (maximales Startgewicht)", value="1800", width=250)),
        self.kostenmodell_inputs.setdefault("x", ft.TextField(label="x (Anteil der Strukturmasse)", value="0.3", width=250)),
    ])

    # Batteriekosten und Antriebssystemkosten (Cpt)
    cpt_section = create_section("Cpt (Batteriekosten und Antriebssystemkosten)", [
        self.kostenmodell_inputs.setdefault("Q", ft.TextField(label="Q (Batteriekapazität)", value="178.2", width=250)),
        self.kostenmodell_inputs.setdefault("UCes", ft.TextField(label="UCes (Kosten pro kWh Batterie)", value="100", width=250)),
        self.kostenmodell_inputs.setdefault("Pds", ft.TextField(label="Pds (Leistung des Antriebssystems)", value="228", width=250)),
        self.kostenmodell_inputs.setdefault("UCds", ft.TextField(label="UCds (Kosten pro kW Antrieb)", value="25", width=250)),
    ])

    # Kosten für Pilot oder autonomes Fliegen (Cplt)
    cplt_section = create_section("Cplt (Kosten für Pilot oder autonomes Fliegen)", [
        self.kostenmodell_inputs.setdefault("Cuad", ft.TextField(label="Cuad (Herstellungskosten UAD-System)", value="60000", width=250)),
        self.kostenmodell_inputs.setdefault("beta", ft.TextField(label="beta (Anteil der Pilotenkosten)", value="0.2", width=250)),
    ])

    # Energiekosten (Ce)
    ce_section = create_section("Ce (Energiekosten)", [
        self.kostenmodell_inputs.setdefault("pE", ft.TextField(label="pE (Stromkosten pro kWh)", value="0.09", width=250)),
        self.kostenmodell_inputs.setdefault("eC", ft.TextField(label="eC (Energieverbrauch pro Mission)", value="59.36", width=250)),
        self.kostenmodell_inputs.setdefault("V", ft.TextField(label="V (Reisefluggeschwindigkeit)", value="234", width=250)),
```

Anhang

```
        self.kostenmodell_inputs.setdefault("ratioDE", ft.TextField(label="ratioDE (Anteil der Strecke für leere Rückflüge)", value="0.2", width=250)),
    ])

    # Infrastrukturkosten (Ci)
    ci_section = create_section("Ci (Infrastrukturkosten)", [
        self.kostenmodell_inputs.setdefault("Cvpc0", ft.TextField(label="Cvpc0 (Baukosten Vertiport)", value="1450000", width=250)),
        self.kostenmodell_inputs.setdefault("VPL", ft.TextField(label="VPL (Lebensdauer Vertiport)", value="20", width=250)),
        self.kostenmodell_inputs.setdefault("Cvpo", ft.TextField(label="Cvpo (Betriebskosten Vertiport)", value="80000", width=250)),
        self.kostenmodell_inputs.setdefault("Crfc0", ft.TextField(label="Crfc0 (Baukosten Energieversorgung)", value="600000", width=250)),
        self.kostenmodell_inputs.setdefault("RFL", ft.TextField(label="RFL (Lebensdauer Energieversorgung)", value="13", width=250)),
        self.kostenmodell_inputs.setdefault("Crfo", ft.TextField(label="Crfo (Betriebskosten Energieversorgung)", value="82000", width=250)),
    ])

    # Steuern, Gebühren, Versicherungen (Ctfi)
    ctfi_section = create_section("Ctfi (Steuern, Gebühren, Versicherungen)", [
        self.kostenmodell_inputs.setdefault("ratioTFI", ft.TextField(label="ratioTFI (Anteil der direkten Betriebskosten)", value="0.2", width=250)),
    ])

    # Wartungskosten (Cm)
    cm_section = create_section("Cm (Wartungskosten)", [
        self.kostenmodell_inputs.setdefault("ratioM", ft.TextField(label="ratioM (Anteil der fixen Wartungskosten)", value="0.075", width=250)),
        self.kostenmodell_inputs.setdefault("UCvm", ft.TextField(label="UCvm (variable Wartungskosten)", value="20", width=250)),
    ])

    # Restwert am Ende der Lebensdauer (Csv)
    csv_section = create_section("Csv (Restwert am Ende der Lebensdauer)", [
        self.kostenmodell_inputs.setdefault("ratioSV", ft.TextField(label="ratioSV (Anteil des Restwerts)", value="0.1", width=250)),
    ])

    # **Ergebnisbereich hinzufügen**
    self.result_irc = ft.Text(value="Amortisationsdauer (IRC): - Jahre", size=16, weight="bold")
    self.result_pyc = ft.Text(value="Gesamtkosten (PYC): - $", size=16, weight="bold")
    self.result_pyr = ft.Text(value="Gesamteinnahmen (PYR): - $", size=16, weight="bold")
```

Anhang

```
        self.result_npv = ft.Text(value="Net Present Value (NPV): - $",
size=16, weight="bold")
        self.result_lc = ft.Text(value="Levelized Cost (LC): - $/Passagier-
km", size=16, weight="bold")

        # Berechnen- und Export-Buttons
        calculate_button = ft.ElevatedButton("Berechnen", on_click=self.be-
rechne_kostenmodell)
        export_button = ft.ElevatedButton("Exportieren", on_click=self.ex-
port_kostenmodell)

        # Alles zusammenfügen
        return ft.Column(
            controls=[
                betriebs_parameter,
                cvbm1_section,
                cpt_section,
                cplt_section,
                ce_section,
                ci_section,
                ctfi_section,
                cm_section,
                csv_section,
                ft.Divider(),
                ft.Row([calculate_button, export_button], spacing=20),
                ft.Divider(),
                self.result_irc,
                self.result_pyc,
                self.result_pyr,
                self.result_npv,
                self.result_lc,
            ],
            spacing=20,
            scroll="auto",
            expand=True
        )

def berechne_kostenmodell(self, e):
    try:
        # Parameter aus der GUI lesen
        param_values = {}
        for key, field in self.kostenmodell_inputs.items():
            try:
                param_values[key] = float(field.value)
            except ValueError:
                self.show_popup("Fehler", f"Ungültiger Wert für '{key}':
{field.value}", success=False)
        return
```

```

# Neues Kostenmodell mit den GUI-Parametern erstellen
kostenmodell = Kostenmodell(
    Cvbm_f=param_values.get("Cvbm_f", 600000),
    Cvbm_v=param_values.get("Cvbm_v", 800),
    mto=param_values.get("mto", 1800),
    x=param_values.get("x", 0.3),
    Q=param_values.get("Q", 178.2),
    UCes=param_values.get("UCes", 100),
    Pds=param_values.get("Pds", 228),
    UCds=param_values.get("UCds", 25),
    Cuad=param_values.get("Cuad", 60000),
    beta=param_values.get("beta", 0.2),
    pE=param_values.get("pE", 0.09),
    eC=param_values.get("eC", 59.36),
    V=param_values.get("V", 234),
    ratioDE=param_values.get("ratioDE", 0.2),
    Cvpc0=param_values.get("Cvpc0", 1450000),
    VPL=param_values.get("VPL", 20),
    Cvpo=param_values.get("Cvpo", 80000),
    Crfc0=param_values.get("Crfc0", 600000),
    RFL=param_values.get("RFL", 13),
    Crfo=param_values.get("Crfo", 82000),
    ratioTFI=param_values.get("ratioTFI", 0.2),
    ratioM=param_values.get("ratioM", 0.075),
    UCvm=param_values.get("UCvm", 20),
    ratioSV=param_values.get("ratioSV", 0.1),
    pP=param_values.get("pP", 1.5),
    np=param_values.get("np", 7),
    ratioP=param_values.get("ratioP", 0.5),
    R=param_values.get("R", 100),
    nM=param_values.get("nM", 7.06),
    nD=param_values.get("nD", 300),
    DR=param_values.get("DR", 0.07),
    VL=param_values.get("VL", 13),
    YT=param_values.get("YT", 2400)
)

# Ergebnisse berechnen
irc = kostenmodell.IRC()
pyc = sum(kostenmodell.calculate_pyc())
pyr = sum(kostenmodell.calculate_pyr())
npv = kostenmodell.NPV()
lc = kostenmodell.LC()

# Überprüfen, ob ein gültiges Ergebnis für die Amortisationsdauer
existiert
if irc == -1 or irc < 0:
    self.show_popup("Fehler", "Keine Amortisation innerhalb der
Lebensdauer erreicht. Prüfen Sie die Eingabewerte.", success=False)

```

Anhang

```
# Ergebnisse anzeigen
self.result_irc.value = f"Amortisationsdauer (IRC): {irc:.2f}
Jahre"

self.result_pyc.value = f"Gesamtkosten (PYC): {pyc:.2f} $"
self.result_pyr.value = f"Gesamteinnahmen (PYR): {pyr:.2f} $"
self.result_npv.value = f"Net Present Value (NPV): {npv:.2f} $"
self.result_lc.value = f"Levelized Cost (LC): {lc:.4f} $/Passag-
ier-km"

# GUI aktualisieren
self.page.update()

except Exception as ex:
    self.show_popup("Fehler", f"Fehler bei der Berechnung des Kosten-
modells: {ex}", success=False)

def export_kostenmodell(self, e):
    try:
        # Parameter aus der GUI lesen
        param_values = {}
        for key, field in self.kostenmodell_inputs.items():
            try:
                param_values[key] = float(field.value)
            except ValueError:
                self.show_popup("Fehler", f"Ungültiger Wert für '{key}':
{field.value}", success=False)
        return

        # Kostenmodelle erstellen (autonom und pilot)
        kostenmodul_autonom = Kostenmodell(
            **param_values
        )
        kostenmodul_pilot = Kostenmodell(
            pilot=True,
            **param_values
        )

        # Berechnungen autonom
        pyc_autonom = kostenmodul_autonom.calculate_pyc()
        pyr_autonom = kostenmodul_autonom.calculate_pyr()
        LC_autonom = kostenmodul_autonom.LC()
        NPV_autonom = kostenmodul_autonom.NPV()
        IRC_autonom = kostenmodul_autonom.IRC()
        cplt_autonom = []
        cplt_pilot = []
```

```

cpt = []

# Berechnungen pilot
pyc_pilot = kostenmodul_pilot.calculate_pyc()
pyr_pilot = kostenmodul_pilot.calculate_pyr()
LC_pilot = kostenmodul_pilot.LC()
NPV_pilot = kostenmodul_pilot.NPV()
IRC_pilot = kostenmodul_pilot.IRC()

for t in range(len(pyc_pilot)):
    cplt_autonom.append(kostenmodul_autonom.Cplt(t))
    cplt_pilot.append(kostenmodul_pilot.Cplt(t))
    cpt.append(kostenmodul_autonom.Cpt(t))

jahr = list(range(len(pyc_pilot)))
daten = {
    "Jahr": jahr,
    "PYC autonom": pyc_autonom,
    "PYC pilot": pyc_pilot,
    "PYR autonom": pyr_autonom,
    "PYR pilot": pyr_pilot,
    "Cvbm1": [kostenmodul_autonom.Cvbm1()] * len(pyc_pilot),
    "Cpt": cpt,
    "Ce": [kostenmodul_autonom.Ce()] * len(pyc_pilot),
    "Cplt autonom": cplt_autonom,
    "Cplt pilot": cplt_pilot,
    "Ci": [kostenmodul_autonom.Ci()] * len(pyc_pilot),
    "Ctfi autonom": [kostenmodul_autonom.Ctfi(param_val-
lues.get("VPL", 20))] * len(pyc_pilot),
    "Ctfi pilot": [kostenmodul_pilot.Ctfi(param_values.get("VPL",
20))] * len(pyc_pilot),
    "Cm": [kostenmodul_autonom.Cm()] * len(pyc_pilot),
    "Csv": [kostenmodul_autonom.Csv()] * len(pyc_pilot)
}

daten2 = {
    "LC autonom": [LC_autonom],
    "LC pilot": [LC_pilot],
    "NPV autonom": [NPV_autonom],
    "NPV pilot": [NPV_pilot],
    "IRC autonom": [IRC_autonom],
    "IRC pilot": [IRC_pilot]
}

# DataFrame erzeugen
df = pd.DataFrame(daten)
df2 = pd.DataFrame(daten2)

# Als Excel speichern

```

Anhang

```
        with pd.ExcelWriter("kostenmodell.xlsx", engine="openpyxl") as
writer:
            df.to_excel(writer, sheet_name="Übersicht kosten und Einnah-
men", index=False)
            df2.to_excel(writer, sheet_name="Übersicht LC,NPV,IRC", in-
dex=False)

            # Erfolgreiche Nachricht anzeigen
            self.show_popup("Erfolg", "Die Excel-Datei 'kostenmodell.xlsx'
wurde erfolgreich exportiert.", success=True)

        except Exception as ex:
            self.show_popup("Fehler", f"Fehler beim Exportieren der Excel-Datei: {ex}", success=False)

def create_optimierung_tab(self):
    # Parameterfelder für die Optimierung
    self.parameter_inputs = {
        "mu_a": ft.TextField(label="Batteriemassenverhältnis (Standard bei
0.3 - 0.8)", value="0.3,0.8,0.1"),
        "e_a": ft.TextField(label="spezifische Energiedichte (Standard bei
100 - 330 Wh/kg)", value="100,330,23"),
        "mam": ft.TextField(label="maximale Abflugmasse (Standard bei 1500
- 3500 kg)", value="1500,3500,100"),
        "ar": ft.TextField(label="Rotorfläche (Standard bei 1 - 50 m²)",
value="1,50,5"),
        "mu_p": ft.TextField(label="Passagiermassenverhältnis (Standard
bei 0.3 - 0.8)", value="0.3,0.8,0.1"),
        "v_R": ft.TextField(label="Reisefluggeschwindigkeit (Standard bei
55 - 100 m/s)", value="55,100,5"),
        "b": ft.TextField(label="Spannweite (Standard bei 5 - 20 m)", va-
lue="5,20,1")
    }

    # Optimierung starten Button
    optimieren_button = ft.ElevatedButton("Optimierung starten",
on_click=self.optimierung_starten)

    # Export Button
    export_button = ft.ElevatedButton("Ergebnisse exportieren",
on_click=self.export_results)

    # Ergebnisanzeige
    self.optimierung_output_area = ft.Column(spacing=10)

    # Layout erstellen
```

Anhang

```
        parameter_section = ft.Column([field for field in self.parameter_in-
puts.values()], spacing=10)
        button_section = ft.Row([optimieren_button, export_button], spac-
ing=10)

# Tab-Inhalt zurückgeben
return ft.Column(
    controls=[
        ft.Text("☒ Optimierung", size=24, weight="bold"),
        parameter_section,
        button_section,
        self.optimierung_output_area
    ],
    spacing=20,
    scroll="auto"
)

def update_dichte_fields(self, e):
    if self.dichte_option.value == "Fester Wert":
        self.inputs["rho"].visible = True
        self.inputs["altitude"].visible = False
    else:
        self.inputs["rho"].visible = False
        self.inputs["altitude"].visible = True
    self.page.update()

def update_input_fields(self, e):
    if self.eingabe_option.value == "Gleitzahl bekannt":
        self.inputs["g_zr"].visible = True
        self.inputs["v_R"].visible = False
        self.inputs["b"].visible = False
        self.inputs["c"].visible = False
    else:
        self.inputs["g_zr"].visible = False
        self.inputs["v_R"].visible = True
        self.inputs["b"].visible = True
        self.inputs["c"].visible = True
    self.page.update()

def optimierung_starten(self, e):
    try:
        # Ladeanzeige hinzufügen
        self.optimierung_output_area.controls.clear()
        self.optimierung_output_area.controls.append(
            ft.Column([
                ft.Text("Optimierung wird gestartet...", size=18,
weight="bold"),
```

Anhang

```
        ft.ProgressRing(width=50, height=50),
        ft.Text("Dies kann einige Minuten dauern, bitte war-
ten...", size=14)
    ], alignment="center")
)
self.page.update()

def run_optimierung():
    try:
        # Parameter sammeln
        param_dict = {}
        for key, field in self.parameter_inputs.items():
            # Werte parsen
            try:
                start, stop, step = map(float,
field.value.split(", "))
                if start > stop or step <= 0:
                    raise ValueError(f"Ungültiger Bereich für
'{key}': Start muss kleiner als Stop sein und Schrittgröße muss positiv
sein.")

                # Parameterbereich speichern
                param_dict[key] = (start, stop, step)
            except ValueError:
                raise ValueError(f"Ungültiger Bereich für '{key}'.
Drei numerische Werte erwartet (start, stop, step).")

        # Optimierung starten
        result = self.optimierer.grid_search_optimierung(pa-
ram_dict=param_dict)

        # Ergebnisse anzeigen
        self.show_results(result)

    except ValueError as ex:
        self.show_popup("Fehler", str(ex), success=False)

    # Run the optimization in a separate thread to avoid freezing the
UI
    threading.Thread(target=run_optimierung).start()
except Exception as ex:
    self.show_popup("Fehler", f"Unerwarteter Fehler beim Starten der
Optimierung: {ex}", success=False)

def show_results(self, result):
    self.optimierung_output_area.controls.clear()

    # Beste Reichweite
    if result["werte_reichweite"]:
```

Anhang

```
self.optimierung_output_area.controls.append(
    ft.Text("Beste Lösung (Reichweite)", weight="bold", size=16)
)
self.optimierung_output_area.controls.append(
    self.ergebnis_karte(
        result["werte_reichweite"],
        result["beste_reichweite"],
        f"km ({result['werte_energieverbrauch_max_reichwei-
te']:.2f} kWh/100km)"
    )
)

# Bester Energieverbrauch
if result["werte_verbrauch"]:
    self.optimierung_output_area.controls.append(
        ft.Text("Beste Lösung (Energieverbrauch)", weight="bold",
size=16)
    )
    self.optimierung_output_area.controls.append(
        self.ergebnis_karte(
            result["werte_verbrauch"],
            result["bester_verbrauch"],
            f"kWh/100km ({result['werte_reichweite_max_energiever-
brauch']:.2f} km)"
        )
    )

# Debug-Infos
self.optimierung_output_area.controls.append(
    ft.Text("✘ Debug-Infos", weight="bold", size=16)
)
self.optimierung_output_area.controls.append(
    ft.Text(f"mu_a + mu_p verworfen: {result['debug']['mu_limit']}")
)
self.optimierung_output_area.controls.append(
    ft.Text(f"c_A_R verworfen: {result['debug']['cAR_limit']}")
)
self.optimierung_output_area.controls.append(
    ft.Text(f"Exceptions: {result['debug']['exceptions']}")
)
self.optimierung_output_area.controls.append(
    ft.Text(f"Kombis getestet: {result['debug']['kombis_getestet']}")
)

# Update the page
self.page.update()

def export_results(self, e):
```

Anhang

```
# Prüfe, ob Ergebnisse vorliegen
if not self.optimierer.best_combination or not self.optimierer.best_combination_energieverbrauch:
    self.show_popup("Fehler", "Keine gültigen Optimierungsergebnisse zum Exportieren vorhanden.", success=False)
    return

# Daten vorbereiten
reichweite_daten = list(self.optimierer.best_combination)
energieverbrauch_daten = list(self.optimierer.best_combination_energieverbrauch)

# Füge die berechneten Reichweiten und Energieverbräuche hinzu
beste_reichweite = self.optimierer.best_reichweite
bester_energieverbrauch = self.optimierer.best_energieverbrauch
energieverbrauch_max_reichweite = self.optimierer.energieverbrauch_max_reichweite
reichweite_max_energieverbrauch = self.optimierer.reichweite_max_energieverbrauch

beschriftungen = [
    "Batteriemassenverhältnis",
    "Energiedichte der Batterie [Wh/kg]",
    "Maximale Abflugmasse [kg]",
    "Gesamtrotorfläche [m²]",
    "Passagiermassenverhältnis",
    "Reisefluggeschwindigkeit [m/s]",
    "Flügelspanweite [m]",
    "Mittlere Flügeltiefe [m]",
    "Gleitzahl"
]

# Ergebnis-Daten zusammenstellen
daten = {
    "Parameter": beschriftungen,
    "Beste Reichweite": reichweite_daten,
    "Bester Energieverbrauch": energieverbrauch_daten
}

# Zusätzliche Ergebnisse hinzufügen
daten["Reichweite (km)"] = [f"{beste_reichweite:.2f} km"] + ["" ] * (len(beschriftungen) - 1)
daten["Energieverbrauch (kWh/100km)"] = [f"{bester_energieverbrauch:.2f} kWh/100km"] + ["" ] * (len(beschriftungen) - 1)
daten["Energieverbrauch bei bester Reichweite (kWh/100km)"] = [f"{energieverbrauch_max_reichweite:.2f} kWh/100km"] + ["" ] * (len(beschriftungen) - 1)
daten["Reichweite bei bestem Energieverbrauch (km)"] = [f"{reichweite_max_energieverbrauch:.2f} km"] + ["" ] * (len(beschriftungen) - 1)
```

Anhang

```
# In DataFrame umwandeln
df = pd.DataFrame(daten)

# Datei speichern
dateiname = "optimierungsergebnisse.xlsx"
df.to_excel(dateiname, index=False)

# Bestätigung anzeigen
self.show_popup("Erfolg", f"Optimierungsergebnisse erfolgreich als
'{dateiname}' exportiert.", success=True)

def ergebnis_karte(self, werte, wert, einheit):
    if not werte:
        return ft.Text("Keine gültige Kombination gefunden.", co-
lor="white")

    # Theme erkennen
    if self.page.theme_mode == ft.ThemeMode.DARK:
        bgcolor = ft.colors.GREY_900
        text_color = "white"
    else:
        bgcolor = ft.colors.SURFACE_VARIANT
        text_color = "black"

    beschriftungen = [
        "Batteriemassenverhältnis",
        "Energiedichte der Batterie [Wh/kg]",
        "Maximale Abflugmasse [kg]",
        "Gesamtrotorfläche [m2]",
        "Passagiermassenverhältnis",
        "Reisefluggeschwindigkeit [m/s]",
        "Flügelspanweite [m]",
        "Mittlere Flügeltiefe [m]",
        "Gleitzahl"
    ]

    zeilen = [
        ft.Text(f"{label}: {val:.2f}", color=text_color) for label, val in
zip(beschriftungen, werte)
    ]
    zeilen.append(ft.Text(f"→ Ergebnis: {wert:.2f} {einheit}",
weight="bold", color=text_color))

    return ft.Card(
```

Anhang

```
        content=ft.Container(
            content=ft.Column(zeilen, spacing=2),
            padding=10,
            bgcolor=bgcolor,
            border_radius=10
        )
    )

def add_flugzeug(self, e):
    try:
        # Flugzeug erstellen
        name = self.flugzeug_name.value.strip()
        if not name:
            self.show_popup("Fehler", "Bitte einen Flugzeugnamen eingeben.", success=False)
            return

        if name in self.manager.flugzeuge:
            self.show_popup("Fehler", f"Ein Flugzeug mit dem Namen '{name}' existiert bereits.", success=False)
            return

        try:
            params = {
                key: float(self.inputs[key].value)
                for key in self.inputs
                if self.inputs[key].visible and self.inputs[key].value
            }
            gzc = params.pop("gzc", 0)
        except ValueError as ex:
            self.show_popup("Fehler", f"Ungültiger Parameterwert: {ex}", success=False)
            return

        try:
            flugzeug = Flugzeug(gzc, **params)
            self.manager.add_flugzeug(name, **params, gzc=gzc)
        except ValueError as ex:
            self.show_popup("Fehler", f"Fehler beim Erstellen des Flugzeugs: {ex}", success=False)
            return

        # Tabelle aktualisieren
        try:
            self.update_table()
        except Exception as ex:
            self.show_popup("Fehler", f"Fehler beim Aktualisieren der Tabelle: {ex}", success=False)
            return
```

Anhang

```
        self.show_popup("Erfolg", f"Flugzeug '{name}' erfolgreich hinzuge-
fügt.")

    except Exception as ex:
        self.show_popup("Fehler", f"Unerwarteter Fehler beim Hinzufügen
des Flugzeugs: {ex}", success=False)

def update_table(self):
    try:
        self.table_container.rows.clear()
        for name, flugzeug in self.manager.flugzeuge.items():
            try:
                self.table_container.rows.append(ft.DataRow(cells=[
                    ft.DataCell(ft.Text(name)),
                    ft.DataCell(ft.Text(f"{flugzeug.reichweite():.2f}")),
                    ft.DataCell(ft.Text(f"{flugzeug.energiever-
brauch():.2f}"))
                ]))
            except Exception as ex:
                self.show_popup("Fehler", f"Fehler beim Hinzufügen der Ta-
belle für '{name}': {ex}", success=False)
                return
        self.page.update()

    except Exception as ex:
        self.show_popup("Fehler", f"Unerwarteter Fehler beim Aktualisieren
der Tabelle: {ex}", success=False)

def show_popup(self, title, message, success=True):
    dialog = ft.AlertDialog(
        title=ft.Text(title, weight="bold", color="green" if success else
"red"),
        content=ft.Text(message),
        actions=[ft.TextButton("OK", on_click=self.close_dialog)],
        on_dismiss=lambda e: self.page.update()
    )
    self.page.dialog = dialog
    self.page.dialog.open = True
    self.page.update()

def close_dialog(self, e):
    self.page.dialog.open = False
    self.page.update()
```

Anhang

```
def add_list_item(self, name, reichweite, energieverbrauch):
    # Prüfen, ob das Flugzeug bereits existiert
    for item in self.list_view.controls:
        if isinstance(item, ft.Row) and isinstance(item.controls[0],
ft.Text):
            if name in item.controls[0].value:
                self.show_popup("Fehler", f"Ein Flugzeug mit dem Namen
'{name}' existiert bereits.", success=False)
                return

    # Listeneintrag hinzufügen
    list_item = ft.Row(
        controls=[
            ft.Text(f"✂ {name} - {reichweite:.2f} km - {energiever-
brauch:.2f} kWh/100km", expand=True),
            ft.IconButton(
                icon=ft.icons.DELETE,
                on_click=lambda e: self.remove_list_item(name)
            )
        ],
        spacing=10
    )

    self.list_view.controls.append(list_item)
    self.page.update()

def remove_list_item(self, name):
    try:
        # Flugzeug aus der ListView entfernen
        try:
            self.list_view.controls = [
                item for item in self.list_view.controls
                if not (isinstance(item, ft.Row) and isinstance(item.con-
trols[0], ft.Text) and item.controls[0].value.startswith(f"✂ {name} -"))
            ]
        except Exception as ex:
            self.show_popup("Fehler", f"Fehler beim Entfernen aus der
Liste: {ex}", success=False)
            return

        # Flugzeug auch aus dem Manager entfernen
        try:
            if name in self.manager.flugzeuge:
                self.manager.flugzeuge.pop(name)
        except KeyError as ex:
            self.show_popup("Fehler", f"Flugzeug '{name}' nicht gefunden:
{ex}", success=False)
            return
```

Anhang

```
# Nur exportieren, wenn noch Flugzeuge vorhanden sind
try:
    if self.manager.flugzeuge:
        self.manager.export_flugzeuge_to_excel()
except Exception as ex:
    self.show_popup("Fehler", f"Fehler beim Exportieren der Daten:
{ex}", success=False)
    return

# GUI aktualisieren
try:
    self.page.update()
except Exception as ex:
    self.show_popup("Fehler", f"Fehler beim Aktualisieren der GUI:
{ex}", success=False)
    return

# Erfolgsmeldung
self.show_popup("Erfolg", f"Flugzeug '{name}' erfolgreich ent-
fernt.", success=True)

except Exception as ex:
    self.show_popup("Fehler", f"Unerwarteter Fehler beim Entfernen des
Flugzeugs: {ex}", success=False)

def create_flight_list(self):
    # ListView für die Flugzeugliste erstellen
    self.list_view = ft.ListView(
        expand=True,
        spacing=10,
        padding=10,
        auto_scroll=False
    )

    # Rückgabe der ListView
    return self.list_view

def create_input_fields(self):
    # Eingabeart Dropdown
    self.eingabe_option = ft.Dropdown(
        label="Eingabeart",
        options=[
            ft.dropdown.Option("Gleitzahl bekannt"),
            ft.dropdown.Option("Gleitzahl unbekannt"),
        ],
        value="Gleitzahl bekannt",
        on_change=self.update_input_fields
```

Anhang

```
)

# Luftdichte Option
self.dichte_option = ft.DropDown(
    label="Luftdichte Option",
    options=[
        ft.dropdown.Option("Fester Wert"),
        ft.dropdown.Option("Höhenbasierte Berechnung")
    ],
    value="Fester Wert",
    on_change=self.update_dichte_fields
)

# Parameterfelder
fields = [
    ("flugzeug_name", "Flugzeugname", True),
    ("gZR", "Gleitzahl (GZR)", True),
    ("eta_r", "Wirkungsgrad Reiseflug ( $\eta_R$ )", True),
    ("mu_a", "Batteriemassenverhältnis AM/MAM ( $\mu_a$ )", True),
    ("e_a", "Energiedichte Batterie (Wh/kg)", True),
    ("ts", "Schwebeflugdauer (ts in Sekunden)", True),
    ("eta_s", "Wirkungsgrad Schwebeflugbetrieb ( $\eta_S$ )", True),
    ("mAM", "Maximale Abflugmasse (MAM in kg)", True),
    ("Ar", "Rotorfläche (Ar in m2)", True),
    ("PM", "Passagiermasse (PM in kg)", True),
    ("v_R", "Reisefluggeschwindigkeit (v_R in m/s)", False),
    ("b", "Spannweite (b in m)", False),
    ("c", "Flügelteiefe (c in m)", False),
    ("rho", "Luftdichte (kg/m3)", True),
    ("altitude", "Höhe (m)", False)
]

self.inputs = {}
for key, label, visible in fields:
    self.inputs[key] = ft.TextField(label=label, visible=visible)

# Rückgabe als Column
return ft.Column([
    self.eingabe_option,
    self.dichte_option,
    *self.inputs.values(),
], spacing=10)

def create_result_fields(self):
    self.result_reichweite = ft.Text(value="Reichweite: - km", size=16,
weight="bold")
    self.result_energie = ft.Text(value="Energieverbrauch: - kWh/100km",
size=16, weight="bold")
    self.calculate_button = ft.ElevatedButton("Berechnen",
on_click=self.berechne)
```

Anhang

```
# Rückgabe als Column
return ft.Column([
    self.result_reichweite,
    self.result_energie,
    self.calculate_button
], spacing=10)

def create_tabs(self, parameter_section, result_section):
    # Info-Panel für den Einleitungstabs
    einleitung_panel = self.create_intro_tab()

    # Tabs erstellen
    return ft.Tabs(
        tabs=[
            ft.Tab(text="Einleitung", content=einleitung_panel),
            ft.Tab(text="Berechnung", content=ft.Column([parameter_section, ft.Divider(), result_section, self.list_view], spacing=20, expand=True)),
            ft.Tab(text="Optimierung", content=self.create_optimierung_tab()),
            ft.Tab(text="Kostenmodell", content=self.create_kostenmodell_tab()),
        ],
        expand=True
    )

def create_intro_tab(self):
    # Bild einfügen
    flugzeug_image = ft.Image(
        src="eVTOL2.png",
        width=500,
        height=300,
        fit="contain"
    )

    # Einleitungstext als Column
    return ft.Column(
        controls=[
            ft.Text("Willkommen zum eVTOL-Berechnungsmodell", size=24, weight="bold"),
            flugzeug_image,
            ft.Text("      (Bild KI-generiert)", size=12),
            ft.Text(
                "Dieses Programm ist im Rahmen der Masterprüfung mit dem Thema 'Entwicklung eines Berechnungsmodells zur Abschätzung des Energiebedarfs"
```

Anhang

von eVTOLs und Konzeption eines wirtschaftlich optimierten eVTOL-Gesamtkonzepts' im Studiengang Flugzeugbau an der HAW Hamburg entstanden. "

"Es unterstützt bei der Abschätzung des Energiebedarfs und der Reichweite von eVTOLs und bietet über eine Optimierungsfunktion eine automatische Ausgabe von Designparametern zur Minimierung des Energieverbrauchs. "

"Zudem ist ein Kostenmodell zur Berechnung von wirtschaftlichen Größen implementiert worden.",

```
        size=16,
    ),
    ft.Text("Hauptfunktionen:", size=20, weight="bold"),
    ft.Text(
        "- Berechnung der Reichweite und des Energiebedarfs im Tab
'Berechnung'\n"
        "- Optimierung der Designparameter zur Minimierung des
Energiebedarfs im Tab 'Optimierung'\n"
        "- Berechnung wirtschaftlicher Größen im Tab 'Kostenmo-
dell'\n"
        "- Export aller Daten als Excel-Datei",
        size=16,
    ),
    ft.Text("Anleitung zur Berechnung:", size=20, weight="bold"),
    ft.Text(
        "1. Geben Sie den Namen des eVTOLs ein und wählen Sie die
Eingabeart aus (Gleitzahl bekannt oder unbekannt, Luftdichte bekannt oder un-
bekannt).\n"
        "2. Geben Sie die Parameter ein.\n"
        "3. Klicken Sie auf 'Berechnen', um die Reichweite und den
Energieverbrauch zu berechnen.\n"
        "4. Verwenden Sie die Liste, um eVTOLs hinzuzufügen oder
zu entfernen.\n"
        "5. Alle Änderungen werden automatisch in einer Excel-Datei
gespeichert.",
        size=16,
    ),
    ft.Text("Anleitung zur Optimierung:", size=20, weight="bold"),
    ft.Text(
        "1. Da Default-Werte existieren, kann die Optimierung so-
fort über 'Optimierung starten' gestartet werden. \n"
        "2. Die Wertebereiche und Intervallgrößen können verändert
werden. Geben Sie dazu die Werte in folgender Reihenfolge ein: 'Anfangs-
wert', 'Endwert', 'Schrittgröße'. \n"
        "3. Die Ergebnisse können über den Button 'Ergebnisse ex-
portieren' als Excel-Datei exportiert werden. ",
        size=16,
    ),
    ft.Text("💡 Hinweise:", size=20, weight="bold"),
    ft.Text(
        "- Bitte verwenden Sie bei der Eingabe von Dezimalzahlen
Punkte statt Kommata. \n"
```

Anhang

```
        "- Beachten Sie, dass die Excel-Datei geschlossen werden
muss, bevor neue Änderungen gespeichert werden können. \n"
        "- Die Excel-Dateien werden im Ordner 'dist' gespeichert.
Dieser befindet sich im gleichen Verzeichnis wie die Programmdatei. ",
        size=16,
    ),
    ft.Text("📄 Weiterführende Informationen:", size=20,
weight="bold"),
    ft.Text(
        "- Programm erstellt von Sohail Karimi \n"
        "- Erstprüfer der Masterprüfung: Prof. Dr.-Ing. Alexander
Frenkel \n"
        "- Zweitprüfer der Masterprüfung: Prof. Dr.-Ing. Benedikt
Plaumann \n"
        "- Kontakt für Unterstützung: sohail.karimi@haw-ham-
burg.de",
        size=16,
    ),
    ft.Text("Viel Erfolg bei der Optimierung Ihrer eVTOLs!",
size=16, weight="bold"),
    ],
    spacing=20,
    scroll="auto",
    expand=True
)
```

9.2.5 kostenmodell.py

```
import pandas as pd

class Kostenmodell:
    def __init__(self, pilot=False, Cvbm_f=400000, Cvbm_v=550, mto=1800,
x=0.3, Q=178.2, UCes=100, Pds=228, UCds=25, Cuad=60000, beta=0.2, pE=0.09,
eC=59.36, V=234, ratioDE=0.2, Cvpc0=1450000, VPL=20, Cvpo=80000, Crfc0=600000,
RFL=13, Crfo=82000, ratioTFI=0.2, ratioM=0.075, UCvm=20, ratioSV=0.1, pP=1.5,
np=7, ratioP=0.5, R=100, nM=7.06, nD=300, DR=0.07, VL=13, YT=2400):
    # Allgemeine Parameter
    self.pyr = [0] # Jährliche Einnahmen
    self.pyc = [0] # Jährliche Gesamtkosten
    self.Cj = [0] # Jährliche Teilkosten
    self.pP = pP # Flugpreis pro Passagier ($/km)
    self.nP = np # Anzahl der Passagiere
    self.ratioP = ratioP # Ladefaktor
```

Anhang

```
self.R = R    # Reichweite pro Flugmission in km
self.nM = nM  # Anzahl der Flugmissionen pro Tag
self.nD = nD  # Anzahl der Werktage im Jahr
self.DR = DR  # Diskontierungsrate
self.VL = int(VL)    # Lebensdauer des eVTOLs
self.YT = YT  # Jährliche Flugzeit in h (Standard: 2000)
self.pilot = pilot

# Spezifische Parameter
self.Cvbm_f = Cvbm_f
self.Cvbm_v = Cvbm_v
self.mto = mto
self.x = x
self.Q = Q
self.UCes = UCes
self.Pds = Pds
self.UCds = UCds
self.Cuad = Cuad
self.beta = beta
self.pE = pE
self.eC = eC
self.V = V
self.ratioDE = ratioDE
self.Cvpc0 = Cvpc0
self.VPL = VPL
self.Cvpo = Cvpo
self.Crxfc0 = Crxfc0
self.RFL = RFL
self.Crfo = Crfo
self.ratioTFI = ratioTFI
self.ratioM = ratioM
self.UCvm = UCvm
self.ratioSV = ratioSV

# Passagieranzahl anpassen, wenn Pilot vorhanden
if self.pilot:
    self.nP -= 1

# Berechnet Investment Return Cycle (IRC, Amortisationsdauer)
def calculate_irc(self):
    cumulative_net = 0
    for t in range(len(self.pyr)):
        cumulative_net += self.pyr[t] - self.pyc[t]
        if cumulative_net >= 0:
            return t + 1 # 1-basiert (Jahr)
    return None # Kein Break-even während der Lebensdauer möglich

def calculate_pyr(self):
```

Anhang

```

    """
    Berechnet die Einnahmen fuer ein bestimmtes Jahr t und fuegt es in die
    Liste ein.\n\n
    Parameter:\n
    t: Jahr fuer das die Einnahmen berechnet werden sollen.\n\n
    Rueckgabewert: Eine Liste mit den Einnahmen für alle Jahre.\n
    """
    self.pyr = [0]
    for t in range(1, self.VL + 1):
        zaehler = self.pP * self.nP * self.ratioP * self.R * self.nM *
self.nD
        nenner = (1 + self.DR) ** t
        self.pyr.append(zaehler / nenner)
    return self.pyr
def calculate_pyc(self):
    """
    Berechnet die jaehrlichen Gesamtkosten im Jahr t.\n\n
    Rückgabe: Liste aller Gesamtkosten pro Jahr.\n
    """
    self.pyc = [0]
    for t in range(1, self.VL + 1):
        if t == 1:
            self.Cj.append(self.Cvbm1() + self.Cpt(t) + self.Cplt(t) +
self.Ce() + self.Ci() + self.Ctfi(t) + self.Cm())
        elif t < self.VL:
            self.Cj.append(self.Cpt(t) + self.Cplt(t) + self.Ce() +
self.Ci() + self.Ctfi(t) + self.Cm())
        elif t == self.VL:
            self.Cj.append(self.Cpt(t) + self.Cplt(t) + self.Ce() +
self.Ci() + self.Ctfi(t) + self.Cm() + self.Csv())
        else:
            raise ValueError("Wert t darf nicht negativ sein.")

        nenner = (1 + self.DR) ** t
        self.pyc.append(self.Cj[t] / nenner)
    return self.pyc

def Cvbm1(self):
    """
    Berechnet die Herstellungskosten der eVTOL-Zelle im ersten Jahr.

    Rückgabe:
    float: Gesamtkosten der eVTOL-Zelle im ersten Jahr in $
    """
    m0vbm = self.x * self.mto
    return self.Cvbm_f + self.Cvbm_v * (self.mto - m0vbm)

def Cpt(self, t):
    """
```

Anhang

```
Berechnet die Gesamtkosten für Batterie und Antriebssystem.\n\nParameter:\n  t: Jahr\n  Q (float): Batteriekapazität in kWh (Standard: 77.25)\n  UCes (float): Kosten pro kWh Batterie in $ (Standard: 100)\n  Pds (float): Leistung des Antriebssystems in kW (Standard: 228)\n  UCds (float): Kosten pro kW Antrieb in $ (Standard: 25)\n\nRückgabe:\n  float: Gesamtkosten in $\n  ""\n\n  batterieAustauschZyklus = 3\n  if t == 1 or (t % batterieAustauschZyklus == 0):\n      return (self.Q * self.UCes) + (self.Pds * self.UCds)\n  return self.Pds * self.UCds\n\ndef Cplt(self, t):\n    ""\n    Berechnet die jährlichen Kosten für Pilot oder autonomes Fliegen.\n\n    Parameter:\n      pilot (bool): True für Pilot, False für autonomes Fliegen (Standard:\nFalse)\n      t (int): Jahr\n      Cuad (float): Herstellungs-Kosten UAD-System im ersten Jahr (Stan-\ndard: 60000)\n      beta (float): Anteil der Pilotenkosten (Standard: 0.2)\n\n    Rückgabe:\n      float: Jährliche Kosten\n      ""\n\n      pilotengehalt = 75000\n      if self.pilot:\n          return pilotengehalt\n      if t == 1:\n          return self.Cuad + self.beta * pilotengehalt\n      return self.beta * pilotengehalt\n\ndef Ce(self):\n    ""\n    Berechnet die jährlichen Energiekosten für den eVTOL-Betrieb.\n\n    Rückgabe:\n      float: Gesamte jährliche Energiekosten in $\n      ""\n\n      return self.pE * (self.eC / self.R) * self.V * self.YT * (1 + self.ra-\ntioDE)\n\ndef Ci(self):
```

Anhang

```

    """
    Berechnet die Kosten für die Infrastruktur im Jahr t in $.\n

    Rückgabe:\n
    Kosten für die Infrastruktur im Jahr t in $.\n
    """
    return (self.Cvpc0 / self.VPL) + self.Cvpo + (self.Crfc0 / self.RFL) +
self.Crfo

def Ctfi(self, t):
    """
    Berechnet die Kosten für Steuern, Gebühren und Versicherungen im Jahr
t in $.\n\n

    Parameter:\n
    t(int): Jahr
    rationTFI (float): der Anteil der Kosten für Steuern, Gebühren und
Versicherungen von den direkten Betriebskosten.\n\n

    Rückgabe:\n
    Kosten für Steuern, Gebühren und Versicherungen im Jahr t in $.\n

    """
    return self.rationTFI * (self.Ce() + self.Cplt(t) + self.Ci())

def Cm(self):
    """
    Berechnet die Wartungskosten im Jahr t in $.\n\n
    Parameter:\n
    ratioM: der Anteil der fixen Wartungskosten an den Herstellungskosten
für die eVTOL-Zelle.\n
    UCvm: die variablen Wartungskosten in $/h.\n\n
    Rückgabe: Wartungskosten im Jahr t in $.\n
    """
    return self.ratioM * self.Cvbm1() + self.YT * self.UCvm

def Csv(self):
    """
    Berechnet den Restwert im letzten Jahr der Lebensdauer VL des eVTOLs
in $.

    Parameter:\n
    ratioSV: der Anteil des Restwerts an den Herstellungskosten für die
eVTOL-Zelle\n\n
    Rückgabe: Restwert im letzten Jahr der Lebensdauer VL des eVTOLs in
$.\n
    """
    return -self.ratioSV * self.Cvbm1()

def LC(self):
    """
```

Anhang

```
Berechnet die nivellierten Betriebskosten in $/Passagierkilometer.\n\n    """\n    pyc = self.calculate_pyc()\n    zaehler = sum(pyc)\n    nenner = self.nP * self.ratioP * self.VL * self.nD * self.nM * self.R\n\n    return zaehler/nenner\n\ndef NPV(self):\n    """\n    Berechnet den Barwert (NPV) ueber die gesamte Lebensdauer.\n\n    Rueckgabewert: Barwert (NPV) in $.\n    """\n    pyc_sum = sum(self.calculate_pyc())\n    pyr_sum = sum(self.calculate_pyr())\n    return pyr_sum - pyc_sum\n\ndef IRC(self):\n    """\n    Berechnet die Amortisationsdauer (investment return cycle).\n\n    Ruckgabe: Amortisationsdauer in Jahren.\n    """\n    profit = -1\n    pyc = self.calculate_pyc()\n    pyr = self.calculate_pyr()\n    for i in range(len(pyr)):\n        profit += (pyr[i] - pyc[i])\n        if profit >= 0:\n            return i\n    return -1
```

9.2.6 flugzeugManager.py

```
import pandas as pd\nfrom flugzeug import Flugzeug\n\nclass FlugzeugManager:\n    def __init__(self):\n        self.flugzeuge = {}\n\n    def add_flugzeug(self, name, gzc, eta_r, mu_a, e_a, ts, eta_s, mam, ar,\nmu_p, pm, rho=None, altitude=None):\n        if name in self.flugzeuge:\n            raise ValueError(f"Ein Flugzeug mit dem Namen '{name}' existiert\nbereits.")
```

Anhang

```
        flugzeug = Flugzeug(gzr, eta_r, mu_a, e_a, ts, eta_s, mam, ar, mu_p,
pm, rho, altitude)
        self.flugzeuge[name] = flugzeug
        print(f"Flugzeug '{name}' erfolgreich hinzugefügt.")

def remove_flugzeug(self, name):
    if name not in self.flugzeuge:
        raise ValueError(f"Kein Flugzeug mit dem Namen '{name}' gefun-
den.")
    del self.flugzeuge[name]
    print(f"Flugzeug '{name}' erfolgreich entfernt.")

def get_flugzeug(self, name):
    if name not in self.flugzeuge:
        raise ValueError(f"Kein Flugzeug mit dem Namen '{name}' gefun-
den.")
    return self.flugzeuge[name]

def list_flugzeuge(self):
    return list(self.flugzeuge.keys())

def export_flugzeuge_to_excel(self, filename="flugzeuge.xlsx"):
    with pd.ExcelWriter(filename, engine="openpyxl") as writer:
        for name, flugzeug in self.flugzeuge.items():
            daten = {
                "Gleitzahl": [flugzeug.gzr],
                "Wirkungsgrad Reiseflug": [flugzeug.eta_r],
                "Akkumassenverhältnis": [flugzeug.mu_a],
                "Energiedichte": [flugzeug.e_a],
                "Schwebeflugdauer": [flugzeug.ts],
                "Wirkungsgrad Schwebeflug": [flugzeug.eta_s],
                "Maximale Abflugmasse": [flugzeug.mam],
                "Rotorfläche": [flugzeug.ar],
                "Personenmassenverhältnis": [flugzeug.mu_p],
                "Passagiermasse": [flugzeug.pm],
                "Luftdichte": [flugzeug.rho],
                "Reichweite (km)": [flugzeug.reichweite()],
                "Energieverbrauch (kWh/100km)": [flugzeug.energiever-
brauch()]
            }
            df = pd.DataFrame(daten)
            df.to_excel(writer, sheet_name=name, index=False)
        print(f"Flugzeuge erfolgreich nach '{filename}' exportiert.")
```

10. Selbstständigkeitserklärung



Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

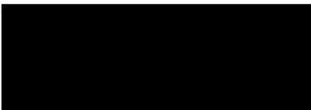
Erklärung zur selbstständigen Bearbeitung einer Abschlussarbeit

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.

<u>Erklärung zur selbstständigen Bearbeitung der Arbeit</u>		
Hiermit versichere ich,		
Name:	Karimi	_____
Vorname:	Sohail	_____
dass ich die vorliegende Masterarbeit _____ bzw. bei einer Gruppenarbeit die entsprechend gekennzeichneten Teile der Arbeit – mit dem Thema: Entwicklung eines Berechnungsmodells zur Abschätzung des Energiebedarfs von eVTOLs und Konzeption eines wirtschaftlich optimierten eVTOL-Gesamtkonzepts		
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.		
- die folgende Aussage ist bei Gruppenarbeiten auszufüllen und entfällt bei Einzelarbeiten -		
Die Kennzeichnung der von mir erstellten und verantworteten Teile der -bitte auswählen- ist erfolgt durch:		
Hamburg	20.05.2025	
Ort	Datum	Unterschrift im Original