A Hybrid Model for the Estimation of Time-Variant Aircraft Parameters

Ein hybrides Modell zur Schätzung zeitvarianter Parameter eines Luftfahrzeugs
Zur Erlangung des akademischen Grades Doktor-Ingenieur (Dr.-Ing.)
Genehmigte Dissertation von Franz A. R. Enkelmann, M.Sc. aus Berlin
Tag der Einreichung: 30. September 2023, Tag der Prüfung: 10. Januar 2024

1. Gutachten: Prof. Dr.-Ing. Uwe Klingauf
2. Gutachten: Prof. Dr.-Ing. Eckhard Kirchner
Darmstadt, Technische Universität Darmstadt

TECHNISCHE
UNIVERSITÄT
DARMSTADT

Fachbereich
Maschinenbau

Institute of Flight
Systems and Automatic
Control

A Hybrid Model for the Estimation of Time-Variant Aircraft Parameters
Ein hybrides Modell zur Schätzung zeitvarianter Parameter eines Luftfahrzeugs

Genehmigte Dissertation von Franz A. R. Enkelmann, M.Sc.

Tag der Einreichung: 30. September 2023
Tag der Prüfung: 10. Januar 2024

Darmstadt, Technische Universität Darmstadt

Für meinen Opa
Dr. Ernstgünter Pettinger
und meinen Großvater
Dipl.-Ing. Walter Enkelmann

Erklärungen laut Promotionsordnung

§ 8 Abs. 1 lit. c PromO

Ich versichere hiermit, dass die elektronische Version meiner Dissertation mit der schriftlichen Version übereinstimmt.

§ 8 Abs. 1 lit. d PromO

Ich versichere hiermit, dass zu einem vorherigen Zeitpunkt noch keine Promotion versucht wurde. In diesem Fall sind nähere Angaben über Zeitpunkt, Hochschule, Dissertationsthema und Ergebnis dieses Versuchs mitzuteilen.

§ 9 Abs. 1 PromO

Ich versichere hiermit, dass die vorliegende Dissertation selbstständig und nur unter Verwendung der angegebenen Quellen verfasst wurde.

§ 9 Abs. 2 PromO

Die Arbeit hat bisher noch nicht zu Prüfungszwecken gedient.

Darmstadt, 30. September 2023

Franz Enkelmann

Abstract

Knowing the health of the aircraft supports safe, efficient, and sustainable operations. It allows maintenance measures and retrofits of the aircraft in service to be used economically. To meet the safety criteria for flight operations, a reliable assessment is required. On the other hand, the assessment methods must become more precise. This ensures technical modifications that even have a minor impact on the health of the aircraft, such as sharkskin, can be classified as significant. If such modifications are used fleet-wide and in a targeted manner, their potential can better be utilized.

To address the demand for increased precision, machine learning methods are used in the context of prognostics and health management. However, the safety criterion in aviation manifests the need for reliable and comprehensive methods, which are experienced in physical model building. Consequently, the combination of both, the hybrid model, found a branch of research to profit from the individual advantages. The hybrid model is considered in this thesis to estimate and predict aircraft system states accurately and physically consistently. The states are set to include time-variant aircraft parameters representing system degradation and recovery within the aircraft life cycle. Their estimation is assumed to be a key element for health assessment.

However, present approaches to hybrid model building are shown to be inappropriate in such state estimation tasks. Therefore, a new hybrid model is developed in this thesis. The developed hybrid model is characterized by a parameter scheduling structure and a recursive filter method for system identification. Thus, a data-driven model, an artificial neural network, can extend a physical model. Furthermore, requirements are defined in initialization, generalization, interpretation, and recovery, addressed within the development process.

The new hybrid model offers stepwise learning while considering measurement noise and joint state estimation. In this way, the parameters and the weights of the artificial neural network are considered states besides the dynamical states. Consequently, both model parts adapt simultaneously. To handle nonlinearities, a modified unscented Kalman filter is implemented. The use of state constraints further improves physical consistency and filter stability.

The developed hybrid model is evaluated using an unmanned aircraft system example. Therefore, a flight test platform is introduced, and a flight simulation environment is developed. A database of flight tests and flight simulations is built, including aircraft modifications that abstract degradation and recovery effects. In four test series, the aircraft is investigated by estimating the system states, including selected parameters, using different physical models and databases. In detail, the unmodified and aerodynamically modified aircraft are considered. In the case of real flight tests, the aircraft's main wing area is extended, and the virtual aircraft's parameters are changed in the case of flight simulations.

The hybrid model can estimate and predict the aircraft system states. The modified parameters are estimated physically consistently in some of the test series. In conclusion, the developed hybrid model can estimate and predict system degradation and recovery in perspective and meets the defined requirements. However, two dilemmas are recognized, which require further improvement. One concerns the hybrid model structure, and the other one the learning algorithm, the filtering procedure. The open hybrid model structure involves an artificial neural network, whose weighting enables high adaptability to the considered database but decreases the model's generalizability. Second, the application-specific initialization of the covariance matrices of the filtering procedure allows for high adaptability but negatively affects the filtering quality. As a result, anomalies within the database cannot be properly detected and separated.

Finally, the new hybrid model is discussed in the context of artificial and natural intelligence. Descriptions learned via data-driven models are compared to the system knowledge that can be physically experienced. The main contribution is extending a physical model by an artificial neural network, which is recursively and simultaneously adapted. For future work, the expansion of physical knowledge using artificial intelligence is proposed, where any dynamical systems can be considered.

Kurzfassung

In der Luftfahrt wird stets nach einem sicheren, effizienten und nachhaltigen Betrieb von Luftfahrzeugen gestrebt. Unterstützt wird dieser durch das Wissen um die Gesundheit der im Betrieb befindlichen Systeme, was eine ökonomische Wartung und Instandhaltung sowie Nachrüstung erlaubt. Um die Sicherheitskriterien für den Flugbetrieb zu erfüllen, ist eine zuverlässige Bewertung notwendig. Andererseits ist eine hohe Genauigkeit der Bewertungsmethoden erforderlich. Dadurch wird sichergestellt, dass technische Änderungen, die sich auch nur geringfügig auf die Gesundheit des Flugzeugs auswirken, wie beispielsweise Haifischhaut, als signifikant eingestuft werden können. Wenn solche Modifikationen flottenweit und gezielt eingesetzt werden, kann ihr Potential besser ausgenutzt werden.

Um die Anforderung einer hohen Genauigkeit zu erfüllen, werden im Kontext der Prognostik und des Gesundheitsmanagements maschinelle Lernverfahren eingesetzt. Die Sicherheitskritikalität von Luftfahrtsystemen manifestiert allerdings den Gebrauch zuverlässiger und nachvollziehbarer Methoden, wie der physikalischen Modellbildung. Folglich bietet sich die Kombination der beiden Ansätze zu einem hybriden Modell an, was einen eigenen Forschungszweig eröffnet und die jeweiligen Vorteile zusammenführt. In dieser Arbeit wird das hybride Modell zur Schätzung und Vorhersage der Systemzustände eines Luftfahrzeugs betrachtet. Dabei werden den Zuständen zeitvariante Parameter zugeordnet, die perspektivisch eine Degradation und Erholung im Lebenszyklus der betrachteten Systeme abbilden können. Vorausgesetzt wird, dass die Kenntnis über die Parameter das Gesundheitsmanagement elementar stützt.

Die bisherigen Ansätze zur hybriden Modellbildung eignen sich jedoch nicht. Daher wird in dieser Arbeit ein neues hybrides Modell entwickelt. Die Neuentwicklung zeichnet sich durch eine Modellstruktur geplanter Parameter aus, die mit einem rekursiven Filterverfahren identifiziert werden. Dieser Ansatz erlaubt die Erweiterung eines physikalischen Modells durch ein datengetriebenes, das künstliche neuronale Netzwerk. Darüber hinaus werden für die Entwicklung Anforderungen im Bereich der Initialisierung, Generalisierung, Interpretation und Adaption definiert.

Das entwickelte hybride Modell erlaubt schrittweises Lernen, die Berücksichtigung von Messrauschen sowie eine gemeinsame Schätzung der beiden beteiligten Modelle. Neben den dynamischen Zuständen werden somit die physikalischen Parameter und Gewichte des künstlichen neuronalen Netzes als Zustände definiert, was eine gleichzeitige Anpassung erlaubt. Nichtlinearitäten werden mit der Wahl des modifizierten Unscented Kalman Filters aufgefangen, dessen Stabilität zusätzlich durch die Begrenzung ausgewählter Systemzustände erhöht wird.

Zur Bewertung des Ansatzes werden ein unbemanntes Luftfahrzeug als Versuchsplattform ausgewählt und eine Simulationsumgebung entwickelt. Mit der Durchführung von Flugversuchen und Flugsimulationen wird eine Datenbasis aufgebaut, die eine Abstraktion von Degradations- und Erholungseffekten beinhaltet. In vier Testserien wird das Luftfahrzeug unter Schätzung der Systemzustände und ausgewählter Parameter untersucht, wobei das physikalische Modell und die verwendete Datenbasis Variationen enthalten. Im Detail wird das unmodifizierte und aerodynamisch modifizierte Luftfahrzeug betrachtet, dessen Hauptflügelfläche technisch erweitert oder dessen Parametrierung innerhalb der Simulationsumgebung verändert wird.

Der Ansatz zur hybriden Modellbildung erfüllt die definierten Anforderungen und ermöglicht eine Schätzung und Vorhersage aerodynamischer Degradation und Erholung. Allerdings werden zwei Dilemmata festgestellt, die zum einen die hybride Modellstruktur und zum anderen den Lernalgorithmus, das Filterverfahren, betreffen. Die offene hybride Modellstruktur beinhaltet ein künstliches neuronales Netz, dessen Gewichtung eine hohe Anpassungsfähigkeit an die betrachtete Datenbasis ermöglicht, die Generalisierbarkeit des Modells jedoch verringern kann. Die anwendungsspezifische Initialisierung der Kovarianzmatrizen des Filterverfahrens ermöglicht ebenfalls eine hohe Anpassungsfähigkeit, diese kann sich aber negativ auf die Filterqualität auswirken, sodass Anomalien innerhalb der Datenbasis nicht mehr sachgemäß erkannt und getrennt werden.

Abschließend wird der neue Ansatz zur hybriden Modellbildung im Kontext künstlicher und natürlicher Intelligenz diskutiert. Im Zuge dessen wird die maschinell erlernte Abbildung dem physikalisch erfahrbaren Systemwissen gegenübergestellt. Wesentliche Beiträge dieser Arbeit bestehen in der Erweiterung eines physikalischen Modells durch ein künstliches neuronales Netz, das rekursiv und gleichzeitig angepasst wird. Für weiterführende Forschungen wird die Erweiterung physikalischen Wissens unter Verwendung künstlicher Intelligenz empfohlen, wobei beliebige dynamische Systeme betrachtet werden können.

Danksagung

Die vorliegende Dissertation ist während meiner Zeit als wissenschaftlicher Mitarbeiter am Institut für Flugsysteme und Regelungstechnik der Technischen Universtität Darmstadt entstanden. In erster Linie möchte ich mich bei *Professor Dr.-Ing. Uwe Klingauf* bedanken, der mich als Institutsleiter bei der Anfertigung der Dissertation fachlich unterstützt hat und mir den notwendigen Rückhalt gab. *Professor Dr.-Ing. Eckhard Kirchner* danke ich für die Übernahme des Korreferats und die kritische Auseinandersetzung mit meinem Dissertationsthema.

Auch möchte ich mich bei allen Kolleginnen und Kollegen bedanken, die mich in dieser Zeit am Institut begleitet haben. Die fachlichen Diskussionen und persönlichen Begegnungen sind für mich von unschätzbarem Wert. Ganz besonderer Dank gilt Professor *Dr.-Ing. Jürgen Beyer*, dessen gemeinsame Forschung im Bereich der Optimalfilter und künstlicher Intelligenz mich und diese Arbeit bereicherte. Für die Diskussionen und gemeinsame Forschung auf dem Gebiet unbemannter Luftfahrzeuge bedanke ich mich bei *Frederik Prochazka*, *Saleh Krüger*, *Tilman Strampe* und *Alexander Joest*. Für die fachlichen Auseinandersetzungen auf dem Gebiet der Prognostik und Zustandsüberwachung bedanke ich mich bei *Henrik Simon*, *Henrik Heier* und *David Hünemohr*. Ihm danke ich besonders für die wertvolle Beratung bei der Verschriftlichung meiner Ergebnisse. Auch möchte ich mich bei *Robert Heigl* der Lufthansa Technik AG bedanken, der mich bis zuletzt über die gemeinsame Projektarbeit hinaus unterstützt hat.

Abschließend bedanke ich mich bei meiner Familie und meinen Freunden. Meinen Großeltern und insbesondere meiner Oma *Helga* danke ich für ihre zeitlose Unterstützung. Meinen Eltern *Andreas* und *Birgit* danke ich für ihre Unterstütztung, ihren moralischen Kompass und ihren Halt in allen Lebenslagen. Meiner Frau *Maike* bedanke ich mich für ihr unermüdliches Wesen und die viele Kraft, die sie mir gegeben hat. Zuletzt bedanke ich mich bei meinem Sohn *Hektor* für seinen wundervollen und phantastischen Blick für unsere Welt.

Frankfurt, September 2023

Franz Enkelmann

Contents

Symbols and abbreviations

Latin Letters

Notation	Description	Unit
A	Rotor surface	m
a	Acceleration	$\frac{m}{s^2}$
alt	Altitude above sea level	m
b	Wingspan	m
C	Physical coefficient	-
c_{p_i}	Motor power coefficient of the respective motor i	-
c_{perm}	Coefficient describing permanent degradation	-
c_{perm_t}	Coefficient describing the period of permanent degradation	s
c_{rev}	Coefficient describing reversible degradation	-
c_{rev_t}	Coefficient describing the period of reversible degradation	s
Dr	Damper ratio	-
D	Aerodynamic drag	N
d	Rotor diameter	m
$\hat{e}$	Residual between measurements and model estimates	-
F	Force	N
$\vec{F}$	Vector of forces	N
$f_{prediction}$	Frequency of state prediction	Hz
f_{update}	Frequency of state update	Hz
GS	Ground Speed (GS)	$\frac{m}{s}$
g	Gravitational constant at the earth surface	$\frac{m}{s^2}$
h_j	Value of the hidden layer neuron with index j	-
I	Current	A
J	Moment of inertia	$\frac{kg}{m^2}$
k	Time step	-
L	Aerodynamic lift	N

Notation	Description	Unit
ℓ	Roll moment	Nm
l_{μ}	Mean Aerodynamic Chord (MAC)	m
$\underline{M}_{be}$	Transformation matrix from experimental to body-fixed CS	-
$\underline{M}_{bg}$	Transformation matrix from geodetic to CS	-
$\underline{M}_{gb}$	Transformation matrix from body-fixed to geodetic CS	-
M_{HI}	Weighting matrix connecting input and hidden layer nodes	-
M_{OH}	Weighting matrix connecting hidden and output layer nodes	-
m	Measurements (introduced in Chapter 2)	-
m	Pitching moment (introduced in Chapter 4)	Nm
$\vec{m}$	Vector of measurements	-
N_i, N_j	Maximum number of the series i and j	-
N_m	Number of measurements	-
N_{nodes}	Number of nodes	-
N_{UKF}	Number of states used by the UKF	-
n^A, n^T	Yaw moment concerning aerodynamics and thrust	Nm
$\vec{P}$	Vector of motor powers	W
$P_{P,1,2,3,4}$	Power of the respective motor P, 1, 2, 3, and 4	W
$\hat{P}_{xx}$	Estimated state covariance	-
$\hat{P}^*_{xx}$	Estimated state covariance without Q_{xx}	-
$\hat{P}_{yy}$	Estimated output covariance	-
$\hat{P}_{ee}$	Estimated residual covariance	-
$\hat{P}_{xy}$	Estimated cross-covariance	-
p	Parameters (introduced in Chapter 2)	-
p	Roll rate (introduced in Chapter 4)	$\frac{rad}{s}$
p_{lat}	Latitude position	deg
p_{long}	Longitude position	deg
p_N, p_E, p_D	Position north, east, down	m
Q_{xx}	System noise covariance matrix	-
$\vec{Q}$	Vector of moments	Nm
q	Pitch rate	$\frac{rad}{s}$
R_{yy}	Measurement noise covariance	-
r	Yaw rate	$\frac{rad}{s}$
r_{xy}	Distance between CG and lift motors	-
$\vec{r}_{yy}$	Standard deviation of measurements	-
S	Wing area	m^2
$\vec{s}$	Vector of position values	m

Notation	Description	Unit
T	Thrust	N
t	Time	s
t_{rev}	Time since recovery of reversible degradation	s
t_{tot}	Total time	s
U	Voltage	V
u	Input (introduced in Chapter 2)	-
u	Airspeed in body-fixed x-direction (introduced in Chapter 4)	$\frac{m}{s}$
$\vec{u}$	Input vector	-
u_k	Input at time step k	-
u_{nn}	Scheduling inputs to the ANN	-
u_i	Input of the index or input layer neuron i	-
$\vec{V}$	Vector of speeds	$\frac{m}{s}$
V_a	True Airspeed (TAS)	$\frac{m}{s}$
V_k	Kalman gain at time step k	-
VS	Vertical Speed (VS)	$\frac{m}{s}$
v	Airspeed in body-fixed y-direction	$\frac{m}{s}$
w	Airspeed in body-fixed z-direction	$\frac{m}{s}$
$w_i^{(c)}$	Weighting of the sigma points	-
w_{ij}	Coefficient of a matrix with row i and column j	-
w_{ji}^{HI}	Weights of the indices i and j connecting the input and hidden layer nodes HI	-
w_{nj}^{OH}	Weights of the indices j and n connecting the hidden and output layer nodes OH	-
$\hat{X}$	Sigma points based on $\hat{P}_{xx}$	-
x	States or transition value in Chapter 2	-
$\hat{x}$	Estimated states	-
$\vec{x}$	State vector	-
$\Delta\hat{x}_k$	State update at time step k	-
Y	Side force	N
$\hat{Y}$	Transformed sigma points based on $\hat{X}$	-
$\hat{y}$	Estimated output	-
$\vec{y}$	Vector of outputs	-
y_n	Output of output layer neuron n	-

Greek Letters

Notation	Description	Unit
α	Angle Of Attack (AOA)	rad
α_{UKF}	Hyperparameter of UKF to adjust the spread of sigma points	-
β	Side Slip Angle (SSA)	rad
β_{UKF}	Hyperparameter of UKF to adjust the mean sigma point	-
$\vec{\delta}$	Vector of deflections	rad
ζ	Rudder deflection	rad
η	Elevator deflection	rad
θ	Pitch angle	rad
κ_{UKF}	Hyperparameter of UKF to adjust the spread of sigma points	-
λ	Hyperparameter of UKF, affecting the weighting of sigma points	-
ξ	Aileron deflection	rad
ρ_0	Air density at standard atmosphere	$\frac{kg}{m^3}$
$\vec{\Phi}$	Vector of attitude angles	$\frac{rad}{s}$
Φ_P, Φ_{Li}	Logical values of active motors	-
ϕ	Roll angle	rad
χ	Ground course	rad
ψ	Yaw angle	rad
$\vec{\Omega}$	Vector of rotational rates	$\frac{rad}{s}$
ω	Motor rotational rate	rpm
ω_0	Eigenfrequency	$\frac{rad}{s}$

Indices

Notation	Description	Unit
b	body-fixed Coordinate System (CS)	-
bat	Value of the battery	-
D	Down	-
d	Value concerns system dynamics	-

Notation	Description	Unit
E	East	-
e	experimental Coordinate System (CS)	-
i,j ,n	Indices of a series or matrix	-
$ideal$	Ideal value	-
ini	Initial value	-
k	Value at time step k	-
$k\|k{-}1$	Value at current time step k given the information at the previous time step $k-1$	-
$k\|k$	Value at current time step k given the information at the current time step k	-
$k{+}1\|k$	Value at next time step $k+1$ given the information at the current time step k	-
Li	Value of a lift motor	-
L	left side	-
lb	Lower boundary	-
max	Maximum value	-
mod	Modified value	-
N	North	-
n	Normed value	-
nn	Value concerns an ANN	-
ops	Operational	-
P	Value of the pusher motor	-
p	Value concerns physical parameters	-
phy	Value of a physical model	-
$prior$	Value of a prior model	-
R	right side	-
red	Reduced	-
rel	Relative value	-
s	Value concerns static moments	-
sim	Value concerns the simulation environment	-
$supplement$	Value of a supplementary model	-
$training$	Value concerns training data	-
ub	Upper boundary	-
UKF	Value concerns the UKF	-
$validation$	Value concerns validation data	-
w	Value concerns wind	-
x ,y ,z	In x-, y-, z-direction	-

Notation	Description	Unit
$_{1,2,3,4}$	Value of the lift motors 1, 2, 3, and 4	-

Superscripts

Notation	Description	Unit
^{A}	Value describes aerodynamics	-
^{c}	Value describes commands	-
$^{(c)}$	Value concerns covariance	-
^{G}	Value describes gravity	-
^{HI}	Connection of the input and hidden layer nodes	-
^{M}	Value describes measurements	-
^{OH}	Connection of the hidden and output layer nodes	-
^{T}	Value describes thrust	-
$^{\mathsf{T}}$	Transposed	-
$^{(x)}$	Value concerns system states	-
*	Value affected by a specification	-

Mathematical operations

Notation	Description	Unit
$\frac{\delta}{\delta u}$	Partial differentiation by signal u	-
$\frac{d}{dt}$	Differentiation by signal t	$\frac{1}{s}$
$f(x, u)$	System state equations as a function of x and u	-
f_h	Activation function of the hidden layer	-
f_y	Activation function of the output layer	-
$h(x, u)$	System measurement equations as a function of x and u	-
$\|M\|_{FN}$	Frobenius Norm of matrix M	-
$max(x)$	Maximum value of signal x	-

Notation	Description	Unit
NRMSE_x	Normed Root Mean Squared Error of est. and act. signal x	-
$R(x)$	Variance of signal x	-
$randn()$	Generation of a set of random normally distributed numbers	-
RMSE_x	Root Mean Squared Error of estimated and actual signal x	-
$\sigma(x)$	Standard deviation of signal x	-
$\bar{x}$	Arithmetic mean of signal x	-
$\dot{x}$	Time derivative of signal x	-

Abbreviations

AC	Alternate Current
ACoRUs	Active fault-tolerant Control for Redundant UAV
ADB	Air Data Boom
AGL	Above Ground Level
AI	Artificial Intelligence
AIAA	American Institute of Aeronautics and Astronautics
AIDS	Airbus Aircraft Integrated Data System
ANN	Artificial Neural Network
AOA	Angle Of Attack
BLDC	Brushless Direct Current
CA	Commanded Airspeed
CAD	Computer-Aided Design
CBM	Condition-Based Maintenance
CFD	Computational Fluid Dynamics
CG	Center of Gravity
CNN	Convolutional Neural Network
CS	Coordinate System
DC	Direct Current
DL	Deep Learning
DM	Digital Model
DR	Desirable Requirement
DS	Digital Shadow
DT	Digital Twin
EASA	European Union Aviation Safety Agency

ECM Engine Condition Monitoring
EKF Extended Kalman Filter
ESC Electronic Speed Controllers
FDIR Fault Detection Isolation and Recovery
FE Feature Engineering
FEM Filter Error Method
FM Flight Mission
FMU Flight Management Unit
FN Frobenius Norm
FNN Feed-forward Neural Network
FNO Fourier Neural Operator
FS Flight Simulation
FaS Fail-Safe
FSR Institute of Flight Systems and Automatic Control
FT Flight Test
FW Fixed Wing
GAN Generative Adversarial Network
GC Ground Course
GPS Global Positioning System
GS Ground Speed
H1 Hybrid Model 1 connecting experience-based and data-driven models
H2 Hybrid Model 2 connecting experience-based and physical models
H3 Hybrid Model 3 connecting different data-driven models
H4 Hybrid Model 4 connecting data-driven and physical models
H5 Hybrid Model 5 connecting experience-based, data-driven, and physical models
IAS Indicated Airspeed
IMU Inertial Measurement Unit
KF Kalman Filter
LHT Lufthansa Technik AG
LS Least Squares
LSTM Long Short-Term Memory neural network
MAC Mean Aerodynamic Chord
ML Machine Learning
MLi Maximum Likelihood
MR Mandatory Requirement
MRO Maintenance Repair and Overhaul
MSR Mean Sample Rate
MTOW Maximum Take-Off Weight

mUKF modified Unscented Kalman Filter

NASA National Aeronautics and Space Administration

NeuralODE Neural Ordinary Differential Equations

NI Natural Intelligence

NRMSE Normalized Root Mean Squared Error

OEM Output Error Method

PENN Physical Encoded Neural Network

PeRCNN Physics-encoded Recurrent Convolutional Neural Network

PGNN Physical Guided Neural Network

PHM Prognostics and Health Management

PINN Physical Informed Neural Network

PLA Polylactide

PMB Power Management Board

PWM Pulse Width Modulated

QC Quadcopter

RLS Recursive Least Squares

RMSE Root Mean Squared Error

RMV Recursive Minimum Variance

RNN Recurrent Neural Network

RUL Remaining Useful Life

SC State Constraints

SCmUKF State Constraint modified Unscented Kalman Filter

SIL Software-in-the-Loop

SSA Side Slip Angle

TAS True Airspeed

TOW Take-Off Weight

TR Transition

TRL Technology Readiness Level

TS Test Series

TU Darmstadt Technical University of Darmstadt

UAS Unmanned Aircraft System

UAV Unmanned Aircraft Vehicles

UKF Unscented Kalman Filter

UT Unscented Transformation

VLOS Visual Line Of Side

VM Virtual Machine

VS Vertical Speed

VTOL Vertically Take-Off and Landing

1. Introduction

With the increasing complexity of technical systems and the associated implementation of sensor and information technologies, the interest in intelligent systems is growing, especially in the aviation industry [AIA16]. The mechanical components of such a technical system experience degradation and recovery in their life cycle. System intelligence now manifests in the ability of diagnosis and prognosis to predict the Remaining Useful Life (RUL) of the components and to enable intelligent maintenance. [Meh22; Ang18; Käh17; Tch+14; Lei14]

Intelligent maintenance relies on finding an optimum between traditional corrective and preventive maintenance procedures for monitoring and maintaining a system's health during operation. The optimization task incorporates information about the system's states and maintenance status to increase safety, cost efficiency, and sustainability. In conclusion, the potential of intelligent maintenance relies on knowledge about the system states in an accurate and physically consistent manner. [VDI22; LS19; PK18]

Today, Machine Learning (ML) methods are used to develop Digital Twins (DTs) or at least aspects of them for the diagnosis and prognosis task. ML has a wide scope of applicability scaling with the considered system's complexity and data access. Accurate results are achieved without requiring certain domain knowledge. But especially for a technical system, ML suffers from interpretability, consistency, and acceptability in a "challenge of trust" [Ful+20]. Based on these criteria, physical models are persuasive but require elaborate modeling procedures enforcing the reduction of model complexity, which results in a loss of accuracy and usability. [Ful+20]

Therefore, the development and use of hybrid models combining different model types have been established as a branch of research in the scientific society [GLL19]. Various hybrid model structures are discussed in the literature, emphasizing the connection between data-driven and physical models. The connection between Artificial Neural Networks (ANNs) and physical models is the most common. However, hybrid models can lack physical consistency and interpretability, which lowers acceptance and usage. [Cha+22; Mur+20; DR20; Hon+20; Jia+19; Cha+19; CPC19]

1.1. Motivation

Adaptive, physically consistent, and reliable system models are required to enable intelligent maintenance of aircraft systems. These models need to be accurate because maintenance measures and aircraft retrofits exist that have a small impact on the aircraft's health state. Hybrid models may address these needs by beneficially combining a physical model and a data-driven approach.

However, current hybrid models are shown to be insufficient. In the literature, data-driven models are focused on being extended by physics. Instead, the motivation in this thesis is to extend physical knowledge by using ML. This way, the required consistency and reliability shall be ensured and accuracy increased.

The research in this thesis contributes to safe, efficient, and sustainable aviation by supporting aircraft maintenance. The intelligent use of maintenance measures and retrofits is addressed in the context of digital platforms [Yan18]. These platforms are fed with flight, maintenance, and operational data and are operated by aircraft manufacturers and airlines today. [Luf21a; Ame21; Air21; Boe22]

1.2. Goals

The goal of the thesis is the development of a new approach to hybrid model building. An extension of a physical model is targeted using an ANN. This provides access to a comprehensible and interpretable model structure, which shall continuously adapt through learning. Insufficient representation of the physical model is to be captured by the ANN in a physically consistent manner.

In contrast to the most common hybrid models used in literature, the introduced structure is intended to provide a deep connection between the physical model and ANN using physical parameters as an interface[1]. Interpretability is desired, which allows the system's behavior to be studied and its understanding to be improved. Including scheduling data, such as maintenance data, can improve state estimation and prediction.

[1] Nells discusses this kind of model structure as a parameter scheduling model, which is rarely used under the topic of Prognostics and Health Management (PHM). The parameter scheduling model structure will be a key element in the design of a new hybrid model. [Nel20]

Estimating states, including time-variant parameters, is essential to the method development. The recursive filter model is chosen for research to handle nonlinearities, stepwise adaption, and the mutual linking of the two model parts. In particular, the Unscented Kalman Filter (UKF) is considered for joint estimation building on the literature from Julier, Uhlmann, Van der Merwe, Wan, Nelson, and Simon. [JUD95; vW03; Sim06a; WN00; WMN00]

For evaluation, flight and maintenance data of aircraft are used. A flight test platform is chosen to ensure knowledge about the aircraft system's characteristics and health. This also allows a retrofit to be applied independently from any other modification, which may take effect in civil aircraft operations. With the ambition to use real flight data acquired without constraints but according to its specifications, a Unmanned Aircraft System (UAS) is used.

Besides flight testing, a Software-in-the-Loop (SIL) environment is developed for flight simulations. Investigating simulated flight data enables the hybrid model to be assessed in a fully observed simulation environment using true states and ideal measurements. Further, time-variant parameters are integrated to be part of the estimation task. The real flight test data is used for plausibility analysis, adjustment of the simulation framework, and, finally, to evaluate the hybrid model when dealing with real sensors and flight conditions.

1.3. Thesis structure

The structure of the thesis is described in the following and shown in Figure 1.1.

After the introduction, the state of the art is presented in Chapter 2. Discussing aspects of aircraft maintenance, the idea of hybrid model building in aviation is derived. Definitions in this field and current hybrid model implementations help clarify the research gap.

The method development is considered in Chapter 3. In Chapter 4, experiments are designed, introducing a flight test platform and a simulation environment to acquire data. The hybrid model is applied and evaluated in Chapter 5. The system states of the non-modified and modified UAS are estimated and predicted, including time-variant parameters.

The thesis closes with a summary, conclusions, and an outlook for future research in Chapter 6.

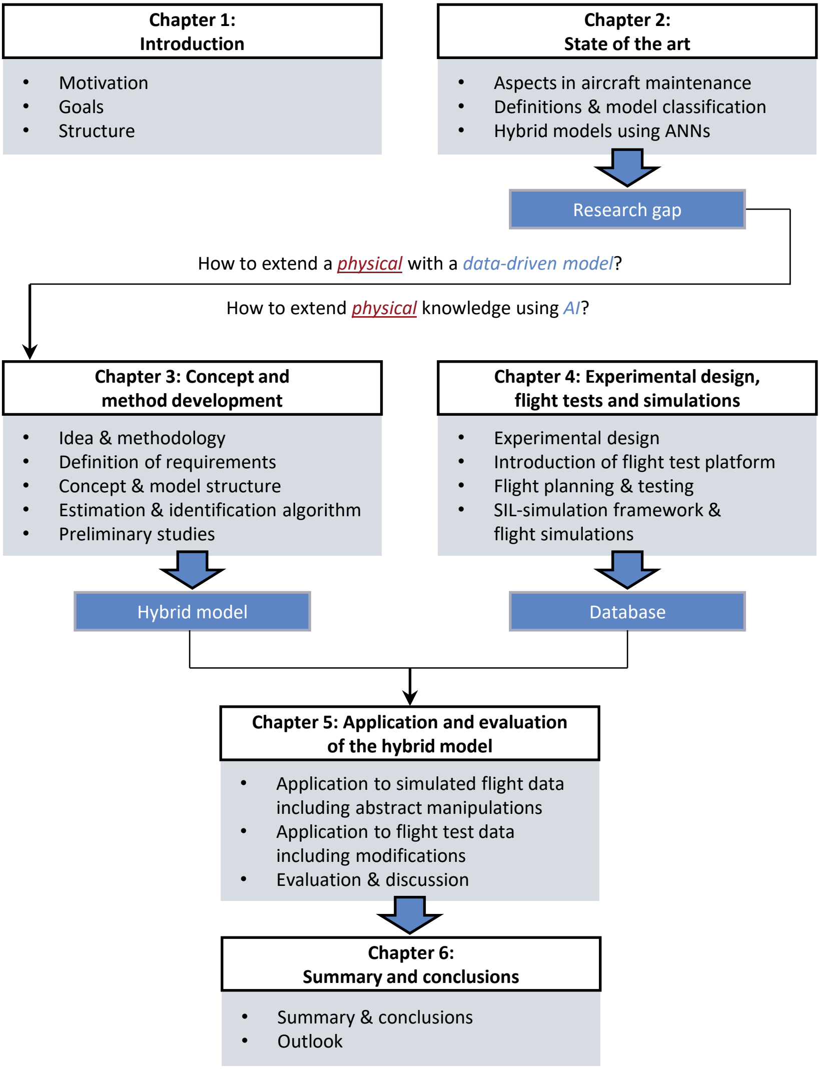

Chapter 1:
Introduction
• Motivation
• Goals
• Structure
Chapter 2:
State of the art
• Aspects in aircraft maintenance
• Definitions & model classification
• Hybrid models using ANNs
Research gap
How to extend a physical with a data-driven model?
How to extend physical knowledge using AI?
Chapter 3: Concept and method development
• Idea & methodology
• Definition of requirements
• Concept & model structure
• Estimation & identification algorithm
• Preliminary studies
Chapter 4: Experimental design, flight tests and simulations
• Experimental design
• Introduction of flight test platform
• Flight planning & testing
• SIL-simulation framework & flight simulations
Hybrid model
Database
Chapter 5: Application and evaluation of the hybrid model
• Application to simulated flight data including abstract manipulations
• Application to flight test data including modifications
• Evaluation & discussion
Chapter 6:
Summary and conclusions
• Summary & conclusions
• Outlook

Figure 1.1.: Structure of the thesis

2. State of the art

This chapter presents a background of the considered use case in aviation and hybrid model building. Starting with the aspects of aircraft maintenance in Section 2.1, a deeper understanding of system models' use and added value in aircraft maintenance is provided. The conventional methods to identify aircraft system models and to evaluate their quality are followed in Section 2.2. Introducing relevant definitions in hybrid models allows the classification of different approaches to creating of hybrid models and their placement in the context of artificial intelligence in Section 2.3. According to the literature, the combination of physical and data-driven models using Artificial Neural Networks (ANNs) is most promising. Therefore, Section 2.4 discusses recent hybrid models using ANNs. This results in the research gap, which is clarified in Section 2.5. The chapter closes with a summary in Section 2.6.

2.1. Aspects of aircraft maintenance

Aircraft system models play an important role in aircraft maintenance. They support flight operations in increasing system safety and reliability. The effort depends on the system's complexity, the maintenance measure, the maintenance strategy, and the quality of the system model used for description. To briefly build an understanding of existing strategies and to emphasize the significance of aircraft system model building, aspects of aircraft maintenance are discussed in the following. [Käh17; Lin05]

> Maintenance is the "combination of all technical, administrative, and managerial actions during the life cycle of an item aimed to retain it in or restore it to a state, in which it can perform the required function." [Deu12]

In its life cycle, an aircraft experiences several maintenance measures, which cause technical modification. In addition, the aircraft's health state varies through the aircraft system's degradation during flight operation. Unless safety is the most striving factor in civil aviation, the health state is reliable for assessing airworthiness. [Meh22; Ang18; Käh17]

However, besides the safety aspect stand different factors in aircraft maintenance [Ang18; Deu12]:

- Safety and security
- Costs
- Availability
- Environmental protection
- Product quality
- Conservation of the item’s value

In particular, the cost factor implies an optimization task. It involves the availability of the aircraft. Especially when the downtime of a component is not expected, unscheduled maintenance and delays occur that cause costs [Ins00]. It further includes aspects of environmental protection as the aircraft’s fuel efficiency and emissions can suffer from a bad health state. This results in increased fuel costs and the need for additional carbon dioxide certificates in aircraft operations [Das17]. These factors increase the repair or breakdown costs with a decreasing degree of preventive actions (thin line), as shown in Figure 2.1.

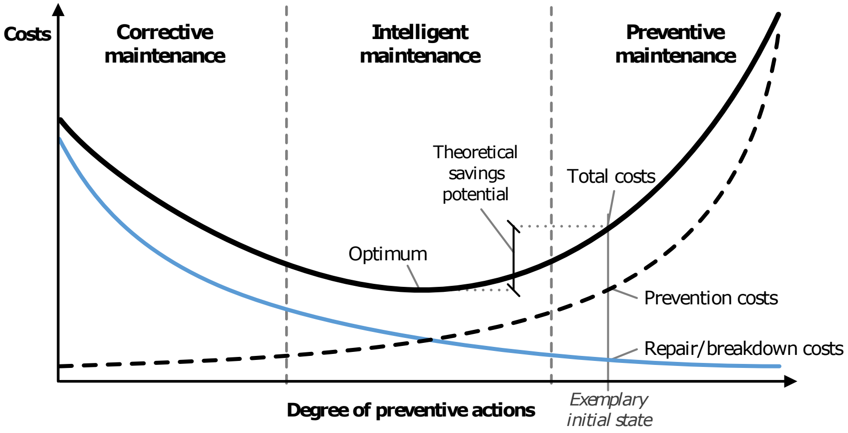


Figure 2.1.: Impact of maintenance strategies on costs, illustrated by [Käh17] and based on [Lei14; Tch+14]

This is offset by the prevention costs (dashed line), which include the monitoring and sensor effort and the unused lifetime of components replaced prior to failure. As a result, an optimum exists, which may grant savings compared to the total costs (thick line) of any exemplary initial state.

Therefore, different maintenance types and strategies exist. On the one hand, corrective maintenance triggers maintenance action whenever a fault or failure is recognized. On the other hand, preventive maintenance is employed to fix any problems in advance. These maintenance types are considered traditional and are briefly discussed in Subsection 2.1.1. If a maintenance procedure enhances the system characteristics beyond its nominal state, perfective maintenance is used [Deu12]. Technical enhancements and modifications are considered retrofits to the aircraft, examples of which are provided in Subsection 2.1.2. Suppose the cost-optimal use of an item is sought. In that case, intelligent maintenance is followed. Here, model-based intelligence is incorporated into the considered system, which is the focus of this thesis and further discussed in Subsection 2.1.3.

2.1.1. Corrective and preventive maintenance

An aircraft system, subsystems, and components adopt different health states within its life cycle. Following [Deu12] and [Ang18], health states are classified into four classes:

- *New state*: the item completely fulfills the requirements, which may lie above the predefined limits for the nominal state.
- *Nominal state*: the item completely fulfills the requirements.
- *Degraded state*: the fulfillment of the required functions is reduced but within defined limits between the nominal and minimum states.
- *Minimum state*: the fulfillment of the required functions is only ensured; further degradation will result in a fault.

Without any maintenance, the health of an item degrades in operation starting from the *new state*, exceeding the *nominal state*, and adopting any *degraded state* until the *minimum state* is passed, a fault occurs, and its required function can no longer be performed. In this case, the reserve of wear-out is fully exhausted during the operating time.

Once a fault is discovered, corrective maintenance can be followed to restore the item to its *nominal state* and, thus, its required function, see Figure 2.2. In this way, the lifetime of an item is fully used. However, since the sudden failure of an item can cause safety issues and unscheduled downtime of the aircraft, additional breakdown costs are incurred, see Figure 2.1.

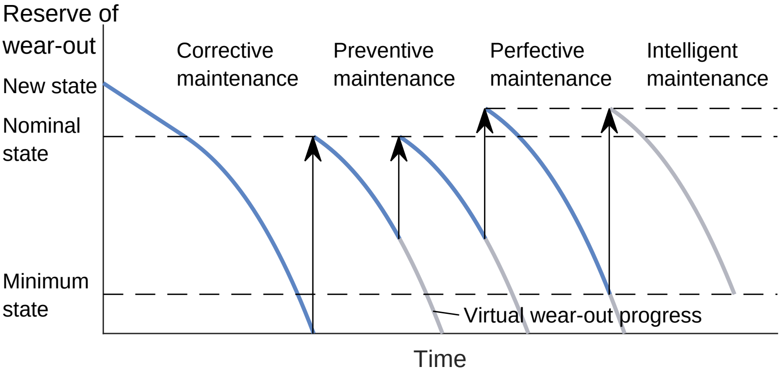


Figure 2.2.: Distinction between corrective, preventive, perfective, and intelligent maintenance, based on [Ang18; Käh17; Deu12].

Preventive maintenance is chosen to avoid additional costs caused by sudden failure of an item. Before the *minimum state* of an item, the *degraded state* of an item is restored to its *nominal state*, see Figure 2.2. Based on the predefined values triggering maintenance action, unused lifetime remains. Consequently, the item is inefficient, and potential cost savings remain unused. This is the case for approximately 85 % of preventive maintenance tasks [Lei14].

As preventive maintenance involves predetermined and condition-based maintenance, values triggering maintenance action can be set differently [Deu10]. For predetermined maintenance, fixed values of, e.g. time intervals, are set to prevent sudden failure and retain required functionality. Maintenance action is triggered condition-based to further approach to the *minimum state* and thus a longer use of an item without risking failure.

Condition-Based Maintenance (CBM) involves monitoring, inspection, testing, and analysis of an item [Deu10]. In particular, using sensor data and system models enables the assessment of the system's health state. The standard procedure is divided into data acquisition, manipulation, state detection, health assessment, prognostic assessment, and advisory generation [Int03].

Various use cases have been elaborated over the past decades [Kar21]. In aircraft maintenance, Engine Condition Monitoring (ECM) has emerged as an important instance, along with various other monitored subsystems. Physical models and ECM data[1] are still preferred to diagnose engine conditions in terms of faults and the ability to assess engine efficiency. Therefore, if declining efficiency is observed, engine maintenance procedures such as overhaul, gas path cleaning, fan blade recontouring, and software updates are considered. [Fen+19; Vol14; KSS14]

The introduction of CBM for aircraft has already prevented sudden failures and reduced costs. Thus, aircraft operations are affected regarding safety and operational efficiency [GSG16; WC11]. Further improvements of CBM are discussed by Chen et al. in the real-time condition monitoring of aircraft in [CWZ12] and by Bernado, Simon, and Schoenhof in the fusion of hardware and software information in [SS21; Ber14].

However, the potential cost savings are offset by the cost of the monitoring itself [Käh17]. While sensor technology evolves and its share of costs decreases, the need for costly expert knowledge remains. This encourages the operator to incorporate Artificial Intelligence (AI) methods, which may reduce the need for expert knowledge. Further, adapting through learning enables one to map any system change accurately. Changes occur through the degradation of the system and the maintenance measures. If maintenance measures enhance the system's health beyond its nominal state, the term perfective maintenance is used, which is discussed in the next Subsection.

2.1.2. Perfective maintenance and aircraft retrofits

Perfective maintenance is handled as optional. With the characteristic to enhance the *nominal state* of an item, perfective maintenance may be combined with any other maintenance type triggering maintenance action. Figure 2.2 shows the recovery of a *degraded state* beyond the item's *nominal state*, an example of preventive and perfective maintenance. However, the perfective maintenance measure may permanently change the item's technical condition, so the *nominal state* must be adjusted. Such measures involve software updates and technical modifications called retrofits when applied to the item in operation. [Käh17; Deu12]

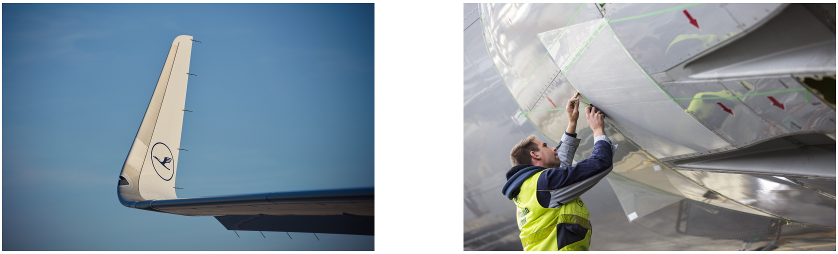
[1] ECM data is aggregated according to certain criteria for takeoff and cruise reports. These can be defined by the operator, by the Maintenance Repair and Overhaul (MRO) supplier, or used as suggested by the manufacturer. [Air02]

Aircraft retrofits are carried out to increase system safety, operational efficiency, and passenger comfort. Depending on the aircraft, retrofit, development, and employment are associated with high costs. Therefore, determining the impact and evaluating retrofit's economic value is of great interest to the aircraft operator.

This thesis considers degradation, recovery, and permanent modification of aircraft aerodynamics. Therefore, some aerodynamic retrofits in civil aviation are discussed in the following.

Aerodynamic retrofits such as winglets, sharklets, and sharkskin are applied to reduce induced, and friction drag. A reduction of drag lowers the fuel consumption. Thus, cost savings arise, and environmentally harmful emissions decrease.

Winglets and sharklets are developed and distributed by the aircraft manufacturers Boeing and Airbus. Historically, winglets were developed based on wind tunnel tests, computer studies, and finally, flight tests by the National Aeronautics and Space Administration (NASA) from 1979 to 1980 promising fuel savings of up to 7 % [Nat04]. Airbus questioned the benefit of winglets versus the weight increase of the aircraft and thus started to develop their design based on the winglets far later in early 2000. They introduced the concept of sharklets in 2012 and started a retrofit program, promising fuel savings of up to 4 %. Today, modern civil transportation aircraft such as the A321 Neo are initially equipped or retrofitted with sharklets, see Figure 2.3a. [Air14; Air13]



(a) Sharklet of an A321 Neo aircraft, with permission from [Deu19]

(b) Application of sharkskin on an aircraft fuselage, with permission from [Luf19]

Figure 2.3.: Aerodynamic retrofits of civil transportation aircraft in service.

Sharkskin is a development of the MRO company Lufthansa Technik AG (LHT) and the chemical company BASF SE. In close cooperation with the Lufthansa Group, the effects of sharkskin were researched through experimental and numerical investigations and flight tests. In addition to the laboratory examination by Leitl et al., flight test data is taken into account utilizing the fuel efficiency analysis provided by the AVIATAR, the digital asset analysis platform of LHT. It measured “variations in fuel consumption before and after the modification“ [Luf21b]. A reduction in frictional drag of 0.8% is determined, resulting in significant fuel savings in flight operations. This has encouraged the Lufthansa Group to equip the Lufthansa cargo fleet of Boeing 777 aircraft with sharkskin, focusing on the aircraft fuselage, see Figure 2.3b. [Luf21b; Lei+19]

Aerodynamic retrofits are performed only once and rarely repeated during an aircraft's life cycle. Therefore, the aircraft and associated flight data are considered separately before and after the retrofit to assess its impact today [Luf21b; Lei+19]. However, when considering aircraft in service as well as flight and maintenance data, influences such as maintenance measures, retrofits, damages, and operating conditions affect the aircraft's health state simultaneously. Their effects overlap in the time domain. To still be able to determine accurate results in retrofit assessment, statistical methods are used. Here, several aircraft of a fleet are considered rather than knowing the individual effect on the efficiency of an aircraft. [EHH22; Luf21b; EHP20; Lei+19]

2.1.3. Intelligent maintenance of aircraft systems

> Intelligent systems sense, interact, and communicate “in an environment with other agents,“ where they “follow [. . .] principles“ and “adapt through learning.“ [Mol20]

Therefore, the intelligence of a technical system, such as an aircraft system, can be assumed by sensors, physically meaningful behavior, and the ability to adapt the system states continuously. However, the aircraft does not necessarily be intelligent in every manner. Only some subsystems need to have AI technologies implemented to classify the aircraft as intelligent, cf. Subsection 2.3.5 [Yin19]. From their employment, an increase in manufacturing, operational efficiency, mission performance, and safety of current and future aerospace systems is expected, following the roadmap of the American Institute of Aeronautics and Astronautics (AIAA) [AIA16].

Intelligent maintenance may be a subdiscipline of AI. It involves CBM and Prognostics and Health Management (PHM) to accurately find the cost optimum in maintaining an ever-changing system, cf. Figure 2.1 [KAC17; AIA16; Joh11]. Consequently, the *minimum state* of an affected item is found to advise maintenance action to restore its *nominal state*, cf. Figure 2.2. If the *nominal state* has changed due to prior technical modifications, intelligent maintenance is set to learn and adjust the *nominal state* during operation.

While the first implementations of AI for aircraft engine condition monitoring were referred to in 1995, the tasks of AI have increased in trust and effort today [AIA16; AJV95]. This occurs through the technological progress of sensor technology and data acquisition and the implementation of physical, data-driven, and hybrid models using state-of-the-art algorithms, see Section 2.3.

With the ability to adapt through learning, AI methods differ from conventional methods used for CBM. It enables system models to adapt to new and unknown effects and inputs, ensuring reliable mapping in operation. Thus, intelligent maintenance uses AI technologies to take over CBM tasks, including on-condition maintenance, predictive diagnostics, and predictive prognostics. Here begins the field of predictive maintenance, where PHM algorithms settle. [KKH18; Deu12; Joh11]

Thus, diagnostics can be carried out to prognostics within the PHM topic, where Lughofer et al., Pecht et al., and Johnson grant an overview [LS19; PK18; Joh11]. Johnson strictly defines prognosis as:

> “Predicting the time at which a component will no longer perform its intended function.“ [Joh11]

In this case, appropriate models predict the future health state until a failure occurs, which is reflected in the Remaining Useful Life (RUL). The health state and its threshold underlying the RUL are system-dependent values determined based on expert knowledge. Knowledge about the degradation and failures of the system is mandatory. [Lei+16; Li+15; Si+12]

In conclusion, a reliable estimation and prediction of system states is required for RUL prediction. This allows variable future operating states to be considered for the RUL prediction, as Mehringskötter studied. To perform such predictions, Mehringskötter assumes the actual and future operating states to be given. Such data may be used to assess of the aircraft’s health and efficiency using today’s data-driven models. [Meh22; EHP20; Bau19; BK19]

Thus, “failure prognosis has been approached [. . .] to artificial intelligence tools and methodologies“ [Joh11]. Conversely, AI has acquired the ability to perform PHM tasks. However, it is evident first to determine the system states correctly before predicting them into the future and using them for PHM [Vac+06]. As the determination of aircraft system states relies on the system model, the next Section discusses the fundamentals of aircraft system identification.

2.2. Fundamentals in aircraft system identification

The process of aircraft system identification is discussed by Klein, Morelli, Stevens, Johnson, Lewis, Brockhaus, Alles, Luckner, Tischler, Remple, and Jategaonkar [KM16; SJL16; Jat15; BAL11; TR06]. They employ the ability of system identification for aircraft, which is considered in two parts. First, system identification consists of a model concerning inputs, architecture, dynamics representation, order, structure, parameters, and validation. Second, feedback to the model involves user interaction or an automatic algorithm. At this point, using experiments overcomes the use of prior knowledge. [Nel20]

The model structure and the algorithm used for identification will be part of the development process in Chapter 3. In defining time-variant parameters to be part of the hybrid model, mapped by a data-driven model, and identified using a recursive filter method, some fundamentals are provided in the following. These involve an overview of conservative aircraft system identification methods in Subsection 2.2.1, focusing on the recursive filter method in Subsection 2.2.2. Since observability turns out to be a key measure of control theory to be considered in hybrid model building, Subsection 2.2.3 discusses its definitions. Finally, some model evaluation techniques and metrics are given in Subsection 2.2.4 to be used later.

2.2.1. Overview of aircraft system identification methods

Physical models are preferred for aircraft system identification. The underlying algorithms to optimize the models are classified and discussed in [Nel20; KM16; BAL11; TR06], where a reduced representation based on the work by Nelles is given below:

- Linear optimization
  - Least Squares (LS)
  - Recursive Least Squares (RLS)
- Nonlinear local optimization
  - Batch and sample adaption
  - General gradient-based algorithms
  - Nonlinear least squares problems

- Nonlinear global optimization
    - Evolutionary algorithms
- Unsupervised learning techniques
- Model complexity optimization
- Recursive Minimum Variance (RMV) algorithms

In the context of aircraft systems, Jategaonkar emphasizes optimizing algorithms within the Output Error Method (OEM) or the Filter Error Method (FEM) for system identification. In the case of the OEM, the LS, as well as the Maximum Likelihood (MLi) method, are commonly used to determine the aircraft system parameters directly from noisy measurements. At the same time, MLi is considered for both linear and nonlinear optimization. [Jat15]

Alternatively, Jategaonkar proposes the FEM to identify the parameters inside a filter structure with a Gauß-Newton optimization following a state update every time step [Jat15]. Another example of using a filter structure is given by Curvo, implementing an Extended Kalman Filter (EKF) to estimate aircraft aerodynamic derivatives in [Cur00].

Recursive filter methods such as the EKF are suitable for system identification. They consider noise, handle nonlinear stochastic problems, and estimate bias-free parameter values, which are discussed in more detail below. [BK94]

2.2.2. The recursive filter method

Recursive filter methods differ from optimization methods using batch and iterative sample adaption. Most importantly, they minimize the variance of residuals recursively (RMV) instead of minimizing a residual directly [BK94]. Filter methods can be used online and enable adaption through learning, which is an important characteristic in the context of intelligent systems, cf. Subsection 2.1.3. [Nel20]

The Kalman filter and its variants belong to the field of recursive filter methods. Giving a brief introduction in the following, further reading of the literature by Gelb et al., Haykin, and Simon is recommended [Gel+06; Sim06a; Hay96]. Since the Unscented Kalman Filter (UKF) will be chosen to identify a hybrid model of an aircraft system in this thesis, more specific literature by Julier, Uhlmann et al., and Van Der Merwe and Wan is recommended [vW03; JUD95]. They introduced the UKF in 1995 [JUD95].

The following is based on their work, where the notation used is based on the work by Krebs [Kre80].

Kalman filters consider a system with the system equation $f(x, u)$, the estimated state, $\hat{x}_{k|k}$, and the input u_k providing the state prediction with $\hat{x}_{k+1|k}$, and the measurement equation $h(x, u)$ estimating the output $\hat{y}_{k|k-1}$. The measurement m_k at timestep k and the estimated model output $\hat{y}_{k|k-1}$ yield the residual $\hat{e}_k$, see Eq. 2.1.

$$\hat{e}_k = m_k - \hat{y}_{k|k-1}, \tag{2.1}$$

The structure of the nonlinear discrete-time Kalman filter is given in Figure 2.4. Variants of the Kalman filter mainly differ in considering the Kalman gain V_k, see Eq. 2.2. V_k is used to update the state to $\hat{x}_{k|k}$ based on the residual $\hat{e}_k$, see Eq. 2.3.

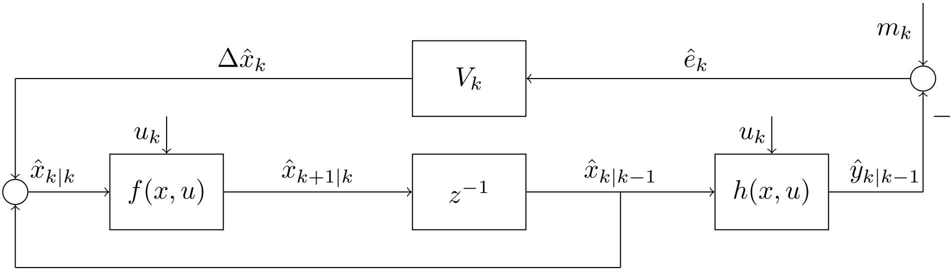


Figure 2.4.: Structure of the nonlinear discrete-time Kalman filter, based on [Hay96].

$$V_k = \hat{P}_{xy_{k|k-1}} \times \hat{P}^{-1}_{ee_{k|k-1}} \tag{2.2}$$

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + V_k \times \hat{e}_k \tag{2.3}$$

The Kalman gain is subject to an optimization problem concerning the covariance estimations of the residual covariance $\hat{P}_{ee}$, output covariance $\hat{P}_{yy}$, state covariance $\hat{P}_{xx}$, and cross-covariance $\hat{P}_{xy}$. The problem includes the assumed uncertainty of the measurements with the measurement noise covariance matrix R_{yy} and the fictional uncertainty of the system with the system noise covariance matrix Q_{xx}:

$$\hat{P}_{ee_{k|k-1}} = \hat{P}_{yy_{k|k-1}} + R_{yy}. \tag{2.4}$$

The state covariance updates follow by using V_k and $\hat{P}_{xy}$:

$$\hat{P}_{xx_{k|k}} = \hat{P}_{xx_{k|k-1}} - V_k \times \hat{P}^{\mathsf{T}}_{xy_{k|k-1}}. \tag{2.5}$$

Finally, the updated states $\hat{x}_{k|k}$ are predicted into the next time step $k+1$ based on $f(x,u)$, see Eq. 2.6.

$$\hat{x}_{k+1|k} = f(\hat{x}_{k|k}, u) \tag{2.6}$$

The definition of the covariance matrices $\hat{P}_{xx}$, $\hat{P}_{yy}$, and $\hat{P}_{xy}$ depend on the Kalman filter variant chosen. The most common variants are the EKF, UKF, and particle filter for nonlinear system equations. They differ due to the computational effort needed and the accuracy of the covariance estimation, see Figure 2.5a.

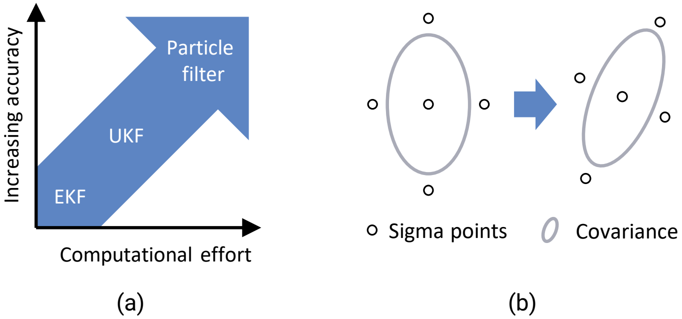


Figure 2.5.: The increasing accuracy and computational effort of the EKF, UKF, and particle filter (a), based on [Sim06a] and the Unscented Transformation (UT) of the sigma points with the corresponding covariance (b), based on [vW03].

The EKF uses partial differentiation in each operating point to handle nonlinear systems. The UKF uses so-called sigma points, which are chosen to represent the covariance of a distribution. The UKF exceeds the accuracy of the EKF in terms of state and covariance estimation, while it can cause an increased computational effort on a reasonable order of magnitude. The particle filter is not considered in this thesis because of a significant increase in the computational effort, although increased accuracy is expected. [Sim06a; vW03; JUD95]

As discussed in Chapter 3, the UKF is well suited for the nonlinear system identification needed in the proposed hybrid model structure. The UKF is characterized by its covariance calculation and state prediction using the UT. For the UT, the sigma points are defined. They are adjusted according to the hyperparameters α_{UKF}, κ_{UKF} and β_{UKF} to capture the covariance of the states. The hyperparameters, α_{UKF}, and κ_{UKF} influence the spread of the sigma points and thus filter performance. The hyperparameter β_{UKF} describes a shift of the mean value and is used to handle distributions different from the normal distribution. With the number of states N_{UKF} and the hyperparameter $\lambda = \alpha_{\text{UKF}}^2(N_{\text{UKF}} + \kappa_{\text{UKF}}) - N_{\text{UKF}}$, the weighting of the sigma points is defined in Eq. 2.7. [Hay04; vW03; JUD95]

$$w_i^{(c)} = \frac{\lambda}{2(N_{\text{UKF}} + \lambda)}, i = 1, ..., 2N_{\text{UKF}} \tag{2.7}$$

Using $w_i^{(c)}$ and the sigma points $\hat{X}_{k|k-1}$ around $\hat{x}_{k|k-1}$, as well as their mapping using $\hat{Y}_{k|k-1} = f(\hat{X}_{k|k-1}, u_k)$, the covariances $\hat{P}_{xy_{k|k-1}}$ and $\hat{P}_{yy_{k|k-1}}$ are calculated in Eq. 2.8 and in 2.9.

$$\hat{P}_{xy_{k|k-1}} = \sum_{i=1}^{2N_{\text{UKF}}} w_i^{(c)}(\hat{X}_{k|k-1} - \hat{x}_{k|k-1})(\hat{Y}_{k|k-1} - \hat{y}_{k|k-1})^{\mathsf{T}} \tag{2.8}$$

$$\hat{P}_{yy_{k|k-1}} = \sum_{i=1}^{2N_{\text{UKF}}} w_i^{(c)}(\hat{Y}_{k|k-1} - \hat{y}_{k|k-1})(\hat{Y}_{k|k-1} - \hat{y}_{k|k-1})^{\mathsf{T}} \tag{2.9}$$

The covariance prediction results from transforming the sigma points following $\hat{X}_{k+1|k} = f(\hat{X}_{k|k}, u_k)$ used in Eq. 2.10.

$$\hat{P}^*_{xx_{k+1|k}} = \sum_{i=1}^{2N_{\text{UKF}}} w_i^{(c)}(\hat{X}_{k+1|k} - \hat{x}_{k+1|k})(\hat{X}_{k+1|k} - \hat{x}_{k+1|k})^{\mathsf{T}} \tag{2.10}$$

The prediction of $\hat{P}_{xx_{k+1|k}}$ finally considers Q_{xx} in Eq. 2.11.

$$\hat{P}_{xx_{k+1|k}} = \hat{P}^{*}_{xx_{k+1|k}} + Q_{xx} \tag{2.11}$$

Parameters are defined as parameter states, and the state vector is extended to enable parameter estimation. The dual and joint estimation methods are suitable for state estimation using the UKF. In the case of dual estimation, the covariance calculation of the dynamical states, which describe the system dynamics, and of the parameters states, which are typically set to be constant are separated.

For joint estimation, a combined covariance calculation takes place. Therefore, the entries of the secondary diagonals connect the parameter states and the dynamical states, which have a significant value for the state update. Further readings and the mathematical background are provided in the literature. [ZLL17; MPA13; Sim06a; WN00; WMN00; Nel00]

In this thesis, a modification of the UKF is used. The modification simplifies the calculation of the mean value of the states and the covariances using sigma points[2]. Also, the state and covariance estimation is separated, which improves filter stability. The modified Unscented Kalman Filter (mUKF) is derived for state estimation within the proposed hybrid model discussed in Chapter 3. A complete description of the filter algorithm used is given in [EK22].

2.2.3. Observability of system states

In this thesis, a dynamic aircraft system is considered, which is nonlinear and stochastic. It is characterized by the nonlinear dependencies between the system states and the presence of noise, as expected for aircraft systems concerning noisy flight measurement data.

The observability of such a system has to be treated differently compared to the linear deterministic system. Therefore, state estimation and stability approval are more challenging, while the separation theorem[3] is no longer applicable. Thus, observability is considered for evaluation and discussion of the hybrid model. This is new in building hybrid models, which are otherwise treated as data-driven, see Section 2.3. [Ada14]

[2] Note that the covariance calculations in Eq. 2.8, Eq. 2.9 and, Eq. 2.10 already take into account the sigma points without the mean state with index $i = 0$.

[3] The separation theorem claims that the states are independent of each other [Ada14].

The two definitions follow from the mathematically shaped formulation given in the literature to which reference is made accordingly:

Deterministic observability

> A system is observable if the initial states are determinable using the given in- and outputs of the system in a defined time interval. [Ada14; LB11]

Stochastic observability

> A system is stochastically observable if the solution for the initial states converges with the given in- and outputs of the system in a defined time interval or if the initial states lie in a defined solution space with the given in- and outputs of the system in a defined time interval. [Ada14; Che80]

In literature, such criteria are also discussed as weak or local observability, understood as stochastic observability [Ada14].

Observation of nonlinear stochastic systems opens another field of research, where contributions are traced back to Chen, who defined stochastic observability and controllability in 1980 [Che80]. Further research is provided by van Handel, Liu, McDonald, and Yuksel, who finally discuss stochastic observability in the context of filter stability till 2018 [MY18; LB11; van09].

2.2.4. Evaluation techniques in model building

Within the hybrid model, an optimization method is considered for state estimation. The performance of the hybrid model is characterized by the accuracy and reliability of the estimated states, particularly the time-variant parameter states. The metrics used to assess the model's performance are given below. Qualitative studies can be added in terms of plausibility and applicability to evaluate the development in Chapter 3.

According to Johnson, model performance is rated by:

- accuracy,
- precision,
- and convergence. [Joh11]

Accuracy is described with an average bias. Precision describes the spread of an error, which the standard deviation can capture. Convergence is the property of the learning algorithm to sufficiently decrease the deviation from the objective function relative to the prior estimates. [Joh11]

The proposed hybrid model is investigated in all three disciplines using observed signals and objectives considering physical parameter states additionally. The consideration of flight data allows for the evaluation of the applicability of the hybrid model and a plausibility analysis. A defined simulation environment enables the acquisition of flight simulation data, including the simulated system's true states and ideal measurements. These have a significant value for assessment.

Thus, quantifying the accuracy, precision, and convergence of both dynamical and parameter states is enabled, which grants insight into the learning behavior of the hidden states.

A metric to describe accuracy is the Root Mean Squared Error (RMSE):

$$\mathrm{RMSE}_x = \sqrt{\frac{1}{N_i}\sum_{i}^{N_i}(x_i - \hat{x}_i)^2} \tag{2.12}$$

Due to the squared error, deviations of greater magnitude are weighted higher. The RMSE appears in the physical dimension of x resp. $\hat{x}$. Thus, a normalization offers the comparison of the RMSE of different magnitudes and resolves the dimensionality:

$$\mathrm{NRMSE}_x = \frac{\mathrm{RMSE}_x}{x_{rel}}, \text{ with } x_{rel} \in [\sigma(x)\, x_{max}]. \tag{2.13}$$

Typically normalized by the standard deviation $\sigma(x)$, the relation by x_{rel} can proceed using the maximum value x_{max} [Oli12].

The standard deviation $\sigma(x)$ can also be used to describe the precision of the model:

$$\sigma(x) = \sqrt{\frac{1}{N_i - 1}\sum_{i}^{N_i}(x_i - \bar{x})^2}. \tag{2.14}$$

Using the standard deviation is advantageous when using a Kalman filter method and analyzing its covariance estimates[4].

Examples of the applied UKF can be found in the literature analyzing convergence criteria [DBA21; FFT07]. Visualization helps to examine convergence behavior, which is performed for residuals and covariances of the measurements and states, respectively.

The number of dynamical and parameter states increases with system model complexity, which is expected to be in a manageable range. Instead, the number of ANN weight states may greatly increase with the number of input, hidden layer, and output nodes. Therefore, the Frobenius Norm (FN) is introduced to examine the convergence behavior while estimating the ANN weight states. [DBA21; Tho+20; Gv96]

$$\|M\|_{FN} := \sqrt{\sum_{i=1}^{N_i}\sum_{j=1}^{N_j}|w_{ij}|^2} \tag{2.15}$$

The FN summarizes any change in a matrix M affected by its coefficients w_{ij}. It is proposed for convergence analysis by Daid et. al in the context of the UKF and by Thoiyab et al. in the context of the ANN. [DBA21; Tho+20]

2.3. Definitions in the field of hybrid models

A hybrid model involves the combination of two or more models of different types to benefit from their advantages [GLL19; KAC17; LK14]. In literature, a wide range of research, combinations, solutions, and implementations of hybrid models exist, partly discussed in the following [Far+22; GLL19; LK14].

[4] Note, the variance ν depends on the standard deviation with $\nu = \sigma^2$

According to Liao and Kottig, three model types provide coupling partners of a hybrid model:

- the experience-based model[5],
- the physical model, see Subsection 2.3.1,
- and the data-driven model, see Subsection 2.3.2.

Any combinations of these result in a hybrid model H1 to H5, see Figure 2.6.

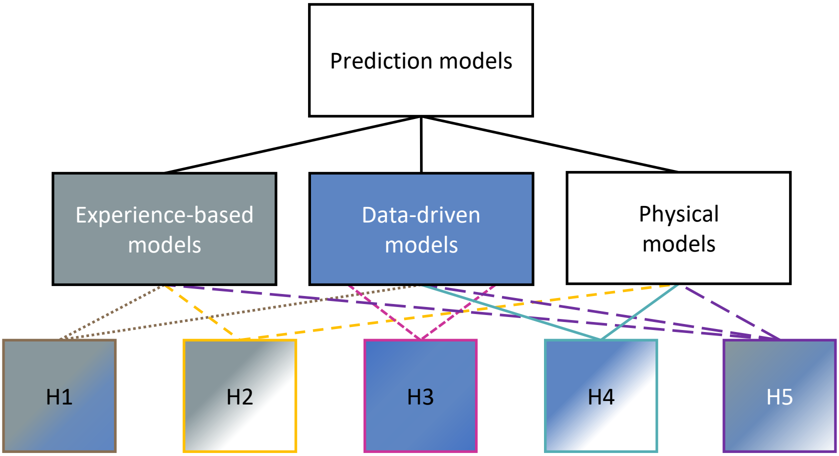


Figure 2.6.: Possible combinations of the three major classes of prediction models, based on [LK14]

The experience-based model occurs in H1 with the data-driven model, in H2 with the physical models, and in H5 with both. These models rely on the capability of the domain expert, who specifies rules for a system. As system complexity increases, experience-based models suffer from applicability and consistency-checking problems. [LK14]

Hybrid Model 3 connecting different data-driven models (H3) covers the combination of two or more data-driven models. The broad availability of different data-driven

[5] "Experience-based models correlate expert knowledge and engineering experience (often documented as IF-THEN rules directly from domain experts) with the observed situation to infer RUL from historical measurements or events." [LK14]

model types comes with various learning strategies that allow combinations with themselves. For example, a decision tree may be combined with an ANN. [LK14]

Hybrid model 4 (H4) combines physical and data-driven models. This combination represents the current understanding of hybrid models:

> “Hybrid approaches are to integrate advantages of both physics-based and data-driven methods to improve the prediction capability.“ [KAC17]

> “Hybrid approaches are the integration of both data-driven and physics-based prognostics. It is an intuitive idea to leverage the strength of [...] data-driven and physics-based prognostics to improve the prediction performance.“ [GLL19]

Therefore, this thesis focuses on hybrid models consisting of a physical and a data-driven model. A uniform understanding is provided because various names are used in the literature to describe such models. Consequently, some definitions of hybrid models are given in the following. These are settled in the context of intelligence, machine learning, and digital models. In the context of Natural Intelligence (NI) and AI, the labels of physical, hybrid, and data-driven models are given in Figure 2.7.

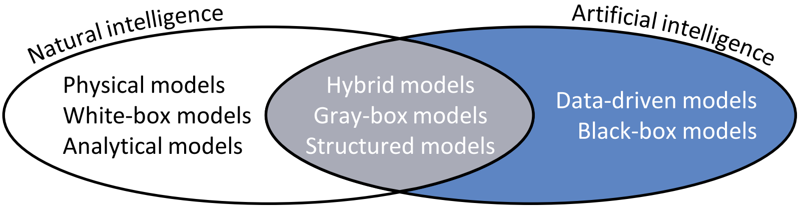


Figure 2.7.: Nomenclature of physical, hybrid, and data-driven models

Hybrid, gray-box, and structured models are referred to as hybrid models. Physical, white-box, and analytical models are called physical models, defined in Subsection 2.3.1. Data-driven and black-box models are called data-driven models, as defined in Subsection 2.3.2. The combination of the physical and data-driven models results in the hybrid model, which is explained in more detail in Subsection 2.3.3.

These models create a digital object that represents a physical object in a virtual environment. Depending on their implementation and linking, they offer the design of a Digital Model (DM), Digital Shadow (DS), or Digital Twin (DT), which are defined in Subsection 2.3.4. Superior to this stands the use of NI and AI, defined in

Subsection 2.3.5, followed by the definition of Machine Learning (ML), Deep Learning (DL), and ANN in Subsection 2.3.6.

2.3.1. Physical models

Physical models are characterized by their model structure and parametrization, which is based on physical laws, experimental work, and a historically grown fund of knowledge. Their comprehensibility and interpretability are compelling. However, their complexity can increase significantly, leading to laborious identification processes and cost inefficiencies, both financial and computational. Reducing complexity can lower costs but negatively impact model accuracy and informative value. As a result, reliability suffers, requiring dexterity to find a reasonable complexity of a physical model that can tolerate systematic errors up to a point. [GT21; Nel20; PLP20; Are+18; Har+16; Ada14]

2.3.2. Data-driven models

The data-driven model structure is an open structure, which can vary through the setting of hyperparameters or the chosen learning strategy. The training process, which describes the learning phases of the algorithm, follows a defined objective. Formulated as a mathematical function, a minimization problem is typically described in the case of supervised learning. In addition to a classification learner used to predict discrete class labels, the regression learner predicts continuous quantities such as the time series data used in this thesis.

Thus, the data-driven model depends on the database provided and behaves according to the objective function. A difficulty occurs when the chosen algorithm perfectly reproduces the individual behavior of the training data, which is called overfitting. Overfitting is avoided by setting termination criteria and data preprocessing.

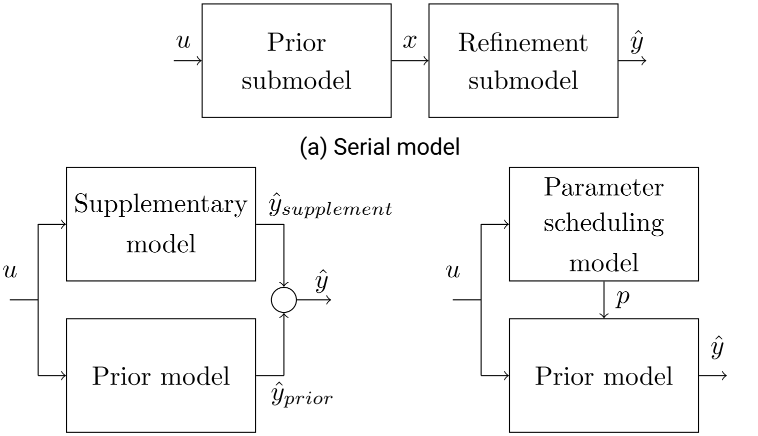
Data-driven approaches are beneficial in their applicability, adaptability, implementation effort, and accuracy. Still, they can suffer from physical consistency, interpretability, comprehensibility, and explainability because the inner mathematics does not provide access to understanding the model behavior. [PLP20; PK18; Are+18; KAC17; Har+16; Roj96]

2.3.3. Hybrid models

In this thesis, a hybrid model is considered a combination of a physical and data-driven model. Using a data-driven model within a hybrid model increases its scope and accuracy. On the other hand, the reliability and interpretability may be increased when incorporating a physical model. However, both data and domain knowledge are required. [GLL19; LK14]

According to Guo et al. and Liao and Kottig, a hybrid model can be connected differently. The connection of the model parts can be in series, in parallel, or in replacing the physical with a data-driven model. [GLL19; LK14]

Nelles considers an additional type of connection: the parameter scheduling model [Nel20]. He introduces the terms of the *prior submodel* and *refinement submodel* for serial hybrid models, the *supplementary model* and *prior model* for parallel hybrid models, and the *parameter scheduling model* and *prior model* for parameter scheduling models, which are structured in Figure 2.8.



(a) Serial model

(b) Parallel model

(c) Parameter scheduling model

Figure 2.8.: Structures of hybrid models: serial model (a), parallel model (b), and parameter scheduling model (c), based on [Nel20]

Physical and data-driven models can be arbitrarily involved in the proposed hybrid model structures, where transition values or states (x) are used to connect both models in the case of a serial structure in Figure 2.8a.

In the case of a parallel model structure, the estimated outputs ($\hat{y}$) of both models are merged, while mergers can be realized additive, multiplicative, or by filtering methods, see Figure 2.8b [Nel20; Sie+19].

In the case of a parameter scheduling model, the inner parameters (p) of the prior model are provided by the parameter scheduling model sharing the input (u) in Figure 2.8c [Nel20]. As discussed in Chapter 3, the parameter scheduling structure has a significant advantage compared to serial and parallel model structures. This is due to the parameters, which are not considered constant but can be variable. Thus, both the data-driven and the physical model can adapt through learning, an important property in AI and DTs discussed below.

2.3.4. Digital models, shadows, and twins

Briefly, definitions of the DM, DS, and DT are provided to guide the derivation of the requirements for the hybrid model and its development in Chapter 3. This is also seen as necessary because of misconceptions and misidentified DTs in the literature, which should be avoided when using the terminology [Ful+20].

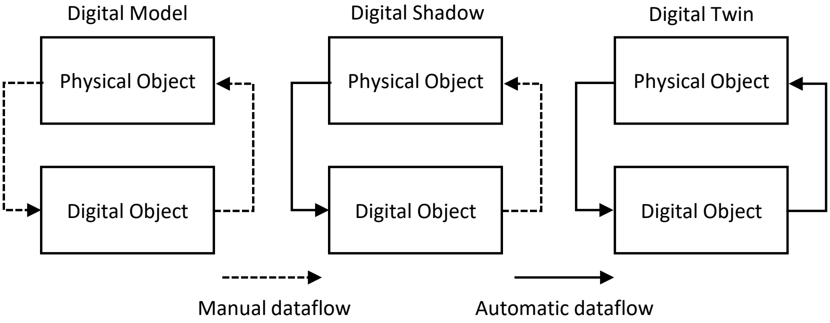


Figure 2.9.: The digital model, shadow, and twin, based on [Ful+20]. Note: In the illustration above, the digital object feeds back the physical object, which differs from the illustration in the literature and reflects the understanding provided in this thesis.

A DM is “described as a digital version of a pre-existing or planned physical object“ [Ful+20]. The DM is characterized by a manual data flow linking the physical and digital objects and does not consider any changes to the physical or digital objects once the learning process is terminated, see Figure 2.9. [Ful+20]

If an automatic dataflow is implemented that feeds the digital object, followed by continuous updates, the criterion of a DS is met. Changes in the physical object influence the digital object. [Ful+20]

A DT requires an automatic dataflow between the digital and physical object and both models to be fully integrated in both directions. Changes in the physical object automatically lead to changes in the digital object and vice versa. As a consequence, learning algorithms that enable such automation are considered in this thesis. [Ful+20]

2.3.5. Natural and artificial intelligence

The development of the hybrid model in Chapter 3 aims to incorporate intelligence into an aircraft system to enable intelligent maintenance. Therefore, the term intelligence is defined and distinguished between NI and AI, which is helpful in the context of hybrid models. This improves understanding of the proposed hybrid model and allows the results to be discussed and placed into the bigger picture of intelligent systems.

In the literature, different definitions are given, which deal with different points of view on the topic. This thesis considers the definitions of intelligence and its subcategories by Wang, Weng, Joshi, and Chollet. [Jos20; Cho17; Wen12; Wan09]

The following definitions are given by Wang:

Intelligence

> “Intelligence is a driving force or an ability to acquire and use knowledge and skills, or to inference in problem solving.“ [Wan09]

Natural Intelligence (NI)

> “Natural intelligence (NI) [. . .] implements intelligent mechanisms and behaviors by naturally grown biological and physiological organisms such as human brains [. . .].“ [Wan09]

Artificial Intelligence (AI)

> “Artificial intelligence (AI) [. . .] implements intelligent mechanisms and behaviors by cognitively-inspired artificial models and man-made systems such as intelligent systems, knowledge systems, decision-making systems, and distributed agent systems.“ [Wan09]

The consideration of NI and AI within this thesis is reduced to their subdiscipline of intelligent maintenance and the topic of technical systems. Therefore it is important to understand NI’s contribution to hybrid model building, which lies in the incorporation of physical laws and physically experienced expertise.

In contrast, the expectation of AI is contextualized by Joshi: “AI does not mean building an extraordinarily intelligent machine that can solve any problem in no time, but rather it means to build a machine that is capable to human-like behavior.“ Instead, “whenever we speak of AI, we mean machines that are capable of performing“ specific tasks such as “solving computer-based complex problems possibly involving large data in very short time [. . .]“. [Jos20]

Consequently, AI does not necessarily imply a human-like robot but describes a machine of far lower intelligence that is sufficient to take over a certain task to relieve the human. [Cho17]

2.3.6. Machine learning, deep learning, and the Artificial Neural Network

The intelligent maintenance of aircraft systems can involve AI, which requires the ability to learn from data. This ability relies on ML theory, an essential AI subcategory, see Figure 2.10. [Jos20]

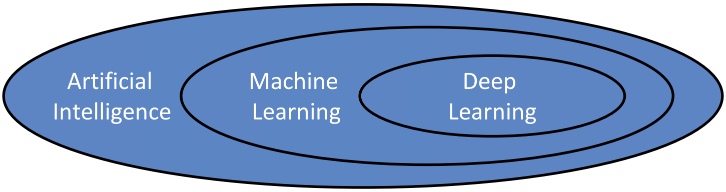


Figure 2.10.: Staging of Artificial Intelligence, Machine Learning, and Deep Learning, based on [Cho17]

ML algorithms are classified according to the characteristics of the input features. Supervised, unsupervised, and reinforcement learning describe the learning paradigm according to labeled, unlabeled, and feedback data. Alternatively, a distinction is made between static and dynamic learning. It builds on the temporal behavior of the system, which is constant or time-variant. From these classes, various ML algorithms have been developed to derive useful representations of data. [Cho17]

A specific subfield of ML is represented by DL, which "is a mathematical framework for learning representations from data" [Cho17] using ANNs typically.

The ANN is a purely data-driven approach, ideally suited to collaborate with physical models, see Section 2.4. Therefore, the ANN is further described in the following.

In the DL domain, various trends and applications of ANNs and further developments are found in the literature [GRD21]. Most common are the Feed-forward Neural Network (FNN), the Recurrent Neural Network (RNN), the Convolutional Neural Network (CNN), and the Long Short-Term Memory neural network (LSTM). The Generative Adversarial Network (GAN) can be classified as a hybrid model combining two data-driven models. [Jos20; Ras16; Roj96]

The hybrid model developed in Chapter 3 proposes using an ANN in general, such that one of the above variants can be implemented, depending on the use case. To keep the complexity concise and to investigate the interaction between the physical model and the ANN in principle, only the FNN is used in this thesis. The following contents and further readings are based on the literature provided by Joshi, Rashid, and Rojas. [Jos20; Ras16; Roj96]

The FNN consists of different layers with a variable number of nodes, whose linkage is inspired by the structure of human neurons, see Figure 2.11.

Input features u are fed into the input layer of the FNN. Processed in one or more hidden layers, the input data finally maps the output y represented by the output layer:

$$y = f_y(M_{OH} \times f_h(M_{HI} \times u)). \tag{2.16}$$

In Eq. 2.16, an FNN with one hidden layer is described, including the matrices M_{HI} and M_{OH}. These carry the FNN weights describing the connection between the layers. An activation function is applied within a node, which is f_h for the hidden layer nodes and f_y for the output layer nodes. Activation functions are defined according to the use case and data provided. Under a variety of possible activation functions, the linear

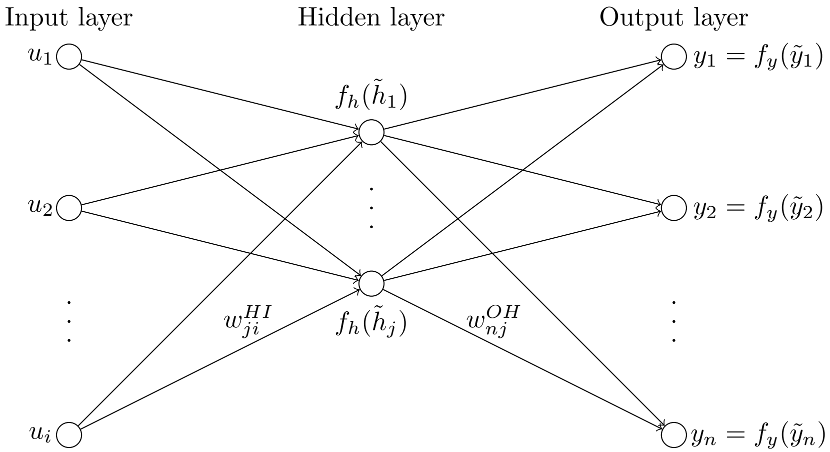


Figure 2.11.: Structure of the FNN with an input, one hidden, and an output layer, the input neurons u_i, hidden neurons h_j, and output neurons y_n connected through the weights w_{ji}^{HI} and w_{nj}^{OH} with the activation functions f_h and f_y.

and rectangular relu66 activation function scaled on $y \in [0, 1]$ are given in Eq. 2.17 and Eq. 2.18.

$$f_{lin}: \quad y = u, \tag{2.17}$$

$$f_{relu}: \quad y = min(max(0, u), 1). \tag{2.18}$$

The most common procedure to determine the FNN weights is the backpropagation algorithm, which relies on the LS method and the redistribution of the resulting error on the FNN weights. Alternatively, recursive filter methods, such as the EKF or UKF, have successfully been used for weight state estimation of an ANN in the literature. [Lim17; VV15; Oli12; WW12; Hay04]

This is important information to be used in developing the approach to hybrid model building in Chapter 3. It is further shown that recursive filters have not been used before in the context of hybrid models using ANNs, see Section 2.4.

2.4. Hybrid models using ANNs

The state of the art in hybrid modeling focuses on the ANN, which is a popular choice for the data-driven part [Far+22; Cha+22; GT21; Cha+19; Jia+19; LK14; And+90]. To address the lack of physical substance, a central question prevails:

> “How to integrate different layers of physics into neural networks?“ [Far+22]

Therefore, different approaches to hybrid model building have been developed and implemented. The most evident procedure is the physical-based reconfiguration of the input layer, called Feature Engineering (FE). Further, three distinct frameworks have emerged to enhance neural networks with physics, which have been reviewed by Faroughi et al. in [Far+22] and are briefly introduced in the following:

- the Physical-based Feature Engineering (FE) [BKK23; Bau19; BK19; Hea16; BCV13; CNL11], Subsection 2.4.1.
- the Physical Guided Neural Network (PGNN) [Mur+20; Jia+19; Daw+17; And+90], Subsection 2.4.2.
- the Physical Informed Neural Network (PINN) [Law+22; VS21; Wol+21; DV20; DV19; NV19; RPK17], Subsection 2.4.3.
- the Physical Encoded Neural Network (PENN) [RSL21; Inn+19], Subsection 2.4.4.

2.4.1. Physical-based Feature Engineering

> “The performance of machine learning methods is heavily dependent on the choice of data representation (or features) on which they are applied.“ [BCV13]

Therefore, input data is often transformed and preprocessed into subsets. FE can be applied for unsupervised and supervised learning and can significantly influence the model´s performance [CNL11]. In particular, the ANN profits from FE [Hea16].

Using FE enables system knowledge by physical laws and dependencies to be incorporated. [BKK23; Bau19; BK19; BCV13]. With a physically meaningful interpretation and ranking of the features, FE is understood as a first step of hybrid modeling.

In addition, FE provides standardization of data to increase convergence speed, which can be realized by referring the data to their standard deviation, mean, or maximum values [Bau19].

2.4.2. Physical Guided Neural Network

The core component of the PGNN consists of an ANN of any type, characterized by two major steps. First, prior estimates of a physical model are added to the input layer of the PGNN. Second, the objective function is extended by a physics-based penalty term, $\hat{e}_{phy}$. The design of the PGNN is shown in Figure 2.12. [Far+22]

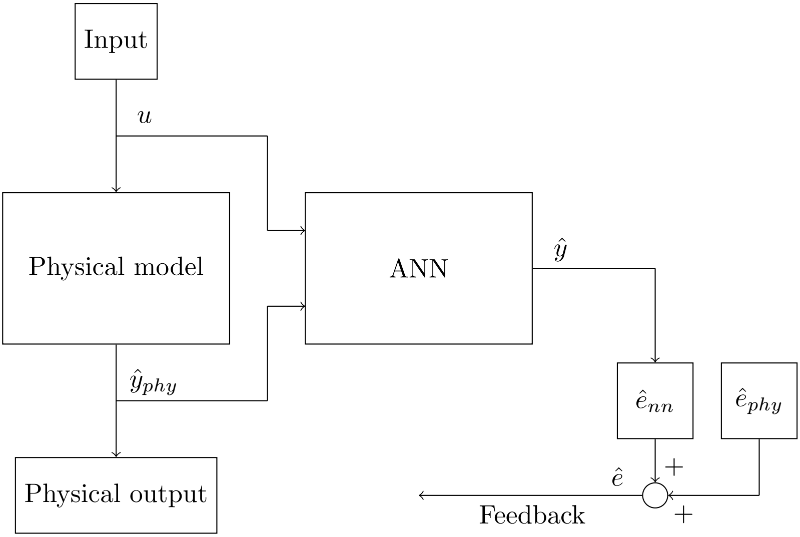


Figure 2.12.: Design of the PGNN using physical-based features $\hat{y}_{phy}$ in the input layer of an ANN and adding a physical-based penalty term $\hat{e}_{phy}$ to the output error of the ANN $\hat{e}_{nn}$, based on [Daw+17]

Following Faroughi et al., the first PGNN has been implemented by Andersen et al. in 1990 investigating solid mechanics through modeling an arc welding process using an ANN [And+90]. Since then, the PGNN has been elaborated for different applications in scientific computing, for which an overview can be found in [Far+22]. A simple example exists for lake temperature modeling with an FNN in [Daw+17] or an RNN in [Jia+19]. In this case, a penalty term is added if water density decreases with increased water depth. In aviation, a more challenging application consists of the prediction of drag force by Muralidhar et al. [Mur+20].

2.4.3. Physical Informed Neural Network

PINNs concentrate on physical phenomena, which differential equations can describe. Again, the type of the ANN can be chosen arbitrarily, and outputs $\hat{y}_{nn}$ are then differentiated from the inputs u. The differentiations form the basis for further terms of the total loss function, resulting in the error $\hat{e}$. Thus, "the underlying physics is incorporated outside the neural network architecture to constrain the model while training, thereby ensuring outputs follow known physical laws" [Far+22], see Figure 2.13.

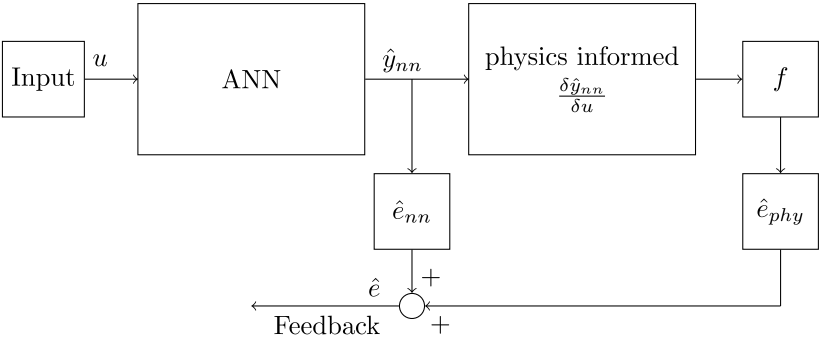


Figure 2.13.: Design of the PINN using a physics-informed approach with a selection of derivatives of the ANN output $\hat{y}_{nn}$ according to the input u to add a physical-based loss function to the output error $\hat{e}_{nn}$, based on [Wol+21]

With the first publication of a PINN in 2017 by Raissi et al. about physics-informed deep learning with nonlinear partial differential equations, the PINN framework found a young research trend [RPK17]. The first applications are described for a corrosion-fatigue prognosis for an aircraft by Dourado and Viana [DV20; DV19] and a fleet prognosis by Nascimento and Viana [NV19]. Since then, publications in PINN research have been increasing significantly [Law+22]. An overview of the historical development and further publications are given in [VS21] and [Law+22].

2.4.4. Physical Encoded Neural Network

PENN is the latest development in incorporating physics into ANNs. The nomenclature is “forcibly encoding the known physics knowledge“ [Far+22]. There are different algorithms and their applications given in the literature. Three of them, the Physics-encoded Recurrent Convolutional Neural Network (PeRCNN), the Fourier Neural Operator (FNO), and the Neural Ordinary Differential Equations (NeuralODE), are discussed by Faroughi et al. in [Far+22]. All of them “strive to hard-encode physics (i.e., prior knowledge) into the core architecture of the neural networks“ [Far+22].

While the PeRCNN introduces the π-block, which allows for an elementwise product and recursion inside the structure, physical knowledge can be integrated through a convolutional layer [RSL21]. Innes et al. describe their algorithmic solution as NeuralODE. They propose a serial hybrid model structure where the output of an ANN is used as input to a physical model, and the output error relies on the physical output exclusively [Inn+19]. Abstracted from the illustration in the literature, the concept is shown in Figure 2.14.

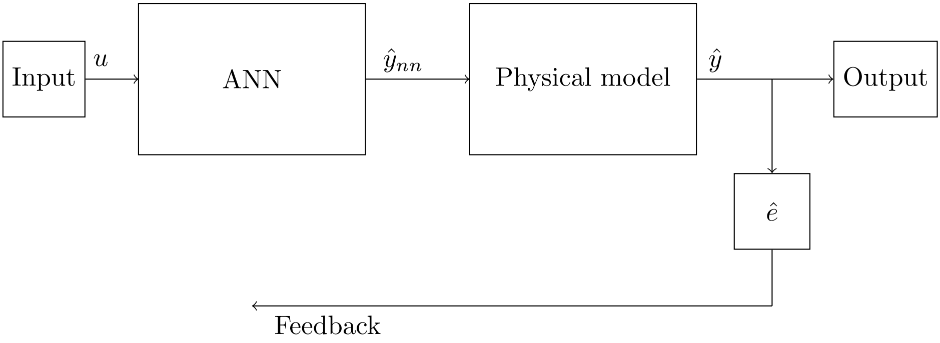


Figure 2.14.: Design of the PENN using the output of the ANN $\hat{y}_{nn}$ as input to the physical model. The output of the physical model equals the final estimation $\hat{y}$ and is considered for the feedback exclusively, based on[Inn+19]

Innes et al. apply the PENN with an FNN to estimate the necessary settings of a trebuchet (the mass of the counterweight and the angle of release) depending on the wind velocity and the target location. The predicted distance is compared to the commanded target location. The FNN weights are adjusted using backpropagation, which runs through both the physical model and the ANN. [Inn+19]

Due to the incorporation of the NeuralODE, the convergence speed is increased, and extrapolation capabilities are observed [Far+22]. Therefore, the PENN addresses the generalization, interpretability, and computational efficiency of the PGNN and PINN [Far+22; Inn+19].

However, two major flaws remain:

1. The parameters of the encoded physical model are assumed to be constant, which assumes the described system to be physically not changing in its life cycle. This is not the case when degradation and recovery of the system occur.
2. The considered backpropagation algorithm for weight adjustment concerns batch-wise optimization. Therefore, adaptability is limited.

2.5. Research gap

As shown above, hybrid models are a couple of physical and data-driven models, preferably using ANNs. The prevailing approaches to hybrid model building deal exclusively with the increasing incorporation of physics into ANNs. In conclusion, a data-driven model is currently preferred to be used as a basis and extended by physics.

However, aircraft systems demand a high safety standard and trust in new technologies before taking effect on aircraft systems and operations. Therefore, the reliability of an aircraft system and the interpretability and comprehensibility of a DM, DS, or DT have a significant value to the operator. It is, therefore, not surprising that monitoring systems still build on physical-based knowledge in the aviation industry, as discussed in Section 2.1.

Instead, it is surprising that our physical knowledge about aircraft systems and their components is considered an add-on rather than a basis to start with in hybrid model building. Second, the physical-based aircraft model being part of the hybrid model is considered fixed when using the approaches to hybrid model building discussed above. This is not the case in the appearance of system degradation and recovery.

Consequently, the model's parameters must be considered time-variant and part of the estimation problem rather than assumed to be constants. In the context of AI, such an estimation must be highly adaptive and affect both model parts simultaneously.

In this field, the research gap is identified. Research is missing in hybrid model building concerning:

1. the physical knowledge and model as a basis,
2. the physical model's parameters as time-variant and subject to the data-driven part,
3. and the learning method to be highly adaptive and effective for both model parts simultaneously.

Further, the integrity of data-driven models is questioned, as well as the utilization of AI to be used to enhance physical knowledge. The development and testing of such a hybrid model and its discussion in the context of AI and control engineering is new to the state of the art.

2.6. Summary

Aircraft systems experience degradation and recovery through maintenance measures in their life cycle. Regarding system safety and operational costs, different types of maintenance exist. When a recovery is triggered, it is crucial to operational efficiency and, thus, maintenance costs. Consequently, intelligent maintenance is desirable, searching for the item's *minimum state* to trigger recovery. Such a task requires intelligence to be incorporated into the system.

Intelligence requires a system model to adapt through learning. The data-driven approach to model building prevails in the context of AI but suffers from a lack of physical substance and trust. On the other hand, conservative aircraft system identification methods have proven reliable and provide insight into a physical model. The idea of merging both sides leads to a hybrid model.

Hybrid model building has emerged as a valuable research branch over the past decade. Three models can be combined, resulting in five possible hybrid model classes. One of them, coupling physical and data-driven models, has emerged as the most promising hybrid model class. The implementation is usually serial or parallel, with both parts of the hybrid model connected in a cascaded structure. This implies physical system knowledge as input to the data-driven part or the assignment of a physically meaningful model output that is merged with the output of a data-driven model.

Since ANNs are usually considered the data-driven part, three major frameworks are discussed: the physical guided (PGNN), informed (PINN), and encoded (PENN) ANN. Implementations in the literature have shown that model accuracy, computational efficiency, and physical consistency can be increased, leading to greater applicability and acceptance. The developments of the past decade show a trend. Physical knowledge is needed to realize physically consistent, comprehensive, and interpretable learning strategies. Thus, physical knowledge is increasingly important for hybrid model building.

However, the current hybrid model structures struggle with interpretability and initialization problems, as is the case for the structure of the PGNN and PINN. Their model structures do not allow insight into the working principles as the open structure of the ANN dominates the structural design. Physical knowledge is incorporated to guide or inform the ANN by extending the input layer of the ANN with physical-based features or the objective function with additional penalty terms. The PENN addresses some of the shortcomings of the PGNN and PINN. It reverses the serial structure of the PGNN and extends the idea of the PINN by encoding a physical model fed by an ANN. Thus, the ANN outputs are physically interpretable and promise a generalizable behavior.

Consequently, current hybrid models rely on data-driven models to be extended by physics. The parameters of the physical models used here are assumed to be constant and are not the target of adaption. But when the degradation and recovery of an aircraft system is considered, the system model parameters are assumed time-variant rather than constant.

Therefore, research is needed to develop a hybrid model, which extends the physical model using a data-driven model and enables the physical parameters to be part of the learning process. Learning is required to be adaptive and affect both model parts simultaneously. The learning strategies currently used for hybrid model building do not fulfill this.

3. Development of a new approach to hybrid model building

The previous chapter has shown that hybrid models have promising characteristics to represent complex systems physically consistent, and accurate. However, it has also been shown that current approaches to hybrid model building do not favor physical knowledge as a basis to be extended. Nor do they implement methods enabling step-wise and recursive learning. These properties are key to mapping reliable, comprehensive and interpretable aircraft systems that can adapt through learning and enable intelligent maintenance.

Therefore, a new approach to hybrid model building is developed and organized in the following Section.

3.1. Development methodology

The procedure of development orientates on the ISO/IEC/IEEE norm 15288:2015 for “Systems and software engineering – System life cycle processes“ and on the VDI/VDE norm 2206:2021-11 for the “Development of mechatronic and cyber-physical systems“ [VDI21; ISO15]. The main components are the requirements, the model design consisting of the model structure and the identification method, and the implementation, see Figure 3.1. The implementation is separated into preliminary studies to gain first feedback on the model design and the evaluation following in Chapter 4 and Chapter 5.

The development of the hybrid model starts with the introduction of the approach and the definition of requirements in Section 3.2. The requirements describe “the overall objectives [. . .], its target environment, [. . .] constraints, assumptions and non-functional requirements“ according to ISO/IEC/IEEE 29148:2018 [ISO18].

The model structure is the topic of Section 3.3. First, the conceptual design is presented as the principle solution according to Pahl et al. [Pah+07]. Second, the hybrid model

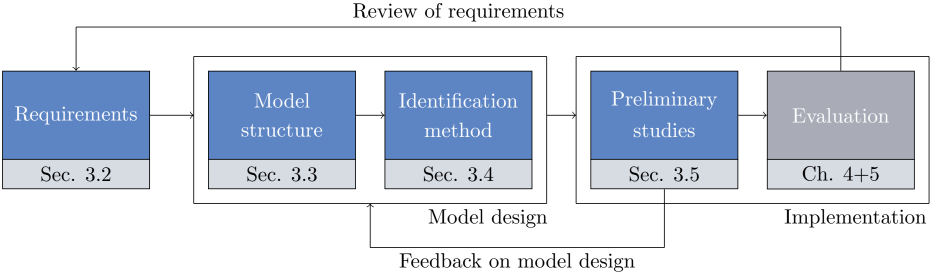


Figure 3.1.: Methodical procedure for the development of the approach to hybrid model building, based on [ISO15].

structure is derived considering a physical model, an ANN, various inputs, and data streams used for connectivity.

To finally identify the system using the derived hybrid model structure, an appropriate estimation method is chosen in Section 3.4.

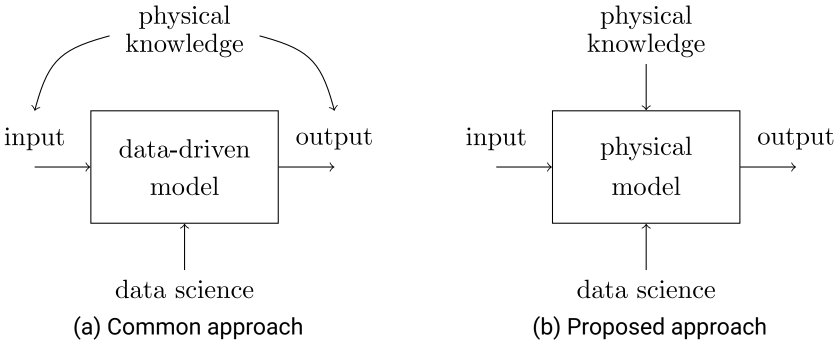
The implementation is first performed in preliminary studies discussed in Section 3.5 based on the publications [EK22] and [EHH22]. They provide relevant feedback on the model design, which is to be considered in the evaluation. In the evaluation, the complexity of the considered system is increased. A Unmanned Aircraft System (UAS) is used as a flight test platform providing dynamic flight test and flight simulation data.

The development is summarized in Section 3.6.

3.2. Introduction of the approach and definition of requirements

Physical knowledge is understood as the foundation of building physically consistent models. Thus, the physical model forms the core structure of the hybrid model, which is to be extended by data science using data-driven models in this thesis. Figure 3.2 illustrates the two approaches leading to a hybrid model based on a data-driven or a physical model.

The common approach assumes a data-driven basis. It intends to incorporate physics into the input features and into the output of the model to affect the learning algorithm,



(a) Common approach

(b) Proposed approach

Figure 3.2.: The common (a) and proposed approach (b) to hybrid model building based on a data-driven or physical model.

see Chapter 2. In contrast, the proposed approach utilizes the physical model as a basis into which data-driven structures are to be integrated. This is innovative in the literature, where methods and procedures are currently developed to incorporate physics into data-driven designs, as shown in Section 2.4.

The development requirements are defined according to [ISO18]. They provide the basic framework for the development process by specifying primary goals. First, Subsection 3.2.1 describes needs related to the hybrid model design. Second, conditions are placed in the context of aircraft systems and extended as needed in Subsection 3.2.2. Finally, the needs are transformed into requirements in Subsection 3.2.3. For clarity, the delimitation of the development is discussed in Subsection 3.2.4.

3.2.1. Needs related to the hybrid model design

Four significant problems with the hybrid model yield the needs for the development of the proposed hybrid model that must be capable of:

- initialization,
- generalization,
- interpretation,
- and adaption.

In detail, the initialization problem of hybrid models, especially of ANNs incorporating physics, is still unsolved. Therefore, an alternative concept for a hybrid model must consider the endowment of the learning algorithm with suitable initial conditions. In this case, the physical model is understood to be helpful when used as the initial model structure.

Even though recent research advances through the PENN contrast the PINN and PGNN in addressing the issues of generalization and interpretability, potentials remain unused. Generalization shall be maintained both in the definition of the model structure and in the estimation method, whose design must allow extrapolation and prevent overfitting. Generalization refers to a system and its operational history individually. Therefore, the system states of the individual systems are to be extrapolated and thus predictable.

The interpretation of models enables comprehensive results and access to the insides of the system behavior. It allows a substantial gain in knowledge and confidence in the hybrid model. Moreover, the ability to interpret the inner states of the system is assumed to be crucial to performance indicators that can be used for RUL prognosis and fault detection and isolation. Therefore, it is surprising that interpretability has been treated even poorly in current hybrid models. Consequently, the new hybrid model must be interpretable and provide access to meaningful inner system states.

The last instance to be considered is the adaptability of the hybrid model, more specifically, the learning algorithm. The ability to adapt through learning is essential for employing an intelligent system (cf. Subsection 2.1.3) and for applying a DT. According to its definition in Subsection 2.3.4, information must be processed continuously. Hence, the learning algorithm of the hybrid model must have adaptive properties.

3.2.2. Needs related to the aircraft system

The four stated needs gain additional weight in the context of aircraft systems. The well-known physical dependencies of the aircraft provide a solid basis to develop and implement a physical model. The model includes flight dynamics, aerodynamics, and aircraft propulsion. Such a model can be considered a core model structure for initialization. Furthermore, the initial conditions of the hybrid model can be set according to the physical model and physical-based assumptions about the initial states. Thus, knowledge of aircraft physics offers an adequate basis to be used for initialization.

The need for a generalizable model is manifested through various of operating conditions, mission requirements, and environmental disturbances that affect the aircraft.

It is unlikely to cover all of the different operational needs within a database nor to have access[1] to it. The validity of data-driven models and most hybrid models, such as the PGNN and the PINN, quickly come to limitations due to the database. Therefore, generalization is fundamental when extrapolation and state prediction is required.

The interpretability of an aircraft system is considered one of the most desirable factors. It enables comprehensive results and access to characteristic parameters. These parameters describe aerodynamic coefficients, thrust, efficiency, and other system-specific states, allowing the behavior of the aircraft system to be analyzed. In addition, modifications can be assessed, and benchmarks can be used from the literature. Finally, the aircraft system model can be evaluated in terms of physical consistency and robustness, which plays a vital role in confidence and certification needs.

In the last step, an essential characteristic of the aircraft must be considered: the aircraft system changes during operation. The degradation of the aerodynamic surfaces and the technical subsystems constantly infects the health of the aircraft system over time. Maintenance measures are carried out to catch or at least weaken the degradation mode, see Section 2.1. Retrofits are implied to enhance the aged aircraft regarding safety, operational efficiency, and passenger comfort, see Subsection 2.1.2. The maintenance measures and retrofits both modify the aircraft's technical condition, and modifications have a recurring character in the case of maintenance. Therefore, adapting to degradation effects and maintenance measures must be enabled, which suggests considering time-variant parameters and additional maintenance data.

3.2.3. Transformation of needs into requirements

The research conducted in this thesis is at an early stage of method development. Since such a basic development is placed in the first two Technology Readiness Levels (TRLs) according to [Nat23], fundamentals are explored.

Therefore, the following requirements are formulated as functional requirements exclusively. They tackle the above-mentioned needs, addressed substantially in the body of this thesis. However, the requirements provide a recent framework to study the feasibility, advantages, and disadvantages of the new approach to hybrid model building,

[1] In the aviation industry, owners, operators, maintenance contractors, manufacturers, and researchers deal with aircraft systems. However, sharing data is a barrier. It requires contractual effort or is prohibited. This leads to a burden on research, since access is limited to both the characteristics of the considered system and its operational data.

as introduced above. Performance and system requirements are not specified. They can potentially be included in future developments once feasibility is demonstrated.

Table 3.1.: Definition, classification, and description of requirements (A) and (B) imposed on the new hybrid model

Group	Sign	Class	Short	Formulation
(A) Initialization	(A1)	(MR)	Physical model	**Requirement**: The hybrid model shall be initialized using a physical model.
				Indicator: The output of the hybrid model initially equals the output of the physical model ($\hat{y}_{hyb_{ini}} = \hat{y}_{phy_{ini}}$).
	(A2)	(MR)	Physical conditions	**Requirement**: The hybrid model shall be initialized based on physical conditions.
				Indicator: The initial system states are based on physical knowledge and expertise ($\hat{x}_{hyb_{ini}} = \hat{x}_{phy_{ini}}$).
(B) Generalization	(B1)	(MR)	Extra-polation	**Requirement**: The hybrid model shall perform the estimation task based on existing trends.
				Indicator: The considered system states are predictable ($\hat{x}^* = f(\hat{x})$).
	(B2)	(MR)	Over-fitting	**Requirement**: The implemented learning algorithm shall prevent overfitting.
				Indicator: Measurement noise is considered ($R_{yy} \neq \underline{0}$).
	(B3)	(MR)	Opera-tional data	**Requirement**: Only operational data shall be used for learning.
				Indicator: The database available for learning is limited. It involves standard operation of the system in service ($u = u_{ops}$, $m = m_{ops}$).

Nevertheless, the requirements are formulated concerning [ISO18]. A description is additionally given. Indicators are provided for later approval. The stated indicators reflect the nomenclature of the recursive filter chosen for learning. In addition, the calculation time needed per time step k is defined t_{calc_k}, used in (D2). Further definitions are given subsequently.

Furthermore, the requirements are classified into Mandatory Requirements (MRs) and Desirable Requirements (DRs) to highlight their priority following [Kir20].

Table 3.2.: Definition, classification, and description of requirements (C) and (D) imposed on the new hybrid model

Group	Sign	Class	Short	Formulation
(C) Interpretation	(C1)	(MR)	Inner states	**Requirement**: The hybrid model shall provide insight into its inner physical principles of functionality.
				Indicator: The (hidden) system states are interpretable/ observable (cf. Subsection 2.2.3 and Section 4.5)
	(C2)	(DR)	Benchmarks	**Requirement**: The hybrid model estimates shall correlate to specified benchmarks.
				Indicator: Aerodynamic aircraft system modification is to be assessed ($\Delta C_{D_0} \in [1, 6]\%$, $\Delta C_{L_0} \in [3, 12]\%$)
(D) Adaption	(D1)	(MR)	Adaption through learning	**Requirement**: The hybrid model shall continuously adapt through learning.
				Indicator: Stepwise learning is enabled; cf. Subsection 2.2.2 ($V_k \times \hat{e}_k \neq \underline{0}$)
	(D2)	(DR)	Automated information flow	**Requirement**: The hybrid model shall process information automatically in operation.
				Indicator: In the context of DTs, the automated learning algorithm is real-time capable ($t_{calc_k} < \frac{1}{f_{prediction}}$).
	(D3)	(MR)	Additional data	**Requirement**: The hybrid model shall consider additional data beyond the scope of the physical model.
				Indicator: A database is provided to be input to the data-driven part ($u_{nn} \neq \underline{0}$)

In conclusion, four major groups are defined in Table 3.1 and Table 3.2 with ten subcategories to be addressed with the hybrid model.

3.2.4. Delimitations

The hybrid model is designed to describe any technical system individually. This means that the derived hybrid model is not suitable for a general description of the type of a system but for a specific[2] system, including its operational history.

For the evaluation, the hybrid model considers a specific aircraft system. This results in a description of an aircraft in operation, influenced by characteristic aging processes, damage, maintenance measures, retrofits, and operating history. The operating history includes maintenance, operational, and flight data, which are acquired according to the functional usage and address degradation and recovery.

The scope of the acquired flight data is limited; therefore, maintenance measures are not available in a long-term observation. To enrich the considered database, abstract manipulations are performed. These represent degradation and recovery in the following, affecting the time-variant parameters. Thus, modeling of the individual aircraft is done using the available database. Operational flight data is addressed in requirement (B3), and additional data, such as maintenance data, is addressed in requirement (D3).

3.3. Design of the concept and model structure

According to the derived requirements, the importance of the physical model is discussed for initialization. It, therefore, forms the basis for the conceptual design. Secondly, the need for interpretability and visualization of the inner states are essential features that shape the conceptual design through the need for a physical model and parameters. However, as stated in Subsection 2.3.1, building a physical model is elaborate and can lack accurate state estimation in the case of insufficient model complexity. Therefore, the use of a data-driven model to compensate for such insufficiencies and further employ various databases is suggested. Consequently, the solution space is limited to a hybrid model consisting of a physical and a data-driven model.

[2] Using the aircraft example, a technical aircraft system carries a specific tail-sign, while the type of a aircraft system describes the aircraft family.

The physical model is intended to incorporate a data-driven model as an extension, as first introduced in Section 3.2.

3.3.1. Concept of integrating an ANN into a physical model

The ANN is a valuable coupling partner of the physical model in building hybrid models, as introduced in Chapter 2. As a result, the ANN is chosen as the data-driven part for the concept creation of the hybrid model in this thesis. The data-driven feature enables adaptability concerning insufficiencies of the physical model and the inclusion of additional data sources with little effort.

Even though the concept is intended to allow the implementation of any ANN, the research within this thesis focuses on an FNN with an input, one hidden, and an output layer to reduce model complexity. Despite its simplicity, the FNN can represent any functional dependency (cf. Subsection 2.3.6) and is therefore considered suitable for basic investigations. As a consequence, the following question is examined:

How do we integrate different layers of neural networks into physics?[3]

With the physical model as the core model structure and the requirement to maintain access and visibility of the physical parameters, a parameter scheduling model (cf. Subsection 2.3.3) is appropriate. The scheduling model structure considers varying conditions of the prior model, represented by the physical model. It includes the inputs u and the parameters p, which result from the parameter scheduling model represented by the ANN. This is accompanied by the assumption that physical parameters are not constant but change during operation due to degradation and technical modification of the system. Consequently, physical parameters are considered time-variant. Unlike previous parameter scheduling models, the ANN is chosen, which can involve additional data in predicting the physical parameters, see Figure 3.3.

In addition, uncertainties of the physical model, such as unknown physical effects or the intentional implementation of a less complex physical model, can potentially be accounted for by the ANN. In this way, systematic deviations can be addressed in the physical parameters to increase the value of the hybrid model.

Choosing a parameter scheduling model structure has advantages over serial and parallel hybrid model concepts prevailing in the literature, see Chapter 2. It enables

[3] Instead of: “How to integrate different layers of physics into neural networks?“ (cf. Section 2.4)

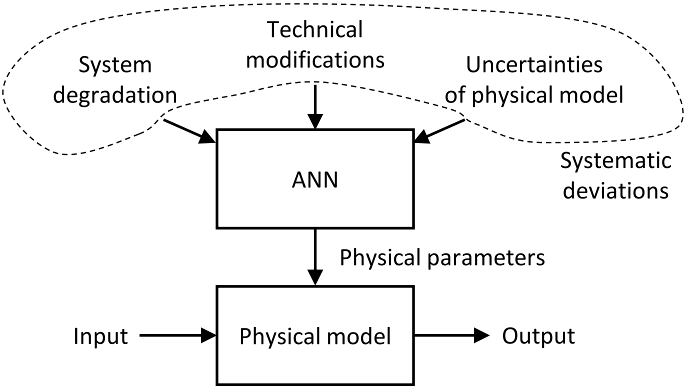


Figure 3.3.: Conceptual design of the hybrid model using an ANN to predict the physical parameters of the physical model.

deep coupling of the physical model and the ANN using the physical parameters as an interface.

The number of nodes in the different layers sets the size of the ANN. Choosing the number of nodes is part of the initialization process. It depends on the system complexity, the physical model used, and the selected physical parameters. Further, the activation functions within the nodes have to be chosen according to the use case in Chapter 5.

3.3.2. Development of the hybrid model structure

The hybrid model structure orientates on establishing a parameter scheduling model as introduced in Subsection 2.3.3 and on the concept designed in the previous Section, which results in the representation given in Figure 3.4.

The available data is divided into the input u, scheduling input u_{nn}, and measurement data m. These are the starting points of the data flow inside the model structure. In the considered case of an aircraft system, input data are commanded values of the flight mission, the flight phase, and aircraft-specific information such as the aircraft weight, which can change according to the flight mission. This data is necessarily input to the

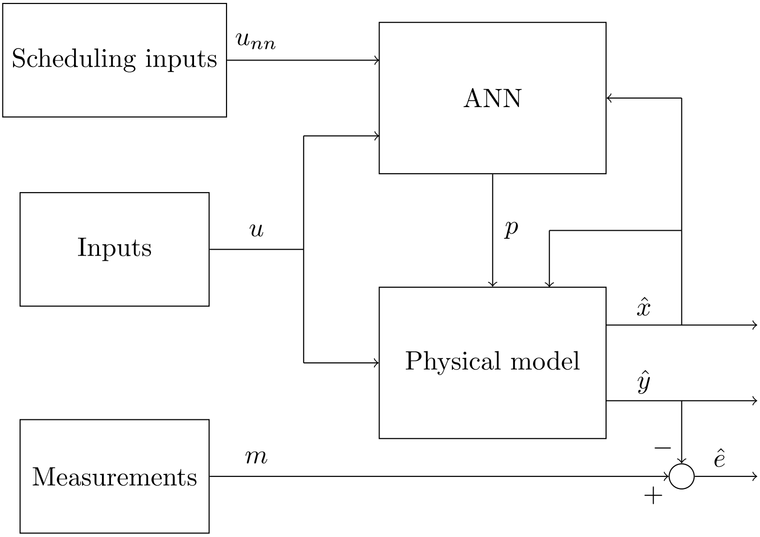


Figure 3.4.: The proposed hybrid model structure includes scheduling inputs u_{nn}, inputs u, and measurements m to be processed by an ANN and a physical model. The residual $\hat{e}$ is based on the estimated output $\hat{y}$ and m. The ANN uses estimated states $\hat{x}$ and the inputs u_{nn} and u to predict the physical parameters p.

physical model and can also be used as input to the ANN for the parameter scheduling part.

Additional data that can be represented as time series data and affect the hybrid model are defined as scheduling input u_{nn} for the parameter scheduling process of the ANN. The data flow of u_{nn} is stated separately within the visualization of the model structure. It can keep information about the history of flight cycles, the elapsed time since the last maintenance measures, and the total operating time. The scheduling input data can be considered maintenance data for the aircraft use case and is further described in Subsection 4.4.4.

Measurement data contain the sensor measurements, which support state estimation significantly but suffer characteristic sensor noise. Notice that the sensor measurements do not serve as input to the physical model or the ANN but exclusively to calculate the residual. In this way, the influence of the sensor noise is prevented from affecting the hidden states, including the parameter states [EHH22; BK94].

The physical model is located in the center of the hybrid model structure. Equipped with the inputs u and the parameters p, the physical model estimates the output $\hat{y}$ and the hidden states $\hat{x}$ using physical laws and dependencies. The states $\hat{x}$ are fed back into the ANN and physical model. They are provided for analysis, the estimated output $\hat{y}$, and the residual $\hat{e}$.

Thus, the ANN receives the inputs u, the scheduling inputs u_{nn}, and the estimated states $\hat{x}$. With the system's increasing complexity, the size of the vectors u, u_{nn}, and $\hat{x}$ can increase significantly. Therefore, a choice of input features is recommended to reduce the size of the input layer to a physically reasonable subset. This improves computational efficiency as described in Subsection 2.4.1 and further discussed in Chapter 5. Including the feature selection, the ANN acts as a parameter scheduling model and determines the physical parameters p used in the physical model.

The presented model structure prepares the implementation of a recursive filter method for system identification and state estimation in the following Section. It attaches to the representation of $\hat{x}$, $\hat{y}$, and $\hat{e}$.

3.4. System identification and state estimation

The established structure of the proposed hybrid model defines the interaction of the physical model, the ANN, and the considered data. In the next step, an adequate identification procedure is required to allow a sufficient estimation of the system states, physical parameters, and ANN weights. The residual vector $\hat{e}$ shall provide the information to adjust the weights of the ANN. Equipped with the updated weights, the ANN is expected to predict the physical parameters more accurately.

Due to the requirement for adaption (D) and the claim for a deep coupling inside the hybrid model, a recursive filter algorithm with a joint estimation approach is chosen, cf. Subsection 2.2.2. The identification of ANN weight states (cf. Subsection 2.3.6), and an aircraft system consisting of dynamical and parameter states (cf. Subsection 2.2.1) was successfully performed in the literature using recursive filter methods.

A recursive filter method is introduced for the first time to build a hybrid model based on a parameter scheduling structure. The mUKF extended by State Constraints (SC) realizes the recursive filter algorithm. The filter variant is defined as the State Constraint modified Unscented Kalman Filter (SCmUKF), derived and discussed below. Joint estimation allows deep and simultaneous coupling of the physical model and the ANN, as mentioned in Subsection 3.4.2.

3.4.1. Choosing the UKF and filter modifications

The UKF, filter modifications, and SC are chosen according to the requirements defined in Section 3.2. The choice is discussed below.

The recursive filter method

Recursions allow the model to adapt through learning (D1). An example is found in the literature by Nascimento and Viana in [NV19], who consider cumulative degradation of an aircraft fleet with using an RNN in the context of a PINN, cf. Subsection 2.4.3. In the case of recursive filters, information can be obtained from new data samples. With every new data sample, adaption and thus an online learning process can be enabled, cf. Subsection 2.2.2.

Integrating new data samples plays a vital role in supplying an automatic data flow (D2) and thus the definition of a DT, see Subsection 2.3.4. Note the online learning process is enabled within the description of the system identification method but roughly examined in terms of real-time capability.

The Unscented Kalman Filter (UKF)

Handling the measurement noise R_{yy} and the system noise Q_{xx} of the Kalman Filter (KF) allows further integration of knowledge beyond the initialization of the hybrid model using a physical model and its initial conditions (A1 and A2). These are distinctive values to tune the filtering process, incorporate system characteristics, and avoid overfitting (B2).

To encounter the expected high nonlinearities of the hybrid model, the UKF is chosen [EK22]. High nonlinearities occur due to the extended state vector $\vec{x}$, which carries states of the dynamical system model x_d, the parameter states x_p, and the ANN weight states x_{nn}. The extension of the state vector leads to the joint estimation mode of the UKF discussed in Subsection 3.4.2. Unlike the EKF, the UKF has shown to be stable[4] and thus applicable to the state estimation problem of the proposed hybrid model.

[4] Testing has shown that linearization of the coupled physical model and ANN can cause filter instability. The linearization solved by numerical methods is mandatory for the use of the EKF in the context of the proposed hybrid model.

This agrees with the literature, which emphasizes the UKF in the case of increased nonlinearities, cf. Subsection 2.2.2.

Moreover, the UKF is formulated as an RMV algorithm. Beyer and Klingauf have shown that the RMV is superior to the LS algorithm because influences of the measurement noise can be isolated from the system, and bias-free estimation is enabled [BK94].

In this thesis, the LS method, typically employed ML application training (see Subsection 2.3.6), is replaced by the RMV method. Any simplification of the KF formulation leading back to an LS method, as proposed by Gäb for the identification of a UAS [Gäb12], is refrained from. As shown in [EHH22], measurements used as inputs to the ANN within the hybrid model insert measurement noise to the predicted physical parameters and thus the system states. Therefore, the UKF formulation strictly follows the RMV method.

Filter modifications

Based on [EK22; SGH19; Qi+18], a modification of the UKF is introduced to increase robustness and stability and to decrease mathematical complexity and computational effort.

The derived modification simplifies the consideration of the mean state value. This also reduces the adjusting hyperparameters of the UKF by β_{UKF}. The hyperparameters α_{UKF} and κ_{UKF} remain to adjust the spread of the sigma points. Sigma points are now used for covariance calculation separately, while the mean value of the estimated output and the estimated state are obtained directly from the measurement and state equations. The separated analysis of the mean value and the covariance benefits the challenging state estimation task of the hybrid model in increasing filter stability. The mathematical formulation is given and discussed in [EK22].

State Constraints (SC)

The use of constraints limits the solution space of the states and can, therefore, increase physical consistency and filter stability [EHH22]. For more complex systems such as an aircraft engine circuit or a holistic aircraft system considering an increased number of system states, SC can enrich and even enable filtering by binding the states into a physically meaningful set $(x \in \mathbb{R} | x_{lb} < x < x_{ub})$. Setting the lower x_{lb} and upper boundary of states x_{ub} represents another opportunity to incorporate physical knowledge and ensure applicability.

A broad variety of methods to implement SC exists, which is collected by Simon et al. in [Sim10; Sim06b; SS05; SC02], discussed by Julier et al. for nonlinear equality constraints in [JL07] and by Teixeira et al. for interval-constrained nonlinear systems in [Tei+08].

In the context of the proposed hybrid model, an implementation of SC based on the method introduced by Kandepu et al. in [KIF08] is proceeded in [EHH22]. However, with the increasing complexity of the considered system methods, implementing State Constraints (SC) has to be chosen carefully.

In detail, shifting the sigma points back into its limits in case limits are violated, as proposed by Kandepu et al. can lead to instability. A more elegant way to keep a state in a defined set is using a “perfect measurement“ [Sim10]. In case of a violation, a pseudo-measurement is set into service with $x = x_{lb}$ or $x = x_{ub}$ and $R = 0$. Thus, SC is implemented due to the adjustment of the filter gain. [Sim10]

Differing from the literature, the idea of the perfect measurement is extended for this thesis. To facilitate the implementation, pseudo-measurements are implemented for each parameter state x_p. These are implemented by setting a mean value $\bar{x}$ and a standard deviation $\sqrt{R(x)}$ spanning a range of values, equivalent to interval constraints, see Eq. 3.1.

$$x \in [\bar{x} - \sqrt{R(x)} \quad \bar{x} + \sqrt{R(x)}] \tag{3.1}$$

The implementation of SC using pseudo-measurements, as stated in Eq. 3.1, has a significant advantage:

> The SC is incorporated using the update step of the state covariance matrix $P_{xx\,k|k}$, which ensures the positive definiteness of $P_{xx\,k|k}$ to be untouched and thus provides filter stability.

Parallels are seen in the truncated UKF, whose advantages are discussed in [Tei+08].

Note that a state value can leave the defined interval by the state prediction step of the SCmUKF. Violation of the limits is corrected when a state update is accomplished using the residuals. Consequently, the limitations must be carefully chosen to allow stability close to and outside the constrained intervals.

3.4.2. Joint estimation of the physical model and ANN

The SCmUKF has been derived for the state estimation of the hybrid model. As introduced in the previous section, the extended state vector holds three different types of states: x_d, x_p, and x_{nn}. The estimation is considered with the SCmUKF regarding nonlinearities and further considered against the joint and dual estimation background in the following, cf. Subsection 2.2.2. The consideration is based on the work of Wan, Merwe, and Nelson in [WN00; WMN00; Nel00], and the explicit formulation of the joint UKF is given by Mahdianfar et al. [MPA13].

To correctly estimate the different types of states and enable the Kalman gain V_k to take effect, the state covariance matrix $\hat{P}_{xx}$ has to be evaluated sufficiently, cf. Subsection 2.2.2. In the case of joint estimation, $\hat{P}_{xx}$ establishes the connection between the states and, thus, between the ANN and the physical model.

In the case of dual estimation, the secondary diagonals describing this connection are not provided. Preliminary tests have shown that the design of the $\hat{P}_{xx}$ matrix and the aircraft system and measurement models are relevant to the desired use case. When the creation of the aircraft models strictly follows the guidelines from the literature, as is the case in this thesis, an absence of state updates occurs for some states using dual estimation. Therefore, state estimation can be insufficient, which requires further research in future work.

Consequently, the joint estimation approach is chosen, which enables the connection of all states within $\hat{P}_{xx}$. The size of $\hat{P}_{xx}$ is affected by the number of dynamical, parameter, and ANN weight states. These are also considered in the state update and prediction.

Defining the parameters and ANN weights as additional states of the considered system represents the first step of joint estimation. In detail, the time-discretized state prediction equation at timestep k now involves the dynamical physical-based model f_d for the prediction of $\hat{x}_d$ and the ANN with f_{nn} for the prediction of $\hat{x}_p$, while the ANN weight states $\hat{x}_{nn}$ are held constant with $\hat{x}_{nn_{k+1|k}} = \hat{x}_{nn_{k|k}}$:

$$\hat{x}_{k+1|k} = f_k(\hat{x}_{k|k}) = \begin{bmatrix} \hat{x}_{d_{k+1|k}} \\ \hat{x}_{p_{k+1|k}} \\ \hat{x}_{nn_{k+1|k}} \end{bmatrix} = \begin{bmatrix} f_d(\hat{x}_{d_{k|k}}, \hat{x}_{p_{k+1|k}}, u_k) \\ f_{nn}(\hat{x}_{d_{k|k}}, \hat{x}_{nn_{k+1|k}}, u_k, u_{nn_k}) \\ \hat{x}_{nn_{k|k}} \end{bmatrix} . \tag{3.2}$$

Eq. 3.2 is introduced in [EK22]. As stated in Subsection 3.4.1, the inputs of the ANN consists of the states $\hat{x}_{d_{k|k}}$, u_k, and u_{nn_k}, while the ANN structure is defined by the weights $\hat{x}_{nn_{k|k}}$. In addition, the parameter states $\hat{x}_{p_{k|k}}$ can be included by f_{nn}.

However, to enable the initialization in Chapter 5 and a reduction of the input layer nodes and thus computational effort, $\hat{x}_{p_{k|k}}$ is suggested to be omitted as an input to the ANN. The input vector can be further reduced and preprocessed to build significant features to the ANN, see Subsection 2.4.1 and Chapter 5.

The joint estimation enables simultaneous adjustment of the states and, thus, of the physical model and the ANN. It leads to an occupation of the secondary diagonals of the state covariance $\hat{P}_{xx}$, which establishes the connection between all system states. This connection finally couples the physical model with the ANN within the hybrid model structure.

3.5. Preliminary studies

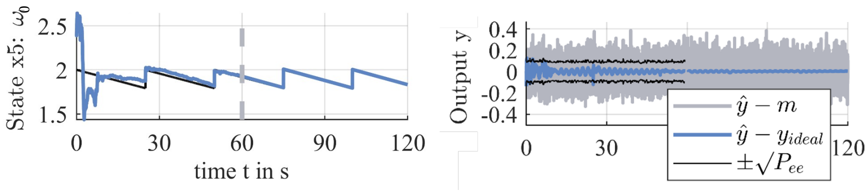
The usability and performance of the proposed hybrid model are first examined in two preliminary studies. They provide significant insight into the initialization and model performance. First, an oscillating system is considered using dynamic and simulated data in Subsection 3.5.1. Second, the efficiency of an aircraft engine is considered with aggregated flight data of a civil transportation A320 aircraft in Subsection 3.5.2.

3.5.1. A study on a basic mass-spring-damper model

A mass-spring-damper model is introduced in [EK22] as a primary simulation test environment for applying recursive filters. It allows for comparing filter performance, supervising the filter tuning, and investigating new developments in this field. The simulation environment provides access to the true states of the system and allows for modifications to the mass-spring-damper model. One modification is the artificial degradation and overhaul of the spring, which is mathematically assumed by a sawtooth trend of the spring eigenfrequency ω_0.

The eigenfrequency is initially set to $\omega_{0_{ini}} = 2rad/s$. It degrades over time and recovers at certain events. The damper ratio is set to $Dr = 0.3$. The input u represents a periodic excitation. Additional data u_{nn} schedules for the overhaul events with the time since the last recovery event to be considered by the ANN part. White noise is added to the output y_{ideal} to obtain the noisy measurement m. A detailed description is found in [EK22].

In applying the hybrid model, the sawtooth trend of ω_0 remains unknown to the filter algorithm. Using m, u, and u_{nn}, the filter accurately estimates the time-variant eigenfrequency ω_0, see Figure 3.5a.



(a)

(b)

Figure 3.5.: Selected representation of the hybrid model's learning behavior estimating the saw tooth trend of the fifth state, the parameter state ω_0 (a). The residual using the noisy simulation output m and the ideal output y_{ideal} is shown in (b). Note that the transition from training to validation data is marked a dashed line at 60 seconds in (a) and with the end of the residual covariance estimation P_{ee} in (b). [EK22]

In the training period of the first 60 seconds, the initial deviations of the states are captured, and the hybrid model is identified. The dynamical and parameter states are sufficiently estimated compared to the true states[5]. Furthermore, the identified model can predict the states' future course without any measurement updates in the period $t \in (60, 120]$. [EK22]

The hybrid model has an RMSE$_{training}$ $= 0.023$ and an RMSE$_{validation}$ $= 0.004$ concerning the output and measurements within the training and validation data. The estimated output converges to the ideal output y_{ideal} after a few oscillations of the system within the first 10 seconds, see Figure 3.5b. The values of the error metrics are considered low, and the application of the hybrid model is successful. Therefore, the subsequent study's complexity is increased.

[5] A complete representation of states, the output, residuals and error metrics are given in [EK22]

3.5.2. A study on the efficiency of an aircraft engine

In cooperation with LHT, the hybrid model is applied to ECM data of an A320 aircraft of the Lufthansa fleet [EHH22]. Over 10 years into service, the conditions of the two engines are investigated in terms of fuel efficiency. The investigation includes the use of flight and maintenance data of the aircraft.

During the period considered, engine gas path cleaning events are performed. Engine overhauls and changes do not occur and are therefore excluded. Other maintenance events are neglected. Consequently, the maintenance events are reduced to engine gas path cleaning events and converted to time-series data[6], added to the aggregated data pool of ECM data. ECM data is aggregated according to stable cruise conditions based on the Airbus Aircraft Integrated Data System (AIDS) for A320 aircraft [Air02].

Since aggregated data are used, the design of the physical model describing the thermodynamic engine circuit process is reduced to the creation of the measurement equations exclusively. The definition of dynamical states and the corresponding state prediction equations are omitted.

For the investigation of fuel efficiency, the fuel flow is used as the objective for the training process. Additional input features extend the inputs to the ANN in three steps, which results in three different hybrid models for comparison. Filter tuning of the hybrid model is performed by using a genetic algorithm for hyperparameter optimization.

The distinction of the models is given through:

- the constant parameter value for the physical model,
- the total operation time used as input feature for hybrid model 1,
- the added information about the maintenance events for hybrid model 2,
- and the added information about the operating point using aggregated sensor measurements for hybrid model 3.

The fuel flow is mapped with increased accuracy, which is particularly visible for the validation and test data set compared to the results of the physical model, see Appendix C.1. The training data set shows slight deviations initially due to the initialization by the physical model and its assumed physical conditions.

[6] For the data conversion the time since the last event is used represented by the flown flight cycles and the flight hours.

Even though higher accuracy is observed in the objective mapping, the observability of the inner states is not given. The inner states of the system consist of eight physical meaningful parameters. While 7 of 8 parameters are shifted into the SC, the fan pressure ratio remains interpretable, see Figure 3.6.

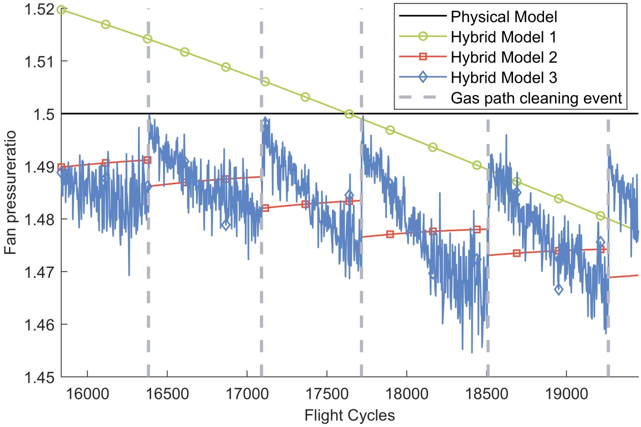


Figure 3.6.: Predicted fan pressure ratio of three hybrid models and the physical model showing an excerpt of the processed test data. [EHH22]

Within the represented section of flight cycles in Figure 3.6, a plausible degradation of the fan pressure over time can be concluded from hybrid model 1 and hybrid model 3, which includes the representation of the gas path cleaning events. Hybrid model 2, on the other hand, does not consistently map the expected impact of gas path cleaning events on the fan pressure ratio, which is related to an alternative initialization found by the genetic algorithm. The alternative initialization includes a variation of the initial covariance matrices. It leads to an estimation of the eight parameters being different from the results provided by the hybrid models 1 and 3. In addition, incorporating measurement data in hybrid model 3 inserts measurement noise into the ANN and thus the state estimation.

Consequently, feedback on the model design is involved in the development process. The conclusions are discussed below and further considered in applying the hybrid model in Chapter 5.

3.5.3. Feedback on the model design

The consideration of the oscillating mass-spring-damper system in Subsection 3.5.1 provides evidence that the concept is feasible for dynamic data of an oscillating system. An aircraft system is such a system that is weakly and strongly damped and experiences high- and low-frequency oscillations during operation. The characteristic movements of the oscillating aircraft are attributable to eigenvalues [BAL11]. Therefore, an aircraft is appropriate for further research, and the hybrid model can be applied to more complex systems for investigation.

However, using the hybrid model for studying a more complex system in Subsection 3.5.2 has revealed issues. First, the use of aggregated event-based data limits the use of the hybrid model. The description of the system is reduced by the loss of the system dynamics and, consequently, by valuable physical dependencies.

Second, the observability of the physical parameters is a significant topic, cf. Subsection 2.2.3. The use of a physical model extended in the context of the hybrid model privileges the use of physical knowledge, and tools and methods from the field of control engineering. In this way a relationship between data-driven model building and observability can be established, which is further explored in this thesis.

Third, the filter tuning process can implement adaptive methods, see [EK22], or randomized optimization methods, see [EHH22]. Both reveal challenges. Adaptive methods require laborious manual filter tuning and can cause filter instability, while the effort can pay off in increased convergence speed. An optimization algorithm can relieve the human, but it can lead to incomprehensible results.

Fourth, measurements used as input to the hybrid model incorporate measurement noise into the state estimation of the system.

Therefore, the following feedback on the model design and its application is derived:

1. It is advantageous to consider dynamic systems with dynamic (non-aggregated) data.
2. The observability of the hybrid model's physical parameters must be considered.
3. The process of filter tuning suggests to be kept simple but physically based.
4. Measurement noise must be excluded from influencing the inner system states.

3.6. Summary

The development of the hybrid model is organized following a methodology based on ISO/IEC/IEEE 15288:2015 and VDI/VDE 2206:2021-11. The scope of this chapter covers the introduction to the new approach, the definition of requirements, the design of the hybrid model consisting of the hybrid model structure and the estimation method, and the implementation using preliminary studies.

The new approach to building a hybrid model includes the available and affordable physical-based knowledge, a data-driven open structure will extend. A derivation of requirements follows, clarifying the research goals and delimiting the hybrid model's scope. The requirements are divided into four groups: initialization, generalization, interpretation, and adaption, each containing specified requirements. The delimitation concerns the usability of the hybrid model, which is designed to describe a particular system, including its operational history.

A concept is designed to address the defined requirements and needs according to the use case of an intelligent aircraft system. Systematic deviations such as system degradation, technical modification, and uncertainties of the physical model are left to be captured by a data-driven model, an ANN. The ANN is connected to a physical model using interpretable physical parameters as an interface. Thus, the hybrid model structure relies on a parameter scheduling model that uses input, scheduling input, and measurement data. The output of the physical model is compared to the measurement data, while the inner states of the system can be fed back to the physical model and ANN. The system states provide insight for analysis.

A state estimation procedure is derived based on existing methods and extensions for system identification. Along with the requirements and claims, arguments are settled for choosing the joint state estimation with a modified Unscented Kalman Filter and State Constraints: the joint SCmUKF.

The hybrid model is implemented and used in two preliminary studies. Based on this, the use of dynamic system data, consideration of observability of states, conservative filter tuning, and the exclusive use of measurement data for the creation of residuals are recommended. These attributes are considered in evaluating the hybrid model prepared in the next Chapter.

4. Experimental design including flight tests and simulation

An exemplary application to flight data of an aircraft system is provided to evaluate the hybrid model. The application aims to provide accurate and physically consistent state estimation and prediction to support future health state monitoring. Furthermore, the applicability in the context of intelligent systems will be evaluated. The evaluation procedures follow an experimental design developed in this Chapter. Part of the experimental design is employing a flight test platform and creating a simulation environment for data acquisition.

To guarantee the availability of dynamical, unprocessed data and to enable the incorporation of modifications, a flight test platform of the Institute of Flight Systems and Automatic Control (FSR) is used and described in Section 4.1. Section 4.2 outlines the experimental design, including the setup for flight tests and simulations.

The flight tests are accomplished for two reasons. First, developing a plausible simulation environment is addressed in supporting its growth using of flight test data. Second, applying the hybrid model to real flight data can be examined. The flight test environment of the Technical University of Darmstadt (TU Darmstadt), the executed flight tests and the recorded data are described in Section 4.3.

Flight simulations are carried out to record flight simulation data containing the system's true states and ideal measurements. In addition, the modification of parameter states is enabled, which is performed according to the planned experiments. The true states and ideal measurements allow us to assess the state estimation accuracy using error metrics. Thus, the hybrid model can be evaluated in Chapter 5. State modifications specify an abstracted behavior of system degradation and recovery, which is part of Section 4.4.

Conclusions about observability follow in Section 4.5. Section 4.6 summarizes the Chapter.

4.1. The Unmanned Aircraft System (UAS) as a flight test platform

UAS are becoming increasingly popular in research, which goes hand in hand with the growing number of UAS use cases, especially in the commercial sector [Fed21; Chr+18; SES17]. Examples of the use of UAS in system identification research are found in a survey of Unmanned Aircraft Vehicles (UAV) system identification in [Hof+14]), the estimation of the aerodynamics of a UAS in [SCJ13], real-time parameter estimation for UAV in [Gäb12], for wind estimation [LAN11] and experimental validation [Che+09]. A common feature of all papers cited is using low-cost sensors and small UAS in research.

The UAS offers the characteristics of an aircraft system. When using a UAS flight test platform of the FSR, the aircraft's characteristics are fully known, and flight tests are executed independently at the flight test site of the TU Darmstadt. Flight testing enables the acquisition of dynamic unprocessed data and the implementation of technical modifications. Therefore, a UAS is evaluated as a flight test platform, assuming transferability to civil aviation[1]. The chosen UAS is described in Subsection 4.1.1. In addition, implemented aircraft modifications are described in Subsection 4.1.2.

4.1.1. Description of the UAS used

The hybrid UAS of the FSR is initially designed at the FSR by Prochazka and Krüger for testing novel concepts for Active fault-tolerant Control for Redundant UAV (ACoRUs) [Pro+21; PS20; PKR20; PRE19; PEK18]. Taking advantage of the payload to implement additional sensors and modifications, the chosen UAS is suitable for flight testing and data acquisition in this thesis. The configuration can be categorized as "Hover Plus Cruise" following [Sta+18], which is characterized by the implementation of static lift motor positioning for the ability to Vertically Take-Off and Landing (VTOL), hover, and Fixed Wing (FW) flight. With its combined skills, the hybrid UAS has become a

[1] Civil aircraft follow the same laws of flight dynamics but can offer more complex systems of aerodynamics and propulsion. Sensors used in civil aviation provide more accurate measurements compared to the low-cost sensors used in small UAS [BM12]. This makes the UAS a suitable candidate for studying less complex flight systems while accounting for increased sensor noise.

subject of research and industry[2] [PRE19; Chr+18; Sta+18]. The configuration used is shown in Figure 4.1.

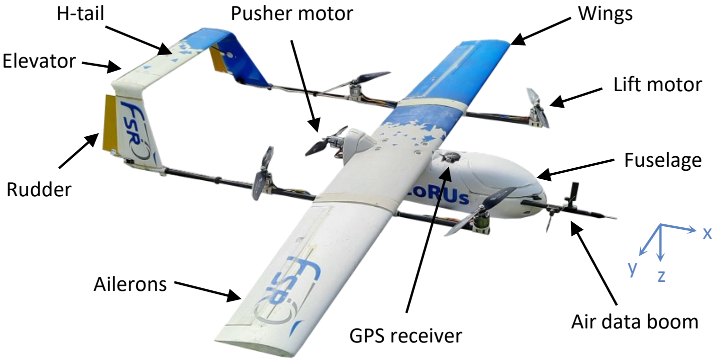


Figure 4.1.: Main components of the UAS used as a flight test platform

The UAS is based on the FW aircraft “Skyhunter“, commercially available from SonicModell. The configuration is also equipped with four lift motors, a high-mounted H-tail, customized control surfaces, fiberglass reinforced wings, and implemented sensor technology named the “Scihunter“ UAS. It is used for various research topics at the FSR today[3].

The Scihunter UAS counts two ailerons, left and right, two synchronous rudders, left and right, and one elevator at the H-tail. Additional sensor equipment, such as the air data boom with two wind vanes, measures the Angle Of Attack (AOA) and Side Slip Angle (SSA). It is connected to the Flight Management Unit (FMU) directly or to an additionally integrated Arduino microcontroller for data acquisition. Both are positioned inside the fuselage next to the Power Management Board (PMB) and the battery packs.

The following details about the system design, geometry, and sensors are needed to derive and implement the experimental design and the physical model. Supplementary information to fully describe the Scihunter UAS is given in Appendix A.

[2] Examples for industrial use of small hybrid UAS are: Wingcopter GmbH and Zipline International Inc. in the civil sector.

[3] Research is contributed in the field of ACoRUs by Prochahzka, Fault Detection Isolation and Recovery (FDIR) by Krüger, PHM by Dingeldein and Weigert, AI and hybrid modeling by Enkelmann

Design of the UAS

The designed Take-Off Weight (TOW) of the Scihunter UAS is $4.39kg$ when fully equipped and ready to fly. Payloads such as weight modules and aerodynamic modifications can be added up to a Maximum Take-Off Weight (MTOW) of $5kg$. The allowed maximum flight altitude of $120m$ can be reached, which is limited by the European Union Aviation Safety Agency (EASA) [Eur22]. The True Airspeed (TAS) of the UAS ranges from 0 to $20m/s$. The different flight modes are divided into:

- Quadcopter (QC) flight with TAS $\in [0, 8]$ in m/s,
- Transition (TR) flight with TAS $\in [0, 11]$ in m/s,
- and FW flight with TAS $\in [11, 20]$ in m/s.

Hovering is a particular case of QC flight with TAS $= 0$.

A Pixhawk 4 flight controller is used as the FMU for control, mission management, and data acquisition, allowing for autonomous flight missions. The battery pack is located in the front part of the fuselage. It consists of four $14.8V$ lithium-polymer cells with a total capacity of $9Ah$. Mainly, depending on the flight modes' composition and the battery's defined capacity to be retained in the case of a Fail-Safe (FaS), the total flight time varies between 5 and 20 minutes.

The Scihunter is equipped with Brushless Direct Current (BLDC) motors. The pusher performs a nominal speed of $700KV$ and $850KV$ by the lift motor. With a power of $750W$, a continuous current of $42A$, and an operation voltage of $10 - 20V$ at a weight of $172g$, the pusher motor is designed to be larger than the lift motor. The lift motor is characterized by a power of $480W$, a continuous current of $35A$, and an operation voltage of $7V$ to $15V$ at a weight of $130g$. This results in a maximum rotation speed of $\omega_{P_{max}} = 10360rpm$ of the pusher and $\omega_{Li_{max}} = 12580rpm$ of the lift motor. An overview of the motor characteristics is given in Appendix A.2.

Geometry of the UAS

The wingspan of the Scihunter UAS is $b = 1.8m$ with a wing area of $S = 0.4271m^2$ and a Mean Aerodynamic Chord (MAC) of MAC $= 0.2334m$.

The lift motors are equidistant from the Center of Gravity (CG). The distance between the motor position and the x-axis and y-axis in the body-fixed Coordinate System (CS) is described by $r_{xy} = 0.32m$. The origin of the body-fixed CS is nested in the CG of the UAS. According to the UAS design, the CG is located in the middle of the fuselage, measured $0.44m$ from the aircraft nose. The deviation in the z-direction is neglected as the impact on the balance of forces and moments is assumed to be small.

The geometry of the rotors is given by a diameter of $d_P = 0.3238m$ and $d_{Li} = 0.3048m$, as well as a rotor surface of $A_P = 0.0823m^2$ and $A_{Li} = 0.073m^2$ for the pusher and lift rotors, respectively.

The moments of inertia are $J_{xx} = 0.2508$, $J_{yy} = 0.2902$, $J_{zz} = 0.5055$, and $J_{xz} = 0.0308$ in kg/m^2. J_{yz} and J_{xy} are set to zero.

For further consideration, technical drawings are given in Appendix A.1.

Sensors of the UAS

The sensors are in the low-cost range, achieving proper measurements with sufficient data quality. A short overview of the sensors and the measurements m used in this thesis is given below and listed in Table 4.2.

The Inertial Measurement Unit (IMU) is part of the FMU. It measures the acceleration and rotational rates in three axes. The IMU is located close to the CG[4].

The pusher and lift motors' Electronic Speed Controllers (ESC) convert Direct Current (DC) to the three-phase Alternate Current (AC) for the BLDC motors according to the Pulse Width Modulated (PWM) signal received. Acquired data are the rotational speed, voltage, and current.

The airspeed sensor is realized by a pitot-static tube integrated into the Air Data Boom (ADB). Both the dynamic pressure and the resulting IAS are available on the FMU. Neglecting an installation error and assuming a constant air density of $\rho_0 = 1.225kg/m^3$ (cf. Subsection 4.4.2), the TAS directly follows from the IAS: TAS $\approx$ IAS. A constant air density can be assumed in this thesis as the flight missions are planned with a maximum of $95m$ Above Ground Level (AGL) and at stable weather conditions. Note the ADB is in line with the body-fixed x-direction of the UAS, which has to be considered in the measurement equations of the physical model. Additionally equipped with two wind vanes, the ADB acquires data about the AOA and SSA.

The used servo motors are closed-loop controllers, which couple the electric motor and position feedback. After their calibration, the deflections of the ailerons, the rudders, and the elevator are obtained.

[4] Adjusting the measured accelerations according to the position of the CG is not performed as suggested by Gäb in [Gäb12]. The effect is assumed to be negligible when using the Scihunter UAS.

Table 4.1.: Sensors and measurements of the Scihunter UAS used

Source	Symbols	Description	Unit	MSR
IMU	a_x,a_y, a_z	Acceleration in x, y, z-direction	m/s^2	249.6
	p, q, r	Roll, pitch, and yaw rate	rad/s	
Pusher-ESC	ω_P	Rotational speed pusher motor ESC	rpm	62.83
	U_P	Voltage pusher motor ESC	V	
	I_P	Current pusher motor ESC	A	
Lift-ESC	ω_{Li}	Rotational speed lift motor ESC	rpm	62.83
	U_{Li}	Voltage lift motor ESC	V	
	I_{Li}	Current pusher motor ESC	A	
Pitot tube	IAS	Indicated Airspeed (IAS)	m/s	62.83
Servo motors	ξ_L	Deflection of the left aileron actuator	rad	62.83
	ξ_R	Deflection of the right aileron actuator	rad	
	η_L	Deflection of the elevator actuator	rad	
	ζ_L	Deflection of the left rudder actuator	rad	
	ζ_R	Deflection of the right rudder actuator	rad	
FMU	ϕ, θ, ψ	Roll, pitch, yaw angle (Euler angles) from internal EKF	rad	20.43
Wind vanes	α	Angle Of Attack (AOA)	rad	8.30
	β	Side Slip Angle (SSA)	rad	
PMB	U_{bat}	Battery voltage	V	8.30
	I_{bat}	Battery current	A	
GPS-receiver	p_{lat}	Latitude position	deg	4.17
	p_{long}	Longitude position	deg	
	alt	Altitude above sea level	m	
	GS	Ground Speed (GS)	m/s	
	VS	Vertical Speed (VS)	m/s	
	χ	Ground Course (GC)	rad	

The Global Positioning System (GPS) receiver measures the latitude and longitude position and the altitude above sea level. Independently, the GS and VS are measured using the Doppler evaluation of the carrier signal [NW94]. This thesis uses the integrated magnetometer of the GPS receiver[5], which delivers the ground course.

The PMB distributes the battery power to all electrical consumers. It measures the battery voltage and current to check system's energy consumption. The power of the PMB and ESCs are directly obtained from the measured voltage and current with $P = UI$. The resulting power signals are treated as measurements with a corresponding noise in Chapter 4 and Chapter 5.

Pseudo measurements are introduced using the EKF running on the FMU to estimate the attitude of the UAS. This way, reference values for the roll, pitch, and yaw angles are obtained.

The sample rates of the sensors are different. In addition, the sample rate can vary due to time delays and specific data errors, which is addressed in the acquisition of flight data in Sections 4.3 and 4.4. The Mean Sample Rate (MSR) is additionally given in Table 4.2 and sorts the sources in descending order.

4.1.2. Implemented modifications for flight testing

One of the advantages of choosing and operating a small UAS is the possibility to integrate modifications under less stringent flight operational regulations; see Appendix A.4. They allow us to vary the operating points in the flight tests due to the technical modification of the UAS and specific changes in the flight mission. Variations are realized by adjusting the Commanded Airspeed (CA) in FW flight, the aircraft TOW, and retrofitting aerodynamic modules to the wingtips. Specific technical solutions are developed for the aerodynamic modification of the aircraft and the adjustment of the TOW in [Enk21; Fic+20].

The additional weights are integrated into the hollow linkage carrying the lift motors and the H-tail. The location of the implemented weight modules and the installation space inside the fuselage are visualized using Computer-Aided Design (CAD), see Figure 4.2.

[5] Located at the top of the fuselage disturbances have less effect on the GPS receiver. The integrated magnetometer of the FMU is neglected.

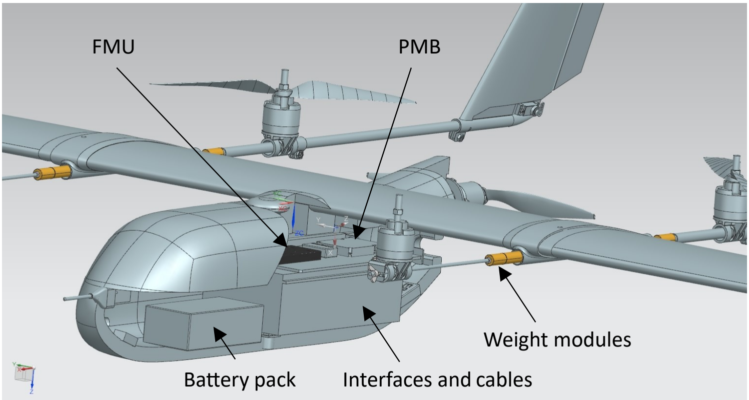


Figure 4.2.: Visualization of the integrated weight modules and the electrical components nested in the fuselage, based on [Enk21; Fic+20].

The electrical components and the battery pack, including access to the components and battery changing, thoroughly use the installation space of the fuselage. Weight modules outside the fuselage can be implemented independently. It allows a stepwise weight increase and the adjustment of the CG weather in the body-fixed x or y direction. The weight modules consist of a threaded rod, cylindrical weights ($50g$ per part), and mounting accessories. The weight modules can be prepared and changed for flight testing.

An aerodynamic modification is integrated with lift modules mounted on the wing tips. Continuing the aerodynamic profile, see Figure 4.3a, the wingspan is extended by an additional $100mm$, see Figure 4.3b.

The total wingspan is increased to $b_{mod} = 2m$ in the case of mounted aerodynamic modules, which enlarges the wing surface area to about $S_{mod} = 0.475m^2$. The modules weigh $82g$ each and are 3D-printed using Polylactide (PLA) material.

According to Ansys flow simulations, one module increases the lift force by $L \in [1.4, 5.4]N$ and of the drag force by $D \in [0.1, 0.9]N$ at a flow velocity of $15m/s$ and when varying the AOA between $\alpha \in [0, 15]deg$. This results in a total increase of the lift force between 3 % to 12 % and the drag force from 1 % to 6 %. These values serve as a reference for the abstracted degradation in the design of the simulation environment

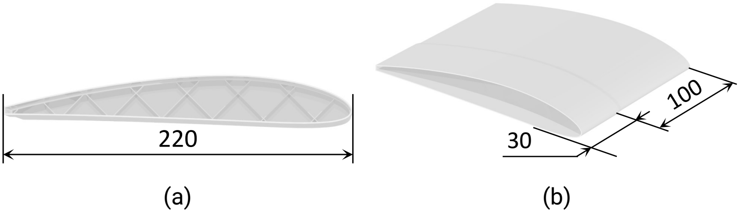


Figure 4.3.: An aerodynamic modification of the UAS wingtips continuing the aerodynamic profile with a hollow 3D printed structure with a chord of 220 mm (a) and increasing the wing length by 100 mm with an overlap of 30 mm for attachment [Enk21; Fic+20].

and as a benchmark for the results obtained from applying the hybrid model to flight test data in Chapter 5. [Enk21]

4.2. Experimental design

For the application and evaluation of the hybrid model, the “true“ experimental design is chosen [CS66]. The experimental design includes flight simulation and testing in terms of computational and physical experiments [DVD17]. It is characterized by manipulation, control, and randomization. Each attribute is considered in the following design, based on the checklist proposed by Dean et al. in [DVD17].

(a) **Objective**
The objective of the experiments is to explore the proposed hybrid model in terms of state estimation and prediction performance using the unmodified and modified UAS example. Further, interpretability, applicability, and learning behavior will be investigated concerning the defined requirements.

(b) **Sources of variation**
The sources of variation are identified as follows:

(i) “Treatment factors“ [DVD17] are the modification of the simulation model for flight simulation, the UAS for flight testing, and the filter model used for estimation and prediction.

(ii) “Experimental units“ [DVD17] are the variations of the TOW and CA within a defined standard flight mission.

(iii) The limited use of operational data is considered a “blocking factor“ [DVD17], which the standard flight mission abstracts. “Nuisance factors“ [DVD17] are the additive white noise in the case of the simulated data. In the case of flight test data, wind, atmospheric, and aircraft conditions as well as sensor errors affecting the records of the sensors.

(c) **Assigning the experimental units**
The variation of experimental units is kept similar while varying the flight simulations, flight testing, and the filter model. The TOW and CA in the FW flight are set to be constant within a single flight mission, see Subsection 4.2.3.

(d) **Specification of measurements**
Measurements are made according to the sensors of the UAS introduced in Subsection 4.1.1 for both the flight simulation and testing. In the case of simulations, the ideal measurements and true states are acquired additionally, which allows for the true experimental design. The execution of flight tests is described in Section 4.3, and flight simulations in Section 4.4.

(e) **Pilot experiments**
Several pilot experiments have been carried out, first of all concerning safety within flight testing. Furthermore, the feasibility of the flight mission and the completeness and the quality of the sensor measurements are successfully tested considering the unmodified UAS.

(f) **Specification of the model**
The hybrid model is subject to experimental investigation. It considers fixed and random effects and involves a data-driven structure, which is to be determined for more reliable state estimation and prediction.

(g) **Outline of the analysis**
The analysis outlines the application of the hybrid model to flight data, focusing on determining unknown modifications of the physical model (considering flight simulation data in Section 5.2) or the physical substance (considering flight test data in Section 5.3) of the UAS. The physical properties of the unmodified UAS are assumed to be known or partly known by the filter model.

(h) **Number of observations**
Several flight tests are required to incorporate modifications between the flights and perform flight tests with varying experimental units. The duration of a flight

test is expected to be between 10 to 15 minutes with sample rates of up to 250 Hz, which intends a maximum volume of 2.25×10^5 data points per flight and sensor.

(i) **Review**
A review of the experimental design follows along with the evaluation and discussion of the results in Sections 5.4 and 5.5.

4.2.1. The architecture of the experimental setup for data generation

The flight tests are defined as physical experiments. Flights simulations are defined as computational experiments that require an adequate simulation environment of the chosen UAS. To simulate a UAS behavior similar to the behavior observed in flight testing, a Software-in-the-Loop (SIL)[6] simulation framework is designed. Consequently, the same FMU characteristics are employed for flight tests and simulations. The physical model used within the simulation framework is also adjusted to the acquired flight test data.

In the following, the architecture of the experimental setup is defined, which organizes the flight tests and flight simulations. The architecture describes the process of data acquisition, see Figure 4.4.

Starting with the UAS used as a flight test platform, the flight test planning, and the execution of flight tests, five links of information are generated:

1. The characteristics of the UAS are provided to the physical model, including the ArduPilot firmware running on the FMU, which is also used in the SIL simulation framework.
2. The operational conditions from the flight test planning are transmitted to the physical model, including the trajectory processed by the FMU in the simulation framework.
3. Selected flight test data is used to adjust the physical model of the simulation framework.
4. The flight test data is fed into the database.

[6] SIL allows advanced and reviewed software to be integrated into the experimental framework. In this way, elaborate processes can be outsourced, while maintaining confidence according to the reviewed status of the used software. Furthermore, it can increase comparability between the real and simulated system as is the case in this thesis.

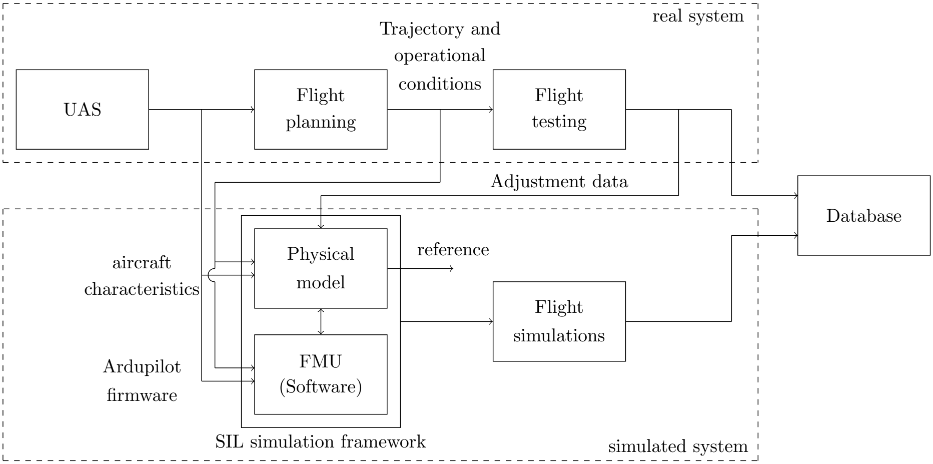


Figure 4.4.: Architecture of the experimental setup for generating flight test and flight simulation data.

5. The flight simulation data is fed into the database.

Thus, the SIL simulation framework consists of two main parts: the FMU software and the physical model describing the UAS flight dynamics and sensor model.

Due to the linkage between the real and the simulated system, especially the adjustment of the physical model using flight test data, the validity of the physical model is limited to the considered flight test data. This is intended for two reasons. On the one hand, a specific system is described, including its operational history (cf. Subsection 3.2.4). On the other hand, a physical model results for reference specifically for the considered flight data. Finally, the SIL simulation framework enables flight simulations to be performed and flight simulation data to be fed into the database.

4.2.2. Treatment factors of the experimental design

Four Test Series (TS) are planned, which successively increase the demands on the hybrid model as the treatment factors are varied. These manipulate the experimental data and hybrid model structure, which enables the exploration of the hybrid model within the defined test and simulation environments. Thus, by specifying the treatment factors in the four TS, the use of the experimental setup for data generation is predetermined. Manipulations within the TS concern the modification of the data source and the hybrid model structure, see Table 4.3.

Table 4.3.: Assignment of the treatment factors, the data source, and the hybrid model structure in four TS within the experimental design.

Test series	Data source	Hybrid model structure
TS1	FS_0	$\mathrm{PM}_0 + \mathrm{ANN}$
TS2	FS_{mod}	$\mathrm{PM}_0 + \mathrm{ANN}$
TS3	FS_{mod}	$\mathrm{PM}_{red} + \mathrm{ANN}$
TS4	$\mathrm{FT}_0 + \mathrm{FT}_{mod}$	$\mathrm{PM}_0 + \mathrm{ANN}$

Concerning the data source, flight simulations are to be conducted without aerodynamic modification and with aerodynamic modification of the UAS, resulting in FS_0 (basic Flight Simulation (FS) data) and FS_{mod} (modified FS data). Flight tests are to be conducted similarly, resulting in FT_0 (basic Flight Test (FT) data) and FT_{mod} (modified FT data).

The hybrid model structure is to be equipped with a basic physical model PM_0 and a reduced physical model PM_{red}, which are further considered in Section 4.4 and Chapter 5.

The simulation framework incorporates a physical model for data generation. Consequently, PM_0 is used to generate FS_0, while the modification of the physical model leads to PM_{mod} and FS_{mod}.

In the case of TS3, the hybrid model structure is manipulated by changing PM_0 and incorporating PM_{red} into the filter environment, see Figure 4.5.

Within the first three TS, the hybrid model is trained using the simulation data. Due to the design of the recursive filter algorithm, the usual separation into training, validation, and test data for ML is not necessary [Arb+22]. Instead, the stepwise update and prediction, the adjustment of the filter covariances, and the examination of the convergence behavior are used to ensure the validity of the filter results, see Chapter 5.

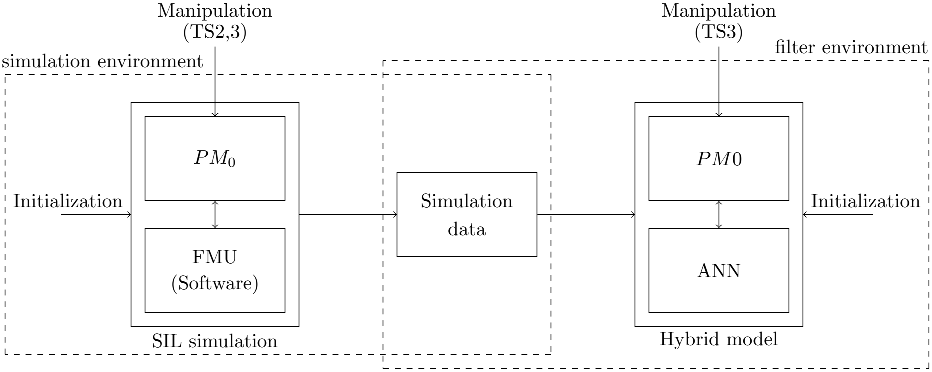


Figure 4.5.: Computational experiments, including the simulation environment and the filter environment. Here, the hybrid model is applied to flight simulation data. In TS1, no manipulation is intended. In TS2, the physical model used for SIL flight simulations is manipulated. In TS3, the SIL simulation and the physical model used within the hybrid model are manipulated.

The modifications emulate the degradation and recovery of the system as it is assumed in operation, which is further described in the design of the simulation environment in Subsection 4.4.4.

The performance in learning the unknown impact of the modification is rated using the:

- true states,
- ideal measurements,
- and physical reference model.

These attributes are unique to the use of simulation environments and provide insight into the behavior of the hybrid model when applied to flight simulation data.

A further step is using real flight test data, closing the gap between the digital object (physical model used in a simulation environment) and the real object (UAS), cf. Subsection 2.3.4. In this case, a deviation from the physical model is inherently included, and the hybrid model is evaluated in applicability and plausibility. Remind true states and ideal measurements are not available.

4.2.3. Operational conditions for flight testing and simulation

In the next step, experimental units are defined according to [DVD17]. This results in the specification and determination of operational conditions and manipulations for flight testing and simulation. Setting the operational conditions is inspired by fight data of commercial transport aircraft experienced in previous publications; see [EHH22; EHP20]. The flight missions are mainly characterized by the predominant stable flight, the airspeed, TOW, and periodically recurring maintenance events.

Consequently, airspeed and TOW vary, while manipulations are employed according to the previous section. The adjustment of the CA realizes the variation in the airspeed of the respective flight mission. The CA is planned for the FW flight as part of the trajectory transmitted to the FMU.

Thus, the CA varies between $12m/s$, $15m/s$, and $18m/s$ in the respective flight mission. The values are determined based on the designed airspeed range of the Scihunter UAS in FW flight, cf. Subsection 4.1.1. In this way, the flight envelope is explored regarding the UAS airspeed, which results in a variety of flight conditions.

To further affect the variety of flight conditions, the TOW of the aircraft is modified with the integration of weight modules, cf. Subsection 4.1.2. Three stages of the TOW are provided for the respective flight mission at $4.39kg$, $4.616kg$, and $4.824kg$. This means payloads of $226g$ and $434g$ increasing the TOW by 5% and 10%. The weight increase considers the limitation of the MTOW of $5kg$, cf. Subsection 4.1.1. The CG is adjusted to stay unchanged for the flight missions, while effects on the moment inertia are neglected.

Finally, one TS is characterized by nine flight missions resulting from three different CA in FW flight and three different TOWs of the UAS. These characteristics of the flight missions remain unchanged for each TS. Manipulations are performed by modification using aerodynamic modules (cf. Subsection 4.1.2) for flight testing or aerodynamic parameters within the simulation environment, see Subsection 4.4.4.

4.3. Flight testing

The chosen flight test platform allows flight tests, which are planned and executed according to the experimental design derived in the previous Section. The flight test data is used to evaluate the hybrid model, configure a physical-based reference model, and learn about actual sensor behavior in flight. The observed sensor behavior is considered for filter tuning and acquiring of simulation data.

The flight tests with the UAS are planned in Subsection 4.3.1. A short overview of the flight executions is given in Subsection 4.3.2. The collected flight data is analyzed in Subsection 4.3.3 before being further used in this thesis.

4.3.1. Flight test planning

A standard flight mission of the UAS is defined according to the accessible flight test environment. The flight test environment specifies a flight geography that is spatially constrained and requires compliance with safety features, see Appendix A.4. The flight mission is planned to be autonomous and inside the defined flight geography. VTOL maneuvers are planned to reach and leave the flight geography. The flight path of the standard flight mission resembles a figure-of-eight-profile consisting of a sequence of different schemes, which are repeated according to the operational conditions planned in Subsection 4.2.3, see Figure 4.6.

In this way, the flight path includes the maneuvers expected when operating a hybrid UAS. In addition, the figure-of-eight profile uses the diagonals of the flight test site used, increasing the distance traveled in horizontal FW flight, ascending or descending. In the figure-of-eight profile of the flight path, two turns of 180 degrees result, which are orientated in both directions, left and right. The turns within the flight path are planned with several waypoints using the entire width of the flight geography volume. This ensures the UAS maintains altitude and the remote pilot closely monitors the trajectory. For visualization, a standard flight mission is given in Figure 4.7 as an example.

Waypoints are used as the basis for planning the flight missions, with details of the specified waypoints given in Appendix A.5 and Appendix A.6. They carry information about GPS coordinates, the FM, CA, and jump commands, which initiate a repetition of maneuvers. Thus, the figure-of-eight-profile can be flown through several times in a loop, including ascent and descent between $75m$ and $95m$ AGL.

Following the experimental design, the flight test planning involves the variation of the operational conditions, cf. Subsection 4.2.3. This includes the implementation of the aerodynamic modifications introduced in Subsection 4.1.2. For the first nine FMs, the UAS aerodynamics remain unmodified. Equipped with the aerodynamic modules

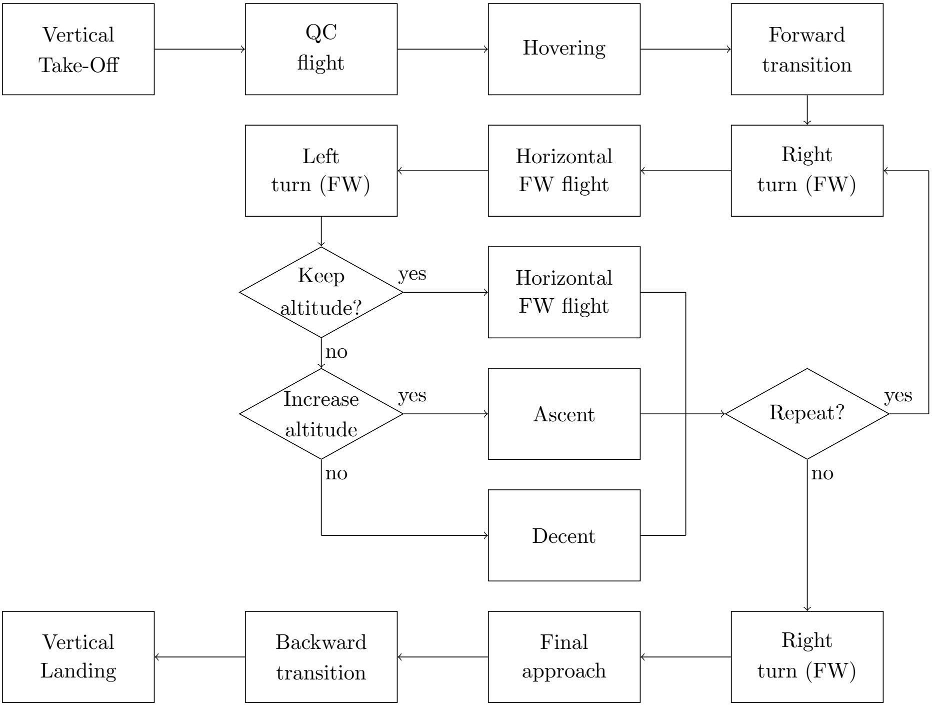


Figure 4.6.: Sequence of maneuvers within the defined standard Flight Mission (FM).

introduced in Subsection 4.1.2, nine additional FMs are planned, bringing the total number of planned FMs to 18.

The number of repetitions is set according to the increase of the CA and the TOW. Since higher power demands are expected, the number of repetitions is adjusted and the length of the scheduled flight path is shortened. This ensures a safe flight without risking a FaS.

Remember for the final approach in the FW flight, and the airspeed is set to $12m/s$ independently of the planned CA in the FW flight to relieve the UAS structure in the backward transition.

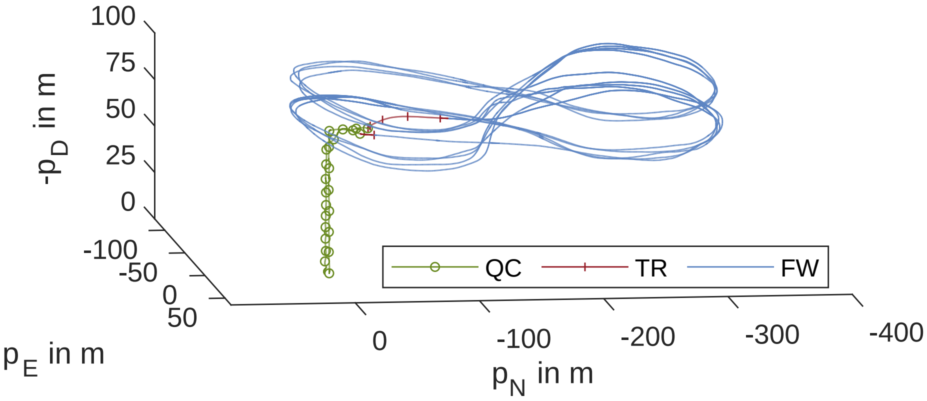


Figure 4.7.: Example of the standard flight mission with a CA of $15m/s$ and a TOW of $4.39kg$.

Starting with the design airspeed of $15m/s$ and without additional payload, the flight tests are planned by setting the number of repetitions according to the operational conditions proposed in Subsection 4.2.3, see Table 4.4.

Table 4.4.: Planned repetitions and distance to fly within the respective FM using the aerodynamically unmodified UAS

FM	1	2	3	4	5	6	7	8	9
TOW in kg	4.39			4.62			4.82		
CA in m/s	15	12	18	15	12	18	15	12	18
Repetitions at $75m$ AGL	4	5	3	5	5	3	4	4	2
Repetitions at $95m$ AGL	4	5	4	5	5	4	4	4	2
Distance in km	7.2	9.8	6.4	9.8	9.8	6.4	7.2	7.2	4

An overview of the flight missions with the implemented aerodynamic modules is given in Appendix A.7.

4.3.2. Flight test execution

The execution of the flight tests is successful, except for FM 8 at CA $= 12m/s$ and TOW $= 4.824kg$. The successful FMs add up to a total flight duration of 92 minutes and a flight distance of $71km$ flown using the aerodynamically unmodified UAS. The total flight distance and duration are distributed unevenly across the different FMs according to the varying CA and TOW, see Table 4.5.

Table 4.5.: Flight distance and duration according to the CA and TOW using the aerodynamically unmodified UAS

(a) Flown flight distance in km

		CA in m/s		
		12	15	18
TOW in kg	4.39	10.30	10.37	-
	4.62	8.63	11.05	8.80
	4.82	8.23	8.25	5.23

(b) Duration of flights in min

		CA in m/s		
		12	15	18
TOW in kg	4.39	15:50	14:49	-
	4.62	11:27	13:00	10:57
	4.82	9:42	9:30	6:44

For a CA of $12m/s$, a flight duration between 15 to 16 minutes is observed at a flight distance of about $10.3km$, while a flight mission at TOW $= 4.824kg$ is missing. Flight duration ranges from 11 to 13 minutes at a CA of $15m/s$ and about 7 to 10 minutes at a CA of $18m/s$. Equivalently, the distances flown lie between $8km$ and $11km$. An exception exists for a flight distance of about 5 km, which results from flight planning in Subsection 4.3.1, according to the increased CA and TOW.

The imbalance of the recorded data occurs due to the increased airspeed in flight trivially. Still, it is also favored by the increased power consumption and the shorter flight path chosen according to Subsection 4.3.1.

In the case of the unsuccessful execution of FM 8, a significantly increased AOA and pitch angle is observed, which led to a stall in stable flight. Consequently, a decrease in maneuverability was observed. This resulted in hitting the geofence and triggering a FaS after a short flight distance of about $1.5km$ and after a flight duration of about 2 minutes. In conclusion, the dataset from FM 8 contains less information and highly dynamic states outside the considered flight conditions and, therefore, is neglected for further consideration.

4.3.3. Data acquisition and analysis

The recorded flight data is analyzed in terms of data quality, completeness, and balance, as highlighted in the previous Section. A plausibility analysis follows in Subsection 4.4.3, when using the acquired flight test data for the adjustment of the physical model derived in Subsection 4.4.2.

In total, the FMU and the implemented Arduino microcontroller collect data of 788 attributes. A large part of the data describes binary processing values and parameter settings of the FMU. For the adjustment of the physical model and the training of the hybrid model, measurements and FMU commands are used. The sensor measurements described in Subsection 4.1.1 and the FMU commands given as PWM signals are therefore considered for analysis.

The measurements used include the 34 attributes of the different sensors with individual time stamps, as introduced in Subsection 4.1.1. This allows the actual sampling rates to be assessed.

The sampling rate of the IMU varies between $163.3Hz$ to $281.6Hz$ and between $2.7Hz$ to $4.7Hz$ for the GPS receiver. As stated in Subsection 4.1.1, the ESC of the pusher motor, the wind vanes, and the aerodynamic actuators (servo motors) are recorded at a sampling rate of $62.83Hz$ on average. The attitude estimation and the command signals of the FMU are available at $20.43Hz$. The barometer, airspeed sensor, and PMB values are provided at $8.30Hz$. Measurements of the four ESCs of the lift motors are recorded at $7.10Hz$ on average, with a minor variance in the sampling rates.

To simplify the handling of the database, preprocessing is introduced to align the different sampling rates to a uniform and constant sampling rate of $100Hz$. Even though the chosen state estimation method of the hybrid model allows varying sampling rates, a uniform, and steady sampling rate reduces the implementation effort, simplifies comparisons, and enables the implementation of ML approaches examined in [Joe22]. Furthermore, the aligned sampling rate of $100Hz$ reduces the computational effort, which is discussed in Chapter 5 in more detail.

The alignment is established by interpolation for all measurements except the yaw angle ψ and the ground course χ, which retain the previous value until an update is recorded[7]. Slight deviations that may occur due to interpolation are neglected and at least taken into account when adjusting the covariance matrices for state estimation.

[7] The recorded values of yaw angle and ground course contain jumps of $360deg$ near the north orientation of the UAS. In the case of an interpolation, physically meaningless values occur.

Joest performs detailed data analysis in [Joe22]. Based on his work and this thesis' data analysis, it is assumed that the described variations of the sampling rates and the occurrence of data errors are within a manageable range.

The acquired database finally enables the balance of the data to be examined. In the previous section, an imbalance of the considered flight data is found, mainly due to the different TOW and CA in the executed TS. With the TAS recorded by the airspeed sensor and the uniform and steady sampling rate of $100Hz$, the balance is investigated, and the data distribution is visualized in Figure 4.8.

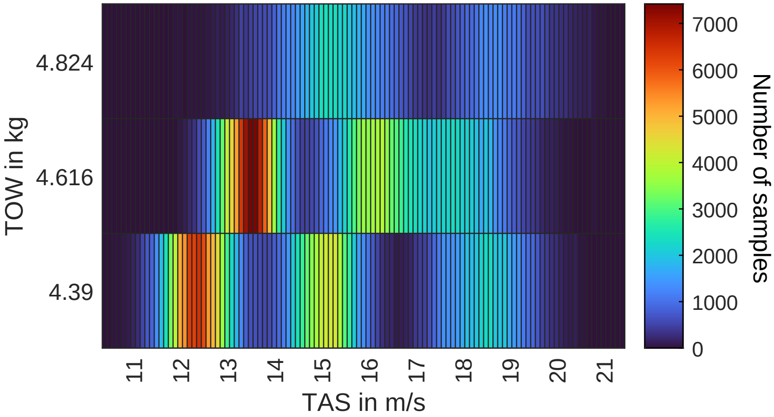


Figure 4.8.: Distribution of the considered flight data in FW flight concerning the TOW and TAS by frequency of occurrence

The visualization highlights a high frequency of occurrence for the lower airspeeds from $12m/s$ to $14m/s$ and a TOW of $4.39kg$ and $4.62kg$. The density of samples decreases with increased TAS and TOW. Samples are rare at the edges of the considered envelope, in the range of $16m/s$ to $17m/s$ at $4.39kg$ and around $17.5m/s$ at $4.82kg$.

The missing flight at CA $= 12m/s$ and TOW $= 4.82kg$ is evident, but in addition, a shift in the data is identified for a TOW of $4.62kg$. In this case, the airspeed commands of the FMU are not sufficiently met in the executed flight tests. The TAS appears to be about $1m/s$ higher than planned, except for the flight data at CA $= 18m/$s, where the TAS appears to be smaller.

Finally, a sufficient representation of the airspeeds is concluded, while lower airspeeds predominate. The physical model's adjustment within the next Section and the evaluation of the hybrid model, especially the underrepresentation of higher TOWs, has to be considered.

Balancing of the data by a selective reorganization of the database is not considered. With a total number of about 550.000 samples and the occurrence of one single flight mission per CA and TOW, the database is considered minor, and the deletion of samples is excluded.

The acquired database contains real sensor noise of the measurements. Using the flight test data to adjust the physical model in Subsection 4.4.3, the sensor noise can be estimated and used to design the simulation environment in the next Section.

4.4. Design of a simulation environment

A significant part of the experimental design is the simulation environment. It generates flight simulation data consisting of true states and ideal measurements. The data generation is based on the flight test platform, planning, and data introduced in the previous Section. The design of the simulation environment is divided into the design of the simulation framework in Subsection 4.4.1, the design of the physical model in Subsection 4.4.2, the adjustment of the physical model using flight test data in Subsection 4.4.3, the abstraction of aerodynamic degradation and recovery in Subsection 4.4.4, and finally the execution of flight simulations in Subsection 4.4.5 with the analysis of the acquired data in Subsection 4.4.6.

4.4.1. Design of the simulation framework

The simulation framework is characterized by a SIL simulation framework based on an implementation given by the ArduPilot Dev Team [Ard09], see Figure 4.9.

The components of the framework are the virtual FMU implemented with the ArduPilot firmware[8] (version 4.2.2) and the physical model of the UAS implemented in Matlab (version 2022b). These two components interact in a loop representing the SIL simulation environment.

[8] The open-source ArduPilot firmware for the Pixhawk 4 flight controller is widespread and recognized in the scientific context. It is well documented and available at: https://ardupilot.org/.

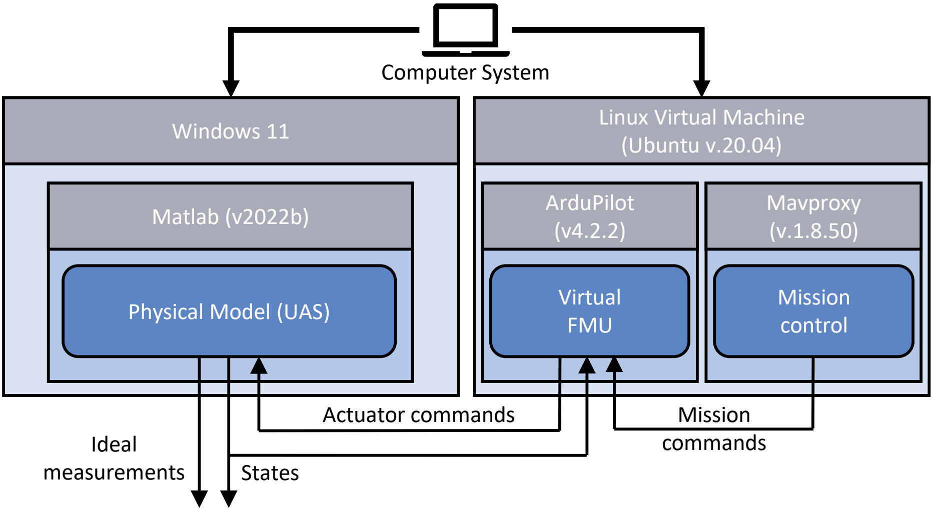


Figure 4.9.: SIL framework using Mavproxy and ArduPilot firmware in a Virtual Machine (VM) environment for mission and flight control with a physical model of a UAS running on Matlab in a Windows environment for data generation.

The framework is implemented on a computer system running Windows 11 and a Linux VM using Ubuntu (version 20.04). In the VM environment, Mavproxy (version 1.8.50) is implemented to perform mission control according to the mission planning by the user. The exact flight path is used for the flight tests (cf. Section 4.3) within the simulation environment, corresponding to the mission commands transmitted to the virtual FMU. The virtual FMU sends the actuator commands and receives the physical states of the simulated UAS in return.

The states, commands, and measurements are saved in a database. It refers to the true states and ideal measurements, which are the unaffected output of the model. White noise is added to the ideal measurements for more realistic sensor behavior. The magnitude of the respective sensor noise is adjusted according to the observations done within the flight tests in the previous Section. Therefore, the numerical values of the added white noise are based on the adjustment of the physical model using flight test data in Subsection 4.4.3 and the physical model designed in the following Section.

4.4.2. Design of the physical model

The design and implementation of the physical model rely on literature, prior developments of the FSR, and aerodynamic formulations suitable for the experimental design, cf. Section 4.2.

Appropriate literature is selected for the implementation of the actuator equations, flight dynamics, and sensor model according to Stevens et al., Beard et al., and Brockhaus et al. [SJL16; BM12; BAL11]. In particular, Stevens, Lewis, and Johnson provide an overview of the nonlinear aircraft model with six degrees of freedom (6DOF) and 12 dynamical states, which is used and extended in this thesis [SJL16].

The main components of the designed physical model are the actuators, the flight dynamics, and the sensors, see Figure 4.10. Using the input u, the forces F and moments M from the actuators block. These affect the system state equations describing the system dynamics. The state vector x is used by the sensor model, resulting in the measurements m. The implemented Matlab code describing the particular algorithms is given in Appendix B.

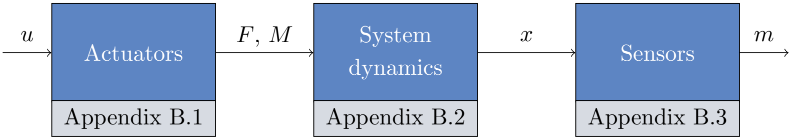


Figure 4.10.: Simplified structure of the physical model used

The design of the physical model of the Scihunter UAS is based on the work of Prochazka, Krüger, and Ribnitzky [PKR20]. A significant number of student works supported the development, implementation, and validation of the considered flight test platform and its mathematical description [Joe22; Fic+20; Frö+19]. The initial parameters are based on Computational Fluid Dynamics (CFD) analysis, wind tunnel tests, and flight tests, see Appendix A.9.

The experimental design offers flight data of a standard FM, cf. Section 4.2. In the FMs performed in Section 4.3, stable flight conditions at similar operating points result in the FW flight of the UAS. Therefore, a simplified physical model is designed in the following, which is shown to provide sufficient mapping.

Actuators and the calculation of forces and moments

The calculation of the forces and moments is mainly considered in terms of aerodynamics and thrust. The command inputs u affecting the actuators are calibrated. In addition, they consider a time delay. The time delay is modeled using a PT1-element and the time constant T_{pwm}, respectively. The actuator states of the previous time step are considered for the time delay and, therefore, part of the system state equations below[9].

The calibrated input signals include the commands for the two ailerons (ξ_L^c, ξ_R^c), the elevator (η^c), and the two rudder deflections (ζ_L^c, ζ_R^c). The calibrated power of the pusher and lift motors[10] are P_P^c, P_1^c, P_2^c, P_3^c, and P_4^c. In addition, u contains the TOW of the UAS and the current flight phase.

To distinguish between the three flight phases (QC, TR, FW), P_P^c, P_1^c, P_2^c, P_3^c, and P_4^c are used[11]. Implemented with logical values in Matlab, two values are introduced: The Φ_P and the Φ_{Li} logic. In the case the pusher motor is active ($P_P^c > 0$), $\Phi_P = 1$; otherwise, $\Phi_P = 0$. In the case one lift motor is active ($P_{1,2,3,4}^c > 0$), $\Phi_{Li} = 1$; otherwise, $\Phi_{Li} = 0$. This results in the following assignments:

$$QC(\Phi_P == 0\,\&\,\Phi_{Li} == 1) = \text{`}true', \tag{4.1}$$

$$TR(\Phi_P == 0\,\&\,\Phi_{Li} == 0\,||\,\Phi_P == 1\,\&\,\Phi_{Li} == 1) = \text{`}true', \tag{4.2}$$

$$FW(\Phi_P == 1\,\&\,\Phi_{Li} == 0) = \text{`}true'. \tag{4.3}$$

It follows that only one flight phase is active at a time.

The flight time in FW flight prevails, cf. Section 4.3. In contrast, the QC flight and, in particular, the TR flight are underrepresented in the flight test data. Therefore, descriptions of coupling effects in TR flight are neglected when considering UAS aerodynamics.

Further, the modeled aerodynamics are valid for the FW flight exclusively and, therefore, only used when the FW flight condition is fulfilled. Otherwise, the aerodynamic forces

[9] The actuator states of the previous time step are transmitted using the state vector. They are defined constant for state prediction. This allows for modeling time-delayed behavior with reduced implementation effort.

[10] The motor assignments are given in Appendix A.2.

[11] Different approaches have been tested against the background of state estimation of the hybrid UAS. Using estimated or measured airspeeds for an abrupt or smooth transition between flight phases lead to instability or unusable results. The flight phases must be specified using an a priori definition. This avoids the state estimation algorithm from shifting its estimates into invalid aerodynamics.

and moments are set to zero.

For the description of the aerodynamics in FW flight, the aerodynamic values are represented as TAS $= V_a$, AOA $= \alpha$, SSA $= \beta$, and MAC $= l_\mu$. From u the deflections ξ_L, ξ_R, η, ζ_L, and ζ_R are used. The rotational rates p, q, and r resulting from the state equations are transformed into the experimental CS to be used in the following.

The drag[12], side, and lift forces (D,Y,L) are determined in the experimental CS with index e. The corresponding coefficients C_D, C_Y, and C_L consist of the parameters C_{D_0}, C_{D_α}, C_{Y_β}, C_{Y_ζ}, C_{Y_p}, C_{Y_r}, C_{L_0}, C_{L_α}, C_{L_η}, and C_{L_q}. Finally, the resulting forces are transformed into the body-fixed CS with index b using the transformation matrix $\underline{M}_{be}$, see Eq. 4.4.

$$\vec{F}_b^A = \begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix}_b^A = \begin{bmatrix} -D \\ Y \\ -L \end{bmatrix}_e = \frac{\rho_0}{2} V_a^2 S \, \underline{M}_{be} \begin{bmatrix} C_D \\ C_Y \\ C_L \end{bmatrix}_e \tag{4.4}$$

$$\text{with} \begin{bmatrix} C_D \\ C_Y \\ C_L \end{bmatrix}_e = \begin{bmatrix} C_{D_0} + C_{D_\alpha}\alpha \\ C_{Y_\beta}\beta + C_{Y_\zeta}\frac{\zeta_L+\zeta_R}{2} + (C_{Y_p}p_e + C_{Y_r}r_e)\frac{b}{2V_a} \\ C_{L_0} + C_{L_\alpha}\alpha + C_{L_\eta}\eta + C_{L_q}q_e\frac{l_\mu}{2V_a} \end{bmatrix}_e$$

The calculation of the aerodynamic roll, pitch, and yaw moments (ℓ_b^A, m_b^A, n_b^A) is divided into two components: the static (index s) and the dynamic (index d) coefficients, which are defined in the body-fixed and the experimental CS. The static coefficients consist of C_{ℓ_β}, C_{ℓ_ξ}, C_{m_0}, C_{m_α}, C_{m_η}, C_{n_β}, C_{n_ζ} and the dynamic coefficients consist of C_{ℓ_p}, C_{ℓ_r}, C_{m_q}, C_{n_p} and C_{n_r}. The resulting dynamic moments are transformed into the body-fixed CS and added to the static moments to obtain the resulting aerodynamic moments, see Eq. 4.5.

$$\vec{Q}_b^A = \begin{bmatrix} \ell^A \\ m^A \\ n^A \end{bmatrix}_b = \frac{\rho_0}{2} V_a^2 S \left(\begin{bmatrix} C_{\ell_s} \\ C_{m_s} \\ C_{n_s} \end{bmatrix}_b + \underline{M}_{be} \begin{bmatrix} C_{\ell_d} \\ C_{m_d} \\ C_{n_d} \end{bmatrix}_e \right) \begin{bmatrix} \frac{b}{2} \\ \text{MAC} \\ \frac{b}{2} \end{bmatrix} \tag{4.5}$$

[12] The drag is assumed to increase linearly with the AOA. Using flight measurement data and an UKF, model fitting has shown that the selected equation sufficiently represents the drag coefficient C_D. Within the considered range $V_a \in [11, 20] m/s$, the linear mapping even performs better than a quadratic mapping and simplifies the model. A quadratic dependence was therefore rejected.

$$\text{with} \begin{bmatrix} C_{\ell_s} \\ C_{m_s} \\ C_{n_s} \end{bmatrix}_b = \begin{bmatrix} C_{\ell_\beta}\beta + C_{\ell_\xi}(\xi_L + \xi_R) \\ C_{m_0} + C_{m_\alpha}\alpha + C_{m_\eta}\eta \\ C_{n_\beta}\beta + C_{n_\zeta}\frac{\zeta_L+\zeta_R}{2} \end{bmatrix}_b \text{ and } \begin{bmatrix} C_{\ell_d} \\ C_{m_d} \\ C_{n_d} \end{bmatrix}_e = \begin{bmatrix} (C_{\ell_p}p_e + C_{\ell_r}r_e)\frac{b}{2\,V_a} \\ C_{m_q}q_e\frac{\text{MAC}}{2\,V_a} \\ (C_{n_p}p_e + C_{n_r}r_e)\frac{b}{2\,V_a} \end{bmatrix}_e$$

The motor thrust depends on the rotation speed, which is calculated based on the respective motor power, defined constants, and the parameters c_p, which are available for each of the five motors. For the calculation of the motor thrust, a squared dependency to the rotation speed of the motor is assumed, represented by the parameter $C_{T_{\omega 2}}$ for the lift motors and by the parameters $C_{T_{\omega 2_P}}$ for the pusher motor. In addition, a dependence on the airspeed in the x-direction is specified for the thrust calculation of the pusher motor described by $C_{T_{u2\omega_P}}$ see Eq. 4.6.

$$\vec{F}_b^T = \begin{bmatrix} F_x^T \\ F_y^T \\ F_z^T \end{bmatrix}_b = \begin{bmatrix} \rho_0\pi(\frac{d_P}{2})^4(C_{T_{\omega 2_P}}\omega_P^2 + C_{T_{u2\omega_P}}u^2\omega_P) \\ 0 \\ -\sum_{i=1}^4 T_i \end{bmatrix}_b \tag{4.6}$$

$$\text{with } T_i = \rho_0\pi(\frac{d_{Li}}{2})^4 C_{T_{\omega 2}}\omega_i^2$$

The motor thrust takes effect on the moment equations. Eq. 4.7 describes the effect on the roll and pitch moment (ℓ^T and m^T). The yaw moment (n^T), which results from the difference in rotational speed and direction of the lift motor, is considered with the parameter C_{n_T}.

$$\vec{Q}_b^T = \begin{bmatrix} \ell^T \\ m^T \\ n^T \end{bmatrix}_b = \begin{bmatrix} r_{xy}(T_2 + T_3 - T_1 - T_4) \\ r_{xy}(T_2 + T_3 - T_1 - T_4) \\ C_{n_T}(\omega_3^2 + \omega_4^2 - \omega_1^2 - \omega_2^2) \end{bmatrix}_b \tag{4.7}$$

The resulting forces and moments consist of an aerodynamic, a thrust, and a gravitation term concerning the UAS attitude, which are additively merged, see Eq. 4.8.

$$\vec{F}_b = \vec{F}_b^A + \vec{F}_b^T + \vec{F}_b^G \tag{4.8}$$

$$\text{with } \vec{F}_b^G = \begin{bmatrix} GW g \sin\theta \\ GW g \cos\theta \sin\phi \\ GW g \cos\theta \cos\phi \end{bmatrix}_b \tag{4.9}$$

The resulting moments are given in Eq. 4.10.

$$\vec{Q}_b = \vec{Q}_b^A + \vec{Q}_b^T \tag{4.10}$$

The constants used describe the geometry and moments of inertia of the Scihunter UAS, introduced in Section 4.1. The gravitation constant is defined by $g = 9.81 m/s^2$ and the air density by $\rho_0 = 1.225 kg/m^3$. The calibration of the actuator command is set to be constant and is specified in Appendix A.10.

System dynamics and the system state equations

The first 12 states describe the flight dynamics with the position $\vec{s}_g = [p_N\, p_E\, p_D]_g^\mathsf{T}$, the body-fixed speed $\vec{V}_b = [u\, v\, w]_b^\mathsf{T}$, the rotational rates $\vec{\Omega}_b = [p\, q\, r]_b^\mathsf{T}$, and the attitude $\vec{\Phi}_b = [\phi\, \theta\, \psi]_b^\mathsf{T}$. Wind estimation is added to handle the real flight test data in Subsection 4.4.3 with $\vec{V}_w = [u_w\, v_w\, w_w]^\mathsf{T}$. Five states describe the aileron deflections, the elevator deflection, and the rudder deflections with $\vec{\delta} = [\xi_L\, \xi_R\, \eta\, \zeta_L\, \zeta_R]^\mathsf{T}$. States 21 to 25 describe the power of the respective motors $\vec{P} = [P_P\, P_1\, P_2\, P_3\, P_4]^\mathsf{T}$.

The time derivation of the dynamical state vector $\dot{\vec{x}}_d$ represents the dynamical states used in the physical model based on [SJL16; BAL11]; see Eq. 4.11.

$$\dot{\vec{x}}_d = \begin{bmatrix} \frac{d\vec{s}_g}{dt} & \frac{d\vec{V}_b}{dt} & \frac{d\vec{\Omega}_b}{dt} & \frac{d\vec{\Phi}_b}{dt} & \frac{d\vec{V}_w}{dt} & \frac{d\vec{\delta}}{dt} & \frac{d\vec{P}}{dt} \end{bmatrix}^\mathsf{T} \tag{4.11}$$

The change of position is given in the geodetic CS (index g). It results from $\vec{V}_b$ using the transformation matrix $\underline{M}_{gb}$ and the current wind field described by $\vec{V}_w$ given in the geodetic CS, see Eq. 4.12.

$$\frac{d\vec{s}_g}{dt} = \begin{bmatrix} \dot{p}_N \\ \dot{p}_E \\ \dot{p}_D \end{bmatrix}_g = \underline{M}_{gb} \begin{bmatrix} u \\ v \\ w \end{bmatrix}_b + \begin{bmatrix} u_w \\ v_w \\ w_w \end{bmatrix}_g \tag{4.12}$$

The accelerations affecting the aircraft system consider the Coriolis term and the total forces $\vec{F}_b = [F_x \, F_y \, F_z]_b^{\mathsf{T}}$ resulting from Eq. 4.8 in Eq. 4.13.

$$\frac{d\vec{V}_b}{dt} = \begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix}_b = \begin{bmatrix} rv - qw \\ pw - ru \\ qu - pv \end{bmatrix}_b + \frac{1}{GW} \begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix}_b \tag{4.13}$$

The derivation of the rotational rates results from the moments taking effect on the aircraft system, cf. Eq. 4.10. They consider the current rotational rates, the active moments $\vec{Q}_b = [\ell \, m \, n]_b^{\mathsf{T}}$, and the moments of inertia of the Scihunter UAS in Eq. 4.14.

$$\frac{d\vec{\Omega}_b}{dt} = \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix}_b = \begin{bmatrix} \frac{J_{xz}(J_{xx}-J_{yy}+J_{zz})}{\Gamma} pq - \frac{J_{zz}(J_{zz}-J_{yy})+J_{xz}^2}{\Gamma} qr + \frac{J_{zz}}{\Gamma} \ell + \frac{J_{xz}}{\Gamma} n \\ \frac{(J_{zz}-J_{xx})}{J_{yy}} pr - \frac{J_{xz}}{J_{yy}} (p^2 - r^2) + \frac{m}{J_{yy}} \\ \frac{(J_{xx}-J_{yy})J_{xx}+J_x z^2}{\Gamma} pq - \frac{J_{xz}(J_x-J_{yy}+J_{zz})}{\Gamma} qr + \frac{J_{xz}}{\Gamma} \ell + \frac{J_x}{\Gamma} n \end{bmatrix}_b \tag{4.14}$$

$$\text{using } \Gamma = J_x J_z - J_{xz}^2 \, ,$$

The change of attitude considers the rotational rates with the directional cosine matrix given in Eq. 4.15.

$$\frac{d\vec{\Phi}_b}{dt} = \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix}_b = \begin{bmatrix} p + \tan\theta(q\sin\phi + r\cos\phi) \\ q\cos(\phi) - r\sin\phi \\ \frac{q\sin\phi + r\cos\phi}{\cos\theta} \end{bmatrix}_b \tag{4.15}$$

The wind velocities are assumed to stay constant within an FM. The deflections and the power of the motors stay constant until new commands take effect, see Eq. 4.16:

$$\frac{d\vec{V}_w}{dt} = [0\,0\,0]^{\mathsf{T}} \, , \; \frac{d\vec{\delta}}{dt} = [0\,0\,0\,0\,0]^{\mathsf{T}} \, , \text{ and } \frac{d\vec{P}}{dt} = [0\,0\,0\,0\,0]^{\mathsf{T}} \, . \tag{4.16}$$

The flight dynamics are used to predict the states for the next time step when used by a recursive filter algorithm. The resulting states are input to the sensor model, which is defined below.

Sensors and the measurement equations

For the design of the sensor model, the representation of Brockhaus, Alles, and Luckner is used [BAL11]. The elaboration by Beard and McLain confirms the usability of the flight mechanics and sensor models for small UAS [BM12].

The output of the sensor model $\vec{y}$ includes 34 values to be compared with the sensor measurements introduced in Subsection 4.1.1; see Eq. 4.17.

$$\vec{y} = \left[\vec{s}_g\, \vec{a}_b\, \vec{\Omega}_b\, \vec{\Phi}_b\, \vec{V}_a\, \vec{\delta}\, \vec{P}\, P_{bat}\, \vec{\omega}_n\, \vec{V}_g\right]^{\mathsf{T}} \tag{4.17}$$

The estimation of the current position $\vec{s}_g$, the rotational rates $\vec{\Omega}_b$, and the aircraft attitude $\vec{\Phi}_b$ directly follow from $\vec{x}_d$, cf. Eq. 4.11. The estimation of the measurements of the accelerations $\vec{a}_b^M$ exclusively considers the aerodynamic forces and the thrust taking effect on the aircraft system. Thus, the total forces given in Eq. 4.8 has to be reduced by the gravitational effect, see Eq. 4.18.

$$\vec{a}_b^M = \begin{bmatrix} a_x^M \\ a_y^M \\ a_z^M \end{bmatrix}_b = \frac{1}{GW} \begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix}_b - \underline{M}_{bg} \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix}_g \tag{4.18}$$

The aircraft speed is measured by the ADB considering the aerodynamics in $\vec{V}_a$ and using the GPS sensor, resulting in $\vec{V}_g$. These measurements are represented by $\vec{V}_b$ using the $atan2$ function from Matlab in Eq. 4.19 and the transformation matrix $\underline{M}_{gb}$ from the body-fixed into the geodetic CS in Eq. 4.20:

$$\vec{V}_a = \begin{bmatrix} V_{a_x} \\ \alpha \\ \beta \end{bmatrix}_b = \begin{bmatrix} u \\ atan2(\frac{w}{u}) \\ atan2(\frac{v}{u}) \end{bmatrix}_b \tag{4.19}$$

$$\text{and } \vec{V}_g = \underline{M}_{gb} \vec{V}_b = \underline{M}_{gb} \begin{bmatrix} u \\ v \\ w \end{bmatrix}_b . \tag{4.20}$$

Remember, the pitot tube introduced in Subsection 4.1.1 measures the IAS in body-fixed x-direction. Therefore, $V_{a_x} = u$ is considered the output of the sensor model.

The deflections $\vec{\delta}$ and motor power $\vec{P}$ result from the actual commands $\vec{u}$ and states $\vec{x}$ of the respective value using a PT1-element. In addition, the measurements of the

PMB can be used considering the total power consumption of the battery, see Eq. 4.21.

$$P_{bat} = P_P + \sum_{i=1}^{4} P_i \tag{4.21}$$

The angular velocities describe the motor rotation rate used in Eq. 4.6 for the calculation of the motor thrust. They are directly measured by the ESC as introduced in Subsection 4.1.1 and, therefore, the functional output of the sensor model. Based on [TR06], the following Eq. 4.22 has been derived. The Equation is used to calculate the angular velocities of the pusher and lift motors. The rotor diameter d_i and the rotor area A_i are chosen, respectively. For consistent units, $v_n = 1m/s$ is used.

$$\omega_{i_n} = \frac{\omega_i}{\omega_{max}} \text{ with } \omega_i = \left(\frac{P_i}{\rho_0 A_i c_{p_i}}\right)^{\frac{2}{3}} \frac{2}{d_i v_n} \text{ and } i \in [P, 1, 2, 3, 4]\,. \tag{4.22}$$

The physical model uses inputs, states, parameters, and constants to estimate forces and moments, states, and sensor signals. These are used for analysis and state estimation in the following.

4.4.3. Plausibility analysis and adjustment of the physical model using flight test data

The derived physical model is adjusted using the acquired flight test data of the unmodified UAS from Section 4.3. The resulting physical model is defined as the standard physical model named PM$_0$. It is used for flight simulation, reference, and initialization of the hybrid model in Chapter 5. Further, PM$_0$ serves as the basis for manipulations and simplifications according to the experimental design.

Plausibility is inferred from three steps:

1. the physical model structure is usable to estimate parameter states,
2. the identified parameters are physically meaningful,
3. and PM$_0$ (equipped with the identified parameters) is usable for SIL flight simulations.

To enable a high degree of adaption to the flight test data and to ensure convergence of the parameter states, the bootstrapping method is chosen. It increases the value of the reasonably small amount of flight test data and is a recent method to handle less data in the context of machine learning [Arb+22]. The bootstrapping method reschedules the flown FMs in an alternative order several times in a row. For the joint state estimation, 100 bootstrap samples of the flight test data are processed. This also allows investigation of the convergence behavior of the parameter states over an increased period. A bootstrap sample includes the whole set of flight test data of the unmodified UAS with a random order of the determined FMs (cf. Section 4.3).

The SCmUKF is used for state estimation. For every processing of a bootstrap sample, the filter is equipped with the parameter states and state covariance matrix estimates of the prior bootstrap sample, starting with initial values.

The filter tuning has been elaborated based on systematic testing and data analysis. A list of the set values for the system noise covariance matrix Q_{xx}, the measurement noise covariance matrix R_{yy}, and the initial state covariance matrix P_{xx} is given in Appendix A.7. The initial dynamical states are assumed and set at the beginning of every flight, while the parameter states are initialized once and based on previous work at the FSR given in Appendix A.9.

The system noise Q_{xx} of the parameter states is set to be small ($Q_{xx\,ij} \in [10^{-5}, 10^{-9}]$), which enables a slow but meaningful convergence of the filter solution. The resulting R_{yy} matrix contains significant information about the measurement noise of the used sensors, which is also used for the data acquisition from the flight simulations in section Subsection 4.4.6. Initialized with the sensor characteristics given in Subsection 4.1.1, the entries of the R_{yy} matrix are tailored to the flight test data.

According to the flight phase, parameter states are selected from the parameter state vector $\vec{x}_p$ for adjustment. Aerodynamic parameters and the pusher motor parameters are estimated in FW flight exclusively. In QC flight, the lift motor parameters are estimated. In TR flight, the parameters of the physical model are considered fixed[13].

Thus, 25 parameters are subject to the parameter estimation in FW flight, see Eq. 4.23.

$$\begin{aligned}\vec{x}_p^{FW} = [&C_{D_0}\, C_{D_\alpha}\, C_{Y_\beta}\, C_{Y_\zeta}\, C_{Y_p}\, C_{Y_r}\, C_{L_0}\, C_{L_\alpha}\, C_{L_\eta}\, C_{L_q}\\ &C_{\ell_\beta}\, C_{\ell_\xi}\, C_{\ell_p}\, C_{\ell_r}\, C_{m_0}\, C_{m_\alpha}\, C_{m_\eta}\, C_{m_q}\, C_{n_\beta}\, C_{n_\zeta}\, C_{n_p}\, C_{n_r}\\ &c_{p_P}\, C_{T_{\omega 2_P}}\, C_{T_{u2\omega_P}}]^\mathsf{T}\end{aligned} \tag{4.23}$$

[13] In the case of simultaneously estimating all parameters in the TR flight, observability is not given.

In QC flight the remaining 6 parameters are chosen for estimation, see Eq. 4.24

$$\vec{x}_p^{QC} = [c_{p1}\; c_{p2}\; c_{p3}\; c_{p4}\; C_{T_{\omega 2}}\; C_{n_T}]^\mathsf{T} \tag{4.24}$$

The targeted parameter estimation is realized as a function of the flight phase by adjusting Q_{xx_p} and the pseudo measurements $\vec{m}_p$ of the parameters, cf. Subsection 3.4.1. In the case of QC flight, the parameters concerning the FW flight ($\vec{x}_p^{FW}$) are forced to stay constant. This is achieved by setting the pseudo measurements $\vec{m}_p^{FW}$ equal to the current parameter states $\vec{x}_p^{FW}$. In addition, the entries of $Q_{xx_p}^{FW}$, which correspond to the parameters in Eq. 4.23, are set to be a zero matrix $\underline{0}$. In agreement, $Q_{xx_p}^{QC}$ and $\vec{m}_p^{QC}$ are set according to the respective flight phase, see Table 4.6.

Table 4.6.: Setting of the pseudo measurements $\vec{m}_p$ of the parameter SC and the system noise covariance matrix Q_{xx} according to the current flight phase.

QC	TR	FW
$\vec{x}_p^{FW} = const.$	$\vec{x}_p = const.$	$\vec{x}_p^{QC} = const.$
$\vec{m}_p^{FW} = \vec{x}_p^{FW}$	$\vec{m}_p = \vec{x}_p$	$\vec{m}_p^{QC} = \vec{x}_p^{QC}$
$Q_{xx_p}^{FW} = \underline{0}$	$Q_{xx_p}^{FW} = \underline{0}$	$Q_{xx_p}^{FW} = Q_{xx_{p_{ini}}}^{FW}$
$Q_{xx_p}^{QC} = Q_{xx_{p_{ini}}}^{QC}$	$Q_{xx_p}^{QC} = \underline{0}$	$Q_{xx_p}^{QC} = \underline{0}$

The frequency of the state prediction is set to $f_{prediction} = 100Hz$; The exact frequency measurements are available after flight data preprocessing. The frequency of state updates using the residuals is set to $f_{update} = 1Hz$. In this way, the filter algorithm uses a range of time steps to predict future states based on the physical model without the support of the measurements, which enables further insight into the prediction performance.

For the application of the SCmUKF, uncorrelated sensor measurements and the ergodicity of the flight test data are assumed. Consequently, R_{yy} is a diagonal matrix. Further, it is assumed that the order of flight missions is uncorrelated, and any aging effects are negligible within the TS of the physical experiments.

This allows to change the order of the flight test data and to apply the bootstrap method. The state estimation results in a fixed parameter set. To assess the validity of the parameter set, the convergence behavior is analyzed. To represent the convergence behavior, the state estimation of the ANN weight states is evaluated using the FN (cf. Subsection 2.2.4), see Figure 4.11.

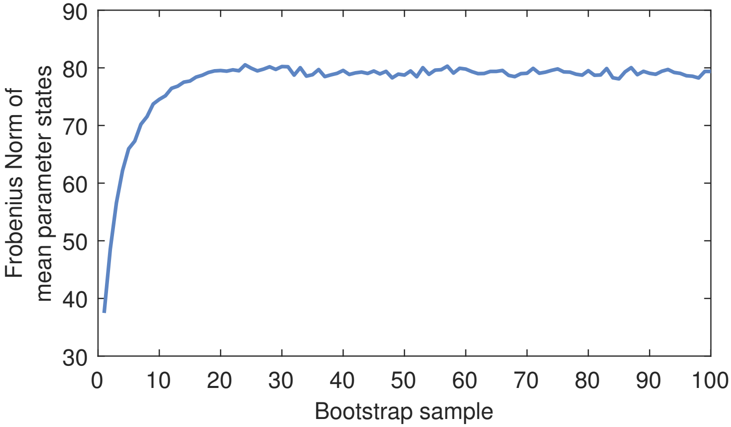


Figure 4.11.: Convergence behavior of the parameter states adjusting the physical model and represented by the FN over 100 bootstrap samples.

The state estimation slowly converges over the first 25 bootstrap samples. Since then, the values of the parameter states have been established within a specific range. The remaining uncertainty is assumed to be at an acceptable level. Thus, the criteria of a stochastic nonlinear system are fulfilled, implying stochastic observability as defined in Subsection 2.2.3.

The resulting parameter set is taken from the mean values of the last bootstrap sample rounded to three significant digits. The parameter set concerning the FW flight is given in Table 4.7.

Besides the TAS, the altitude h is used to constrain the validity of the parameter sets. The altitude h describes AGL at the air density ρ_0. The parameter set concerning the QC flight is given in Table 4.8.

Compared to the initial set of parameters given in Appendix A.9, some of the initially defined parameter values change. In particular, the damping coefficients C_{Lq} and C_{Mq} significantly increase their absolute value.

Table 4.7.: Adjusted parameters of PM$_0$ using the acquired flight test data in FW flight. The parameter set is valid for FW flight at $V_a \in [11, 20]m/s$ and $h \in [75, 95]m$.

C_{D_0}	0.132	C_{L_0}	0.614	C_{ℓ_p}	-4.32	C_{n_β}	0.125
C_{D_α}	0.451	C_{L_α}	3.68	C_{ℓ_r}	0.794	C_{n_ζ}	-0.151
C_{Y_β}	-0.819	C_{L_η}	0.784	C_{m_0}	$-4.61 * 10^{-3}$	C_{n_p}	-0.466
C_{Y_ζ}	0.485	C_{L_q}	25.4	C_{m_α}	-2.44	C_{n_r}	-0.358
C_{Y_p}	0.210	C_{ℓ_β}	-0.226	C_{m_η}	-2.81	c_{p_P}	2.49
C_{Y_r}	1.05	C_{ℓ_ξ}	-0.695	C_{m_q}	-74.7	$C_{T_{\omega 2_P}}$	$2.15 * 10^{-2}$
						$C_{T_{u2\omega_P}}$	$-2.93 * 10^{-2}$

Table 4.8.: Adjusted parameters of PM$_0$ using the acquired flight test data in QC flight. The parameter set is valid for QC flight at $V_a \in [0, 11]m/s$ and $h \in [0, 75]m$.

c_{p_1}	2.26	c_{p_3}	2.24	$C_{T_{\omega 2}}$	$1.33 * 10^{-2}$
c_{p_2}	2.34	c_{p_4}	2.34	C_{n_T}	$-1.41 * 10^{-6}$

In addition, the parameter values are compared to parameter sets of UAS given in the literature. Scale and sign are concluded to be in a reasonable range considering the work from Gäb, Beard et al. and previous publications [Enk19; Gäb12; BM12].

Strikingly, the coefficient C_{M0} is negative and small, leading to instability in stable FW flight. Such a value is also found for the Aerosonde UAS in [BM12]. In this case, the elevator deviation compensates for the resulting pitching moments, as was observed in the flight tests and data considered in this thesis. However, the estimated parameter set contains reasonable values and is therefore considered physically meaningful.

A detailed consideration of the observed flight dynamics and estimated states is conducted in Subsection 4.4.6 when comparing the flight test data to the flight simulation data using PM$_0$. Before, modifications of the physical model are derived and discussed below.

4.4.4. Abstraction of aerodynamic degradation and recovery

The experimental design intends a systematic manipulation of the physical model used for the generation of flight simulation data or within the filter environment, cf. Subsection 4.2.2. In this Section, the manipulation of the physical model used within the simulation environment is considered, leading to the modified physical model PM_{mod}. It provides the basis to explore the ability of the hybrid model to identify the manipulated behavior of the system. Thus, the hybrid model is proven if it is able to learn and predict the degradation of a system and the effects of recovery measures. Therefore, degradation and recovery are abstracted by deriving mathematical formulations. These are used to manipulate PM_0.

Based on the preliminary studies discussed in Section 3.5 and, in particular, the consideration of commercial transport aircraft in [EHH22], three major influences are assumed. They are classified as an abstract representation of first-degree rational curves:

- the permanent degradation over time, → Random constant
- the reversible degradation over time, → Random ramp
- and the recovery effect at a particular time step. → Random jump

The three resulting mathematical abstractions are shown as ergodic sets of curves in Figure 4.12.

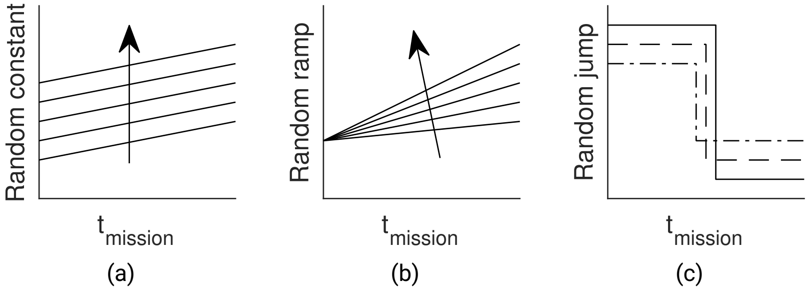


Figure 4.12.: Abstracted parameter behavior considering the set of curves of the random constant (a), the random ramp (b), and the random jump (c), based on [Gel+06].

Concerning ergodicity, the abstracted parameter behaviors are represented as sets of curves, which can be transferred into time behavior considering the total operation time.

A sufficient magnitude of manipulation is defined to ensure visibility and enable the exploration of the hybrid model. In contrast, the complexity of the modified physical model is increased to a tolerable extent. Therefore, the magnitude of manipulation does not necessarily meet the criteria of a realistic use case. A practical use case may involve a slow degradation and less frequent recovery, which is not feasible using the acquired database of nine or 18 FMs in this thesis. Thus, the application to flight data and evaluation of the hybrid model in Chapter 5 are performed to explore the applicability and usability of the approach based on the following abstracted modifications.

For manipulation, the parameters C_{D_0} and C_{L_0} are selected. They represent the independent influences on the aerodynamic drag and lift forces, modified to include the abstracted degradation and recovery effects. While drag increases, lift production is assumed to decrease. To define the manipulation magnitude, the aerodynamic modules introduced in Subsection 4.1.2 are concerned. The following definition is set, including the expected nominal change of drag and lift forces and the flight duration of the executed flight tests.

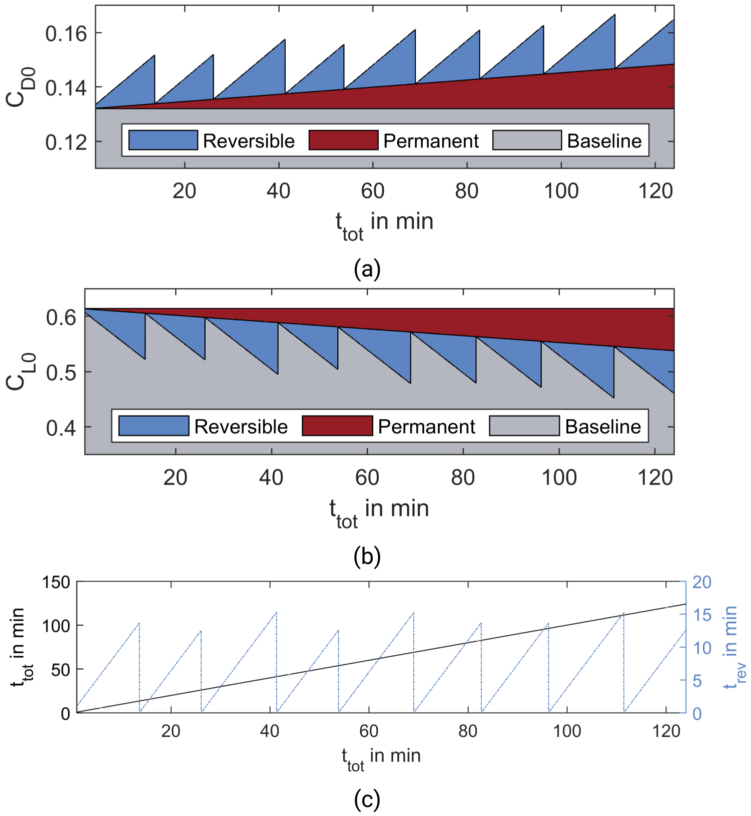
The degradation of the aircraft system aerodynamics is divided into a permanent degradation with a nominal charge of 10 % ($c_{perm} = 0.1$) over 100 minutes ($c_{perm_t} = 6000s$) and into a reversible degradation of 10 % ($c_{rev} = 0.1$) over 10 minutes ($c_{rev_t} = 600s$). The reversible degradation will be fully recovered at the beginning of every flight. The resulting equations of the modified parameter values $C_{D_{0_{mod}}}$ and $C_{L_{0_{mod}}}$ are defined in Eq. 4.25 and Eq. 4.26:

$$C_{D_{0_{mod}}} = C_{D_0} \left(1 + \frac{c_{perm}}{c_{perm_t} t_{tot} + \frac{c_{rev}}{c_{rev_t}}} t_{rev} \right), \tag{4.25}$$

$$C_{L_{0_{mod}}} = C_{L_0} \left(1 - \frac{c_{perm}}{c_{perm_t} t_{tot} - \frac{c_{rev}}{c_{rev_t}}} t_{rev} \right). \tag{4.26}$$

For the implementation, two input values are used: The total time t_{tot} of the flight operations within a TS and the time t_{rev} since the beginning of each TS. The characteristics are represented in Figure 4.13, considering the total time in flight operation.

The aircraft-specific baseline (gray) is a constant value that experiences change through permanent degradation (red) and reversible degradation (blue). The characteristic



(a)

(b)

(c)

Figure 4.13.: Reversible and permanent modification of the drag and lift production in flight operation concerning the aerodynamic coefficients C_{D_0} in (a) and C_{A_0} (b), and the time domain with t_{tot} and t_{rev} in (c).

sawtooth trend results from the occurrence of the recovery effects.

The time data t_{tot} and t_{rev} is used as input data u_{nn} for the ANN being part of the hybrid model, which is scaled with the maximum values $max(t_{tot})$ and $max(t_{rev})$ within the database. The recovery measures are handled as maintenance events, transcribed to time series data using t_{rev}. The representations in Eq. 4.25 and Eq. 4.26 replace C_{D_0} and C_{L_0} in PM_0, leading to the definition of PM_{mod}.

Simon does further research at the FSR, considering different transcriptions of event-based data into time-series data and its effect on ML capabilities [SS21]. In this thesis, the abstracted formulation above is used for the execution of flight simulations concerning the aerodynamically modified UAS in the next Section.

4.4.5. Execution of flight simulations

For the execution of flight simulations, the simulation framework developed in Subsection 4.4.1, the characteristics of the UAS from Subsection 4.1.1, and the flight test planning from Subsection 4.3.1 are used. This way, the same aircraft characteristics and trajectory are implemented for the flight simulations.

The definition of an FaS and emergency procedures is neglected for flight simulations because the safety criterion expires, cf. Appendix A.4. Therefore, the planned flight path remains unchanged for the flight simulations and the variation of the TOW and CA in the respective FM. Flight simulations are executed according to the experimental design using PM_0 (Subsection 4.2.2 and Subsection 4.4.3) and PM_{mod} (Subsection 4.4.4).

Due to the increasing computational effort according to an increased model complexity observed in [EHH22] and pilot experiments within this thesis, the simulation frequency is set to $100Hz$. Lowering the frequency is limited by the design of the SIL environment, where a stimulation frequency of $100Hz$ is needed to maintain stable flight conditions within the flight simulations. Consequently, simulation data appear as frequently as the preprocessed flight test data.

The simulated flight durations and distances are given in Appendix A.8. Unlike the executed flight test, the flight simulation is successfully conducted at a TOW of $4.82kg$ and a CA of $12m/s$. This is attributed to the absence of environmental disturbances such as wind. Similar to the flight test data in Subsection 4.3.3, the flight duration of the performed flight simulations decreases with increased CA, leading to an imbalance of the acquired data analyzed in the next Section.

4.4.6. Data acquisition and analysis

The acquired database contains true states and ideal measurements of the simulated flight missions. The data does not lack information nor has varying sampling rates and sensor errors, as noted in the flight test data in Subsection 4.3.3.

To pretend a slight approximation to reality and for the further use of the flight simulation data, white noise is added to the $N_m = 34$ ideal measurements m_{ideal_k} at timestep k. The standard deviation $\vec{r}_{yy}$ is set according to the observations made in Subsection 4.4.3 listed in Appendix A.11. Using $randn()$ from Matlab Eq. 4.27 results:

$$\vec{m}_{sim_k} = \vec{m}_{ideal_k} + randn(N_m)\vec{r}_{yy} \tag{4.27}$$

$$\text{with } \vec{r}_{yy} = diag\left(\sqrt{R_{yy}}\right). \tag{4.28}$$

The resulting measurements $\vec{m}_{sim_k}$ are fed to the database, which finally fulfills the randomization requirement of the simulation data within the experimental design.

The balance of the obtained flight simulation data is increased, cf. Subsection 4.3.3. However, the data still suffers an uneven distribution of the TAS, see Figure 4.14.

The flight test and the flight simulation data provide a high level of agreement. For comparison, an excerpt of flight and simulation data is chosen considering the design airspeed of CA $= 15m/s$ and the TOW $= 4.39kg$. The data excerpt includes QC, TR, and FW flight covering the figure-of-eight profile without repetition.

Aligned at the beginning of the FW flight, the two resulting flight paths based on the measurements of the flight test (a) and the flight simulation (b) are compared in Figure 4.15.

The flown flight paths appear with high similarity in both the flight test and simulation. Three significant differences are observed concerning the QC flight, the forward transition, and the added noise of the simulation data:

- In the flight test, a higher QC flight speed is observed in horizontal flight, leading to an increased representation of the flown QC flight path. The QC flight in the flight simulation appears to be shorter.
- Based on a detailed examination of the flight altitude given in Appendix C.2, the UAS behavior is compared in forward transition. While transitioning, the UAS loses height in the flight test (from $alt \approx 75m$ to $alt \approx 65m$) and in the flight simulation (from $alt \approx 75m$ to $alt \approx 62m$). This is attributed to the transition

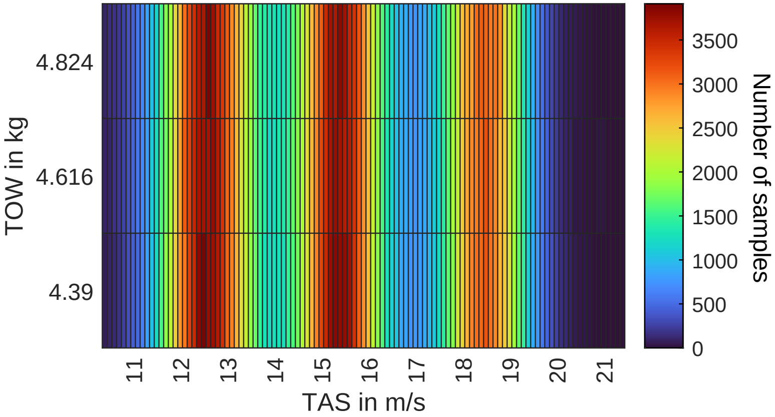


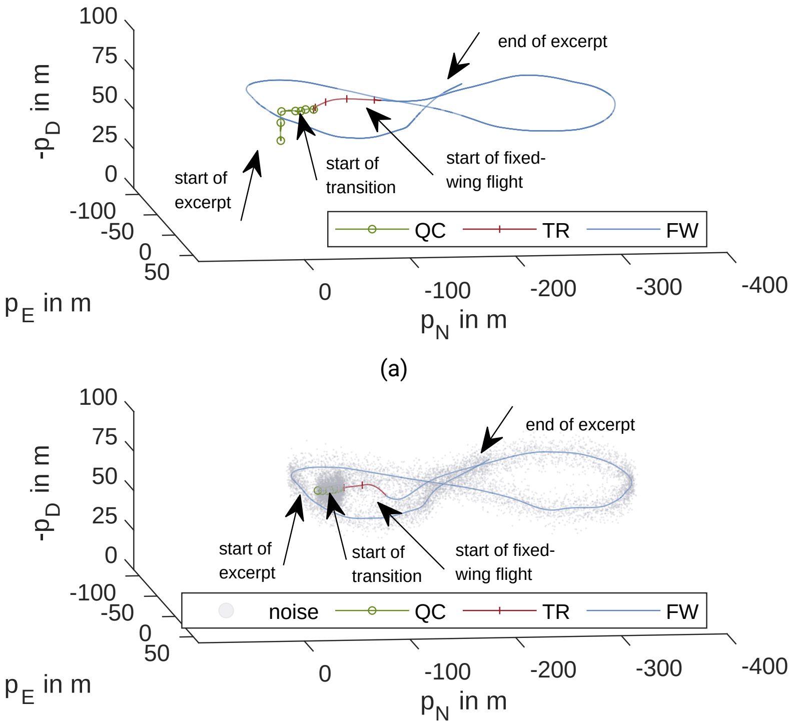
Figure 4.14.: Distribution of the considered flight simulation data in FW flight concerning the take-off weight and true airspeed by frequency of occurrence.

controller used by the FMU. The difference between flight test and simulation is explained by the absence of aerodynamic forces in the simulation environment during TR.

- The noise of the GPS signal observed in the flight test appears to be small. The uncertainty addressed in the covariance R_{yy} in Subsection 4.4.3 and the resulting white noise represented in Figure 4.15b appears to be higher.

The corresponding measurements beginning with $t_{mission} = 0$ at the start of the excerpt of the flight path are considered with the rotational rates and the accelerations below. Starting with the rotational rates in Figure 4.16, similar characteristics within the QC, TR, and FW flight, especially the right and left turn, are observed.

In QC and TR flight, the measurement data of the flight test slightly differ from the data acquired in flight simulation. The flight simulation does not map aerodynamic forces and moments in QC and TR flight nor the influence of environmental disturbances. Thus, the simulation data are less noisy within the two flight phases. Within the transition from TR to FW flight, the values show peaks within a short period before aerodynamic forces and moments are considered.



(a)

(b)

Figure 4.15.: Excerpt of the flight path flown with a CA $= 15m/s$ and TOW $= 4.39kg$ based on flight test data (a) and flight simulation data (b). In (b), ideal measurements are used to represent the flight path, where the additional white noise is represented in light gray.

In particular, the yaw rate r oscillates in QC flight in Figure 4.16f. This is explained by the unstable lateral movement in QC flight and the limited control of the yaw rate using different rotation speeds of the lift motors described by the parameter C_{n_T}, cf. Subsection 4.4.2. In comparison, the oscillations in the flight test data in Figure 4.16e occur with variable amplitude and phase. But especially in the left and right turn within the FW flight, the yaw rate is higher, which affects the recorded accelerations within the flight simulation, see Figure 4.17.

The yaw rate of the simulated UAS leads to high accelerations in the direction of the y-axis Figure 4.17d. Within the flight test data, the accelerations in the y-direction stay in a particular range, see Figure 4.17c.

In the x- and z-direction, the accelerations appear similar in the considered excerpt of flight test and simulation data. Strikingly different are the peaks, which occur within the transition between TR and FW flight. Again, the abrupt transition from non-aerodynamic to aerodynamic flight within the simulation environment is considered for an explanation.

The simulation runs at $100Hz$, which enables the data to be acquired at a sampling rate of $100Hz$ as well. As shown, the chosen frequency is sufficient to represent the flight dynamics in FW flight but is close to limits when representing the QC and TR flight. Thus, the resulting flight conditions remain stable while flight simulation and the low sampling rate favor the computational effort later on.

Remember the physical model used within the simulation environment considers wind in estimating the wind speeds with u_w, v_w, and w_w. Even though the variation of wind is not subject to the performed flight simulations according to the experimental design, wind is part of the state estimation[14] in Chapter 5.

4.5. Conclusions in the field of observability

In conclusion, the implemented simulation environment is assumed to be plausible and valuable for the application and evaluation of the hybrid model in Chapter 5. However, the design of PM_0 and the adjustment using flight test data offer insights into the system behavior and the observability of the system states, including the parameter states. These insights concern: the behavior of the dynamical states, the functionality of the physical model implemented, and the state estimation with respect to the QC, TR, and FW flight of the hybrid Scihunter UAS. The three aspects are discussed together in terms of the joint estimation below.

However, the joint estimation of the dynamical and parameter states of the UAS has revealed challenges. These are assigned to the topics of filter stability and robustness,

[14] The states of the wind speeds add additional degrees of freedom to the filter algorithm. Thus, the success of the state estimation can be approved if the wind speeds are considered to be zero by the filter algorithm.

computational effort, and observability. The stability and robustness of the filter can be ensured by filter tuning of the SCmUKF and the separation of the flight phases discussed in Subsection 4.4.3. The computational effort has to be reduced by considering fewer parameters when applying the hybrid model in Chapter 5.

A selection of parameters can favor their observability. Observability has emerged as a critical factor in the interpretability of the physical model, as it is required by (C) in Subsection 3.2.3. Transferred to the topic of hybrid model building, two expectations are followed:

Expectation 1

> In the case a state of a physical model is observable, it is also observable in the hybrid model, built on the same physical model.

Expectation 2

> In the case a state is observable, it is interpretable.

The joint estimation of the physical model discussed in Subsection 4.4.3 indicates a nonlinear system that lacks deterministic observability. The observability problem is solved assuming slight system noise of the parameter states ($Q_{xx_p} \in [10^{-5}, 10^{-9}]$), which leads to slow but existing convergence of the parameter states. Furthermore, the bootstrapping method is applied to determine a parameter set successfully.

Remind the standard procedure to identify one factor at a time is neglected as the separation theorem is no longer applicable [Ada14; Fis74]. The need for specific data and to specify the procedures of data acquisition taking effect on flight operations contrast requirement (B3) to use operational data exclusively. Therefore, the observability of all dynamic and parameter states is considered at a time.

In the hybrid model, the nonlinearity of the system increases with additional states, which are additively and multiplicatively connected. At the same time, a slight system noise Q_{xx} cannot be chosen since it leads to a slow convergence behavior and restricted adaptability, which contradicts the adaptability requirement (D1).

Consequently, it is convenient to distinguish between deterministic and stochastic observability, which opens the research field of “observability for stochastic nonlinear systems“, as discussed by Chen et al. in [MY18; LB11; van09; Che80]. This topic addresses the filter dilemma discussed by Beyer in [WB18], which is considered with increased attention in the application of the hybrid model and within the discussion of the results in Chapter 5.

4.6. Summary

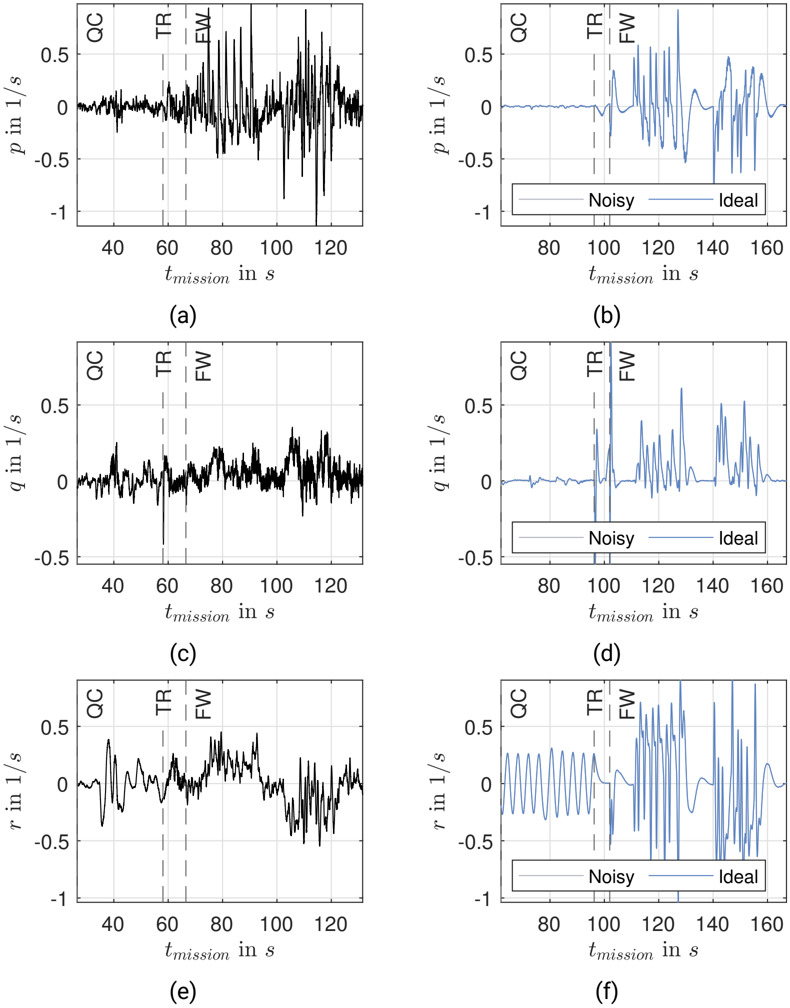
In this Chapter, a UAS is introduced to serve as a flight test platform. Therefore, the UAS design, geometry, and sensors are stated. Included are the aircraft modifications, which allow the change of the TOW to take effect on the aircraft's aerodynamics. The flight test platform, modifications, and knowledge about the aircraft characteristics enable an experimental design, including flight tests and flight simulations.

The goal of the experimental design is to explore the applicability of the hybrid model and its performance in state estimation when applied to flight data. The flight test data covers an aerodynamic modification of the UAS by increasing the wing span and area at the wingtips. The flight simulation data implements abstracted degradation and recovery effects described by time-variant aircraft parameters in PM$_{mod}$. Four TS are provided with a systematic change of the data origin; the UAS used in the flight tests and simulations. In addition, the initial hybrid model structure is planned to be varied according to the TS conducted. Thus, there are systematic deviations specified to be learned by the hybrid model.

The flight test and simulation data are obtained from flight tests and simulations at three different TOWs ($4.39kg$, $4.62kg$, $4.82kg$) and three different CAs ($12m/s$, $15m/s$, $18m/s$). A standard flight path is defined for both the execution of the flight tests and simulations.

For flight simulations, a simulation environment is developed. The simulation environment is based on a SIL framework, a physical model describing the UAS, an adjustment using flight test data leading to PM$_0$, and an abstraction of degradation and recovery leading to PM$_{mod}$. In the SIL framework, the virtual FMU is represented by the same ArduPilot firmware, taking care of the flight controls in the flight tests.

The generated data of flight tests and simulations is considered usable for the application and evaluation of the hybrid model. Furthermore, the use of knowledge about the considered aircraft system is evident for the state estimation task and is taken into account in Chapter 5.



(a)

(b)

(c)

(d)

(e)

(f)

Figure 4.16.: Rotational rates recorded in the flight test (left) and the flight simulation (right) with the roll rate p in (a) and (b), the pitch rate q in (c) and (d), and the yaw rate r in (e) and (f).

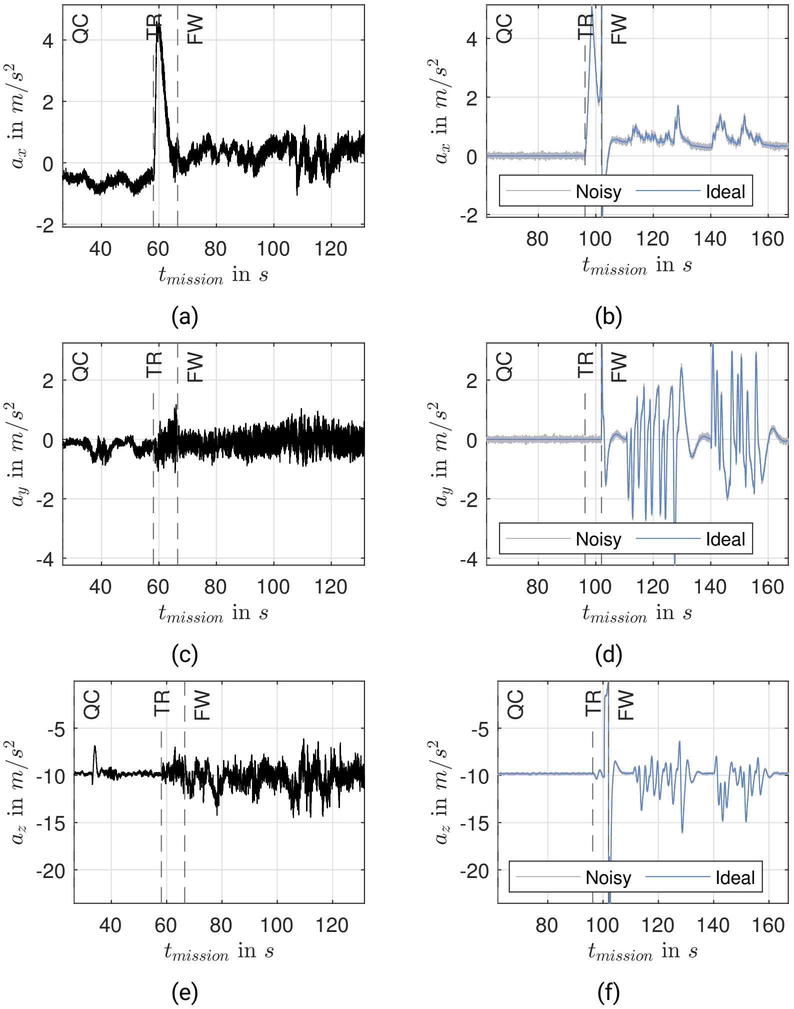


Figure 4.17.: Accelerations recorded in the flight test (left) and the flight simulation (right), in body-fixed x-direction (a) and (b), y-direction (c) and (d), and z-direction (e) and (f).

5. Application and evaluation of the hybrid model, results and discussion

In this chapter, the new hybrid model from Chapter 3 and the experimental design from Chapter 4 are merged. The four TS are conducted to apply and evaluate the hybrid model, focusing on aerodynamic parameters in FW flight.

The initialization of the hybrid model is described in Section 5.1. First, the application to simulated flight data is performed in Section 5.2, followed by the application to flight test data in Section 5.3. Based on the results, the evaluation and discussion are carried out in Section 5.4 and Section 5.5.

The investigation within a single TS is structured as follows:

- **Objective**
  The objective of the TS is to highlight the focus of the investigation considering the state estimation task and the capability of the hybrid model to be assessed.
- **Expectation**
  To provide further background of the investigation, expectations are given that involve results of previous work considering the hybrid model structure or benchmarks from flow simulations.
- **Initialization**
  According to the initialization procedure introduced in Section 5.1, an initialization is performed for the TS respectively with reference to the Appendix.
- **Results**
  Within the results, the convergence behavior, the estimation of the model output, dynamical states, and parameter states are considered in the learning process.
- **Consistency**
  A consistency analysis is provided in addition to evaluating the prediction performance of the converged hybrid model.

5.1. Initialization procedure

The application of the hybrid model, in particular the hybrid model structure and the state estimation algorithm, the SCmUKF, requires initialization, see Figure 5.1.

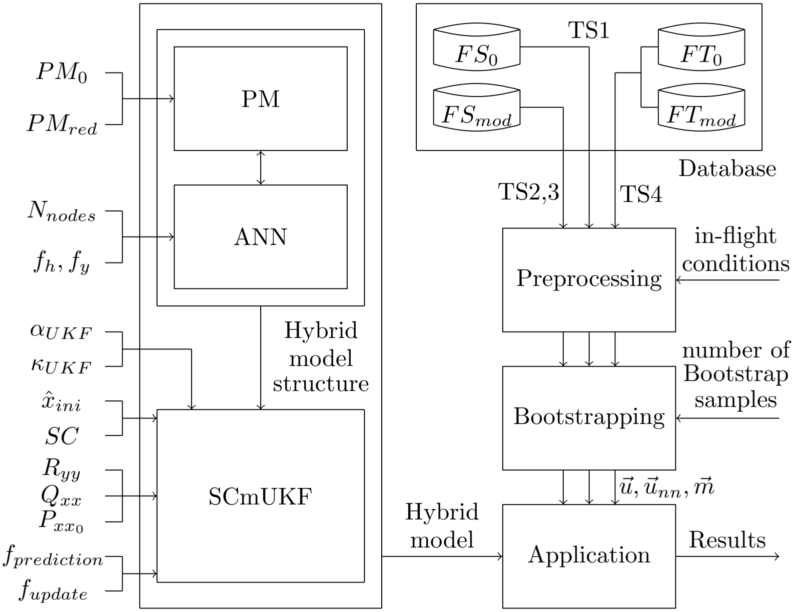


Figure 5.1.: Initialization procedure of the hybrid model when applied to the flight database. The initialization of the hybrid model structure and the filter algorithm is stated on the left. The data of the respective TS is chosen from flight simulations (FS$_0$, FS$_{mod}$) and flight tests (FT$_0$, FT$_{mod}$) on the right.

First, a physical model is chosen to be part of the hybrid model structure. The two choices consist of PM$_0$, cf. Subsection 4.4.2, and of the reduced physical model PM$_{red}$, introduced in Subsection 5.2.3. The ANN[1] is equipped with input, hidden, and output layer nodes, including bias nodes, as well as activation functions for the hidden and output layer (N_{nodes}, f_h, f_y), cf. Subsection 2.3.6.

[1] Remind, the FNN with one hidden layer is considered in this thesis.

The SCmUKF is initialized with the hybrid model structure, the initial states $\hat{x}_{ini}$, SC of the parameters, the covariance matrices Q_{xx}, R_{yy}, and P_{xx_0}, and the filter frequencies $f_{prediction}$ and f_{update}[2]. Further, the SCmUKF is equipped with $\alpha_{\text{UKF}} = 10^{-4}$ and $\kappa_{\text{UKF}} = 3 - N_{\text{UKF}}$ using the number of states N_{UKF} based on [Hay04].

The SC referring to the parameter states and the covariance matrices are adjusted according to the flight phase, see Appendix A.11. This enables the aerodynamic parameter states to be solely estimated within the FW flight and to stay unchanged within QC and TR flight, following the procedure used in Subsection 4.4.3.

The database is preprocessed by manually determining the airtime of the UAS based on the recorded flight data when the UAS is detached from the ground. Thus, the flight data maps in-flight conditions exclusively to avoid numerical issues when ground contact affects the balance of forces[3].

The covariance matrices of the prior FM, as well as the parameter and ANN weight states, initialize the subsequent FM. The initial dynamical state vector $\vec{x}_{d_{ini}}$ is set at the beginning of every flight, including deviations described in the respective TS.

The bootstrapping method is applied to investigate the convergence behavior over a randomized repetition of the recorded flight data scheduled by flights. As will be shown, bootstrapping further enables the ANN weight states to converge but increases computational effort. A TS of 9 flights (or 8 flights in the case of flight test data) is defined as a bootstrap sample based on Subsection 4.4.3.

The hybrid model is applied to the preprocessed and bootstrapped flight data, including the inputs $\vec{u}, \vec{u}_{nn}$, and measurements $\vec{m}$ in the respective TS.

Thus, in each TS, the approach to hybrid model building developed in Chapter 3 is used to build a new hybrid model. The resulting hybrid models depend on the specifications made according to the TS, where they are initialized and trained.

In the following, the ANN estimates selected parameter states as described in Section 3.3. The physical models are implemented based on Subsection 4.4.2 and Subsection 4.4.3. The resulting state vector merges the dynamical $\vec{x}_d$, parameter $\vec{x}_p$, and

[2] Note, the update frequency f_{update} is set for the FW flight. In QC and TR flight, $f_{update} = 100Hz$ is fixed.

[3] To enable the flight simulations, ground contact is considered with an if condition and within the balance of forces, see Appendix B. In case of ground contact during VTOL, inconsistencies occur in the recorded data. These are attributed to numerical issues of the physical model that are considered unrealistic and cause filter instability.

ANN weight states $\vec{x}_{nn}$ as introduced in Eq. 3.2 in Subsection 3.4.2. The joint estimation finally couples the physical model with the ANN, as discussed in Chapter 3 and examined below.

5.2. Application to flight simulation data

The goal of the application of the hybrid model to flight simulation data is to investigate the state estimation performance using the true states known from the simulation environment. The estimation of the model output is additionally compared to the noisy and ideal measurements.

In particular, the performance in estimating the hidden and time-variant parameter states by using an ANN is examined. The parameter states used within the flight simulations and the recorded dynamical states are considered true states and used for comparison.

According to the experimental design in Section 4.2, the parameters are supposed to be:

- constant, in TS1 (Subsection 5.2.1),
- manipulated in terms of degradation and recovery, in TS2 (Subsection 5.2.2),
- manipulated in terms of degradation and recovery, while the physical model is reduced in describing the aircraft aerodynamics, in TS3 (Subsection 5.2.3).

5.2.1. Test series 1: Investigation of the unmodified UAS

In TS1, the constant parameters C_{D_0}, C_{L_0}, and C_{M_0} are subject to the investigation, while the other parameters introduced in Subsection 4.4.2 are considered fixed. Thus, the longitudinal movement of the unmodified UAS is under consideration, with a small subset of three parameters selected.

Objective

The objective is to investigate the hybrid model in estimating three constant parameters initialized with a deviation from the actual value.

Expectation

There is no structural discrepancy designed between the physical model used for simulation and the physical model used within the hybrid model for state estimation. However, compared to the reference physical model PM$_0$, the hybrid model uses a state vector, including ANN weight states. The additional degrees of freedom due to the adjustment of the ANN weight states in the joint state estimation may cause observability problems.

Initialization

Following the initialization procedure from Section 5.1 and the experimental design from Section 4.2, the flight simulation data FS$_0$ is used, which is based on PM$_0$. For state estimation, the hybrid model structure is equipped with the same physical model PM$_0$. Consequently, the dynamic states $\vec{x}_d$ equal the state vector derived in Subsection 4.4.2. The parameter state vector contains $\vec{x}_p = [C_{D_0}\, C_{L_0}\, C_{M_0}]^\mathsf{T}$. These are subject to the estimation using the ANN.

The ANN within the hybrid model is initialized with one input, one hidden, and one output layer, and three input, two hidden, and three output layer nodes, including a bias node in the input and output layer. A linear activation function is chosen to be used in the hidden and the relu66 activation function in the output layer, cf. Subsection 2.3.6. Relu66 is chosen to constrain the output between 0 and 1, which is scaled to the range of parameter values, cf. Subsection 3.3.2. The range of the parameter values for scaling and the implementation of state constraints of the SCmUKF is given in Table A.11. Thus, the state vector $\vec{x}_{nn} = [x_{nn_1} \dots x_{nn_9}]^\mathsf{T}$ contains 9 ANN weights.

The initial dynamical states $\vec{x}_{d_{ini}}$ are known from the SIL simulation environment. They are set to be the true states at the defined starting point after data preprocessing. The initial parameter states $\vec{x}_{p_{ini}}$ is set to be the parameters used in the simulation environment, see Subsection 4.4.3. The initial ANN weight states $\vec{x}_{nn_{ini}}$ are set to be small ($x_{nn_{ini_i}} = 10^{-4}$), except for the weights connecting the bias node of the hidden layer and the parameter states. These are set to be $x_{nn_0}^{OH} = 0.5$ to make the ANN map the mean parameter values using the SC initially, cf. Subsection 3.4.1. In this way, the ANN is initialized to predict the physical parameters correctly.

Finally, the whole state vector $\vec{x}_{ini} = [\vec{x}_{d_{ini}}\, \vec{x}_{p_{ini}}\, \vec{x}_{nn_{ini}}]^\mathsf{T}$ is initialized, and a deviation of 20 % to the whole state vector is added. The initial position $(p_{N_{ini}}, p_{E_{ini}}, p_{D_{ini}})$, the initial body speeds $(u_{ini}, v_{ini}, w_{ini})$, and the initial attitude (ψ) are chosen to be set individually, as given in Appendix A.11.

The measurement covariance matrix R_{yy} is used from Subsection 4.4.3. To account for numerical issues, an increase of 10 % is applied. The entries of the system noise covariance matrix Q_{xx_d} referring to the dynamical states are used from Subsection 4.4.3. The entries of the main diagonal referring to the parameters and ANN weight states are set to be small. The initial state covariance matrix P_{xx_0} is set according to the defined deviations of the initial states. The initial covariance matrices used are given in Appendix A.11.

The prediction frequency of the filter is set to $f_{prediction} = 100Hz$, while measurement updates are provided with $f_{update} = 1Hz$. The maximum number of Bootstrap samples is set to 30 following the convergence behavior observed in Subsection 4.4.3.

The input vector $\vec{u}$ includes the system inputs according to the virtual FMU commands extended by the TOW and the flight phase as described in Subsection 4.4.2. The input to the ANN u_{nn} covers the scaled flight time per mission t_{rev} and the total operation time t_{tot} using the respective maximum value, cf. Subsection 4.4.4.

The 34 measurements are used to create the residuals with the hybrid model estimates and support the state update as introduced in Subsection 2.2.2.

Results

The hybrid model converges in the training process, see Figure 5.2. Within the 30 bootstrap samples, 270 flights are processed, including about 2.2×10^7 data points.

The input layer weight states remain small, and the influence from the input layer nodes is negligible. The hidden layer weight states are significantly adjusted within the first bootstrap sample and slightly change in the further course. Thus, the hidden layer is considered fully converged after about 1.5×10^7 or 20 bootstrap samples.

While converging, both the estimated measurements and states show physically meaningful behavior. Residuals of greater magnitude are observed at the beginning of the learning process of the hybrid model. Since then, the measurement residuals mostly stay in the range of the square root of the estimated covariance $\sqrt{\hat{P}_{ee}}$, the estimated standard deviation defined in Subsection 2.2.2. The behavior of the measurement and state residuals are considered at the beginning and the end of the learning process with the examination of the 9 FMs within one TS.

The second flight of the 30th Bootstrap sample is retrieved for analysis, showing the example of the acceleration in the x-direction in Figure 5.3.

The magnitude of the measurement residuals $\hat{e}$ is similar in QC and TR flight. The residuals are in the range of the estimated covariance. In FW flight, the residuals

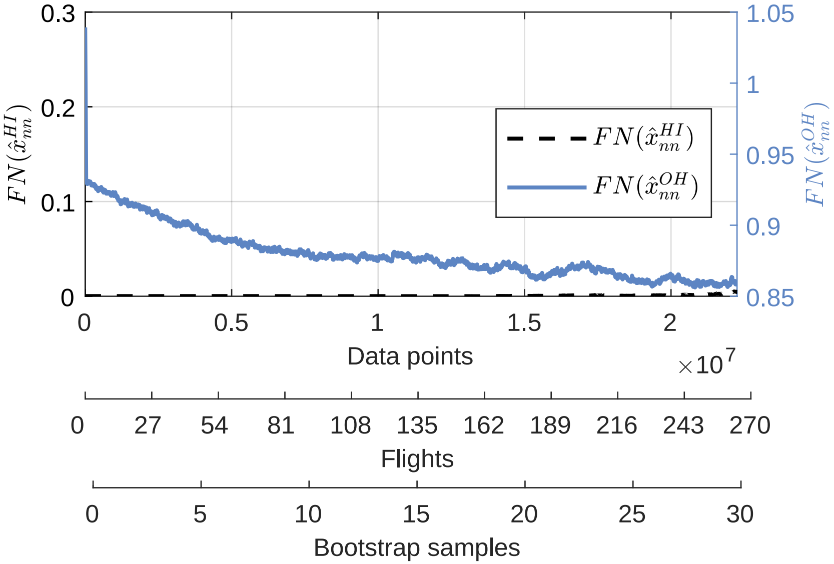


Figure 5.2.: Convergence behavior of the ANN weight states in TS1 showing the FN of the input and hidden layer weights over the processed data points, flights, and bootstrap samples.

slightly increase. $\sqrt{\hat{P}_{ee}}$ also increases in FW flight and decreases before the backward transition enclosing the residuals in FW flight.

The rotational rates behave similarly. Considering the pitch rate q, the residuals are small within a range of $q \in [-0.02, 0.02]$ in rad/s compared to the estimated standard deviation, see Figure 5.4.

Peaks of deviations are observed during the initialization of every FM and in the transition between TR and FW flights, which are captured after a short time. In the FW flight, the residuals of q increase compared to the QC and TR flight.

The estimation of the covariance $\hat{P}_{ee}$ appears to be conservative, which is attributed to the following:

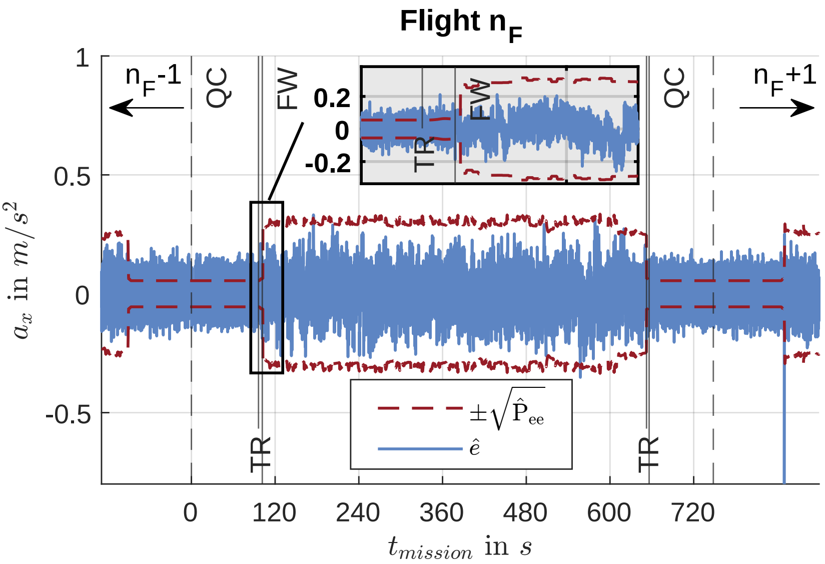


Figure 5.3.: Measurement residuals of the acceleration in the x-direction of flight $n_F = 263$, the second flight of the 30th Bootstrap sample in TS1.

- The physical model within the filter equals the model used for acquired flight simulation data.
- The system noise covariance matrix Q_{xx} remains unchanged according to its use with flight test data in Subsection 4.4.3.

Further examination of the innovation covariance is performed with a numerical analysis of the estimated ($\hat{P}_{ee}$) and actual ($\tilde{P}_{ee}$) covariances, introduced and discussed in Appendix C.3. In addition, the simulation data allows for assuming the actual output covariance $\tilde{P}_{yy}$, using the ideal measurements y_{ideal}. This provides insight into the filter performance, detached from the measurement noise according to Eq. 2.4. A comparison of the estimated and actual covariances yields correlation for most of the signals considered. In some cases, the bias between the estimated covariances appear to be more significant than the actual covariances. This ensures filter stability but offers potential for improvement.

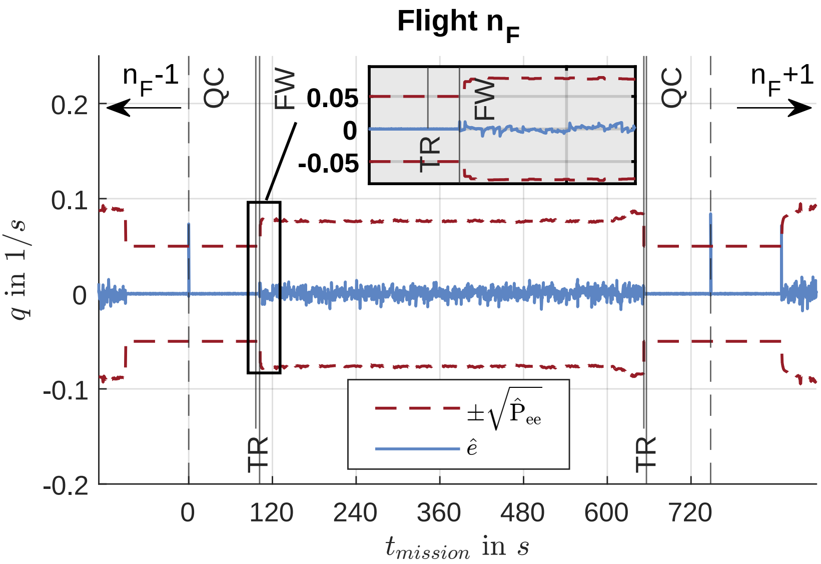


Figure 5.4.: Measurement residuals of the pitch rate of flight $n_F = 263$, the second flight of the 30th Bootstrap sample in TS1.

Thus, the model output, dynamical states, and covariances are sufficiently estimated with further examinations given in Appendix C.3. So far, an increase in filter performance can be expected when adjusting the initial covariance matrices due to filter tuning.

The estimated parameter states show high agreement with the true values, as shown in the example of the drag coefficient C_{D_0} in Figure 5.5. While the parameter states C_{D_0} and C_{L_0} are well estimated within the first data points of the first bootstrap sample, the estimation of C_{M_0} converges more slowly. Consequently, the slow convergence of the FN observed in Figure 5.2 can be attributed to the estimation of C_{M_0}.

Consistency

The hybrid model provides a converged and physically meaningful solution in accurately estimating measurements, dynamical states, and parameter states, based on the adjusted ANN weight states. To investigate the consistency of the resolution, the hybrid model, including the ANN, and its prediction of the parameters is considered fixed in

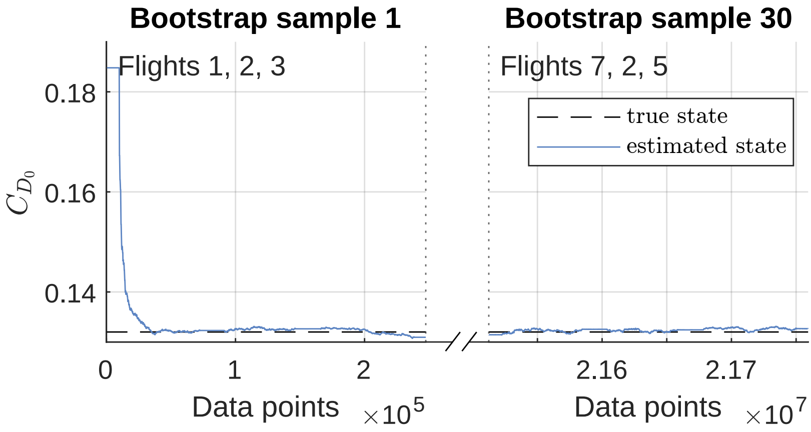


Figure 5.5.: Comparison of the true and estimated parameter state C_{D_0} considering the first three flights in the first and the last Bootstrap sample in the case of constant parameters.

the following. Consequently, state updates concern the dynamical states exclusively: $\vec{x} = \vec{x}_d$.

Thus, the known flight simulation data of the first bootstrap sample of TS1 is processed a second time. For consistency analysis, the initial dynamical states without deviation are considered. The adjusted ANN weight states are considered fixed.

The Normalized Root Mean Squared Error (NRMSE) is used to evaluate the performance of the hybrid model considering the measurements, dynamical states, and predicted parameters normed by the respective maximum value, see Appendix C.3. The normalization of the output measure RMSE_y by the respective standard deviation $\sqrt{R_{yy}}$ is given in the Appendix. Further, the NRMSE metric is exclusively considered in FW flight building a mean value of the 34 measurements, the 25 dynamical states, and the three parameters predicted, see Table 5.1.

The predicted parameters stay in a specific range of the actual values and do not diverge. The error metrics are considered small with $\overline{\text{NRMSE}}_{x_p} = 0.29\text{e-}2$. Thus, the hybrid model can accurately estimate the measurements and dynamical states. Moreover, the

Table 5.1.: Mean NRMSE of the outputs y, dynamical states x_d, and parameters x_p in FW flight resulting from using the hybrid model, including parameter prediction and from the physical reference model in TS1.

Model	$\overline{\text{NRMSE}}_y$	$\overline{\text{NRMSE}}_{x_d}$	$\overline{\text{NRMSE}}_{x_p}$
Hybrid	$3.21\text{e}{-2}$	$0.53\text{e}{-2}$	$0.29\text{e}{-2}$
Physical	$3.21\text{e}{-2}$	$0.53\text{e}{-2}$	0

prediction of the parameters by the ANN using the adjusted and fixed ANN weights is considered sufficient.

NRMSE metrics are obtained using the physical model PM_0 for the same estimation task for reference. In conclusion, the results of the hybrid model are similar to those of the physical model. A physical model is sufficient for constant parameters and even outperforms the hybrid model in terms of simplicity, convergence speed, and adaptability, cf. Subsection 4.4.3).

5.2.2. Test series 2: Investigation of the modified UAS

In TS2, C_{D_0}, C_{L_0}, and C_{M_0} are considered time-variant, while the other parameters introduced in Subsection 4.4.2 are fixed. Similar to the investigation in TS1, the longitudinal movement is investigated while the modified UAS is under consideration.

Objective

The objective is to investigate the hybrid model in estimating three time-variant parameters initialized with a deviation from the actual value.

Expectation

In this TS, a structural discrepancy between the physical model used for simulation and the physical model used within the hybrid model occurs for state estimation. The distinction is expected to be learned and mapped by the hybrid model.

Initialization

The flight simulation data FS_{mod} based on PM_{mod} is used to initialize the structural discrepancy, while the physical model within the hybrid model remains PM_0.

Further, the initialization of TS2 remains unchanged compared to the initialization of TS1. Consequently, the system noise covariance matrix Q_{xx} is intentionally kept to discuss the performance of the hybrid model independently from filter tuning.

Results

The hybrid model converges in estimating time-variant parameters, see Figure 5.6.

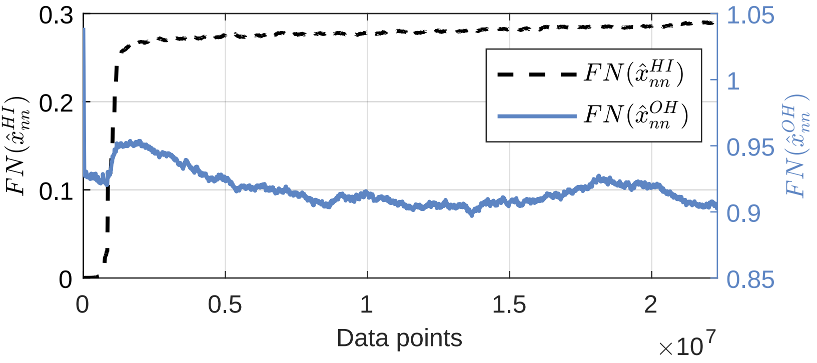


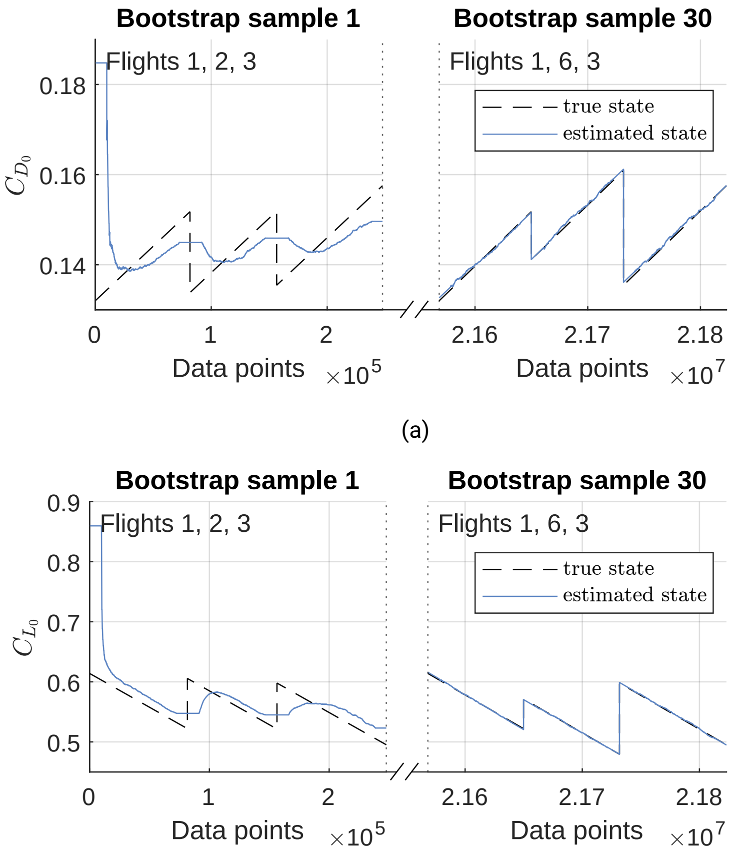
Figure 5.6.: Convergence behavior of the ANN weights in TS2 showing the FN of the input and hidden layer weights over the processed data points.

After the second bootstrap sample, the input layer weight states are adjusted to be completed after 0.2×10^7 data points. Minor adjustments follow, which form a trend of the FN having a low gradient in an acceptable range. The hidden layer weight states remain in a specific range of FN $\in [0.9, 0.93]$ after about 0.5×10^7 data points. According to the stochastic nature of the estimation problem, the solution is considered converged.

In the case of a converged solution, the measurement and state residuals and their associated covariances show a similar behavior compared to the results provided in Subsection 5.2.1.

The hybrid model estimates the modified parameter states accurately, see Figure 5.7.

Both the increasing drag in Figure 5.7a and decreasing lift coefficients in Figure 5.7b and their recovery recurring every FM can be determined. At the beginning of the learning process, the estimated states roughly map the behavior of the true states



(a)

(b)

Figure 5.7.: Comparison of the true and estimated parameter states C_{D0} (a) and C_{L0} (b) considering the first three flights of the first and the last bootstrap sample in the case of modified parameters in TS2.

without capturing the recovery (within the first Bootstrap sample). The recovery can be assessed accurately after the input layer weight states converge, see Figure 5.6. At the same time, C_{M_0} provides a constant estimate, with a negligible influence coming from the input layer of the ANN.

Consistency

Finally, the learned behavior of the time-variant parameters is investigated in terms of consistency. The following investigation further approves the ability of the hybrid model to predict the future impact of degradation and recovery in the considered use case. Based on the procedure in TS1, the ANN weights are considered fixed. These and the parameters, predicted by the ANN, are not part of the state vector that undergoes state updates.

Based on an initialization without state deviation and the normalization provided in Subsection 5.2.1, the mean NRMSE of the measurements, dynamical states, and parameters are given below, see Table 5.2.

Table 5.2.: Mean NRMSE of the outputs y, dynamical states x_d, and parameters x_p in FW flight resulting from using the hybrid model, including parameter prediction and from the physical reference model in TS2.

Model	$\overline{\text{NRMSE}}_y$	$\overline{\text{NRMSE}}_{x_d}$	$\overline{\text{NRMSE}}_{x_p}$
Hybrid	3.21e−2	0.52e−2	0.34e−2
Physical	3.98e−2	2.05e−2	9.41e−2

In conclusion, the hybrid model can accurately predict the hidden, modified, and time-variant parameters by successfully combining the physical model and the ANN. Regarding time-variant parameters or modifications of the UAS at hand, the hybrid model outperforms the physical model in accuracy and interpretability. The physical parameters are sufficiently predicted. Measurements and dynamical states are estimated more accurately than the physical model.

5.2.3. Test series 3: Investigation of the modified UAS with a reduced filter model

In TS3, the six coefficients C_D, C_Y, C_L, C_ℓ, C_m, and C_n are initialized as constant parameter states and without the derivatives describing a dependency on AOA, SSA, deflections, or rotational rates. Consequently, the complexity of the physical model is reduced in describing the aerodynamics of the UAS, leading to the definition of PM$_{red}$ below.

Objective

The objective of the hybrid model is to describe and learn the aerodynamics of the UAS, including the abstracted degradation and recovery. The resulting model is examined in estimating the measurements, dynamical states, and parameters of the reduced physical model. The investigation includes the recursion of dynamical states and virtual FMU inputs to be part of the input layer of the ANN, as intended in Chapter 3.

Expectation

A decreased accuracy in state estimation is expected when using the reduced physical model PM$_{red}$, while it offers less insight into the physics of aerodynamic flight. However, with an increased learning rate of f_{update}, the hybrid model is expected to learn the aerodynamic behavior of the UAS in FW flight using relevant inputs and dynamical states.

Initialization

The initialization is based on TS1 and TS2. Similar to TS2, the flight simulation data of the modified UAS FS$_{mod}$ is chosen, while a reduced physical model PM$_{red}$ is considered within the hybrid model. In PM$_{red}$, Eq. 4.4 and Eq. 4.5 are considered simplified.

Consequently, the state vector $\vec{x}_d$ still describes the aircraft system dynamics as described by PM$_0$. However, the parameter state vector considers the aerodynamic coefficients constant: $\vec{x}_p = [C_D \, C_Y \, C_L \, C_\ell \, C_m \, C_n]^\mathsf{T}$. These are subject to the ANN.

To compensate for the lack of aerodynamic description in FW flight, the scope of the ANN is increased. Including bias nodes, the ANN holds 22 input, 16 hidden, and 6 output layer nodes, resulting in 426 ANN weight states. The linear activation function within the hidden and the relu66 activation function within the output layer are maintained.

The input vector to the ANN contains $\vec{u}_{nn} = [t_{rev} \, t_{tot} \, n_{FM} \, \vec{u}^c_{nn} \, \vec{u}^{(x)}_{nn}]^\mathsf{T}$ with the number of the respective FM n_{FM}, the commanded deflections $\vec{u}^c_{nn} = [\xi^c_L \, \xi^c_R \, \eta^c \, \zeta^c_L \, \zeta^c_R]^\mathsf{T}$, and the system states of the previous time step $\vec{u}^{(x)}_{nn} = [u \, v \, w \, p \, q \, r \, \phi \, \theta \, \xi_L \, \xi_R \, \eta \, \zeta_L \, \zeta_R]^\mathsf{T}$. These are selected to support the ANN in estimating the UAS aerodynamics. In addition, the dynamical states are scaled within the input layer of the ANN using the respective maximum value for reference, see Appendix C.5.

The initial states are set similarly to TS2 except for the parameter states. The parameter states of the reduced model are initially set to be constant and based on the observation of the aerodynamic coefficients made in TS2. They equal the reference values given in Appendix C.5.

The covariance matrices correspond to those chosen for TS1 and TS2, except for $Q_{xx_{nn}}$. Entries of the system noise covariance corresponding to the ANN weight states are increased to a value of $Q_{xx_{nn_{ij}}} = 2.5$e-7 in FW flight. In addition, the prediction and update frequencies of the filter are set to $f_{prediction} = 100Hz$ and $f_{update} = 100Hz$ to allow a stable filtering process[4].

The 34 measurements of the sensor model described in Subsection 4.4.2 are still used. Based on the measurements, the residuals are created to support the state update within the hybrid model.

Results

The hybrid model approximates the UAS aerodynamics in FW flight. With the increased learning rate, the training process finally converges. In the training process the hybrid model sufficiently estimates measurements and states including parameter states' prediction. Convergence is achieved after about 2.1×10^7 data points, corresponding to 252 flights and 28 bootstrap samples, see Figure 5.8. Thus, the time the algorithm needs to converge increases compared to TS1 and TS2.

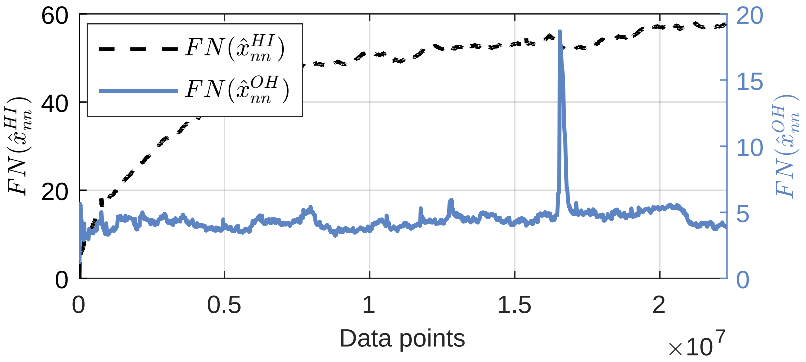


Figure 5.8.: Convergence behavior of the ANN weights in TS3 showing the FN of the input and hidden layer weights over the processed data points.

[4] Note, the reduced physical model is not capable of sufficient state estimation. Therefore an increased update frequency is necessary.

The converged solution includes a population of the weighting matrices with non-zero weight states connecting the input and hidden and the hidden and output layers. Consequently, dependencies between the full order of the input and the output of the ANN exist. In addition, the training process revealed some instability as the FN of the hidden layer weight states peaks after about 1.7×10^7 data points, which is considered detailed in the following.

While learning, estimates of the measurements are adjusted with a frequency of $f_{update} = 100Hz$, which results in small residuals and adequate estimation of the measurement covariance matrix $\hat{P}_{ee}$, see Figure 5.9.

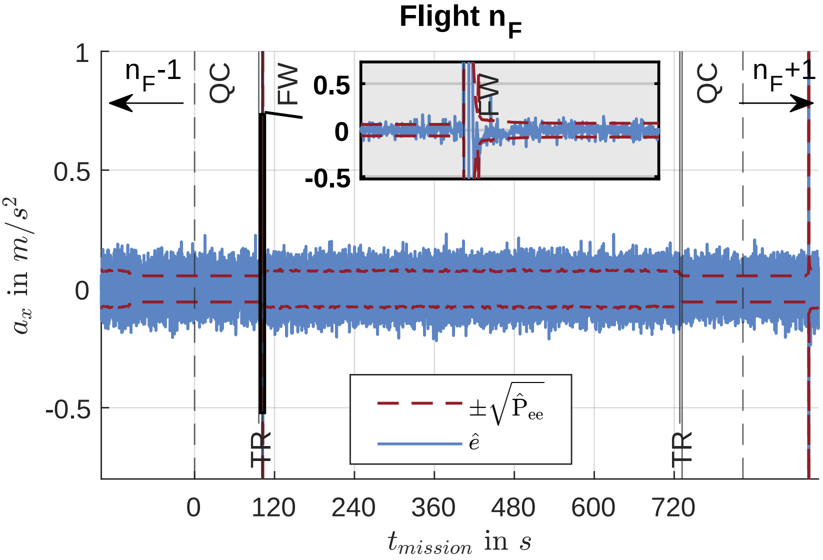


Figure 5.9.: Measurement residuals of the acceleration in the x-direction of flight $n_F = 263$, the second flight of the 30th Bootstrap sample in TS3.

The magnified view of the transition between TR and FW flight shows an issue that occurs frequently during the transition into FW flight. At the beginning of the FW flight, the aerodynamics start to take a sudden effect on calculating the balance of forces and moments. As a result, the estimation of the measurements and states significantly diverge, which is captured in the further course.

The ANN within the hybrid model does not allow for a differentiated view of the influences on the six coefficients C_D, C_Y, C_L, C_ℓ, C_m, and C_n. Therefore, the effects on the parameters describing aerodynamic drag, side, and lift forces and aerodynamic roll, pitch, and yaw moments can not be interpreted further. Dependencies occur between the input layer nodes and the parameter states, which are not intended to exist. These are excluded within the simulation data based on the modified physical model PM$_{mod}$. However, as an example, rotational rates now have an impact on the drag coefficient.

Therefore, the validity of the converged solution has to be considered critically. To visualize and possibly prevent such unexpected dependencies in future work, alternative ways must be found, which may involve the definition of subnets[5].

However, a single ANN is considered in this thesis to describe the full aerodynamics in both longitudinal and lateral motion. Based on the used simulation data, the predicted parameter states correspond to the true parameter values. Similar to TS2, the absolute values of the parameter states are considered to visualize and assess the effect of degradation and recovery, see Figure 5.10.

The increased learning rate and the adjustment of Q_{xx} increase the adaptability of the hybrid model. As a result, the model adapts the coefficients C_D, C_Y, and C_L in the first bootstrap sample. More minor deviations occur in the first bootstrap sample at the beginning of every flight, which happens to be fully captured in bootstrap sample 30. While Figure 5.10 represents the drag coefficient C_D as an example, similar C_Y and C_L behavior is observed.

In TR and QC flight, the parameter state estimation is paused. Here, the ANN still delivers estimates that drift into the SC and are assumed invalid. The deviations are captured at the beginning of the FW flight.

Besides the coefficients of the translational motion, the coefficients C_ℓ, C_m, and C_n describing the rotational movement caused by aerodynamic moments are accurately estimated. For example, the coefficient of the pitching moment C_m is shown in Figure 5.11.

The rotational coefficients adapt more slowly than the translational coefficients but finally agree with the true values in bootstrap sample 30.

[5] The definition of multiple coexisting ANNs describing selected subsets of parameters allows an explicit assignment of input layer nodes to the corresponding parameter subsets. The applicability of such a model structure is successfully tested in preliminary experiments and requires further research.

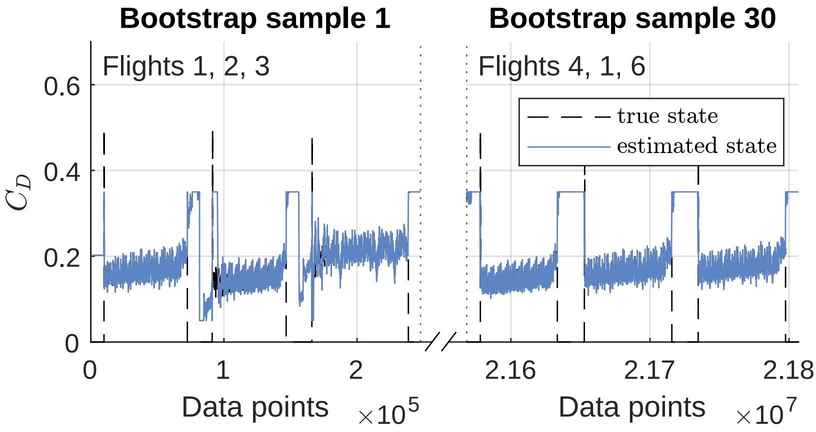


Figure 5.10.: Comparison of the true and estimated parameter states of C_D considering translational motion in the first three flights of the first and the last Bootstrap sample in the case of modified parameters in TS3.

The representation of the absolute values does not provide insight into the degradation and recovery of the system as intended. Compared to TS2, where the impact of degradation and recovery is sufficiently estimated, and the hybrid model provides access to the modified parameters, the influences of degradation and recovery are not observed in TS3.

The parameters C_D, C_Y, C_L, C_ℓ, C_m, and C_n significantly change according to the various flight conditions in the standard FM. Thus, the operation data and modification effect cannot be identified within the learning process.

Consistency

To further investigate the performance of the hybrid model in TS3, the converged solution of the ANN weight states and the usability in assessing modifications are analyzed. Based on the procedure used in TS1 and TS2, the ANN weights are fixed in delivering the parameter predictions. The dynamical states are subject to state updates through the filter algorithm exclusively.

For comparison, the PM$_{red}$ is used. With the initialization used in TS1 and TS2, PM$_{red}$

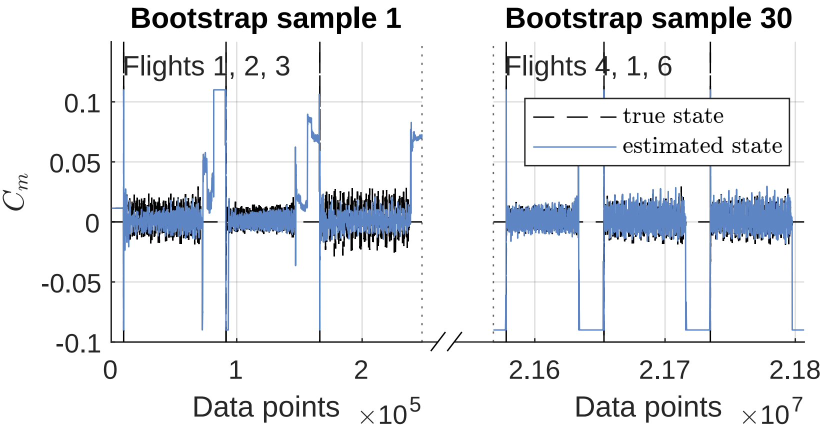


Figure 5.11.: Comparison of the true and estimated parameter states of C_m considering rotational motion in the first three flights of the first and the last Bootstrap sample in the case of modified parameters in TS3.

leads to filter instability, which cancels the state estimation. Therefore, the frequency of the state updates $f_{update} = 100Hz$ is adjusted using PM$_{red}$.

With the application of the converged hybrid model, the state update frequency of $f_{update} = 100Hz$ is also required to maintain filter stability. Instabilities are attributed to physically inconsistent state prediction. In conclusion, the hybrid model cannot intercept the insufficiency of PM$_{red}$.

Even though the hybrid model approximates the physical parameters, resulting in a 4-times lower NRMSE than PM$_{red}$ using constant parameters; the performance is rated low. Consequently, the estimation of the dynamical states and measurements suffers, see Table 5.3.

However, the performance in estimating the output and dynamical states using the hybrid model is lower than the performance of the physical model.

When examining the individual error metrics of the parameters, higher deviations of the parameter value C_D and C_L appear, describing an inadequate representation of the aerodynamic drag and lift forces, see Appendix C.5. The coefficients C_Y, C_ℓ, C_m,

Table 5.3.: Mean NRMSE of the outputs y, dynamical states x_d, and parameters x_p in FW flight resulting from using the hybrid model, including parameter prediction and from the physical reference model in TS3.

Model	$\overline{\text{NRMSE}}_y$	$\overline{\text{NRMSE}}_{x_d}$	$\overline{\text{NRMSE}}_{x_p}$
Hybrid	$7.25\text{e}{-2}$	$21.84\text{e}{-2}$	$69.97\text{e}{-2}$
Physical	$4.60\text{e}{-2}$	$7.71\text{e}{-2}$	$272.77\text{e}{-2}$

and C_n are predicted more accurately compared to PM_{red} leading to an error metric of $\overline{\text{NRMSE}}_{x_p} \approx 0.7$ given in Table 5.3. However, the prediction of C_D and C_L appears insufficient and is therefore considered responsible for the limited estimation quality of the hybrid model.

Contrary to the expectation, the hybrid model insufficiently learns the aerodynamic behavior of the UAS in FW flight. One reason for the discrepancies is the choice of the system noise covariance matrix Q_{xx}, leading to poor estimations of C_D and C_L. The entries of Q_{xx} concerning the ANN weight states are increased in value compared to TS1 and TS2 to enable an increased adaption speed and convergence within the 30 bootstrap samples. However, the increase in adaptability results from a decrease in filter quality. The rise in the value of Q_{xx} seems to correspond with a loss of stochastic observability as the uncertainty of the system state prediction is too to high.

In return, using the ANN with the increased number of nodes in TS3 significantly increases the computational effort. Therefore, the database provided for learning cannot arbitrarily be increased to account for less adaptability. Compared to TS1 and TS2, where the time to compute the 2.23×10^7 data points of the database is about 3 hours, the computational time in TS3 is increased to about 6 days using the same computer system. The reason is the high number of states, which ranks $N = 457$ states in the case of TS3, compared to $N = 37$ states in TS1 and TS2.

In conclusion, the choice of the system noise covariance matrix, the scope of the database, and the computational efficiency affect the hybrid model. Further research is required to fully discover the influence of filter tuning on the hybrid model. However, using the hybrid model in TS3 revealed the significance of a sufficient physical model initializing the hybrid model structure, which is subject to the evaluation performed in Section 5.4.

5.3. Application to flight test data

Applying the hybrid model to flight simulation data delivered insights into estimating the actual system behavior. Accurate flight test data further elaborates on the real object to be mapped by a digital object. The definition of the experimental design in Section 4.2 provides flight test data using the same FMU algorithm for flight controls and the exact flight path. Therefore, the following considerations are not affected by differences in the flight operations compared to the flight simulations.

The flight test data involves the UAS behavior recorded by sensors and influenced by environmental disturbances in the real world. In addition to the unmodified UAS used in Subsection 4.4.3, the modified UAS is part of the consideration. Finally, applying the hybrid model to flight test data allows the investigation of the proposed hybrid model's applicability, usability, and limits. These attributes are subject to the evaluation in Section 5.4.

5.3.1. Test series 4: Investigation of the unmodified and modified UAS

In TS4, the flight test data concerning the unmodified UAS, FT_0, and the modified UAS, FT_{mod}, are merged using a binary attribute to decide whether a modification is applied ($u_{nn} = 1$) or not ($u_{nn} = 0$). Such a behavior corresponds to a random jump discussed in Subsection 4.4.4.

Based on the results obtained in the previous Section, the PM_0 is used in this TS to describe the UAS aerodynamics sufficiently. The aerodynamic modification of the wing tips introduced in Subsection 4.1.2 is expected to impact the UAS drag and lift in FW flight [Enk21]. Therefore, the parameters C_{D_0} and C_{L_0} are selected for investigation below. Consequently, the hybrid model trained in TS4 considers the fully available physical knowledge extended by an ANN concerning flight test data.

Objective

The objective is to identify a dependency of C_{D_0} and C_{L_0} on the employed aerodynamic modification by using a binary attribute to inform the ANN of the current modification. Thus, the ability to continuously adapt to upcoming technical modifications is to be investigated.

Expectation

A physically meaningful adaptation of the drag and lift coefficient is expected according to the employed aerodynamic modification. C_{D_0} and C_{L_0} are expected to stay unchanged in the unmodified case and slightly increase by 1-6 % in drag and 3-12 % in lift production in the modified case, based on flow simulations in [Enk21] introduced in Subsection 4.1.2.

Initialization

The considered flight test data consists of FT$_0$ and FT$_{mod}$ describing flight tests with the unmodified and modified UAS. The data include different CA$s \in [12, 15, 18]\, m/s$, TOW$s \in [4.39, 4.55, 4.62, 4.78, 4.82, 4.99]\, kg$, and flight data concerning aerodynamic modifications with a weight of $0.16kg$. In total, 16 flight tests are considered, 8 with and 8 without aerodynamic modification. Flights at a CA of $12m/s$ and a TOW of $4.82kg$ or $4.99kg$ with modification are unavailable, as discussed in Section 4.3.

Based on the first three TS, using flight simulation data, the hybrid model is initially equipped with the basic physical model PM$_0$. The dynamical state vector is therefore given in Subsection 5.2.1. The parameter state vector involves $\vec{x}_p = [C_{D_0}\, C_{L_0}]^{\mathsf{T}}$, estimated by the ANN.

The ANN is kept small with 2 input, 2 hidden, and 2 output layer nodes, including a bias node in the input and hidden layer each. The chosen activation functions equal the setup used previously in Section 5.2. The state vector $\vec{x}_{nn} = [x_{nn_1} \dots x_{nn_6}]^{\mathsf{T}}$ contains 6 ANN weights.

The initial states are based on measurement data and assumptions without considering the additional deviations, cf. Subsection 5.2.1. The measurement covariance matrix R_{yy}, the system noise covariance matrix Q_{xx}, and the initial state covariance matrix $P_{xx_{ini}}$ equal the setup used with simulation data in Section 5.2 as described in the initialization of TS1, see Appendix A.11.

The prediction frequency of the filter is set to $f_{prediction} = 100Hz$, while measurement updates are provided with $f_{update} = 1Hz$. The number of bootstrap samples is set to 30.

The input vector $\vec{u}$ includes the system inputs according to the FMU commands extended by the TOW and the FM described in Subsection 4.4.2. The input to the ANN, $\vec{u}_{nn}$, consists of the binary attribute describing the unmodified or modified state of the UAS. Additional inputs to the ANN within the hybrid model structure are not provided.

The measurements of the 34 sensor signals are used for training, cf. Subsection 4.1.1.

Results

The hybrid model produces comparable results to PM$_0$ used in Subsection 4.4.3. This is traced back to the initialization using PM$_0$ and the flight test data of the unmodified UAS. Differences consist in the employment of FT$_{mod}$, including u_{nn}, which is used to address the impact of the aerodynamic modification of the UAS on the selected parameters C_{D_0} and C_{L_0}.

Data analysis reveals divergence of ANN weight states during learning, see Figure 5.12. The filter algorithm continuously increases ANN weighs in value affecting the hidden layer ($\hat{x}_{nn}^{HI}$). In the same time, the hidden layer weights affecting the output layer ($\hat{x}_{nn}^{OH}$) converge. A behavior that has not been observed using simulated flight data. Using the real flight test data eventually requires an adjustment of the initial covariance matrices but is not considered the only reason.

In fact, the structure of the ANN can potentially cause divergence. In particular, the bias nodes in both the input and hidden layers are connected to the output layer via the respective states x_{nn_i}. Consequently, the observability of the ANN weight states can be negatively affected, resulting in the divergence observed in Figure 5.12.

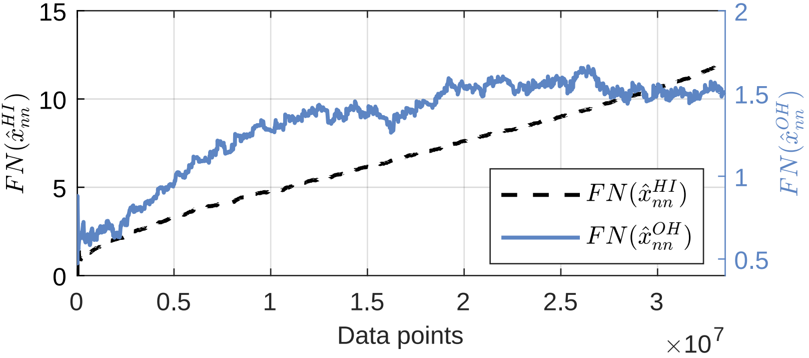


Figure 5.12.: Convergence behavior of the ANN weights in TS4 showing the FN of the input and hidden layer weights over the processed data points.

However, the resulting parameters estimated by the ANN converge within the first bootstrap sample. To further explore their convergence behavior, the mean value of the two parameters, and their standard deviation (uncertainty) according to the respective bootstrap sample are given in Figure 5.13. The uncertainty of the parameter estimation is attributed to the stochastic nature of the estimation problem. With $C_{D_0} \in [0.1, 0.2]$ in Figure 5.13a and $C_{L_0} \in [0.55, 0.75]$ in Figure 5.13b, significant uncertainty is considered. The mean value instead retains its value once it has converged.

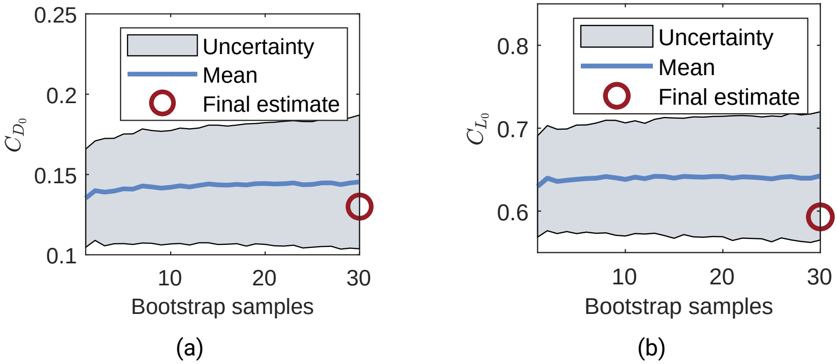


Figure 5.13.: The sharklet of a A321 Neo civil transportation aircraft (a) and the application of sharkskin on an aircraft fuselage (b).

Therefore, the lack of convergence is attributed to the unobservable ANN weight states. Further research is required to identify the impact of the initialization and the ANN structure on the hybrid model and its convergence behavior.

Nevertheless, the hybrid model provides a sufficient estimation of the measurements. Compared to the investigation of flight simulation data in Section 5.2, the residuals of the flight test data appear to be slightly higher; see Appendix C.6.

However, the estimation of $\hat{P}_{ee}$ corresponds to $\tilde{P}_{ee}$; see Appendix C.6. Consequently, innovation and output covariance are also sufficiently estimated.

Considering the measured acceleration of the UAS in body-fixed x-direction, a bias in QC flight is observed, recovered in TR flight, see Figure 5.14. The residuals recorded in FW flight show a slight influence of the flown trajectory but are overall characterized by Gaussian noise.

Similarly, the rotational rate q is estimated in FW flight while the estimation performance in QC and TR flight allows minor residuals without bias, see Figure 5.15.

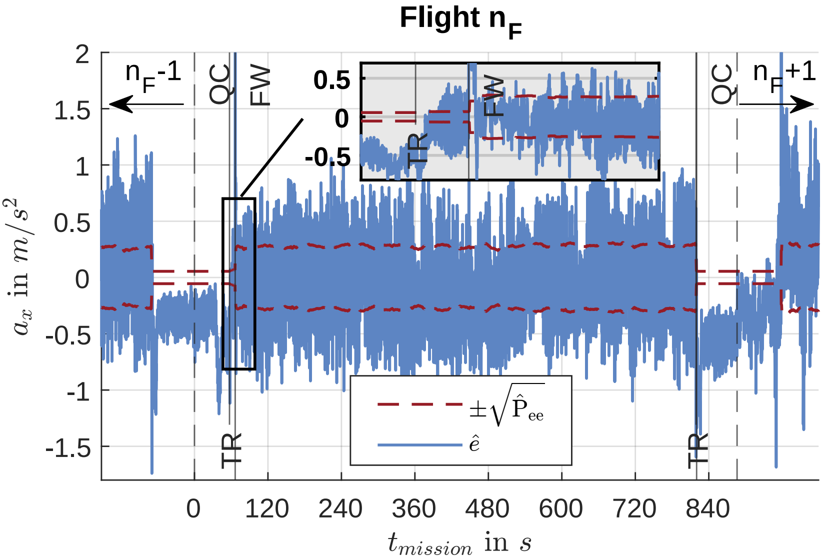


Figure 5.14.: Measurement residuals of the acceleration in the x-direction of flight $n_F = 263$, the second flight of the 30th Bootstrap sample in TS4.

The accelerations and rotational rates, especially in longitudinal motion, are sufficiently estimated concerning the aerodynamic modification with the parameter states C_{D_0} and C_{L_0}. The presented data of flight $n_F = 263$ considers a CA $= 12m/s$ and a TOW $= 4.62kg$ without aerodynamic modification. Measurement residuals resulting from flights with aerodynamic modification do not differ significantly. Instead, the effect of the conducted change is small and difficult to identify, which is further discussed below.

Consistency

As in Section 5.2, the ANN weight states resulting from the training of the hybrid model above are considered fixed. This allows for the investigation of the modification's impact on C_{D_0} and C_{L_0}. Consequently, the final estimate of the ANN is used, cf. Figure 5.13.

Due to the use of real flight test data, true states are not available for analysis. Therefore, comparing the hybrid and physical model's state estimation performance is omitted.

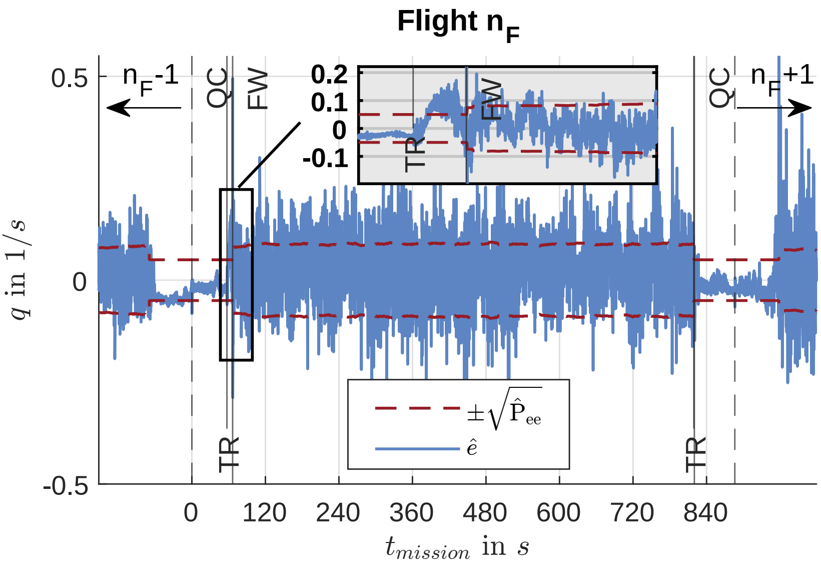


Figure 5.15.: Measurement residuals of the pitch rate of flight $n_F = 263$, the second flight of the 30th Bootstrap sample in TS4.

Further, the physical model cannot learn the modification's effect depending on u_{nn}. The only variables that can be used for comparison are the estimated outputs. Based on the observations made in Section 5.2 and with flight test data analysis, the measurements are assumed to be too noisy. Consequently, considering the measurements is assumed to be insufficient for assessing the aerodynamic modification and is therefore neglected.

Instead, the parameter states are considered. In addition, benchmarks are provided, introduced in Subsection 4.1.2, and used in the following.

The hybrid model attributes a parameter states' dependency on u_{nn}. The simple design of the ANN allows a distinction between the unmodified and modified UAS, as shown in Figure 5.16.

Despite noisy measurements reflected by the residuals and estimated covariances, the hybrid model identifies an increase in the lift coefficient C_{L_0}. The last update of the

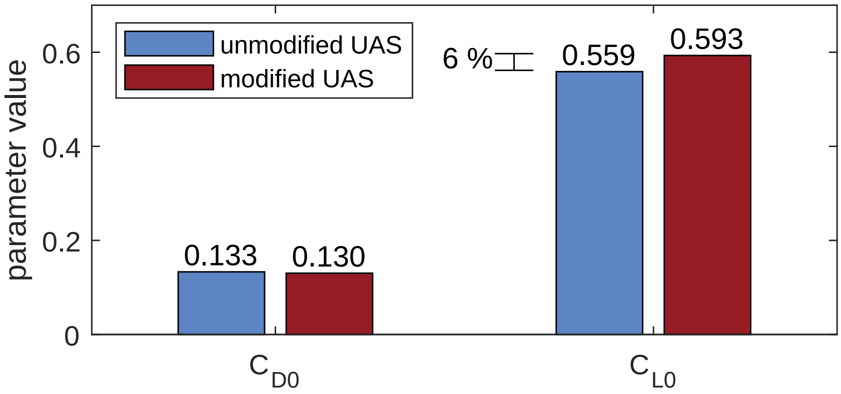


Figure 5.16.: Comparison of the unmodified and modified UAS concerning the parameter states C_{D0} and C_{L0} in TS4.

ANN weight states determines an increase of the lift coefficient by 6 %. In the same time, no increase in drag results from the final estimate.

However, the parameter estimation involves uncertainty, cf. Figure 5.13. Both the lift and drag coefficients vary during learning. They take an average value of a bootstrap sample that lies above the final estimate. The mean values of the last bootstrap sample are $\bar{C}_{D_0} = 0.145$ for the drag and $\bar{C}_{L_0} = 0.642$ for the lift coefficient.

According to the benchmarks and the prevailing flight conditions, the results mainly correspond to the expected values based on flow simulations. In the case of the expected drag increase, the effect of the aerodynamic modification is too small and estimation uncertainty too high to be determined within the range of 1 to 6 %. Thus, the impact on C_{D_0} can not be determined sufficiently using the provided UAS sensors, and the flight test procedure in Section 4.3. The increase of the lift coefficient C_{L0} by 6 % lies in the expected range of 3 to 12 %.

Therefore, the results are physically meaningful, and the hybrid model successfully applies to real flight test data. However, using the flight test data also reveals issues in the convergence behavior of the implemented ANN and the state estimation within QC and TR flight. In addition, the consideration of a binary attribute to inform an ANN about a modification requires further research. This may also involve the consideration of multiple amendments of different kinds in future work.

5.4. Evaluation of the approach to hybrid model building

Based on the approach to hybrid model building developed in Chapter 3, four hybrid models are successfully initialized and trained. They extend the physical model in the presence of time-variant physical parameters and insufficiencies of the physical model to a tolerable extent. The physical models implemented in the hybrid model structures enable physical consistency according to the validity of the physical model used. Accurate state estimation is provided even in the hybrid models' early learning phase, including the prediction of hidden states such as time-variant parameter states. The estimated measurements, hidden states, and corresponding covariances provide insight into the hybrid models' inner working principles, respectively.

Systematic deviations between the real and digital object can be addressed within the data-driven structure, the ANN, being part of the hybrid models. Because the ANN alone cannot predict hidden states nor allow and prove physical consistency, a comparison to a data-driven model using an ANN exclusively is not part of the evaluation.

Four main groups of requirements are defined for the development in Chapter 3. Each of the requirements is addressed in the development process. The successful application and training of the hybrid models using flight simulation and test data enables an evaluation of aircraft systems. It further allows an outlook for verification and validation or improvements in future work.

As the requirements definition is based on the identified research gap, a linkage exists to the state of the art in hybrid model building. The linkage is used to elaborate on the novelty of the proposed approach to hybrid model building and the contributions in the field of AI and NI. The fulfillment of the requirements is considered below.

5.4.1. Fulfillment of the initialization requirements (A)

The first two requirements consider the initialization of the new hybrid model (A): the implementation of a physical model (A1) and the use of physical-based conditions or assumptions (A2).

It has been shown that a physical model can initialize the hybrid model. In this case, the physical model does not solely improve physical consistency as it appears in the literature but builds the starting point of state estimation ($\hat{y}_{hyb_{ini}} = \hat{y}_{phy_{ini}}$), which experiences extension in a learning process. Knowledge about the considered system in terms of the physical model structure, sensor noise, and assumptions about the system noise or confidence in the physical model can, therefore, be fully incorporated into the hybrid model. Furthermore, the dynamical and parameter states of the system can be initially set according to prior measurements and research results or physical-based assumptions ($\hat{x}_{hyb_{ini}} = \hat{x}_{phy_{ini}}$). The hybrid model captures deviations of the initial states, while evidence is provided for a variation of 20 % within the TS concerning flight simulation data.

In addition, SC can be initially provided, which limits the solution space of the selected states and improves filter stability. The sharp limits of the chosen relu66 output activation function further improve filter stability by limiting the prediction horizon of the ANN and supporting the filtering process[6].

The TS shows the closer the implemented physical model is to the described system, the better the hybrid model performs initially. Starting with TS1, the hybrid model has no structural discrepancy to be learned and thus obtains good results since the early phase of state estimation. The hybrid model still performs well if a structural discrepancy between the model and the described system is added in TS2. It separates the learning of the manipulated parameter states C_{D_0} and C_{L_0}, which converge within 20 flights toward the actual values. The dynamical states find sufficient estimates since initialization.

In the case of TS3, the stated discrepancy is increased, resulting in an insufficient mapping due to a reduced physical model PM$_{red}$. Consequently, the hybrid model initialized with the reduced physical model loses the ability of sufficient state estimation and prediction compared to TS1 and TS2. The hybrid model's learning rate, manifested in f_{update}, is therefore increased to provide applicability.

Finally, the initialization requirement (A) and its sub-requirements (A1) and (A2) are fulfilled with the limitation of the learning rate f_{update}, which needs to be adjusted in the case an insufficient physical model is used for initialization.

5.4.2. Fulfillment of the generalization requirements (B)

The generalization requirement (B) is evaluated in terms of the hybrid model´s ability to extrapolate (B1) and to avoid overfitting (B2) using a limited database with operational

[6] Preliminary tests have shown that the choice of sigmoid or tangent hyperbolic functions can lead to instability. They do not provide sharp limits leading to divergent behavior at their edges.

data exclusively (B3).

An extrapolation is performed when predicting the system states concerning new operational conditions, which are unknown to the hybrid model. The operational conditions change with the TOW, and CA included in the operational data. However, as the hybrid model is designed to adapt to such changes, the effect on the prediction performance is automatically captured within the state estimation process. Remind the state update frequency in TS1 and TS2 is set $f_{update} = 1Hz$ with minor adjustments of the ANN weight states ($Q_{xx_{nn_{ij}}} = 10^{-8}$). This enables state prediction and thus extrapolation for 99 time steps without a state correction ($\hat{x}^* = f(\hat{x})$). Therefore, a statement of extrapolation and generalization is possible to a limited extent. TS that allow further investigation of extrapolation are not performed.

To the extent of the changing operational conditions within the considered database and the slow adaption speed of the ANN within the hybrid model, the performance in state estimation and prediction is sufficient. In both the learning phase and the consistency analysis handling a converged solution of the hybrid model, the estimation of states, measurements, and covariances show minor residuals and physically meaningful values. Thus, the extrapolation requirement (B1) is fulfilled according to the use case.

In particular, the measurement residuals and the corresponding covariance estimation reflect the initially set sensor noise ($R_{yy} \neq \underline{0}$). Therefore, overfitting is successfully avoided as required with (B2) and intended in the design of the hybrid model.

Operational data are exclusively considered using the acquired flight simulation and flight test data, which reflect standard FMs as expected in operation. Thus, the application of the hybrid model is successfully performed concerning (B1) and (B2) with a limited database of operational flight data ($u = u_{ops}$, $m = m_{ops}$) as required in (B3).

For future work, some conclusions in extrapolation and PENN are given in the following. The proposed hybrid model enables a data-driven structure to take effect on the parameters of a physical model. Consequently, the hybrid model can potentially unlearn the ability to extrapolate when modifying the physical model with which it is initialized. On the other hand, learning the PENN does not affect the physically encoded model and thus extrapolation. However, it limits the adaptability of the hybrid structure, a dilemma discussed further in Section 5.5 and should be addressed in future work.

5.4.3. Fulfillment of the interpretation requirements (C)

For the hybrid model's interpretability (C), access to inner system states is required (C1) and consistency with benchmarks of system properties is desired (C2).

The hybrid model design enables access to hidden states, including dynamical and parameter states, as intended in Chapter 3. Implementing a parameter scheduling model structure combining the physical model with an ANN provides sufficient estimates of the parameters observed in TS1 and TS2. Using a reduced physical model in TS3 lowers the validity of state estimation and limits insight into the system to visualize and prove the impact of modifications. In the case of TS4, the true states of the system are unknown for comparison, but the results are considered physically meaningful and in agreement with the provided benchmarks.

Concerning the use of a sufficient physical model for initialization, the parameter state prediction by the ANN provides physically consistent results compared to the true parameter states contained in the flight simulation data. In conclusion, the hybrid model offers insight into the system states while learning and delivers a sufficient prediction of states once converged. Prediction results can potentially be used to simulate the further development of system degradation and recovery measures in operation and before FMs, including the effect on system dynamics and energy demand in future work.

The hybrid model is, therefore, interpretable (C), provides access to the inner states (C1), and allows the use of benchmarks (C2) according to the formulation of the physical model and the selected parameter states.

However, as already concluded from using the physical reference model in Section 4.6, the observability of states of a complex system such as the UAS under consideration is essential. Such a system meets the criteria of a stochastic nonlinear system and suffers from a lack of deterministic observability, as shown in Subsection 4.4.3. Therefore, the dynamical states, particularly the parameter states, must be chosen wisely to allow at least stochastic observation.

In this thesis, a selection of parameter states is considered, enabling sufficient estimates of the parameters and, thus, interpretability. In TS1 and TS2, the longitudinal motion of the UAS is regarded with the parameters C_{D_0}, C_{L_0}, and C_{M_0}. In TS3, both the longitudinal and lateral movement are described using C_D, C_Y, C_L, C_ℓ, C_m, and C_n within a reduced physical model PM$_{red}$. In TS4, C_{D_0} and C_{L_0} are exclusively estimated using an ANN.

Consequently, the selected aerodynamic parameters can be distinguished by describing each translational and rotational motion in one axis direction. Therefore, a translational or rotational motion in the three directions is uniquely assigned to a parameter state, allowing stochastic observability and state prediction, as shown in Section 5.2.

Assignments to multiple parameters are handled with the adjustment of the physical model in Subsection 4.4.3. The sufficient estimation of the 31 parameters is achieved with a loss of adaptability. Further research is needed to explore the behavior of the hybrid model with an increased parameter state vector describing similar motion and axis.

So far, the following findings can be stated to be further examined in future work:

- The hybrid model assumes the observability of the physical model used for initialization, as expected in Section 4.6 (Expectation 1).
- The interpretability of states is linked to their observability, as expected in Section 4.6 (Expectation 2).

However, observability and, thus, interpretability are related to adaptability, resulting in a filter dilemma discussed in Section 5.5.

5.4.4. Fulfillment of the adaption requirements (D)

The adaption requirements (D) are the ability to adapt through learning (D1), to process automated information flow (D2), and to concern additional data (D3) using the hybrid model.

The hybrid model uses a recursive filter method for state estimation. The state vector contains dynamical, parameter, and ANN weight states, which are jointly estimated. Within the joint estimation process, the filter algorithm continuously adjusts the state vector and the covariance estimation to minimize the variance of the measurement residuals ($V_k \times \hat{e}_k \neq \underline{0}$). Consequently, the hybrid model continuously adapts through learning (D1), saves knowledge about prior estimates, and updates the state covariance matrix P_{xx}, cf. Eq. 2.5.

Restrictions appear in TS1 and TS2, considering $f_{update} = 1Hz$. In this case, adaption is paused for a specific period so that the hybrid model's extrapolation can be studied, cf. Subsection 5.4.2.

The adaptability of the approach can be influenced by the learning rate f_{update} and the choice of the system noise covariance matrix Q_{xx}. In particular, setting Q_{xx} initially small increases the filter quality but decreases adaptability. A dilemma has been observed in adjusting the physical model in Subsection 4.4.3 and in applying the hybrid model in Sections 5.2 and 5.3. The application to flight simulation and test data is performed with small values of Q_{xx} that allow the parameter states to be sufficiently estimated. However, it increased the time to learn and converge to the set of ANN weight states. In the case of TS3, the values of Q_{xx} concerning the ANN weight states are increased to allow adaptability and enable the hybrid model's applicability. However, the results are reported to be poor due to decreased filter quality.

The dilemma is addressed by applying the bootstrap method to allow convergence and functional predictions of the parameter states in stochastic observability. Thus, concerning stochastic systems and stochastically observable states, sufficient filter quality is required for which losses in adaptability must be accepted. Even though bootstrapping is implemented in TS3, adjustment of Q_{xx} is needed as the number of bootstrap samples is limited and increases computational time[7]. Therefore, the aspect of the computational effort has to be considered in the context of adaptability and filter quality.

Processing automated information flow is not explicitly tested within the provided TS. However, based on the designed filter algorithm and its application, statements towards applicability and real-time capability are made to guide future work on automated information flow.

The design of the filter method includes time-discretized steps, which contain:

- input data u shared by the physical model and the ANN,
- additional data u_{nn} feeding the ANN exclusively,
- and the prior state estimates $\hat{x}_{k|k-1}$, which are used within the filter algorithm and also provided the ANN input layer.

Therefore, new information about the real object can be automatically linked to the hybrid model so that estimates and predictions can be exchanged with the digital object and vice versa (D2).

[7] Remind, in TS3 an ANN is used covering 426 weights, which causes the hybrid model to process the flight simulation data with 30 bootstrap samples in 6 days. In comparison, TS1, TS2, and TS4 are conducted in approximately 3 hours using the same computer system.

The real-time capability must be enabled to allow automated information flow and testing in future work. The chosen filter algorithm enables new information to be processed when available but suffers from the high computational effort corresponding to the size of the state vector. The computational time of learning the hybrid model exceeds the frequencies of the filter used for predictions and updates ($t_{calc_k} >> \frac{1}{f_{prediction}}$. This falls short of the desired real-time capability for automating information flow in flight operations (D2).

Therefore, the approach to hybrid model building must be revised or extended to decrease computational effort. Possible solutions consist of adapting the methodology and modifying the implementation besides upgrading the computing hardware, which is detailed in the outlook of this thesis.

Lastly, the hybrid model involves additional data u_{nn} into the input layer of the ANN. The additional data describes the total operation time and the time since the recovery in TS1, TS2, and TS3 or the binary information whether a modification is active or not in TS4. The data supplies significant information to the hybrid model to sufficiently learn and predict the parameter states. Therefore, the hybrid model enables the inclusion of additional data ($u_{nn} \neq \underline{0}$) and benefits from using such data (D3).

In conclusion, the resulting hybrid models are adaptive and satisfy requirements (D1) and (D3), while requirement (D2) needs further research to enable real-time capability and testing of automated information flow.

5.5. Discussion of the results

The developed approach to hybrid model building addresses some shortcomings of the prevailing concepts in the literature. These concern model structure and optimization. The evaluation is valid for the considered use case and the database used but serves as an outlook for further verification and validation in future work.

The innovation of the resulting hybrid model lies in its parameter scheduling structure, combining a physical model and an ANN paired with a recursive optimal filter method, enabling continuous adaption through learning and recursions that emphasize a deep coupling of the two model parts.

Requirements include the hybrid model's initialization, generalization, interpretation, and adaption. A initialization issue occurs when the complexity of the initial physical model is insufficient. In this case, the learning rate can be increased to ensure adaptability and filter stability.

In terms of generalization, interpretation, and adaption, the hybrid model runs into a structural and a filter dilemma. Both are discussed in detail in the next Section.

5.5.1. The structural dilemma and the filter dilemma

The structural dilemma occurs due to the parameter scheduling structure of the proposed hybrid model and is first mentioned in Subsection 5.4.2. The ANN within the hybrid model is a data-driven open structure attached to the physical parameters used as an interface between the data-driven and physical models. While the ANN enables the hybrid model´s system to be adapted to the provided data, the physical parameters adjust and may distance from their general meaning initially given. For example, aerodynamic derivatives, which flow simulations, wind tunnel, and flight tests have elaborated, may be used for initialization. These are adjusted in the learning process of the hybrid model according to the operational data used.

Consequently, the hybrid model structure is adapted to increase the accuracy and precision of mapping the data used for learning. However, it can unlearn the mapping of the whole envelope as it may be initially granted. Thus, the adaptability of the structure of the proposed hybrid model is provided to the extent that a loss of generalizability is tolerated. This forces the need for expert knowledge to sufficiently design the hybrid model structure initially and use the results later on.

The filter dilemma is part of the evaluation in Subsection 5.4.4 and addresses adaptability in compared to filter quality. While adaptability is affected by filter properties (f_{update} and Q_{xx}), these also affect filter quality. In detail, an increase of Q_{xx} increases the ability to adapt to anomalies within the provided data. Still, if needed, it decreases the filter's effect to compensate for such aberrations. Consequently, the hybrid model is challenged to distinguish anomalies as they occur due to technical modification on the one hand or environmental or sensor disturbances on the other hand.

In particular, the presence of stochastically observable states requires a sufficient filter effect, typically provided by using a physical model. Whether the corresponding set of physical parameters is considered fixed or slightly adjustable to a tolerable extent, the physical model affords the required filter effect.

It is obvious to solve both the structural and filter dilemmas described above by replacing the data-driven model structure within the hybrid model with physical-based dependencies. The data-driven model consequently provides analysis of the model´s behavior to finally conduct physical laws to be used instead of the ANN.

In conclusion, replacing data-driven structures with a physical model is suggested for future work, eventually transforming the hybrid model into a physical one. In this case, the new approach to hybrid model building is not only used to expand a physical model by a data-driven model as intended but also extends physical-based knowledge. This procedure finally offers an alternative view of the use of AI and NI in modeling technical systems and intelligent maintenance, where the following thoughts and conclusions are contributed.

5.5.2. Contributions in the context of artificial intelligence

At the beginning of the thesis, the topic of AI is introduced, where ML and DL settle with the use of data-driven models such as the ANN. In the literature, replacing physical models with data-driven models has become popular. A need for implementing physics has been recognized against the background of technical systems following rules based on natural phenomena. But still, the development of AI focuses on data-driven modeling, which eventually involves physical-based features or a modification of the optimization algorithm as discussed in Chapter 2.

In contrast, this thesis questions how modeling of physically experienceable and experienced systems can and should be based on physical models in the context of AI. It has been shown that physical model building does not exclude the use of AI. Instead, AI can be used to extend a physical model or find physical laws and dependencies. In this case, AI is considered a tool. Such an approach to using AI is a young research trend[8], a promising research field to settle the developed technique for hybrid model building.

Furthermore, the increased use of physical models in the context of AI opens up access to several methods and considerations that have already been successfully used in system identification. One of these, the observability of states, whether deterministic or stochastic, is placed for the first time in hybrid model building using ANNs. It questions the integrity of data-driven approaches in technical systems when physical interpretability and consistency are required.

[8] Contributions are made by Cornelio et al. in 2023. They successfully derive physical laws explaining natural phenomena “by combining logical reasoning with symbolic regression“ [Cor+23].

In conclusion, the research within this thesis contributes to the field of AI and NI, where the use of AI argues for using the NI term in the mechanical engineering domain as defined in Subsection 2.3.5. In the sense of NI, knowledge can be gained due to AI's use. Therefore, AI has to rely on existing knowledge, which implies that AI should be knowledge-based rather than data-driven. Consequently, the proposed approach to hybrid model building finds a basis to use available knowledge and to explore new knowledge in future work.

5.5.3. Contributions in the field of aircraft maintenance

The motivation for this thesis lies in extending physical knowledge using aspects of AI. However, the use case is considered in the field of aircraft maintenance. Consequently, the contributions of the developed hybrid model in the field of aircraft maintenance will be discussed in the following.

The application and evaluation of the hybrid model have shown that its initialization using a sufficient physical model and valuable assumptions of the aircraft states contribute significantly to the model performance. Further, describing the physics of the aircraft and its states cannot easily be replaced by data-driven models. Conversely, knowledge about the aircraft system is fundamental to the creation of hybrid models. Success in intelligent maintenance of aircraft, which depends on modeling performance, therefore builds on the understanding of the aircraft physics.

In terms of extrapolation, the hybrid model has shown accurate predictions of states, which are either supported by measurements or hidden. In particular, the prediction of the aerodynamic coefficients of the aircraft can have a useful value for the evaluation of the aircraft's health and its evolution in future operations. Due to the structure of the hybrid model, these parameters are available for interpretation and thus for understanding and explaining the digital model and its real counterpart. This attribute is crucial for the safety criticality of aircraft maintenance and thus operations (cf. Section 2.1).

Lastly, the integration of a data-driven model part and the use of a time-discrete Kalman filter variant enables a higher adaptivity of the hybrid model. New information can be used incrementally to update the model, while additional data, such as maintenance data, can be utilized. Secondly, the aircraft parameters can be considered non-constant allowing degradation and maintenance effects to be mapped. Consequently, the use of a data-driven part within the hybrid model makes a valuable contribution when used in dosed quantities.

6. Summary and conclusions

This thesis develops and applies a new approach to hybrid model building to flight test and simulation data of an Unmanned Aircraft System (UAS). The goal is the estimation of time-variant aircraft parameters to account for degradation and recovery effects and to support intelligent maintenance. Intelligent maintenance requires optimization, which has to account for physical consistency and interpretability as well as accuracy and adaptability in the aviation domain. Consequently, a hybrid model promising such characteristics is appropriate.

The first chapter introduces aspects of aircraft maintenance and its contribution to aviation safety, efficiency, and sustainability. Intelligent maintenance turns out to be the most promising in the respective areas. It can rely on system models, which result from system identification. Therefore, fundamentals in aircraft system identification are given, including the recursive filter method and the definition of observability.

If a physical model describing an aircraft system is insufficient and fails to map time-variant parameters, data-driven models should be coupled. This requires an approach to hybrid model building. Definitions in the hybrid model building field deliver a basic understanding of the model parts and the different classes where they are coupled. In addition, the Digital Twin (DT), Artificial Intelligence (AI), and, in particular, the Artificial Neural Network (ANN) are discussed to address requirements on the hybrid model later. The combination of a physical model and an ANN is emphasized in the literature with four prevailing concepts. These are the physical-based Feature Engineering (FE), the physical guided (PGNN), informed (PINN), and encoded neural network (PENN). They address the need for physical consistency.

However, these concepts argue for integrating physics into neural networks rather than vice versa. They struggle with initialization, generalization, interpretation, and adaption, leading to identifying the research gap. The research gap concerns the hybrid model structure and the learning method. Therefore, an alternative to the prevailing parallel and serial model structures and the batch-wise and least-squares learning strategies is offered.

Three primary goals are specified and achieved:

- the extension of a physical model using an ANN,
- the access to an interpretable model structure,
- and the adaption through learning.

Chapter 3 introduces a development methodology that defines requirements, the conceptual design, the determination of a model structure, and the implementation of a state estimation algorithm for system identification. Based on the research gap and in the context of intelligent aircraft systems, requirements are defined in terms of initialization, generalization, interpretation, and adaption. The requirements and defined subcategories are addressed within the conceptual design of the new hybrid model. The concept considers the physical parameters as time-variant and as the interface between the physical model and an ANN. Systematic deviations, which occur due to insufficient mapping by the physical model are accounted for when using the ANN.

Based on the concept, a parameter scheduling model structure is chosen to combine the physical model and the ANN, which enables the learning of time-variant parameters. The parameter scheduling structure defines parameters as the output of the ANN and input to the physical model. Further, the input layer of the ANN can include additional data, such as maintenance data and recursions of system states.

A joint state estimation procedure is derived to allow for deep coupling of the hybrid model structure, adaptive optimization, and the incorporation of further a priori knowledge. Based on a recursive filter method, adaption is enabled as soon as information about a new time step is available while considering sensor noise. The joint estimation of states considers both models simultaneously. To address nonlinearities and filter instabilities in state estimation, a modified Unscented Kalman Filter is chosen and extended by State Constraints (SCmUKF). Preliminary studies confirm the applicability and usability of the resulting hybrid model in the case of less complex systems.

In this thesis, the complexity of the use case is increased by considering the flight data of a UAS. The UAS is introduced as a flight test platform in Chapter 4. It is employed within an experimental design involving flight tests and simulation data. Flight tests are executed to acquire data for the application of the hybrid model and to support the development of an SIL simulation environment.

The choice of the UAS used as a flight test platform enables the integration of aircraft modifications into the real and simulated system. These modifications are subject to time-variant aircraft parameters to be learned by the hybrid model in Chapter 5. Four Test Series (TS) are executed to apply and evaluate the hybrid model. The state estimation task involves the consideration of the unmodified and modified UAS and different physical models.

Using flight simulation data, the hybrid model converges. Noisy measurements, dynamical states, and parameter states are sufficiently estimated. In particular, the ANN learns to predict the parameter states in FW flight in the case of constant (TS1) and time-variant parameters (TS2). In the case of using a reduced physical model for initialization (TS3), limitations are uncovered in terms of adaptability versus filter quality.

The application to flight test data confirms the applicability and usability of the hybrid model concerning flight behavior and environmental and sensor disturbances in the real world. Unless convergence of some ANN weights states has not been observed, the ANN offered a physically meaningful prediction of the parameters compared to the provided benchmarks. However, the convergence issues indicate a lack of observability concerning the ANN.

In the evaluation, the requirements from Chapter 3 and the results from Chapter 5 are merged, followed by the discussion and the main conclusions summarized below.

6.1. Conclusions

The developed approach to hybrid model building fulfills the defined requirements primarily. Physical knowledge can be fully used for initialization, extrapolation granted, and overfitting prevented using operational data exclusively. The hybrid model structure is interpretable under the restriction of stochastic observability. Adaptability is provided due to the ability to learn and to account for additional data but lacks the real-time capability to enable automated information flow in operation.

A dilemma in the structural design and the applied filter method is observed. While the parameter scheduling structure can increase adaptability, the initially provided system of the physical model can be affected and thus lose generalizability. Second, a filter dilemma occurs due to the balancing act between adaptability and the filter effect concerning any anomalies.

It is therefore discussed to enhance physical knowledge by using AI to infer physical dependencies replacing data-driven structures. Consequently, the hybrid model should consider Natural Intelligence (NI) to be supported by AI in knowledge gain.

Consequently, the use of physical models in the context of intelligent maintenance of aircraft is suggested. They can form the basis of hybrid model, which is to be extended with a data-driven part, if the inclusion of maintenance data and increased adaptability is required. This can increase model performance, which finally contributes to safe and more efficient aircraft operations.

6.2. Outlook

For future work, improvements of the recursive filter algorithm used within the hybrid model are recommended before verification and validation can be achieved. Whether the hybrid model is used to extend a physical model as intended or to infer physical dependencies, finally, learning is based on the recursive filter method. The filter algorithm can be further developed regarding adaptability and computational efficiency.

Adaptability can be addressed by examining the covariances Q_{xx} and R_{yy} to understand their effect within the hybrid model fully. Furthermore, the use of adaptive covariance algorithms and hyperparameter optimization can be proven, which are promising features [EK22; EHH22].

The computational efficiency is affected by the joint estimation procedure used, where the computational time scales with the size of the system states vector. Therefore, merging the Unscented Kalman Filter (UKF) and the Extended Kalman Filter (EKF) and implementing dual estimation can reduce the computational time by simplifying the state prediction task of the constant ANN weights states.

Finally, improving the convergence behavior and robustness and enabling the real-time capability of the hybrid model form the basis for verification, validation, and the implementation of a DT. In perspective, the DT can visualize degradation and recovery effects in future aircraft operations based on time-variant parameters. Their supporting role in predicting any technical system’s Remaining Useful Life (RUL) and gaining knowledge is subject to future research.

Literature

[Ada14] Jürgen Adamy. *Nichtlineare Systeme und Regelungen*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014. ISBN: 978-3-642-45012-9. DOI: 10.1007/978-3-642-45013-6.

[AIA16] AIAA Intelligent Systems Technical Committee. *Roadmap for intellgent systems in aerospace*. Ed. by American Institute of Aeronautics and Astronautics (AIAA), Intelligent Systems Technical Committee. Version First Edition. 2016. URL: https://aiaa-istc.github.io/IS_roadmap.html (visited on 23. 03. 2023).

[Air21] Airbus. *Airbus Annual Report 2021. Connecting and protecting*. Ed. by Airbus. 2021. URL: https://www.airbus.com/en/investors/financial-results-annual-reports/annual-report-2021-quick-read (visited on 14. 03. 2023).

[Air02] Airbus Flight Operations. *Getting to grips with aircraft performance*. Ed. by Airbus S.A.S. 2002. URL: https://www.skybrary.aero/bookshelf/books/2263.pdf (visited on 10. 11. 2021).

[Air13] Airbus Flight Operations. *Airbus launches Sharklet retrofit for in-service A320 Family aircraft*. Ed. by Airbus S.A.S. 2013. URL: https://www.airbus.com/en/newsroom/press-releases/2013-10-airbus-launches-sharklet-retrofit-for-in-service-a320-family (visited on 14. 12. 2022).

[Air14] Airbus Flight Operations. *Getting to grips with Performance retention and fuel Saving. A320 Family*. Ed. by Airbus S.A.S. 2014. URL: https://pdfslide.net/documents/a320-performance-retention-and-fuel-savings-v3.html?page=1 (visited on 14. 12. 2022).

[Ame21] American Airlines Group. *ESG Report 2021*. Ed. by American Airlines Group. 2021. URL: https://www.aa.com/content/images/customer-service/about-us/corporate-governance/esg/aag-esg-report-2021.pdf (visited on 14. 03. 2023).

[And+90] Kristinn Andersen, George E. Cook, Gabor Karsai, and Kumar Ramaswamy. "Artificial neural networks applied to arc welding process modeling and control". In: *IEEE Transactions on Industry Applications* 26.5 (1990), pp. 824–830. ISSN: 00939994. DOI: 10.1109/28.60056.

[Ang18] Christoph Anger. "Hidden semi-Markov Models for Predictive Maintenance of Rotating Elements". PhD thesis. Darmstadt: Technische Universität Darmstadt, 2018.

[Arb+22] Rosa Arboretti, Riccardo Ceccato, Luca Pegoraro, and Luigi Salmaso. "Design of Experiments and machine learning for product innovation: A systematic literature review". In: *Quality and Reliability Engineering International* 38.2 (2022), pp. 1131–1156. ISSN: 0748-8017. DOI: 10.1002/qre.3025.

[Ard09] ArduPilot Development Team and Community. *ArduPilot. (Version 4.2.2)*. 2009. URL: http://ardupilot.org (visited on 07. 03. 2023).

[Are+18] Krzysztof Arentdt, Muhyiddine Jradi, Hamid R. Shaker, and Christian Veje. "Comparative Analysis of white-, gray- and black-box models for thermal simulation of indoor environment: teaching building case study". In: *Proceedings of the 2018 Building Performance Modeling Conference and SimBuild co-organized by ASHRAE and IBPSA-USA*. 2018, pp. 173–180.

[AJV95] M. Aznar Fernández-Montesinos, P. Janssens, and R. A. Vingerhoeds. "ENHANCING AIRCRAFT ENGINE CONDITION MONITORING". In: *Safety, Reliability and Applications of Emerging Intelligent Control Technologies*. Elsevier, 1995, pp. 161–166. ISBN: 9780080423746. DOI: 10.1016/B978-0-08-042374-6.50029-8.

[Bau19] Sebastian Baumann. "Using Machine Learning for Data-Based Assessing of the Aircraft Fuel Economy". In: *2019 IEEE Aerospace Conference* (Big Sky, MT, USA, 2nd Mar.–9th2019). IEEE, 2019, pp. 1–13. ISBN: 978-1-5386-6854-2. DOI: 10.1109/AERO.2019.8742011.

[BK19] Sebastian Baumann and Uwe Klingauf. "Modeling of aircraft fuel consumption using machine learning algorithms". In: *CEAS Aeronautical Journal* 11.1 (2019), pp. 277–287. ISSN: 1869-5582. DOI: 10.1007/s13272-019-00422-0.

[BM12] Randal W. Beard and Timothy W. McLain. *Small Unmanned Aircraft. Theory and Practice*. The University Press Group Ltd, 2012. ISBN: 978-0-691-14921-9.

[BKK23] Florian Michael Becker-Dombrowsky, Quentin Sean Koplin, and Eckhard Kirchner. “Individual Feature Selection of Rolling Bearing Impedance Signals for Early Failure Detection”. In: *Lubricants* 11.7 (2023), p. 304. DOI: 10.3390/lubricants11070304.

[BCV13] Yoshua Bengio, Aaron Courville, and Pascal Vincent. “Representation learning: a review and new perspectives”. In: *IEEE transactions on pattern analysis and machine intelligence* 35.8 (2013), pp. 1798–1828. DOI: 10.1109/TPAMI.2013.50.

[Ber14] Joseph T. Bernardo. “HARD/SOFT INFORMATION FUSION IN THE CONDITION MONITORING OF AIRCRAFT”. College of Information Sciences and Technology. PhD thesis. Pennsylvania: Pennsylvania State University, 2014.

[BK94] Jürgen Beyer and Uwe Klingauf. “Rekursives Verfahren zur biasfreien Parameterschätzung mit minimaler Schätzfehlerkovarianz (RMV)”. In: *at - Automatisierungstechnik* 42.1-12 (1994), pp. 346–355. DOI: 10.1524/auto.1994.42.112.346.

[Boe22] Boeing. *2022 Sustainability Report. Sustainable Aerospace Together*. Ed. by The Boeing Company. 2022. URL: https://www.boeing.com/resources/boeingdotcom/principles/sustainability/assets/data/2022_Boeing_Sustainability_Report.pdf (visited on 14. 03. 2023).

[BAL11] Rudolf Brockhaus, Wolfgang Alles, and Robert Luckner. *Flugregelung*. Berlin, Heidelberg: Springer, 2011. ISBN: 978-3-642-01442-0. DOI: 10.1007/978-3-642-01443-7.

[CS66] Donald T. Cambell and Julian C. Stanley. *Experimental And Quasi-Experimental Design For Research*. Boston: Houghton Mifflin Company, 1966. ISBN: 978-0395307878.

[CPC19] Diogo V. Carvalho, Eduardo M. Pereira, and Jaime S. Cardoso. “Machine Learning Interpretability: A Survey on Methods and Metrics”. In: *Electronics* 8.8 (2019), p. 832. DOI: 10.3390/electronics8080832.

[Cha+19] Manuel Arias Chao, Chetan Kulkarni, Kai Goebel, and Olga Fink. “Hybrid deep fault detection and isolation: Combining deep neural networks and system performance models”. In: *International Journal of Prognostics and Health Management*. Vol. 10, no. 4. 2019.

[Cha+22] Manuel Arias Chao, Chetan Kulkarni, Kai Goebel, and Olga Fink. “Fusing Physics-based and Deep Learning Models for Prognostics”. In: *Reliability Engineering & System Safety 2022*. Vol. 217. 2022. DOI: 10.1016/j.ress.2021.107961.

[CWZ12] Dehuang Chen, Xiaowei Wang, and Jing Zhao. “Aircraft Maintenance Decision System Based on Real-time Condition Monitoring”. In: *Procedia Engineering* 29 (2012), pp. 765–769. ISSN: 18777058. DOI: 10.1016/j.proeng.2012.01.038.

[Che80] Han-Fu Chen. “On stochastic observability and controllability”. In: *Automatica* 16.2 (1980), pp. 179–190. ISSN: 00051098. DOI: 10.1016/0005-1098(80)90053-9.

[Che+09] X. Q. Chen et al. “Flight Dynamics Modelling and Experimental Validation for Unmanned Aerial Vehicles”. In: *Mobile Robots - State of the Art in Land, Sea, Air, and Collaborative Missions*. Ed. by XiaoQi Chen, Y. Q. Chen, and J. G. Chase. InTech, 2009. ISBN: 978-953-307-001-8. DOI: 10.5772/6994.

[Cho17] François Chollet. *Deep Learning with Python*. New York: Manning Publications, 2017. ISBN: 9781617294433.

[Chr+18] Markus Christen, Michel Guillaume, Maximilian Jablonowski, Peter Marcus Lenhart, and Kurt Moll. *Zivile Drohnen - Herausforderungen und Perspektiven*. [1. Auflage]. Vol. 66. TA-SWISS. Zürich: vdf, 2018. ISBN: 9783728138941. DOI: 10.3218/3894-1.

[CNL11] Adam Coates, Andrew Ng, and Honglak Lee. “An Analysis of Single-Layer Networks in Unsupervised Feature Learning”. In: *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*. Ed. by Geoffrey Gordon, David Dunson, and Miroslav Dudík. Vol. 15. Proceedings of Machine Learning Research. Fort Lauderdale, FL, USA: PMLR, 2011, pp. 215–223.

[Cor+23] Cristina Cornelio et al. “Combining data and theory for derivable scientific discovery with AI-Descartes”. In: *Nature communications* 14.1 (2023), p. 1777. DOI: 10.1038/s41467-023-37236-y. eprint: 37045814.

[Cur00] M. Curvo. “Estimation of aircraft aerodynamic derivatives using Extended Kalman Filter”. In: *Journal of the Brazilian Society of Mechanical Sciences* 22.2 (2000), pp. 133–148. DOI: 10.1590/S0100-73862000000200001.

[DBA21] Assia Daid, Eric Busvelle, and Mohamed Aidene. "On the convergence of the unscented Kalman filter". In: *European Journal of Control* 57 (2021), pp. 125–134. ISSN: 09473580. DOI: 10.1016/j.ejcon.2020.05.003.

[DR20] Arun Das and Paul Rad. *Opportunities and Challenges in Explainable Artificial Intelligence (XAI): A Survey*. 2020. DOI: 10.48550/arXiv.2006.11371.

[Das17] Das Europäische Parlament und der Rat der Europäischen Union. *VERORDNUNG (EU) 2017/2392 DES EUROPÄISCHEN PARLAMENTS UND DES RATES. zur Änderung der Richtlinie 2003/87/EG zur Aufrechterhaltung der derzeitigen Einschränkung ihrer Anwendung auf Luftverkehrstätigkeiten und zur Vorbereitung der Umsetzung eines globalen marktbasierten Mechanismus ab 2021*. 2017. URL: https://www.dehst.de/DE/Europaeischer-Emissionshandel/Luftfahrzeugbetreiber/Emissionshandel/emissionshandel-im-luftverkehr_node.html (visited on 25.08.2023).

[Daw+17] Arka Daw, Anuj Karpatne, William Watkins, Jordan Read, and Vipin Kumar. *Physics-guided Neural Networks (PGNN): An Application in Lake Temperature Modeling*. 2017.

[DVD17] Angela Dean, Daniel Voss, and Daniel Draguljic. *Design and analysis of experiments*. Second edition. Cham: Springer International Publishing, 2017. ISBN: 978-3319522487.

[Deu19] Deutsche Lufthansa AG. *Photography taken on 22.05.2019 by Oliver Rösler*. 2019.

[Deu10] Deutsches Institut für Normung DIN e.V., ed. *Begriffe der Instandhaltung*. DIN EN 13306. 2010.

[Deu12] Deutsches Institut für Normung DIN e.V., ed. *Grundlagen der Instandhaltung*. DIN 31051, Sheet 1. 2012.

[DV19] Arinan Dourado and Felipe A. C. Viana. "Physics-informed neural networks for corrosion-fatigue prognosis". In: *Proceedings of the Annual Conference of the PHM Society*. Vol. 11. 2019.

[DV20] Arinan Dourado and Felipe A. C. Viana. "Physics-Informed Neural Networks for Missing Physics Estimation in Cumulative Damage Models: A Case Study in Corrosion Fatigue". In: *Journal of Computing and Information Science in Engineering* 20.6 (2020). ISSN: 1530-9827. DOI: 10.1115/1.4047173.

[Enk19] Franz Enkelmann. “Vergleich von Parameterschätzverfahren bei diskontinuierlichen Eingangsdaten in Abhängigkeit der Modellkomplexität”. In: *Deutscher Luft- und Raumfahrtkongress (DLRK)* (Darmstadt, 30th Sept.–2nd Oct. 2019). Bonn, Germany: Deutsche Gesellschaft für Luft- und Raumfahrt (DGLR) - Lilienthal-Oberth e.V, 2019. DOI: 10.25967/490143.

[Enk21] Franz Enkelmann. “Untersuchung der Degradation eines hybriden unbemannten Luftfahrzeugs im Flugbetrieb unter künstlich eingebrachten Degradationseffekten”. In: *Deutscher Luft- und Raumfahrtkongress (DLRK)* (Aachen & Virtual, 31st Aug.–2nd Sept. 2021). Bonn, Germany: Deutsche Gesellschaft für Luft- und Raumfahrt (DGLR) - Lilienthal-Oberth e.V, 2021. DOI: 10.25967/530079.

[EHP20] Franz Enkelmann, Robert Heigl, and Kai C. Pfingsten. “Comparison of a physical model and a machine learning approach for a more accurate assessment of fuel efficiency measures”. In: *AIAA Scitech 2020 Forum* (Orlando, FL). Reston, Virginia: American Institute of Aeronautics and Astronautics, 2020. ISBN: 978-1-62410-595-1. DOI: 10.2514/6.2020-0029.

[EHH22] Franz A. R. Enkelmann, Robert Heigl, and David Hünemohr. “A hybrid model approach for more accurate and reliable assessment of aircraft fuel efficiency”. In: *AIAA AVIATION 2022 Forum* (Chicago, IL & Virtual, 27th June–1st July 2022). Reston, Virginia: American Institute of Aeronautics and Astronautics, 2022. ISBN: 978-1-62410-635-4. DOI: 10.2514/6.2022-3236.

[EK22] Franz A. R. Enkelmann and Saleh H. Krüger. “A new hybrid model approach coupling a physical model and an artificial neural network through joint estimation”. In: *The 6th CEAS Conference on Guidance, Navigation and Control (EuroGNC)* (Berlin, Germany, 3rd May–5th2022). Brussels, Belgium: Council of European Aerospace Societies (CEAS), 2022.

[Eur22] European Union Aviation Safety Agency. *Easy Access Rules for Unmanned Aircraft Systems. (Regulations (EU) 2019/947 and 2019/945)*. Ed. by European Union. 2022. URL: https://www.easa.europa.eu/en/document-library/easy-access-rules/easy-access-rules-unmanned-aircraft-systems-regulations-eu (visited on 02. 03. 2023).

[Far+22] Salah A. Faroughi et al. *Physics-Guided, Physics-Informed, and Physics-Encoded Neural Networks in Scientific Computing*. Ed. by ArXiv. Cornell University. 2022. DOI: 10.48550/arXiv.2211.07377. URL: https://arxiv.org/pdf/2211.07377 (visited on 13. 09. 2023).

[Fed21] Federal Aviation Administration. *FAA Aerospace Forecast. Fiscal Years 2021-2041*. 2021. URL: https://www.faa.gov/data_research/aviation/aerospace_forecasts/media/FY2021-41_FAA_Aerospace_Forecast.pdf (visited on 20. 09. 2023).

[FFT07] Jiuchao Feng, Hongjuan Fan, and Chi K. Tse. "Convergence Analysis of the Unscented Kalman Filter for Filtering Noisy Chaotic Signals". In: *2007 IEEE International Symposium on Circuits and Systems* (New Orleans, LA, USA, 27th May 2017–30th May 2007). IEEE, 2007, pp. 1681–1684. ISBN: 1-4244-0920-9. DOI: 10.1109/ISCAS.2007.377916.

[Fen+19] Fentaye, Baheta, Gilani, and Kyprianidis. "A Review on Gas Turbine Gas-Path Diagnostics: State-of-the-Art Methods, Challenges and Opportunities". In: *Aerospace* 6.7 (2019), p. 83. DOI: 10.3390/aerospace6070083.

[Fis74] Ronald Aylmer Fisher. *The design of experiments*. 9. ed. New York: Hafner Press, 1974. ISBN: 9780028446905.

[Ful+20] Aidan Fuller, Zhong Fan, Charles Day, and Chris Barlow. "Digital Twin: Enabling Technologies, Challenges and Open Research". In: *IEEE Access* 8 (2020), pp. 108952–108971. DOI: 10.1109/ACCESS.2020.2998358.

[Gäb12] Andreas Gäb. "Real-Time Parameter Estimation Real-Time Parameter Estimation for Mini Aerial Vehicles using Low-Cost Hardware". PhD thesis. Aachen: RWTH Aachen, 2012.

[GRD21] K. G. Gayathri Devi, Mamata Rath, and Nguyen Thi Dieu, eds. *Artificial intelligence trends for data analytics using machine learning and deep learning approaches*. Artificial intelligence (AI). Elementary to advanced practices. Boca Raton, FL: CRC Press, 2021. ISBN: 9780367854737.

[GT21] A. René Geist and Sebastian Trimpe. "Structured learning of rigid–body dynamics: A survey and unified view from a robotics perspective". In: *GAMM-Mitteilungen* 44.2 (2021). ISSN: 0936-7195. DOI: 10.1002/gamm.202100009.

[Gel+06] Arthur Gelb, Joseph F. Kasper, Raymond A. Nash, Charles F. Price, and Arther A. Sutherland, eds. *Applied optimal estimation*. [19. printing]. Cambridge, Mass.: M.I.T. Press, 2006. ISBN: 0-262-20027-9.

[GSG16] Mike Gerdes, Dieter Scholz, and Diego Galar. “Effects of condition-based maintenance on costs caused by unscheduled maintenance of aircraft”. In: *Journal of Quality in Maintenance Engineering* 22.4 (2016), pp. 394–417. ISSN: 1355-2511. DOI: 10.1108/JQME-12-2015-0062.

[Gv96] Gene Howard Golub and Charles F. van Loan. *Matrix Computations. 3rd ed.* John Hopkins Studies in the Mathematical Sciences. Baltimore (MD): John Hopkins University Press, 1996. ISBN: 978-0-8018-5414-9.

[GLL19] Jian Guo, Zhaojun Li, and Meiyan Li. “A Review on Prognostics Methods for Engineering Systems”. In: *IEEE Transactions on Reliability* 69.3 (2019), pp. 1110–1129. ISSN: 0018-9529. DOI: 10.1109/TR.2019.2957965.

[Har+16] Michael Haranen, Pekka Pakkanen, Risto Kariranta, and Jouni Salo. “White, Grey and Black-Box Modelling in Ship Performance Evaluation”. In: *1st Hull Performance & Insight Conference (HullPIC’16). Castello di Pavone, Italy, 13-15 April 2016*. Ed. by Volker Bertram. Red Hook, NY: Curran Associates Inc, 2016. ISBN: 978-1-7138-1891-5.

[Hay04] Simon Haykin. *Kalman Filtering and Neural Networks*. Vol. v.47. Adaptive and Learning Systems for Signal Processing, Communications and Control Series. Hoboken: John Wiley & Sons Inc, 2004. ISBN: 9780471369981.

[Hay96] Simon S. Haykin. *Adaptive filter theory*. 3. ed. Prentice-Hall information and system sciences series. Upper Saddle River, NJ: Prentice-Hall, 1996. ISBN: 013322760X.

[Hea16] Jeff Heaton. “An empirical analysis of feature engineering for predictive modeling”. In: *SoutheastCon 2016* (Norfolk, VA, USA, 30th Mar.–3rd Apr. 2016). IEEE, 2016, pp. 1–6. ISBN: 978-1-5090-2246-5. DOI: 10.1109/SECON.2016.7506650.

[Hof+14] Nathan V. Hoffer, Calvin Coopmans, Austin M. Jensen, and YangQuan Chen. “A Survey and Categorization of Small Low-Cost Unmanned Aerial Vehicle System Identification”. In: *Journal of Intelligent & Robotic Systems* 74.1-2 (2014), pp. 129–145. ISSN: 0921-0296. DOI: 10.1007/s10846-013-9931-6.

[Hon+20] Chang Woo Hong et al. “Remaining Useful Life Prognosis for Turbofan Engine Using Explainable Deep Neural Networks with Dimensionality Reduction”. In: *Sensors (Basel, Switzerland)* 20.22 (2020). DOI: 10.3390/s20226626.

[Inn+19] Mike Innes et al. “A Differentiable Programming System to Bridge Machine Learning and Scientific Computing”. In: *Neural Information Processing Systems* (8th Dec.–14th2019). Vancouver, 2019. DOI: 10.48550/arXiv.1907.07587.

[Ins00] Institut du Transport Aérien. *Costs of Air Transport Delay in Europe. Final Report*. 2000. URL: https://www.eurocontrol.int/sites/default/files/content/documents/single-sky/pru/publications/other/cost-of-air-transport-delay-in-eu-ita.pdf (visited on 25. 08. 2023).

[Int03] International Organization for Standardization, ed. *Condition monitoring and diagnostics of machines*. ISO 13374-1. 2003.

[ISO15] ISO/IEC/IEEE, ed. *Systems and software engineering - System life cycle processes*. ISO/IEC/IEEE 15288. Geneva, CH, 1st May 2015.

[ISO18] ISO/IEC/IEEE, ed. *Systems and software engineering — Life cycle processes — Requirements engineering*. ISO/IEC/IEEE 29148. Geneva, CH, 1st Nov. 2018.

[Jat15] Ravindra V. Jategaonkar. *Flight Vehicle System Identification: A Time-Domain Methodology, Second Edition*. Progress in Astronautics and Aeronautics. Reston, VA: American Institute of Aeronautics and Astronautics, 2015. ISBN: 978-1-62410-279-0.

[Jia+19] Xiaowei Jia et al. “Physics Guided RNNs for Modeling Dynamical Systems: A Case Study in Simulating Lake Temperature Profiles”. In: *Proceedings of the 2019 SIAM International Conference on Data Mining*. Ed. by Tanya Berger-Wolf and Nitesh Chawla. Philadelphia, PA: Society for Industrial and Applied Mathematics, 2019, pp. 558–566. ISBN: 978-1-61197-567-3. DOI: 10.1137/1.9781611975673.63.

[Joh11] Stephen B. Johnson. *System health management. With aerospace applications*. Aerospace Series. Hoboken, N.J.: Wiley, 2011. ISBN: 978-1-119-99404-6.

[Jos20] Ameet V. Joshi. *Machine learning and artificial intelligence*. 1st ed. 2020. Springer eBook Collection. Cham: Springer, 2020. ISBN: 9783030266226. DOI: 10.1007/978-3-030-26622-6.

[JL07] Simon J. Julier and Joseph J. LaViola. “On Kalman Filtering With Nonlinear Equality Constraints”. In: *IEEE Transactions on Signal Processing* 55.6 (2007), pp. 2774–2784. ISSN: 1053-587X. DOI: 10.1109/TSP.2007.893949.

[JUD95] Simon J. Julier, Jeffrey K. Uhlmann, and Hugh F. Durrant-Whyte. “A new approach for filtering nonlinear systems”. In: *Proceedings of 1995 American Control Conference - ACC’95*. Vol. 3. 1995, 1628–1632 vol.3. DOI: 10.1109/ACC.1995.529783.

[Käh17] Alexander Kählert. “Specification and Evaluation of Prediction Concepts in Aircraft Maintenance. Spezifikation und Evaluierung von Prädiktionskonzepten in der Flugzeuginstandhaltung”. Institut für Flugsysteme und Regelungstechnik. PhD thesis. Darmstadt: Technische Universität Darmstadt, 2017.

[KIF08] Rambabu Kandepu, Lars Imsland, and Bjarne A. Foss. “Constrained state estimation using the Unscented Kalman Filter”. In: *2008 16th Mediterranean Conference on Control and Automation*. Automation (MED 2008) (Ajaccio, France, 25th June–27th2008). IEEE, 2008, pp. 1453–1458. ISBN: 978-1-4244-2504-4. DOI: 10.1109/MED.2008.4602001.

[Kar21] Hamid Reza Karimi, ed. *Fault Diagnosis and Prognosis Techniques for Complex Engineering Systems*. San Diego: Elsevier Science & Technology, 2021. ISBN: 9780128224885.

[KKH18] Hyeonmin Kim, Jung Taek Kim, and Gyunyoung Heo. “Prognostics for integrity of steam generator tubes using the general path model”. In: *Nuclear Engineering and Technology* 50.1 (2018), pp. 88–96. ISSN: 17385733. DOI: 10.1016/j.net.2017.10.006.

[KAC17] Nam-Ho Kim, Dawn An, and Joo-Ho Choi. *Prognostics and Health Management of Engineering Systems*. Cham: Springer International Publishing, 2017. ISBN: 978-3-319-44740-7. DOI: 10.1007/978-3-319-44742-1.

[Kir20] Eckhard Kirchner. *Werkzeuge und Methoden der Produktentwicklung. Von der Idee zum erfolgreichen Produkt*. In collab. with Herbert Birkhofer. Berlin and Heidelberg: Springer Vieweg, 2020. ISBN: 978-3-662-61762-5.

[KM16] Vladislav Klein and Eugene A. Morelli. *Aircraft system identification. Theory and practice*. Second edition. Williamsburg, VA: Sunflyte Enterprises, 2016. ISBN: 9780997430615.

[KSS14] Joern Kraft, Vishal Sethi, and Riti Singh. “Optimization of Aero Gas Turbine Maintenance Using Advanced Simulation and Diagnostic Methods”. In: *Journal of Engineering for Gas Turbines and Power* 136.11 (2014). ISSN: 0742-4795. DOI: 10.1115/1.4027356.

[Kre80] Volker Krebs. *Nichtlineare Filterung*. Methoden der Regelungstechnik. München and Wien: Oldenbourg, 1980. ISBN: 3486247514.

[LAN11] Jack W. Langelaan, Nicholas Alley, and James Neidhoefer. "Wind Field Estimation for Small Unmanned Aerial Vehicles". In: *Journal of Guidance, Control, and Dynamics* 34.4 (2011), pp. 1016–1030. ISSN: 0731-5090. DOI: 10.2514/1.52532.

[Law+22] Zaharaddeen Karami Lawal, Hayati Yassin, Daphne Teck Ching Lai, and Azam Che Idris. "Physics-Informed Neural Network (PINN) Evolution and Beyond: A Systematic Literature Review and Bibliometric Analysis". In: *Big Data and Cognitive Computing* 6.4 (2022), p. 140. DOI: 10.3390/bdcc6040140.

[Lei+16] Yaguo Lei et al. "A Model-Based Method for Remaining Useful Life Prediction of Machinery". In: *IEEE Transactions on Reliability* 65.3 (2016), pp. 1314–1326. ISSN: 0018-9529. DOI: 10.1109/TR.2016.2570568.

[Lei14] Bernhard J. G. Leidinger. *Wertorientierte Instandhaltung. Kosten senken, Verfügbarkeit erhalten*. Wiesbaden: Springer Gabler, 2014. ISBN: 978-3-658-04401-5.

[Lei+19] Peter A. Leitl, Stefan Kuntzagk, Andreas Flanschger, and Kai Pfingsten. "Experimental and numerical investigation of the reduction in skin friction due to riblets applied on the surface of a Taylor-Couette cell". In: *AIAA Scitech 2019 Forum* (San Diego, California). Reston, Virginia: American Institute of Aeronautics and Astronautics, 2019. ISBN: 978-1-62410-578-4. DOI: 10.2514/6.2019-1625.

[Li+15] Naipeng Li, Yaguo Lei, Jing Lin, and Steven X. Ding. "An Improved Exponential Model for Predicting Remaining Useful Life of Rolling Element Bearings". In: *IEEE Transactions on Industrial Electronics* 62.12 (2015), pp. 7762–7773. ISSN: 0278-0046. DOI: 10.1109/TIE.2015.2455055.

[LK14] Linxia Liao and Felix Kottig. "Review of Hybrid Prognostics Approaches for Remaining Useful Life Prediction of Engineered Systems, and an Application to Battery Life Prediction". In: *IEEE Transactions on Reliability* 63.1 (2014), pp. 191–207. ISSN: 0018-9529. DOI: 10.1109/TR.2014.2299152.

[Lim17] Denis Pereira de Lima. "Neural Network Training Using Unscented and Extended Kalman Filter". In: *Robotics & Automation Engineering Journal* 1.4 (2017). DOI: 10.19080/RAEJ.2017.01.555568.

[Lin05] Antje Linser. “Performance Measurement in der Flugzeuginstandhaltung”. PhD thesis. St. Gallen: Universität St. Gallen, 2005.

[LB11] Andrew R. Liu and Robert R. Bitmead. “Stochastic observability in network state estimation and control”. In: *Automatica* 47.1 (2011), pp. 65–78. ISSN: 00051098. DOI: 10.1016/j.automatica.2010.10.017.

[Luf21a] Lufthansa Group. *Sustainability in 2021. Fact sheet*. Ed. by Lufthansa Group. 2021. URL: https://www.lufthansagroup.com/en/responsibility (visited on 14. 03. 2023).

[Luf19] Lufthansa Technik AG. *Photography taken on 30.10.2019 by Sonja Brüggemann*. 2019.

[Luf21b] Lufthansa Technik AG. *AeroSHARK. Shrinking the carbon footprint and improving the bottom line*. 2021. URL: https://www.lufthansa-technik-broadcast.com/files/assets/downloads/inspiration/sustainability/AeroSHARK_Carbon-Footprint.pdf (visited on 04. 07. 2023).

[LS19] Edwin Lughofer and Moamar Sayed-Mouchaweh. *Predictive Maintenance in Dynamic Systems. Advanced Methods, Decision Support Tools and Real-World Applications*. 1st ed. 2019. Cham: Springer International Publishing and Imprint: Springer, 2019. ISBN: 978-3-030-05645-2.

[MPA13] Hessam Mahdianfar, Alexey Pavlov, and Ole Morten Aamo. “Joint unscented Kalman filter for state and parameter estimation in Managed Pressure Drilling”. In: *2013 European Control Conference (ECC)* (Zurich, 17th July–19th2013). IEEE, 2013, pp. 1645–1650. ISBN: 978-3-033-03962-9. DOI: 10.23919/ECC.2013.6669753.

[MY18] Curtis McDonald and Serdar Yuksel. *Stochastic Observability and Filter Stability under Several Criteria*. 2018. DOI: 10.48550/arXiv.1812.01772.

[Meh22] Simon Mehringskötter. “Verschleiß- und Lebensdauerprognose unter Berücksichtigung variabler zukünftiger Betriebszustände. Prediction of Degradation and Remaining Useful Life under Consideration of Variable Future Operating States”. Institut für Flugsysteme und Regelungstechnik. PhD thesis. Darmstadt: Technische Universität Darmstadt, 2022.

[Mol20] Martin Molina. *What is an intelligent system?* 2020. DOI: 10.48550/arXiv.2009.09083.

[Mur+20] Nikhil Muralidhar et al. “PhyNet: Physics Guided Neural Networks for Particle Drag Force Prediction in Assembly”. In: *Proceedings of the 2020 SIAM International Conference on Data Mining*. Ed. by Carlotta Demeniconi and Nitesh Chawla. Philadelphia, PA: Society for Industrial and Applied Mathematics, 2020, pp. 559–567. ISBN: 978-1-61197-623-6. DOI: 10.1137/1.9781611976236.63.

[NV19] Renato Giorgiani Nascimento and Felipe A. C. Viana. *Fleet Prognosis with Physics-informed Recurrent Neural Networks*. 2019. DOI: 10.48550/arXiv.1901.05512.

[Nat04] National Aeronautics and Space Administration. *Technology Facts. Winglets*. Ed. by NASA Dryden Flight Research Center. 2004. URL: https://www.nasa.gov/pdf/89234main_TF-2004-15-DFRC.pdf (visited on 14. 12. 2022).

[Nat23] National Aeronautics and Space Administration. *Procedural Requirements. NASA Systems Engineering Processes and Requirements*. NPR7123.1D. 2023. URL: https://nodis3.gsfc.nasa.gov/npg_img/N_PR_7123_001D_/N_PR_7123_001D__AppendixE.pdf (visited on 23. 09. 2023).

[NW94] Alexander V. Nebylov and Joseph Watson. *Aerospace navigation systems. (Les systèmes de navigation aérospatiaux)*. Vol. 331. AGARDograph. Neuilly sur Seine, 1994. ISBN: 92-836-1018-0.

[Nel20] Oliver Nelles. *Nonlinear System Identification. From Classical Approaches to Neural Networks, Fuzzy Models, and Gaussian Processes*. 2nd ed. 2020. Cham: Springer International Publishing, 2020. ISBN: 978-3-030-47439-3.

[Nel00] Alex T. Nelson. “Nonlinear estimation and modeling of noisy time-series by Dual Kalman filtering methods”. PhD thesis. Oregon: Oregon Health & Science University, 2000.

[Oli12] Mauri Aparecido de Oliveira. “An Application of Neural Networks Trained with Kalman Filter Variants (EKF and UKF) to Heteroscedastic Time Series Forecasting”. In: *Applied Mathematical Sciences*. Vol. 6, no. 74, 3675 - 3686. 2012.

[Pah+07] Gerhard Pahl, Wolfgang Beitz, Jörg Feldhusen, and Karl-Heinrich Grote. “Conceptual Design”. In: *Engineering Design*. Ed. by Gerhard Pahl, Wolfgang Beitz, Jörg Feldhusen, and Karl-Heinrich Grote. London: Springer

London, 2007, pp. 159–225. ISBN: 978-1-84628-318-5. DOI: 10.1007/978-1-84628-319-2_6.

[PK18] Michael Pecht and Myeongsu Kang. *Prognostics and health management of electronics. Fundamentals, machine learning, and internet of things*. Second edition. Hoboken, NJ, USA: Wiley-IEEE Press, 2018. ISBN: 9781119515302.

[PLP20] Emmanuel Pintelas, Ioannis E. Livieris, and Panagiotis Pintelas. “A Grey-Box Ensemble Model Exploiting Black-Box Accuracy and White-Box Intrinsic Interpretability”. In: *Algorithms* 13.1 (2020), p. 17. DOI: 10.3390/a13010017.

[PEK18] Karl F. Prochazka, Hugo Eduardo, and Sebastian R. Klein. “Integrated Fault-Tolerant Control of an Over-Actuated Aircraft Using Optimal Control Allocation and Robust Sliding Mode Observers”. In: *2018 IEEE Conference on Control Technology and Applications (CCTA)* (Copenhagen, 21st Aug.–24th2018). IEEE, 2018, pp. 171–178. ISBN: 978-1-5386-7698-1. DOI: 10.1109/CCTA.2018.8511538.

[PKR20] Karl F. Prochazka, Saleh Krüger, and Daniel Ribnitzky. “Aerodynamic parameter identification of a hybrid unmanned aerial vehicle by using wind tunnel and free flight tests.” In: *DLRK* (Aachen (virtual), 1st Sept.–3rd2020). 2020.

[Pro+21] Karl F. Prochazka, Saleh H. Krüger, Gosta Stomberg, and Maximilian Bauer. “Development of a hardware-in-the-loop demonstrator for the validation of fault-tolerant control methods for a hybrid UAV”. In: *CEAS Aeronautical Journal* 12.3 (2021), pp. 549–558. ISSN: 1869-5582. DOI: 10.1007/s13272-021-00509-7.

[PRE19] Karl F. Prochazka, Tobias Ritz, and Hugo Eduardo. “Over-Actuation Analysis and Fault-Tolerant Control of a Hybrid Unmanned Aerial Vehicle”. In: *5th CEAS Conference on Guidance, Navigation & Control* (Milan, Italy). 2019.

[PS20] Karl F. Prochazka and Gosta Stomberg. “Integral Sliding Mode based Model Reference FTC of an Over-Actuated Hybrid UAV using Online Control Allocation”. In: *2020 American Control Conference (ACC)* (Denver, CO, USA, 1st July–3rd2020). IEEE, 2020, pp. 3858–3864. ISBN: 978-1-5386-8266-1. DOI: 10.23919/ACC45564.2020.9147655.

[Qi+18] Junjian Qi, Kai Sun, Jianhui Wang, and Hui Liu. “Dynamic State Estimation for Multi-Machine Power System by Unscented Kalman Filter With Enhanced Numerical Stability”. In: *IEEE Transactions on Smart Grid* 9.2 (2018), pp. 1184–1196. ISSN: 1949-3053. DOI: 10.1109/TSG.2016.2580584.

[RPK17] Maziar Raissi, Paris Perdikaris, and George Em Karniadakis. *Physics Informed Deep Learning (Part I): Data-driven Solutions of Nonlinear Partial Differential Equations*. 2017. DOI: 10.48550/arXiv.1711.10561.

[RSL21] Chengping Rao, Hao Sun, and Yang Liu. *Hard Encoding of Physics for Learning Spatiotemporal Dynamics*. 2021. DOI: 10.48550/arXiv.2105.00557.

[Ras16] Tariq Rashid. *Make Your own neural network*. Scotts Valley, California , US: CreateSpace Independent Publishing Platform, 2016. ISBN: 9781530826605.

[Roj96] Raúl Rojas. *Neural Networks. A Systematic Introduction*. 1st ed. Berlin, Heidelberg: Springer, 1996. ISBN: 9783642610684.

[SCJ13] Kirk Y. Scheper, Girish Chowdhary, and Eric N. Johnson. “Aerodynamic System Identification of Fixed-wing UAV”. In: *AIAA Atmospheric Flight Mechanics (AFM) Conference* (Boston, MA). Reston, Virginia: American Institute of Aeronautics and Astronautics, 2013. DOI: 10.2514/6.2013-4920.

[SES17] SESAR Joint Undertaking. *European drones outlook study: unlocking the value for Europe*. Publications Office, 2017. DOI: 10.2829/085259.

[Si+12] Xiao-Sheng Si, Wenbin Wang, Chang-Hua Hu, Dong-Hua Zhou, and Michael G. Pecht. “Remaining Useful Life Estimation Based on a Nonlinear Diffusion Degradation Process”. In: *IEEE Transactions on Reliability* 61.1 (2012), pp. 50–67. ISSN: 0018-9529. DOI: 10.1109/TR.2011.2182221.

[Sie+19] Philipp Maximilian Sieberg, Sebastian Blume, Nele Harnack, Niko Maas, and Dieter Schramm. “Hybrid State Estimation Combining Artificial Neural Network and Physical Model”. In: *2019 IEEE Intelligent Transportation Systems Conference (ITSC)* (Auckland, New Zealand, 27th Oct.–30th2019). IEEE, 2019, pp. 894–899. ISBN: 978-1-5386-7024-8. DOI: 10.1109/ITSC.2019.8916954.

[Sim06a] Dan Simon. “A game theory approach to constrained minimax state estimation”. In: *IEEE Transactions on Signal Processing* 54.2 (2006), pp. 405–412. ISSN: 1053-587X. DOI: 10.1109/TSP.2005.861732.

[Sim06b] Dan Simon. *Optimal State Estimation. Kalman, H [infinity] and nonlinear approaches*. Hoboken, NJ, USA: John Wiley & Sons, Inc, 2006. ISBN: 0471708585. DOI: 10.1002/0470045345.

[Sim10] Dan Simon. "Kalman filtering with state constraints: a survey of linear and nonlinear algorithms". In: *IET Control Theory & Applications* 4.8 (2010), pp. 1303–1318. ISSN: 1751-8644. DOI: 10.1049/iet-cta.2009.0032.

[SC02] Dan Simon and Tien Li Chia. "Kalman filtering with state equality constraints". In: *IEEE Transactions on Aerospace and Electronic Systems* 38.1 (2002), pp. 128–136. ISSN: 00189251. DOI: 10.1109/7.993234.

[SS05] Dan Simon and Donald L. Simon. "Aircraft Turbofan Engine Health Estimation Using Constrained Kalman Filtering". In: *Journal of Engineering for Gas Turbines and Power* 127.2 (2005), pp. 323–328. ISSN: 0742-4795. DOI: 10.1115/1.1789153.

[SS21] Henrik Simon and Sascha Schoenhof. "Enhancing the diagnostic performance of Condition Based Maintenance through the fusion of sensor with maintenance data". In: *Annual Conference of the PHM Society* 13.1 (2021). ISSN: 2325-0178. DOI: 10.36001/phmconf.2021.v13i1.2981.

[SGH19] Martin Skoglund, Fredrik Gustafsson, and Gustaf Hendeby. "On Iterative Unscented Kalman Filter using Optimization". In: *22th International Conference on Information Fusion (FUSION)*. Linköping University, Automatic Control and Linköping University, Faculty of Science & Engineering. 2019. ISBN: 978-0-9964527-8-6.

[Sta+18] Philipp Stahl, Thomas Seren, Christian Roessler, and Mirko Hornung. "Development and Performance Comparison of Optimized Electric Fixed-Wing VTOL UAV Configurations". In: *31st Congress of the International Council of the Aeronautical Sciences* (Belo Horizonte, Brazil). 2018.

[SJL16] Brian L. Stevens, Eric N. Johnson, and Frank L. Lewis. *Aircraft control and simulation. Dynamics, controls design, and autonomous systems*. Third edition. Hoboken, N.J.: John Wiley & Sons, 2016. ISBN: 978-1-118-87099-0.

[Tch+14] Pierre Tchakoua et al. "Wind Turbine Condition Monitoring: State-of-the-Art Review, New Trends, and Future Challenges". In: *Energies* 7.4 (2014), pp. 2595–2630. DOI: 10.3390/en7042595.

[Tei+08] Bruno O. S. Teixeira, Leonardo A. B. Torres, Luis A. Aguirre, and Dennis S. Bernstein. “Unscented filtering for interval-constrained nonlinear systems”. In: *2008 47th IEEE Conference on Decision and Control* (Cancun, Mexico, 9th Dec.–11th2008). IEEE, 2008, pp. 5116–5121. ISBN: 978-1-4244-3123-6. DOI: 10.1109/CDC.2008.4739141.

[Tho+20] N. Mohamed Thoiyab et al. “Global Stability Analysis of Neural Networks with Constant Time Delay via Frobenius Norm”. In: *Mathematical Problems in Engineering* 2020 (2020), pp. 1–14. ISSN: 1024-123X. DOI: 10.1155/2020/4321312.

[TR06] Mark B. Tischler and Robert K. Remple. *Aircraft and Rotorcraft System Identification*. Reston, VA: American Institute of Aeronautics and Astronautics, 2006. ISBN: 978-1-56347-837-6. DOI: 10.2514/4.861352.

[Vac+06] George Vachtsevanos, Frank Lewis, Michael Roemer, Andrew Hess, and Biqing Wu. *Intelligent Fault Diagnosis and Prognosis for Engineering Systems*. Hoboken, NJ, USA: John Wiley & Sons, Inc, 2006. ISBN: 9780470117842. DOI: 10.1002/9780470117842.

[vW03] Rudolph van der Merwe and Eric Wan. “Sigma-Point Kalman Filters for Probabilistic Inference in Dynamic State-Space Models”. In: *In Proceedings of the Workshop on Advances in Machine Learning*. 2003.

[van09] Ramon van Handel. “Observability and nonlinear filtering”. In: *Probabilty Theory and Related Fields* 145 (2009), pp. 35–75. DOI: 10.48550/arXiv.0708.3412.

[VDI22] VDI. *Intelligente Zustandsprognose und vorausschauende Instandhaltung. Prognostics and Health Management VDI-Statusreport*. 2022. URL: https://www.vdi.de/ueber-uns/presse/publikationen/details/intelligente-zustandsprognose-und-vorausschauende-instandhaltung (visited on 25. 08. 2022).

[VDI21] VDI/VDE, ed. *Entwicklung mechatronischer und cyber-physischer Systeme*. VDI/VDE 2206. Düsseldorf, DE, 1st Nov. 2021.

[VV15] Shijoh Vellayikot and Mathew V. Vaidyan. “ANN Approach for State Estimation of Hybrid Systems and Its Experimental Validation”. In: *Mathematical Problems in Engineering* (2015), pp. 1–13. ISSN: 1024-123X. DOI: 10.1155/2015/382324.

[VS21] Felipe A. C. Viana and Arun K. Subramaniyan. “A Survey of Bayesian Calibration and Physics-informed Neural Networks in Scientific Modeling”. In: *Archives of Computational Methods in Engineering* 28.5 (2021), pp. 3801–3830. ISSN: 1134-3060. DOI: `10.1007/s11831-021-09539-0`.

[Vol14] Allan J. Volponi. “Gas Turbine Engine Health Management: Past, Present, and Future Trends”. In: *Journal of Engineering for Gas Turbines and Power* 136.5 (2014). ISSN: 0742-4795. DOI: `10.1115/1.4026126`.

[WMN00] Eric Wan, Rudolph Merwe, and Alex Nelson. “Dual Estimation and the Unscented Transformation”. In: *Advances in Neural Information Processing Systems* 12 (2000).

[WN00] Eric Wan and Alex Nelson. “Dual Kalman Filtering Methods for Nonlinear Prediction, Smoothing, and Estimation”. In: *Advances in Neural Information Processing Systems* (2000).

[Wan09] Yingxu Wang. “On Abstract Intelligence”. In: *International Journal of Software Science and Computational Intelligence* 1.1 (2009), pp. 1–17. ISSN: 1942-9045. DOI: `10.4018/jssci.2009010101`.

[WC11] Zhongsheng Wang and Xiaolei Chen. “Condition Monitoring of Aircraft Sudden Failure”. In: *Procedia Engineering* 15 (2011), pp. 1308–1312. ISSN: 18777058. DOI: `10.1016/j.proeng.2011.08.242`.

[Wen12] Juyang Weng. *Natural and artificial intelligence. Introduction to computational brain-mind*. Okemos, Mich.: BMI Press, 2012. ISBN: 978-0-9858757-2-5.

[WB18] Burkard Wigger and Jürgen Beyer. “Grundlagen der Navigation I&II. Skript zur Vorlesung”. Darmstadt, 5th Dec. 2018.

[Wol+21] Taco de Wolff, Hugo Carrillo, Luis Martí, and Nayat Sanchez-Pi. “Assessing Physics Informed Neural Networks in Ocean Modelling and Climate Change Applications”. In: *AI: Modeling Oceans and Climate Change Workshop at ICLR 2021* (Santiago (Virtual), Chile, 7th May 2021). 2021.

[WW12] Xuedong Wu and Yaonan Wang. “Extended and Unscented Kalman filtering based feedforward neural networks for time series prediction”. In: *Applied Mathematical Modelling* 36.3 (2012), pp. 1123–1131. ISSN: 0307904X. DOI: `10.1016/j.apm.2011.07.052`.

[Yan18] Yann Cambier. “Big Data: Racing to Platform Maturity”. In: *Aircraft IT MRO* Winter 2018-2019 (2018), pp. 38–46.

[Yin19] Joe Yin. “A Conceptual Intelligent Aircraft System”. In: *2019 IEEE/AIAA 38th Digital Avionics Systems Conference (DASC)* (San Diego, CA, USA, 8th Sept.–12th2019). IEEE, 2019, pp. 1–7. ISBN: 978-1-7281-0649-6. DOI: 10.1109/DASC43569.2019.9081719.

[ZLL17] Huiping Zhuang, Jieying Lu, and Junhui Li. “Joint estimation of state and parameter with maximum likelihood method”. In: *2017 36th Chinese Control Conference (CCC)* (Dalian, China, 26th July–28th2017). IEEE, 2017, pp. 5276–5281. ISBN: 978-988-15639-3-4. DOI: 10.23919/ChiCC.2017.8028190.

Related Student Theses

All theses listed were prepared at the Institute of Flight Systems and Automatic Control in cooperation with this work or supervised by the author.

[Fic+20] Uwe Fickenscher, Alexander Fuchs, Andrés Larena Valdes, Stefan Schließ, and Johannes Stiller. “Entwicklung und Integration von Degradations-Modulen zur Modifikation eines hybriden unbemannten Luftfahrzeugs für die künstliche Einbringung von Degradationseffekten”. Institut für Flugsysteme und Regelungstechnik. Advanced Design Project Report. Darmstadt: Technische Universität Darmstadt, 2020.

[Frö+19] Tilman Fröhlich, Richard Gerspach, Clemens Janzharik, Jan Mölleney, and Daniel Ribnitzky. “Entwicklung eines hybriden unbemannten Luftfahrzeugs (VTOL-UAV)”. Institut für Flugsysteme und Regelungstechnik. Advanced Design Project Report. Darmstadt: Technische Universität Darmstadt, 2019.

[Joe22] Alexander Joest. “Flight data-based evaluation of the energy efficiency of a hybrid unmanned aerial vehicle using machine learning methods”. Institut für Flugsysteme und Regelungstechnik. Master’s thesis. Darmstadt: Technische Universität Darmstadt, 2022.

Appendices

A. Supplementary information to the UAS flight test platform

A.1. Technical drawing of the hybrid Scihunter UAS

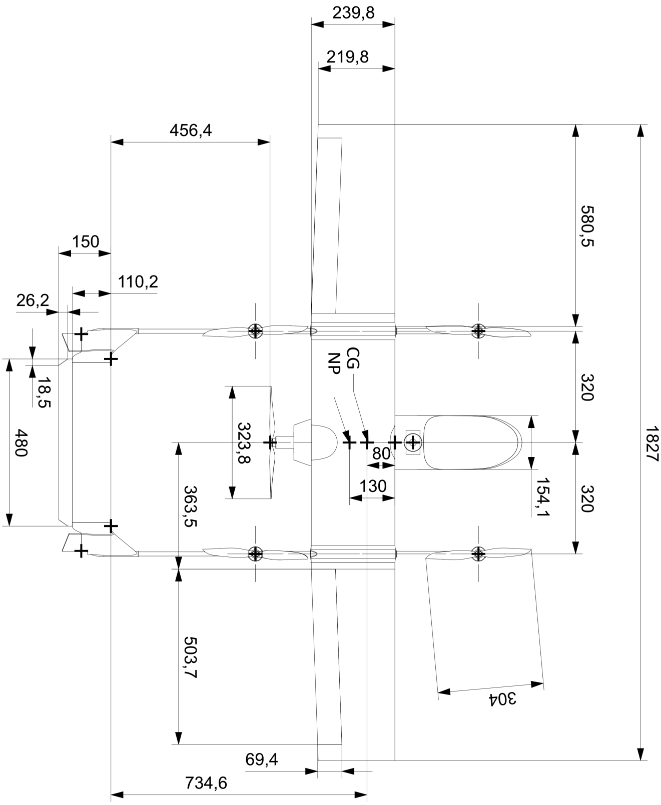


Figure A.1.: Technical drawing of the hybrid Scihunter UAS, top view [Frö+19]

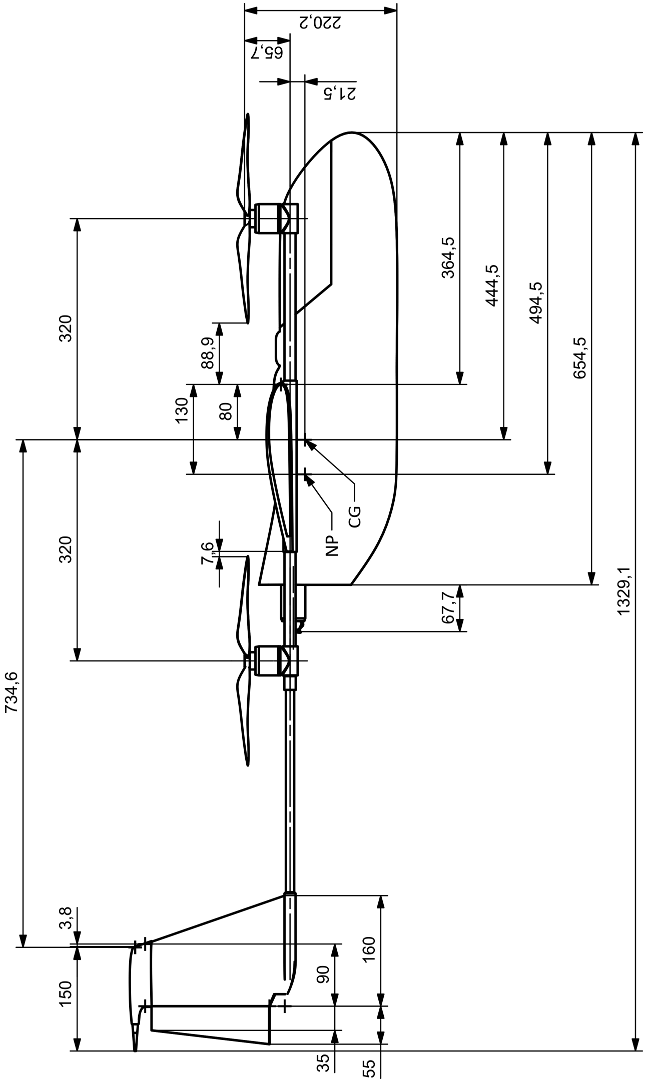


Figure A.2.: Technical drawing of the hybrid Scihunter UAS, side view [Frö+19]

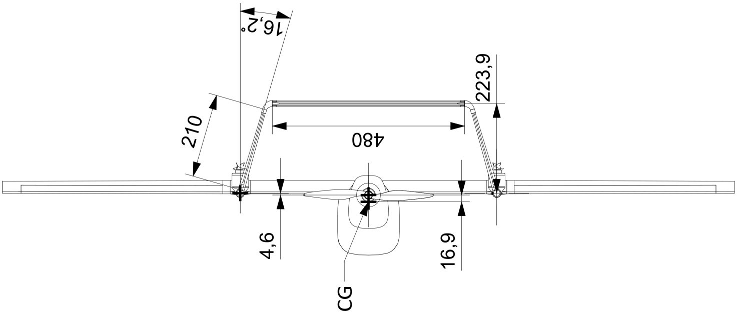


Figure A.3.: Technical drawing of the hybrid Scihunter UAS, rear view [Frö+19]

A.2. Motorization of the hybrid Scihunter UAS

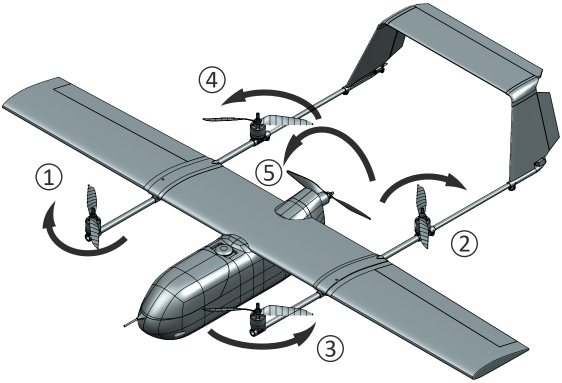


Figure A.4.: Illustration of motor assignments and direction of rotation

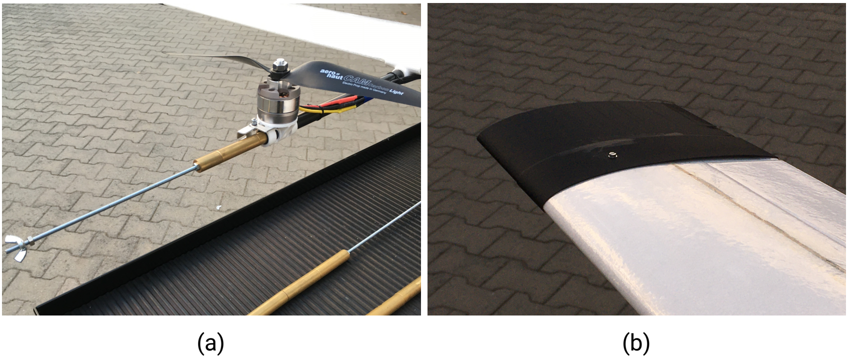
Table A.1.: Motor assignments and direction of rotation

Motor position	motor type	direction of rotation
1	lift motor	clockwise (viewed from top)
2	lift motor	clockwise (viewed from top)
3	lift motor	counter clockwise (viewed from top)
4	lift motor	counter clockwise (viewed from top)
5	pusher motor	clockwise (viewed from rear)

Table A.2.: Technical description of the lift and pusher motors used

	Pusher motor	Lift motor
Description	Roxxy BL Outrunner C35-48-06 (700KV)	Roxxy BL Outrunner C42-40-12 (850KV)
Manufacturer	Multiplex	Multiplex
Nominal rotation speed	$850kV$	$700kV$
Operating voltage	$7V$ - $15V$	$10V$ - $20V$
Continuous current	$35A$	$42A$
Power	$480W$	$750W$
Mass	$130g$	$172g$
Max. rotation speed	$12580rpm$	$10360rpm$

A.3. Mounting the technical modifications of the UAS



(a) (b)

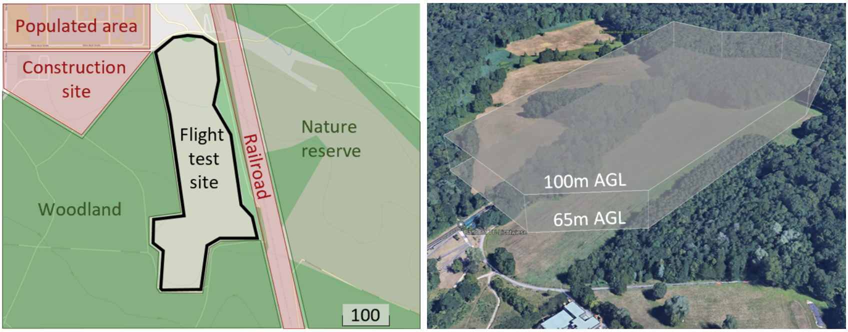
Figure A.5.: Mounting of weight modules on the right wing of the UAS (a) and the mounted lift module on the right wing tip (b) [Fic+20].

A.4. Flight test environment

A designated flight test site at the TU Darmstadt is used as flight test environment. It is located at Campus Lichtwiese in Darmstadt, near the university building L1|05, where the FSR operates its test facilities and a workshop for various flight test platforms. The close logistical connection allows flight testing including the recharging of battery packs, the implementation of modifications, repairs, and access to the power supply for operating a ground station.

The UAS weights less than $5kg$ and is operated in a Visual Line Of Side (VLOS), in a "controlled ground area that might be located in a sparsely populated area" and "[...] with low risk to encounter with manned aircraft" [Eur22].

The flight test site is limited in length and width to about $410m \times 115m$, describing the dimensions of the flight geography area on the ground; see Figure A.6a. The meadow on the ground is surrounded by woods up to $35m$ high, leading to the decision to operate the UAS at a minimum of $75m$ AGL in FW flight; see Figure A.6b. The upper



(a)

(b)

Figure A.6.: Location of the flight test site of the TU Darmstadt, based on openaip[1](a) and implemented geofence to control the operational volume, based on [Joe22] and google earth[2](b)

limit is set by a maximum AGL of $95m$ for the planned flight operations following the rules defined by TU Darmstadt and EASA [Eur22].

The contingency volume is set, considering the flight test environment and the EASA rules. When leaving the planned flight path[3], the UAS returns to the flight path, using contingency volume and increasing alertness. To ensure the UAS not leaving the contingency and thus the operational volume in flight operations, emergency procedures have to be defined.

The emergency procedures implement FaSs of the FMU using the ArduPilot Mission Planner software. A FaS is triggered when leaving the operational volume, realized by implementing a geofence[4]. The defined geofence describes an operational volume of about $480m$ in length, $215m$ in width, and from $65m$ AGL to $100m$ AGL in height, see Figure A.6b.

[1] https://www.openaip.net/map#15.15/49.857512/8.694495

[2] Map basis: Darmstadt Lichtwiese [online], 49°51‘35“N 8°41‘06“E, Altitude 178m, Google Earthfor Chrome, GeoBasis-DE/BKG (©2009), URL: http://www.google.com/earth

[3] Planning the flight path inside the flight geography volume is seen as a contingency procedure as it is suggested by EASA [Eur22].

[4] A geofence defines the measured GPS position to be in a certain space. In case of leaving the defined space specific procedures are triggered to return into the defined space, based on https://ardupilot.org/.

A.5. Specification of the flight path using waypoints

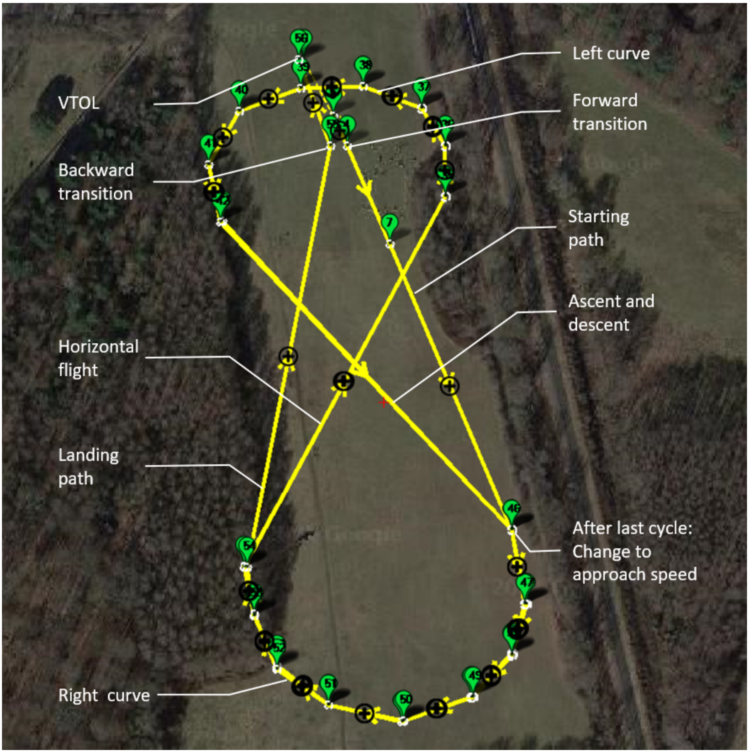


Figure A.7.: Specified flight path using waypoints in ardupilot mission planner [Joe22]

A.6. Definition of the waypoints used in ardupilot mission planner

The waypoints describe the planned trajectory of the UAS. The standard FM covers 10 repetitions of the figure-of-eight-profile. The *.waypoints*-file is given below to reproduce flight tests and simulations if needed.

```
QGC WPL 110
0 1 0 16 0 0 0 0 49.860388 8.686638 176.060000 1
1 0 3 84 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 75.000000 1
2 0 0 3000 3.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.000000 1
3 0 3 16 0.00000000 0.00000000 0.00000000 0.00000000 49.86015630 8.68682470 75.000000 1
4 0 3 16 0.00000000 0.00000000 0.00000000 0.00000000 49.86003010 8.68690790 75.000000 1
5 0 0 3000 4.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.000000 1
6 0 0 178 0.00000000 12.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.000000 1
7 0 3 16 0.00000000 0.00000000 0.00000000 0.00000000 49.85962550 8.68719220 75.000000 1
8 0 3 16 0.00000000 0.00000000 0.00000000 0.00000000 49.85843240 8.68797530 75.000000 1
9 0 3 16 0.00000000 0.00000000 0.00000000 0.00000000 49.85812110 8.68807200 75.000000 1
10 0 3 16 0.00000000 0.00000000 0.00000000 0.00000000 49.85791360 8.68797530 75.000000 1
11 0 3 16 0.00000000 0.00000000 0.00000000 0.00000000 49.85773380 8.68771790 75.000000 1
12 0 3 16 0.00000000 0.00000000 0.00000000 0.00000000 49.85763010 8.68726730 75.000000 1
13 0 3 16 0.00000000 0.00000000 0.00000000 0.00000000 49.85769920 8.68679520 75.000000 1
14 0 3 16 0.00000000 0.00000000 0.00000000 0.00000000 49.85785820 8.68646260 75.000000 1
15 0 3 16 0.00000000 0.00000000 0.00000000 0.00000000 49.85807270 8.68631240 75.000000 1
16 0 3 16 0.00000000 0.00000000 0.00000000 0.00000000 49.85827330 8.68625340 75.000000 1
17 0 3 16 0.00000000 0.00000000 0.00000000 0.00000000 49.85982260 8.68754630 75.000000 1
18 0 3 16 0.00000000 0.00000000 0.00000000 0.00000000 49.86003010 8.68754630 75.000000 1
19 0 3 16 0.00000000 0.00000000 0.00000000 0.00000000 49.86018920 8.68739600 75.000000 1
20 0 3 16 0.00000000 0.00000000 0.00000000 0.00000000 49.86028080 8.68702040 75.000000 1
21 0 3 16 0.00000000 0.00000000 0.00000000 0.00000000 49.86027220 8.68661280 75.000000 1
22 0 3 16 0.00000000 0.00000000 0.00000000 0.00000000 49.86017540 8.68621590 75.000000 1
23 0 3 16 0.00000000 0.00000000 0.00000000 0.00000000 49.85995400 8.68602280 75.000000 1
24 0 3 16 0.00000000 0.00000000 0.00000000 0.00000000 49.85971890 8.68610860 75.000000 1
25 0 0 177 8.00000000 2.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.000000 1
26 0 3 16 0.00000000 0.00000000 0.00000000 0.00000000 49.85843070 8.68797530 95.000000 1
27 0 3 16 0.00000000 0.00000000 0.00000000 0.00000000 49.85811770 8.68807200 95.000000 1
```

```
28 0 3 16 0.00000000 0.00000000 0.00000000 0.00000000 49.85791020 8.68798080 95.000000 1
29 0 3 16 0.00000000 0.00000000 0.00000000 0.00000000 49.85773210 8.68771790 95.000000 1
30 0 3 16 0.00000000 0.00000000 0.00000000 0.00000000 49.85763180 8.68726460 95.000000 1
31 0 3 16 0.00000000 0.00000000 0.00000000 0.00000000 49.85769920 8.68679790 95.000000 1
32 0 3 16 0.00000000 0.00000000 0.00000000 0.00000000 49.85786180 8.68646260 95.000000 1
33 0 3 16 0.00000000 0.00000000 0.00000000 0.00000000 49.85807450 8.68631780 95.000000 1
34 0 3 16 0.00000000 0.00000000 0.00000000 0.00000000 49.85827680 8.68625340 95.000000 1
35 0 3 16 0.00000000 0.00000000 0.00000000 0.00000000 49.85982260 8.68754090 95.000000 1
36 0 3 16 0.00000000 0.00000000 0.00000000 0.00000000 49.86003010 8.68754630 95.000000 1
37 0 3 16 0.00000000 0.00000000 0.00000000 0.00000000 49.86019260 8.68740140 95.000000 1
38 0 3 16 0.00000000 0.00000000 0.00000000 0.00000000 49.86027910 8.68701520 95.000000 1
39 0 3 16 0.00000000 0.00000000 0.00000000 0.00000000 49.86027050 8.68661020 95.000000 1
40 0 3 16 0.00000000 0.00000000 0.00000000 0.00000000 49.86017710 8.68621310 95.000000 1
41 0 3 16 0.00000000 0.00000000 0.00000000 0.00000000 49.85995580 8.68602010 95.000000 1
42 0 3 16 0.00000000 0.00000000 0.00000000 0.00000000 49.85971540 8.68610050 95.000000 1
43 0 0 177 26.00000000 2.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.000000 1
44 0 0 177 8.00000000 2.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.000000 1
45 0 0 178 0.00000000 12.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.000000 1
46 0 3 16 0.00000000 0.00000000 0.00000000 0.00000000 49.85843240 8.68797530 75.000000 1
47 0 3 16 0.00000000 0.00000000 0.00000000 0.00000000 49.85811770 8.68806660 75.000000 1
48 0 3 16 0.00000000 0.00000000 0.00000000 0.00000000 49.85790670 8.68797530 75.000000 1
49 0 3 16 0.00000000 0.00000000 0.00000000 0.00000000 49.85773380 8.68772860 75.000000 1
50 0 3 16 0.00000000 0.00000000 0.00000000 0.00000000 49.85763690 8.68727270 75.000000 1
51 0 3 16 0.00000000 0.00000000 0.00000000 0.00000000 49.85769920 8.68679790 75.000000 1
52 0 3 16 0.00000000 0.00000000 0.00000000 0.00000000 49.85785480 8.68646260 75.000000 1
53 0 3 16 0.00000000 0.00000000 0.00000000 0.00000000 49.85807620 8.68631240 75.000000 1
54 0 3 16 0.00000000 0.00000000 0.00000000 0.00000000 49.85826990 8.68626420 75.000000 1
55 0 3 16 0.00000000 0.00000000 0.00000000 0.00000000 49.86002670 8.68680060 75.000000 1
56 0 3 85 0.00000000 0.00000000 0.00000000 0.00000000 49.86039150 8.68660740 0.000000 1
```

A.7. Flight tests using the aerodynamically modified UAS

Table A.3.: Planned repetitions and distance to fly within the respective FM using the aerodynamically modified UAS

FM	1	2	3	4	5	6	7	8	9
TOW in kg	4.55			4.78			4.99		
CA in m/s	15	12	18	15	12	18	15	12	18
Repetitions at $75m$ AGL	5	5	5	5	5	3	4	-	3
Repetitions at $95m$ AGL	5	5	5	5	5	3	4	-	3
Planned distance in km	9.8	9.8	9.8	9.8	9.8	5.6	7.2	-	5.6

Table A.4.: Flight distance and duration according to the CA and TOW using the aerodynamically modified UAS

(a) Flown flight distance in km

		CA in m/s		
		12	15	18
TOW in kg	4.55	10.67	10.88	10.78
	4.78	10.24	11.26	7.27
	4.99	-	8.96	7.14

(b) Duration of flights in min

		CA in m/s		
		12	15	18
TOW in kg	4.55	14:59	12:54	11:10
	4.78	15:41	12:26	8:11
	4.99	-	11:06	8:42

A.8. Flight simulations using the aerodynamically modified and unmodified UAS

Within the simulation environment safety measures are discarded. In conclusion, all flight simulations were planned and executed with 5 repetitions at both $75m$ AGL and $95m$ AGL with a flight distance of $9.8km$ per mission. Since the recorded flight distance and duration of the aerodynamically modified and unmodified UAS are comparable, only the values for the flight simulations performed with the aerodynamically unmodified UAS are given in Tab. A.5.

Table A.5.: Flight distance and duration according to the CA and TOW using the aerodynamically unmodified UAS within the simulation environment

(a) Flown flight distance in km

		CA in m/s		
		12	15	18
TOW in kg	4.39	9.85	10.10	10.54
	4.62	9.84	10.10	10.53
	4.82	9.84	10.10	10.50

(b) Duration of flights in min

		CA in m/s		
		12	15	18
TOW in kg	4.39	15:17	13:40	12:31
	4.62	15:14	13:39	12:31
	4.82	15:11	13:37	12:28

A.9. Initial parameter set describing the Scihunter UAS

The initial parameter set given in Table A.6 involves the parameters based on prior flight testing and wind tunnel tests. They are used in the respective flight phase. The parameter set is valid for the physical model introduced and used by Prochazka, Krüger, and Ribnitzky in [PKR20]. As the physical model used in this thesis has been changed by neglecting some minor influences on the aircraft dynamics, the number of physical parameters reduces and the validity of the initial parameter set below is limited.

However, the initial parameters serve as a starting point for parameter estimation. The parameters concerning the FW flight are valid in $V_a \in [11, 20] m/s$ and $h \in [0, 100] m$. Besides the TAS, the altitude h is used to constrain the validity of the parameter sets. The altitude h describes AGL at an air density of about $\rho_0 = 1.225 kg/m^3$. The parameters concerning the QC flight are valid at $V_a \in [0, 11] m/s$ and $h \in [0, 100] m$.

Table A.6.: Initial parameter set used for the adjustment of the physical model of the UAS, based on [PKR20]

C_{D_0}	0.0754	C_{L_η}	0.2693	C_{m_η}	-1.2955	c_{p_3}	2.3478
C_{D_α}	0.1719	C_{L_q}	11.3302	C_{m_q}	-22.110	c_{p_4}	2.4074
C_{Y_β}	-0.6337	C_{ℓ_β}	-0.0688	C_{n_β}	0.2939	$C_{T_{\omega 2}}$	$0.41 * 10^{-2}$
C_{Y_ζ}	0.1799	C_{ℓ_ξ}	-0.4085	C_{n_ζ}	-0.1564	C_{n_T}	$-5.47 * 10^{-7}$
C_{Y_p}	0.0813	C_{ℓ_p}	-2.0780	C_{n_p}	-0.1649	c_{p_P}	2.3987
C_{Y_r}	0.3700	C_{ℓ_r}	0.3242	C_{n_r}	-0.1550	$C_{T_{\omega 2_P}}$	$1.37 * 10^{-2}$
C_{L_0}	0.4075	C_{m_0}	$-4.9 * 10^{-2}$	c_{p_1}	2.3480	$C_{T_{u 2\omega_P}}$	$-1.29 * 10^{-2}$
C_{L_α}	5.4889	C_{m_α}	-1.3573	c_{p_2}	2.4738		

A.10. Calibration of the aerodynamic control surfaces of the Scihunter UAS

Table A.7.: Calibrated control surfaces of the UAS using linear regression functions

Control surface	Calibration of the deflection related to the PWM signal
Aileron left	$\xi_l = 0.0669 \frac{\circ}{\mu s} u_{pwm} - 97.78\,\text{deg}$
Aileron right	$\xi_r = 0.0688 \frac{\circ}{\mu s}\ u_{pwm} - 105.6\,\text{deg}$
Rudder left	$\zeta_l = -0.0666 \frac{\circ}{\mu s}\ u_{pwm} - 98.64\,\text{deg}$
Rudder right	$\zeta_r = -0.0646 \frac{\circ}{\mu s}\ u_{pwm} - 97.58\,\text{deg}$
Elevator	$\eta = 0.0473 \frac{\circ}{\mu s}\ u_{pwm} - 67.98\,\text{deg}$

A.11. Additional information on the initialization of the SCmUKF

Table A.8.: Additional deviation of selected dynamical states used for initialization

Description	Sign	Value	Unit
Position north	x_{p_N}	$x_{p_{N_{ini}}} + 1$	m
Position east	x_{p_E}	$x_{p_{E_{ini}}} + 1$	m
Position down	x_{p_D}	$x_{p_{D_{ini}}} + 0.5$	m
Body speed in x-direction	x_u	$x_{u_{ini}} + 0.1$	m/s
Body speed in y-direction	x_v	$x_{v_{ini}} + 0.1$	m/s
Body speed in z-direction	x_w	$x_{w_{ini}} + 0.1$	m/s
Yaw angle	x_ψ	$x_{\psi_{ini}} +$ 5e-2	rad

Table A.9.: Standard deviation r_{yy} of the UAS aerodynamic sensor measurements related to the flight phase

Description	Sign	Value in flight phase			Unit
		QC	TR	FW	
Flow velocity in x-dir.	r_{yy_u}	15	2.5	5e-1	m/s
Angle of attack	r_{yy_α}	360	4π	$\pi/180$	rad
Sideslip angle	r_{yy_β}	360	4π	$\pi/180$	rad

Table A.10.: Standard deviation r_{yy} of the Scihunter UAS sensor measurements based on sensor specifics and flight data analysis

Description	Sign	Value	Unit
Position north	$r_{yy_{pN}}$	2	m
Position east	$r_{yy_{pE}}$	2	m
Position down	$r_{yy_{pD}}$	5	m
Acceleration in x-direction	$r_{yy_{accx}}$	5e-2	m/s^2
Acceleration in y-direction	$r_{yy_{accy}}$	5e-2	m/s^2
Acceleration in z-direction	$r_{yy_{accz}}$	5e-2	m/s^2
Roll rate	r_{yy_p}	4e-4	rad/s
Pitch rate	r_{yy_q}	4e-4	rad/s
Yaw rate	r_{yy_r}	4e-4	rad/s
Roll angle	r_{yy_ϕ}	1e-1	rad
Pitch angle	r_{yy_θ}	1e-1	rad
Yaw angle	r_{yy_ψ}	1e-1	rad
Flow velocity in x-direction	r_{yy_u}	5e-1	m/s
Angle of attack	r_{yy_α}	$\pi/180$	rad
Sideslip angle	r_{yy_β}	$\pi/180$	rad
Aileron deflection left	$r_{yy_{\xi l}}$	5e-2	rad
Aileron deflection right	$r_{yy_{\xi r}}$	5e-2	rad
Elevator deflection	r_{yy_η}	5e-2	rad
Rudder deflection left	$r_{yy_{\zeta l}}$	5e-2	rad
Rudder deflection right	$r_{yy_{\zeta r}}$	5e-2	rad
Power pusher motor	$r_{yy_{PP}}$	5e-2	kW
Power lift motor 1	$r_{yy_{P1}}$	4e-2	kW
Power lift motor 2	$r_{yy_{P2}}$	4e-2	kW
Power lift motor 3	$r_{yy_{P3}}$	4e-2	kW
Power lift motor 4	$r_{yy_{P4}}$	4e-2	kW
Power overall	$r_{yy_{Pbat}}$	4e-2	kW
Rotational rate pusher motor (normed)	$r_{yy_{\omega P}}$	1e-2	—
Rotational rate lift motor 1 (normed)	$r_{yy_{\omega 1}}$	1e-2	—
Rotational rate lift motor 2 (normed)	$r_{yy_{\omega 2}}$	1e-2	—
Rotational rate lift motor 3 (normed)	$r_{yy_{\omega 3}}$	1e-2	—
Rotational rate lift motor 4 (normed)	$r_{yy_{\omega 4}}$	1e-2	—
GPS speed in x-direction	$r_{yy_{ug}}$	0.5	m/s
GPS speed in y-direction	$r_{yy_{vg}}$	0.5	m/s
GPS speed in z-direction	$r_{yy_{wg}}$	1	m/s

Three parameter states are considered with SC. The mean value equals the initially used parameter value and is considered a pseudo measurement. The noise of the pseudo measurement is set to be $r_{yy} = 2$ as shown in Table A.11. In addition, the resulting lower and upper limit of the parameter SC are given.

Table A.11.: State constraints (SC) of the parameter states

Parameter state	Mean	Lower limit	Upper limit	r_{yy}
C_{D_0}	0.132	-0.132	0.396	2
C_{L_0}	0.614	-0.614	1.842	2
C_{M_0}	-4.61e-3	-13.83e-3	4.61e-3	2

Table A.12.: Main diagonal of the system noise $P_{xx_{ini}}$ of the parameter states of the physical model of the Scihunter UAS

Description	Sign	Value	Unit
Parameter C_{D0}	$P_{xx\,CD0}$	7e-4	–
Parameter C_{L0}	$P_{xx\,CL0}$	1.5e-2	–
Parameter C_{M0}	$P_{xx\,CM0}$	1e-4	–

The noise concerning the system's dynamical states is set based on filter tuning and flight data analysis, see Table A.13. The system noise of the parameter states depends on the initial parameters:

$$q_{xx_p} = \left|\hat{x}_{p_{ini}}\right| 1\text{e-}2. \tag{A.1}$$

The system noise of the ANN weight states is set to be small with $q_{xx_{nn}} = 5\text{e-}3$. In summary, the system noise covariance matrix is defined $Q_{xx} = q_{xx}^2 1\text{e-}2$, considering entries of the main diagonals exclusively.

Table A.13.: Main diagonal of the system noise q_{xx} of the dynamical states of the Scihunter UAS based on flight data analysis

Description	Sign	Value	Unit
Position north	$q_{xx_{p_N}}$	1	m
Position east	$q_{xx_{p_E}}$	1	m
Position down	$q_{xx_{p_D}}$	3	m
Body speed in x-direction	q_{xx_u}	0.5	m/s
Body speed in y-direction	q_{xx_v}	0.5	m/s
Body speed in z-direction	q_{xx_w}	0.5	m/s
Roll rate	q_{xx_p}	0.5	rad/s
Pitch rate	q_{xx_q}	0.5	rad/s
Yaw rate	q_{xx_r}	0.5	rad/s
Roll angle	q_{xx_ϕ}	1e-3	rad
Pitch angle	q_{xx_θ}	1e-3	rad
Yaw angle	q_{xx_ψ}	1e-3	rad
Wind speed in geo x-direction	$q_{xx_{u_w}}$	1e-3	m/s
Wind speed in geo y-direction	$q_{xx_{v_w}}$	1e-3	m/s
Wind speed in geo z-direction	$q_{xx_{w_w}}$	1e-3	m/s
Aileron deflection left	$q_{xx_{\xi_l}}$	1e-9	rad
Aileron deflection right	$q_{xx_{\xi_r}}$	1e-9	rad
Elevator deflection	q_{xx_η}	1e-9	rad
Rudder deflection left	$q_{xx_{\zeta_l}}$	1e-9	rad
Rudder deflection right	$q_{xx_{\zeta_r}}$	1e-9	rad
Power pusher motor	$q_{xx_{P_P}}$	1e-9	kW
Power lift motor 1	$q_{xx_{P_1}}$	1e-9	kW
Power lift motor 2	$q_{xx_{P_2}}$	1e-9	kW
Power lift motor 3	$q_{xx_{P_3}}$	1e-9	kW
Power lift motor 4	$q_{xx_{P_4}}$	1e-9	kW

Table A.14.: Main diagonal of the system noise $P_{xx_{ini}}$ of the dynamical states of the Scihunter UAS

Description	Sign	Value	Unit
Position north	$P_{xx_{p_N}}$	3	m^2
Position east	$P_{xx_{p_E}}$	3	m^2
Position down	$P_{xx_{p_D}}$	9	m^2
Body speed in x-direction	P_{xx_u}	1	$(m/s)^2$
Body speed in y-direction	P_{xx_v}	1	$(m/s)^2$
Body speed in z-direction	P_{xx_w}	1	$(m/s)^2$
Roll rate	P_{xx_p}	0.25	$(rad/s)^2$
Pitch rate	P_{xx_q}	0.25	$(rad/s)^2$
Yaw rate	P_{xx_r}	0.25	$(rad/s)^2$
Roll angle	P_{xx_ϕ}	1e-6	rad
Pitch angle	P_{xx_θ}	6.5e-5	rad
Yaw angle	P_{xx_ψ}	2.5e-3	rad
Wind speed in geo x-direction	$P_{xx_{u_w}}$	1e-6	m/s
Wind speed in geo y-direction	$P_{xx_{v_w}}$	1e-6	m/s
Wind speed in geo z-direction	$P_{xx_{w_w}}$	1e-6	m/s
Aileron deflection left	$P_{xx_x i_l}$	7.2e-3	rad
Aileron deflection right	$P_{xx_{\xi_r}}$	2.0e-3	rad
Elevator deflection	P_{xx_η}	3.4e-3	rad
Rudder deflection left	$P_{xx_{\zeta_l}}$	3.9e-3	rad
Rudder deflection right	$P_{xx_{\zeta_r}}$	12.2e-3	rad
Power pusher motor	$P_{xx_{P_P}}$	6.4e-4	kW
Power lift motor 1	$P_{xx_{P_1}}$	1.6e-6	kW
Power lift motor 2	$P_{xx_{P_2}}$	3.6e-4	kW
Power lift motor 3	$P_{xx_{P_3}}$	6.2e-5	kW
Power lift motor 4	$P_{xx_{P_4}}$	2.1e-2	kW

B. Matlab code of the physical model implemented

The physical model of the hybrid Scihunter UAS is implemented in Matlab environment. The Matlab code is optimized to efficiently process the SCmUKF using the functions of the forces and moments (B.1), the flight dynamics (B.2), and the sensor model (B.3) by implementing the sigma points as a vector. The Euler discretization used for the state prediction, the transformations between the CSs used, the direction cosine matrix, and the implementation of the mUKF are given in the literature and are not included in the Matlab code below [EK22; SJL16; BAL11]. The structure of the code connecting the different Matlab functions is shown in Figure B.1.

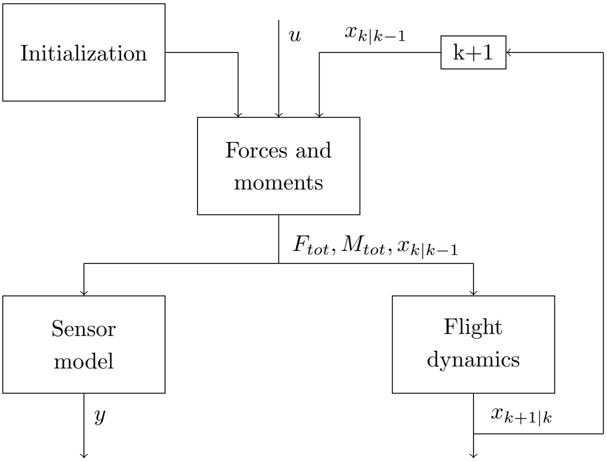


Figure B.1.: Structure of the Matlab code of the physical model using Euler time discretization at timestep k.

In the initialization block constants, physical parameters and initial state conditions are defined. In total 46 constants, 31 parameters, 25 states, 12 inputs, 34 measurements, and the resulting forces and moments are considered. Parts of the the forces and moments block are used in both the calculation of the flight dynamics and the sensor model.

In the Matlab code a calibration value of the pusher motor is considered an additional input. The calibration value had to be adjusted in the FMU during flight testing in order to proceed safely.

B.1. Forces and moments

```
function [F_Total, M_Total, Omega_speeds, u_k1_calb] = ...
    HybUAS_forces_moments_simple5_rad_fct(x_k,Parameters_k, u_k, ...
    delta_t, constants,eq,n_sigma)
% The calculation of the forces and moments of the hybrid UAS

Parameters_k = [Parameters_k ones(size(Parameters_k,1),1).*...
    constants(sum(constants(end-2:end-1))+1:sum(constants(end-2:end)))];

%% Constants

% S         = 0.41;             % m^2 Reference wing area
% b         = 0.9135;           % m Half wingspan
% mac       = 0.23;             % m Mean Aerodynamic chord
% r_xy      = 0.32;             % m x and y distance to rotors from CG
% r_z       = 0.0872;           % m z distance to rotors from CG
% d_pusher  = 0.3302;           % m diameter of pusher rotor
% d_liftr   = 0.3048;           % m diameter of lift rotors
% A_R_pusher= 0.0823;           % m^2
% A_R_liftr = 0.073;            % m^2

% Moments of Inertia (UAS without payload)
% (constants(19:24))
% J_x_ini = 0.2508;           % Moment of Inertia in x-direction in kg/m^2
% J_y_ini = 0.2902;           % Moment of Inertia in y-direction in kg/m^2
% J_z_ini = 0.5055;           % Moment of Inertia in z-direction in kg/m^2
% J_xz_ini = 0.0308;          % Moment of Inertia in xz-direction in kg/m^2
% J_yz_ini = 0;               % Moment of Inertia in yz-direction in kg/m^2
% J_xy_ini = 0;               % Moment of Inertia in xy-direction in kg/m^2

% omega_Puhser_max = 1085;      % max. rot. speed of pusher motor in rad/s
```

```
                                % (10360 rpm *pi/30)
% omega_Liftr_max = 1317;       % max. rot. speed for lift motor in rad/s
                                % (12580 rpm *pi/30)

S           = constants(1);     % m^2 Reference wing area
b           = constants(2);     % m d(1) Half wingspan
mac         = constants(3);     % m Mean Aerodynamic chord
r_xy        = constants(4);     % m d(3) x and y distance to rotors from CG
% r_z       = constants(5);     % m d(4) z distance to rotors from CG
d_pusher    = constants(6);     % m d(5) diameter of pusher rotor
d_liftr     = constants(7);     % m d(5) diameter of pusher rotor
A_R_pusher  = constants(8);     % m^2 rotor surface
A_R_liftr   = constants(9);     % m^2 rotor surface
g           = constants(10);    % m/s^2 gravitation constant
rho         = constants(11);    % kg/m^3 air density at sea level

% Input u_pwm coefficients
% gradient, u_calb_base, K_pwm, T_pwm parameters
% for calibration and PT1 correction

parameters_xi_L_pwm = constants(28:30);     % Aileron left pwm signal
parameters_xi_R_pwm = constants(31:33);     % Aileron right pwm signal
parameters_eta_pwm = constants(34:36);      % Elevator pwm signal
parameters_zeta_L_pwm = constants(37:39);   % Rudder left pwm signal
parameters_zeta_R_pwm = constants(40:42);   % Rudder right pwm signal

parameters_P_P_pwm = constants(43:45);     % Pusher motor
parameters_P1_pwm = constants(46:48);      % Front right lift rotor pwm
parameters_P2_pwm = constants(49:51);      % Back left lift rotor pwm
parameters_P3_pwm = constants(52:54);      % Front left lift rotor pwm
parameters_P4_pwm = constants(55:57);      % Back right lift rotor pwm

%% Input u

% pwm signals
xi_L_pwm    = u_k(1); % aileron left pwm
xi_R_pwm    = u_k(2); % aileron right pwm
eta_pwm     = u_k(3); % elevator pwm
zeta_L_pwm  = u_k(4); % rudder left pwm
zeta_R_pwm  = u_k(5); % rudder right pwm

P_P_pwm = u_k(6);   % target power value pusher pwm
P1_pwm  = u_k(7);   % target power value front right lift rotor pwm
P2_pwm  = u_k(8);   % target power value back left lift rotor pwm
P3_pwm  = u_k(9);   % target power value front left lift rotor pwm
P4_pwm  = u_k(10);  % target power value back right lift rotor pwm
```

```

GW = u_k(11);       % mass of the UAS in kg
fm = u_k(12);       % flightmode: 0 = QC, 1 = TR, 2 = FW

pwm_Pusher_min = u_k(13); % pusher pwm lower limit

%% States

% p_N = x_k(:,1);       % longitude in deg
% p_E = x_k(:,2);       % latitude in deg
% p_D = x_k(:,3);       % position down, neg. altitude in m

u = x_k(:,4);           % velocity in x direction in m/s
v = x_k(:,5);           % velocity in y direction in m/s
w = x_k(:,6);           % velocity in z direction in m/s

p = x_k(:,7);           % rollrate in x direction in rad/s
q = x_k(:,8);           % pitchrate in y direction in rad/s
r = x_k(:,9);           % yawrate in z direction in rad/s

phi = x_k(:,10);        % roll angle, Euler angle in  rad
theta = x_k(:,11);      % pitch angle, Euler angle in rad
% psi = x_k(:,12);      % yaw angle, Euler angle in rad

% u_w_nav = x_k(:,13);  % wind velocity in x direction in m/s in nav CS
% v_w_nav = x_k(:,14);  % wind velocity in y direction in m/s in nav CS
% w_w_nav = x_k(:,15);  % wind velocity in z direction in m/s in nav CS

%% Coefficients

% Drag coefficients
C_D0        = Parameters_k(:,1);        % -
C_Dalpha    = Parameters_k(:,2);        % 1/rad

% Sideforce coefficients
C_Ybeta     = Parameters_k(:,3);        % 1/rad
C_Yzeta     = Parameters_k(:,4);        % 1/rad
C_Yp        = Parameters_k(:,5);        % s/rad
C_Yr        = Parameters_k(:,6);        % s/rad

% Lift coefficients
C_L0        = Parameters_k(:,7);        % -
C_Lalpha    = Parameters_k(:,8);        % 1/rad
C_Leta      = Parameters_k(:,9);        % 1/rad
C_Lq        = Parameters_k(:,10);       % s/rad

```

```
switch eq
    case ’fstate’ % moment coefficients used for state prediction
% Roll Moment coefficient
C_Lmbeta    = Parameters_k(:,11);       % 1/rad
C_Lmxi      = Parameters_k(:,12);       % 1/rad
C_Lmp       = Parameters_k(:,13);       % s/rad
C_Lmr       = Parameters_k(:,14);       % s/rad

% Pitch Moment coefficient
C_M0        = Parameters_k(:,15);       % -
C_Malpha    = Parameters_k(:,16);       % 1/rad
C_Meta      = Parameters_k(:,17);       % 1/rad
C_Mq        = Parameters_k(:,18);       % s/rad

% Yaw Moment coefficient
C_Nbeta     = Parameters_k(:,19);       % 1/rad
C_Nzeta     = Parameters_k(:,20);       % 1/rad
C_Np        = Parameters_k(:,21);       % s/rad
C_Nr        = Parameters_k(:,22);       % s/rad
end

% Thrust Coefficients Lift Rotors
C_Tomega2   = Parameters_k(:,23);       % Thrust Coefficient, omega squared

% Moment coeffients due to rotor rotation
C_Nmt       = Parameters_k(:,24);       % Yaw coeffient, lift rotors

% Lift Power efficiency coefficients
c_p1        = Parameters_k(:,25);       % Lift rotor 1 propulsion
c_p2        = Parameters_k(:,26);       % Lift rotor 2 propulsion
c_p3        = Parameters_k(:,27);       % Lift rotor 3 propulsion
c_p4        = Parameters_k(:,28);       % Lift rotor 4 propulsion

% Thrust Coefficients Pusher Rotor
C_T_omega2_push   = Parameters_k(:,29); % in N/(rad/sec)^2
C_T_u2_omega_push = Parameters_k(:,30); % in N/((rad/sec) * (m/s)^2)

% Pusher Power efficiency coefficients
c_p_P       = Parameters_k(:,31);       % Pusher rotor propulsion

% Quadcopter and Transition aerodynamics forces and moments
% (Parameters initialized to zero and may be learned
% using the hybrid model approach)
C_Xa_qc     = Parameters_k(:,32);       % force coefficient in x-body CS
C_Ya_qc     = Parameters_k(:,33);       % force coefficient in y-body CS
C_Za_qc     = Parameters_k(:,34);       % force coefficient in z-body CS
```

```

C_La_qc     = Parameters_k(:,35);       % moment coefficient in x-body CS
C_Ma_qc     = Parameters_k(:,36);       % moment coefficient in y-body CS
C_Na_qc     = Parameters_k(:,37);       % moment coefficient in z-body CS

%% Precalculations aerodynamics

u_a = u;   % aerodynamic flow velocity in x-direction in body CS in m/s
v_a = v;   % aerodynamic flow velocity in y-direction in body CS in m/s
w_a = w;   % aerodynamic flow velocity in z-direction in body CS in m/s

% True airspeed
TAS = sqrt(u_a.^2 + v_a.^2 + w_a.^2);  % aerodynamic flow velocity in m/s

alpha = atan2(w_a,u_a);                % AOA in rad
beta = atan2(v_a,u_a);                 % SSA in rad

%% Calibration and PT1 correction of input pwm signals

% previous calibrated input signal
% previous deflection of aerodynamical control surface (k-1)
xi_L_k0   = x_k(:,16);   % previous aileron left deflection in rad
xi_R_k0   = x_k(:,17);   % estimated aileron right deflection in rad
eta_k0    = x_k(:,18);   % estimated elevator deflection in rad
zeta_L_k0 = x_k(:,19);   % estimated rudder left deflection in rad
zeta_R_k0 = x_k(:,20);   % estimated rudder right deflection in rad

% previous motor power (k-1)
P_P_k0 = x_k(:,21);      % estimated pusher motor power in kW
P1_k0  = x_k(:,22);      % estimated lift motor 1 power in kW
P2_k0  = x_k(:,23);      % estimated lift motor 2 power in kW
P3_k0  = x_k(:,24);      % estimated lift motor 3 power in kW
P4_k0  = x_k(:,25);      % estimated lift motor 4 power in kW

% calibrate input signal and estimate actual actuator deflections
xi_L = input_calibration_fct(xi_L_pwm, xi_L_k0,parameters_xi_L_pwm, ...
    delta_t, 1500);                          % aileron left in rad
xi_R = input_calibration_fct(xi_R_pwm, xi_R_k0, parameters_xi_R_pwm, ...
    delta_t, 1500);                          % aileron right in rad
eta = input_calibration_fct(eta_pwm, eta_k0, parameters_eta_pwm, ...
    delta_t, 1500);                          % elevator in rad
zeta_L = input_calibration_fct(zeta_L_pwm, zeta_L_k0,...
    parameters_zeta_L_pwm, delta_t, 1500);   % rudder left in rad
zeta_R = input_calibration_fct(zeta_R_pwm, zeta_R_k0,...
    parameters_zeta_R_pwm, delta_t, 1500);   % rudder right in rad

```

```
% calibrate input signal and estimate actual motor power
P_P = input_calibration_fct(P_P_pwm, P_P_k0, parameters_P_P_pwm, ...
    delta_t, pwm_Pusher_min);  % pusher power consumption in kW
P1  = input_calibration_fct(P1_pwm, P1_k0,parameters_P1_pwm, ...
    delta_t, 1000);        % power consumption front right lift motor in kW
P2  = input_calibration_fct(P2_pwm, P2_k0,parameters_P2_pwm, ...
    delta_t, 1000);        % power consumption back left lift motor in kW
P3  = input_calibration_fct(P3_pwm, P3_k0,parameters_P3_pwm, ...
    delta_t, 1000);        % power consumption front left lift motor in kW
P4  = input_calibration_fct(P4_pwm, P4_k0,parameters_P4_pwm, ...
    delta_t, 1000);        % power consumption back right lift motor in kW

% ensure motor power to be greater zero
P_P(P_P<0) = 0;
P1(P1<0) = 0;
P2(P2<0) = 0;
P3(P3<0) = 0;
P4(P4<0) = 0;

% new calibrated input signal array
u_k1_calb = [xi_L, xi_R, eta, zeta_L, zeta_R, P_P, P1, P2, P3, P4];

%% Rotational rates and motor speeds

% rotational rates p,q,r in experimental frame (alpha rotated)
[p_e, q_e, r_e] = Transformation_ef_rad_fct(p,q,r,alpha);

% motor angular velocities
omega_P = motor_rotational_speed_fct(d_pusher,A_R_pusher,c_p_P,P_P,...
    rho); % rotationrate pusher in rad/s
omega1 = motor_rotational_speed_fct(d_liftr,A_R_liftr,c_p1,P1,...
    rho); % rotationrate front right lift rotor in rad/s
omega2 = motor_rotational_speed_fct(d_liftr,A_R_liftr,c_p2,P2,...
    rho); % rotationrate back left lift rotor in rad/s
omega3 = motor_rotational_speed_fct(d_liftr,A_R_liftr,c_p3,P3,...
    rho); % rotationrate front left lift rotor in rad/s
omega4 = motor_rotational_speed_fct(d_liftr,A_R_liftr,c_p4,P4,...
    rho); % rotationrate pusher back right lift rotor in rad/s

%% Aerodynamic Forces and Moments

switch fm
    case 2 % flightmode = ’fw’
% dynamic pressure
q_press = rho/2*TAS.^2;

```

```
% Calculation of forces in experimental coordinate system

% Drag
C_D =   C_D0 + C_Dalpha.*alpha;

% Sideforce
C_Y =  C_Ybeta.*beta +...
       C_Yzeta .* (zeta_L + zeta_R)/2 + ...
       C_Yp.*b./(2*TAS+1e-3).*p_e + C_Yr.*b./(2*TAS+1e-3) .* r_e;

% Lift
C_L = C_L0 + C_Lalpha.*alpha + C_Leta.*eta + C_Lq .* mac./(TAS+1e-3) .*...
    q_e /2; % /2 coming from StephenLewis (difference to Brockhaus)

% Aerodynamic Forces (in experimental frame)
F_D = -q_press.*S.*C_D;
F_Y =  q_press.*S.*C_Y;
F_L = -q_press.*S.*C_L;

% Transformation from experimental to body frame
[F_Ax, F_Ay, F_Az] = Transformation_fe_rad_fct(F_D,F_Y,F_L,alpha);

    switch eq
        case 'fstate' % moments only calculated for state prediction
% Calculation of moments
% static derivatives in body coordinate system
% dynamic derivatives in experimental coordinate system

% static Roll moment coefficient in body frame
C_Lm_static_f = C_Lmbeta .* beta + ...
                C_Lmxi  .*(xi_R + xi_L);

% dynamic roll moment coefficient in experimental frame
C_Lm_dynamic_e = C_Lmp.*b./(2*TAS+1e-3) .* p_e + ...
    C_Lmr.*b./(2*TAS+1e-3).* r_e;

% static pitch moment coefficient in body frame
C_M_static_f = C_M0 + C_Malpha.*alpha + C_Meta .*eta;

% dynamic pitch moment coefficient in experimental frame
C_M_dynamic_e = C_Mq.*mac./(TAS+1e-3) .* q_e /2;
% /2 coming from Stephen and Lewis (difference to Brockhaus)

% static yaw moment coefficient in body frame
C_N_static_f = C_Nbeta .* beta + C_Nzeta  .* (zeta_L + zeta_R)/2;

```

```
% dynamic yaw moment coefficient in experimental frame
C_N_dynamic_e = C_Np.*b./(2*TAS+1e-3) .* p_e + C_Nr.*b./(2*TAS+1e-3).* r_e;

% Aerodynamic moments in experimental frame
M_L_dynamic_e =  q_press * S .* C_Lm_dynamic_e *b/2;
M_M_dynamic_e =  q_press * S .* C_M_dynamic_e *mac;
M_N_dynamic_e =  q_press * S .* C_N_dynamic_e *b/2;

% Transformation from experimental to body frame
[M_L_dynamic_f, M_M_dynamic_f , M_N_dynamic_f] = ...
    Transformation_fe_rad_fct(M_L_dynamic_e, M_M_dynamic_e, ...
    M_N_dynamic_e,alpha);

% Aerodynamic Moments with static derivatives (in body frame)
M_L_static_f = q_press*S.*C_Lm_static_f*b/2;
M_M_static_f = q_press*S.*C_M_static_f*mac;
M_N_static_f = q_press*S.*C_N_static_f*b/2;

% Sum up static and dynamic moments
M_Ax = M_L_static_f + M_L_dynamic_f;
M_Ay = M_M_static_f + M_M_dynamic_f;
M_Az = M_N_static_f + M_N_dynamic_f;

        case 'hmeas' % Set moments to zero when using measurement equations
M_Ax = 0;
M_Ay = 0;
M_Az = 0;
    end
end

%% Propulsion Forces and Moments + Gravitation

% Gravitation Force
    F_G = GW * g;

% Motor set-up,  view from above
%          front
%       T3      T1
%        |      |
%        |      |
%       T2      T4
%          rear

% Motor thrust, forces and moments
switch fm  % flight mode
    case {0,1}      % QC or TR
```

```
% Thrust of lift motors
T = rho .* pi*(d_liftr/2)^4 .* C_Tomega2.*[omega1 omega2 omega3 omega4].^2;

% Yaw moment of lift motors
Nmt = C_Nmt.*(omega3.^2+omega4.^2-omega1.^2-omega2.^2);

% Resulting moments of lift motors
M_Hx = r_xy*(T(:,2) + T(:,3) - T(:,1) - T(:,4));
M_Hy = r_xy*(T(:,1) + T(:,3) - T(:,2) - T(:,4));
M_Hz = Nmt;

% Quadcopter aerodynamics, related to gravitation force
% (initialized to be zero, buy may be learned by ANN)
F_Ax =  C_Xa_qc.*F_G; % Aerodynamic force in x-direction in QC flight
F_Ay =  C_Ya_qc.*F_G; % Aerodynamic force in y-direction in QC flight
F_Az =  C_Za_qc.*F_G; % Aerodynamic force in z-direction in QC flight

M_Ax =  C_La_qc.*F_G*b/2; % Aerodynamic moment in x-direction in QC flight
M_Ay =  C_Ma_qc.*F_G*mac; % Aerodynamic moment in y-direction in QC flight
M_Az =  C_Na_qc.*F_G*b/2; % Aerodynamic moment in z-direction in QC flight
end

switch fm % flight mode
    case 0
% Resulting forces of lift motors
F_Tx = 0;
F_Ty = 0;
F_Tz = - sum(T,2);

    case 1 % ’transition’
% Thrust of pusher motor
T_P = rho*pi*(d_pusher/2)^4 * omega_P .* ...
      (C_T_omega2_push .* omega_P + C_T_u2_omega_push .*  u_a.^2);

% Resulting forces of pusher and lift motors
F_Tx = T_P;
F_Ty = 0;
F_Tz = -sum(T,2);

    case 2   % ’fw’
% Resulting moments of lift motors
M_Hx = 0;
M_Hy = 0;
M_Hz = 0;

% Thrust of pusher motor
```

```
T_P = rho*pi*(d_pusher/2)^4 * omega_P .* ...
      (C_T_omega2_push .* omega_P + C_T_u2_omega_push .*  u_a.^2);

% Resulting forces of pusher motor
F_Tx = T_P;
F_Ty = 0;
F_Tz = 0;
end

%% Gravity

F_Gx = - F_G * sin(theta);
F_Gy =   F_G * cos(theta) .* sin(phi);
F_Gz =   F_G * cos(theta) .* cos(phi);

%% Total Forces and Moments in Body Frame

F_Total =                        [F_Ax  + F_Tx + F_Gx, ...
                    F_Ay  + F_Ty + F_Gy, ...
                    F_Az  + F_Tz + F_Gz];
switch eq
    case 'fstate'  % Calculated moments for state prediction
M_Total =           [M_Ax + M_Hx, M_Ay + M_Hy, M_Az + M_Hz];
    case 'hmeas' % Set moments to zero when using measurement equations
M_Total = zeros(n_sigma,3);
end

% array of angular velocities of all motors
Omega_speeds = [omega_P, omega1, omega2, omega3, omega4];
end

function omega = motor_rotational_speed_fct(d_rotor,A_rotor,c_p,P,rho)
% Motor angular velocity related to the motor power
c_p(c_p<1e-6) = 1e-6;  % ensure c_p to be greater 0 (numerical stability)
omega = ((P*1000./ (rho * A_rotor .* c_p)).^(2/3))./(d_rotor/2); % in rad/s
end

function u_calb_k1 = input_calibration_fct(u_pwm_k1, u_calb_k0, ...
    parameters_pwm_k , delta_t,pwm_base)
% PT1 correction of input value for approximation of current value
u_calb_k1 = (parameters_pwm_k(:,1) .* (u_pwm_k1 - pwm_base) .* ...
    parameters_pwm_k(:,2)./ parameters_pwm_k(:,3) * delta_t + ...
    u_calb_k0)./ (1 + delta_t./ parameters_pwm_k(:,3));
end
```

B.2. System state equations (Flight dynamics)

```
function [x_dot] = HybUAS_dynamic_states_dot_simple5_rad_fct(x_k, ...
Parameters_k, u_k, ~, delta_t, constants)
% HybUAS_dynamic_states_step_fct desribes the state (propagation) equations
% of the hybrid Scihunter UAS from FSR TUDa

% Get number of sigma points to accelerate state prediction using UKF
[n_sigma_dyn, ~] = size(x_k);
[n_sigma,~] = max([n_sigma_dyn,size(Parameters_k,1)]);

%% Input u

GW = u_k(11);               % mass of the UAS in kg

%% Constants

J = constants(19:24);       % Moments of Inertia

%% States:

u = x_k(:,4);               % body CS speed (x_axis) in m/s
v = x_k(:,5);               % body CS speed (y_axis) in m/s
w = x_k(:,6);               % body CS speed (z_axis) in m/s

p = x_k(:,7);               % rotational speed (roll rate) in rad/sec
q = x_k(:,8);               % rotational speed (ptich rate) in rad/sec
r = x_k(:,9);               % rotational speed (yaw rate) in rad/sec

phi     = x_k(:,10);        % attitude angle (roll angle) in rad
theta   = x_k(:,11);        % attitude angle (pitch angle) in rad
psi     = x_k(:,12);        % attitude angle (yaw angle) in rad

u_w_nav = x_k(:,13);        % wind velocity in x direction in m/s in geo CS
v_w_nav = x_k(:,14);        % wind velocity in y direction in m/s in geo CS
w_w_nav = x_k(:,15);        % wind velocity in z direction in m/s in geo CS

%% Forces and moments in body CS

[F_Total, M_Total,~, u_k1_calb] = HybUAS_forces_moments_simple5_rad_fct(...
    x_k,Parameters_k, u_k, delta_t, constants,'fstate',n_sigma);

%% Moment of inertia

Gamma0 = J(1)*J(3)-J(4)^2;
```

```
Gamma  = [(J(4)*(J(1)-J(2)+J(3)))/(Gamma0);
          (J(3)*(J(3)-J(2))+J(4)^2)/(Gamma0);
           J(3)/(Gamma0);...
           J(4)/Gamma0;...
          (J(3)-J(1))/J(2);...
           J(4)/J(2);...
         ((J(1)-J(2))*J(1)+J(4)^2)/Gamma0;...
           J(1)/Gamma0];

%% Accelerations in body CS

acc_x_body = r.*v - q.*w + F_Total(:,1) /GW;
acc_y_body = p.*w - r.*u + F_Total(:,2) /GW;
acc_z_body = q.*u - p.*v + F_Total(:,3) /GW;

%% State prediction

x_dot = zeros(size(x_k)); % initialization

% Position change (north, south, down) transforming body speeds into geo CS
[u_g, v_g, w_g]  = Transformation_gf_rad_fct(u, v, w, phi, theta, psi);

% Chaonge of position adding wind in geo CS
x_dot(:,1) = u_g + u_w_nav;     % Change of position North in m/s
x_dot(:,2) = v_g + v_w_nav;     % Change of position East in m/s
x_dot(:,3) = w_g + w_w_nav;     % Change of position Down in m/s

% Change of speeds
% Activate for SITL simulation
% (check ground contact condition)
% [acc_x_earth, acc_y_earth, acc_z_earth] = Transformation_gf_rad_fct(...
% acc_x_body,acc_y_body,acc_z_body,phi,theta,psi);
%  acc_z_earth(p_D >= -1e-4 & acc_z_earth > 0) = 0;
% [u_dot, v_dot, w_dot] = Transformation_fg_rad_fct(acc_x_earth,...
% acc_y_earth,acc_z_earth,phi,theta,psi);
% x_dot(:,4) = u_dot;        % body fixed speed (x_axis) in m/s
% x_dot(:,5) = v_dot;        % body fixed speed (y_axis) in m/s
% x_dot(:,6) = w_dot;        % body fixed speed (z_axis) in m/s

% Comment out for SITL simulation (see above)
x_dot(:,4) = acc_x_body;     % Change of body fixed speed (x_axis) in m/s^2
x_dot(:,5) = acc_y_body;     % Change of body fixed speed (y_axis) in m/s^2
x_dot(:,6) = acc_z_body;     % Change of body fixed speed (z_axis) in m/s^2

% Change of rotational speed
x_dot(:,7) = Gamma(1)*p.*q - Gamma(2)*q.*r + Gamma(3)*M_Total(:,1) ...
```

```
                + Gamma(4)*M_Total(:,3);       % dot p roll acc in rad/s^2
x_dot(:,8) = Gamma(5)*p.*r - Gamma(6)*(p.^2-r.^2) + M_Total(:,2)/J(2);
                                               % dot q pitch acc in rad/s^2
x_dot(:,9) = Gamma(7)*p.*q - Gamma(1)*q.*r + Gamma(4)*M_Total(:,1) ...
                + Gamma(8)*M_Total(:,3);       % dot r yaw acc in rad/s^2

% Change of attitude (using direction cosine matrix)
[phi_dot, theta_dot, psi_dot] = Transformation_Eulerpqr_rad_fct(p,q,r,...
                    phi,theta);     % with p,q,r in rad/s

x_dot(:,10) = phi_dot;    % change of attitude angle (roll angle) in rad/s
x_dot(:,11) = theta_dot;  % change of attitude angle (pitch angle) in rad/s
x_dot(:,12) = psi_dot;    % change of attitude angle (yaw angle) in rad/s

% Change of wind speeds (Wind assumed to be constant)
x_dot(:,13:15) = zeros(n_sigma,3);      % change of wind speeds in m/s^2

% Actuator deflections
% (preprocessed to tansmit acutal state, no prediction)
x_dot(:,16:20) = (u_k1_calb(:,1:5) - x_k(:,16:20))./delta_t;
% change of deflections of ailerons, elevator, rudders in rad/s

% Motor power (preprocessed to tansmit acutal state, no prediction)
x_dot(:,21:25) = (u_k1_calb(:,6:10) - x_k(:,21:25))./delta_t;
% change of motor power in kW/s
end
```

B.3. Measurement equations (Sensor model)

```
function [y_k] = HybUAS_measurement_model_simple5_rad_fct(x_k, ...
    Parameters_k, u_k, delta_t, constants)
% Desribes the sensor model with measurement equations of the
% hybird Scihunter UAS from FSR TUDa

% Get number of sigma points to accelerate state prediction using UKF
[n_sigma_dyn, ~] = size(x_k);
[n_sigma,~] = max([n_sigma_dyn,size(Parameters_k,1)]);

%% Input u

GW = u_k(11);

```

```
%% States

p_N = x_k(:,1);             % postion north in m
p_E = x_k(:,2);             % postion east in m
p_D = x_k(:,3);             % postion down in m (from sea level)

u = x_k(:,4);               % body fixed speed (x_axis) in m/sec
v = x_k(:,5);               % body fixed speed (y_axis) in m/sec
w = x_k(:,6);               % body fixed speed (z_axis) in m/sec

p = x_k(:,7);               % rotational speed (roll rate) in rad/sec
q = x_k(:,8);               % rotational speed (ptich rate) in rad/sec
r = x_k(:,9);               % rotational speed (yaw rate) in rad/sec

phi     = x_k(:,10);        % attitude angle (roll angle) in rad
theta   = x_k(:,11);        % attitude angle (pitch angle) in rad
psi     = x_k(:,12);        % attitude angle (yaw angle) in rad

u_w_nav = x_k(:,13);        % wind velocity in x direction in m/s in geo CS
v_w_nav = x_k(:,14);        % wind velocity in y direction in m/s in geo CS
w_w_nav = x_k(:,15);        % wind velocity in z direction in m/s in geo CS

%% Constants

g = constants(10);                  % Graviation constant m/s^2
omega_Puhser_max = constants(26);   % maximum rotational speed for pusher
                                    % in rad/s, (10360 rpm *pi/30)
omega_Liftr_max = constants(27);    % maximum rotational speed for lift
                                    % motor in rad/s (12580 rpm *pi/30)

%% Forces, motor speeds, motor power and deflections

[F_Total, ~, Omega_speeds, u_k1_calb] = ...
    HybUAS_forces_moments_simple5_rad_fct(x_k,Parameters_k, ...
    u_k, delta_t, constants,'hmeas',n_sigma);

%% Calculations

u_a = u;          % aerodynmic speeds in body fixed KOS (x-axis)
v_a = v;          % aerodynmic speeds in body fixed KOS (y-axis)
w_a = w;          % aerodynmic speeds in body fixed KOS (z-axis)

% Coriolis acceleration in m/s^2
acc_coriolis = [r.*v - q.*w,...
                p.*w - r.*u,...
                q.*u - p.*v];
```

```
% Body acceleration in m/s^2
acc_x_body = acc_coriolis(:,1) + F_Total(:,1) / GW;
acc_y_body = acc_coriolis(:,2) + F_Total(:,2) / GW;
acc_z_body = acc_coriolis(:,3) + F_Total(:,3) / GW;

% Transform body acceleration into geo CS
[acc_x_earth, acc_y_earth, acc_z_earth] = Transformation_gf_rad_fct(...
acc_x_body,acc_y_body,acc_z_body,phi,theta,psi);

% Activate for SITL simulation
% Check ground condition
% acc_z_earth(p_D >= -1e-4 & acc_z_earth > 0) = 0;

% Transform geo acceleration into body CS substracting graviation in z-dir.
[acc_meas_x_body, acc_meas_y_body, acc_meas_z_body] = ...
Transformation_fg_rad_fct(acc_x_earth,acc_y_earth,acc_z_earth-g,phi,...
theta,psi);

% Substracting coriolis accelertion
acc_meas_x_body = acc_meas_x_body - acc_coriolis(:,1);
acc_meas_y_body = acc_meas_y_body - acc_coriolis(:,2);
acc_meas_z_body  = acc_meas_z_body - acc_coriolis(:,3);

% Geo (GPS) speeds, transform body fixed speeds in geo CS
[u_g, v_g, w_g] = Transformation_gf_rad_fct(u,v,w,phi,theta,psi);

% Add wind in geo CS
u_g = u_g + u_w_nav;    % Geo (GPS) speeds in x-direction in m/s
v_g = v_g + v_w_nav;    % Geo (GPS) speeds in y-direction in m/s
w_g = w_g + w_w_nav;    % Geo (GPS) speeds in z-direction in m/s

% Aerodynamic angles
alpha = atan2(w_a,u_a);      % AOA in rad
beta = atan2(v_a,u_a);       % SSA in rad

% Total power consumption
P_ges = sum(u_k1_calb(:,6:10),2); % Overall energy power consumption in kW

% Normed rotational speed of the pusher motor
omega_Pn_est = Omega_speeds(:,1)/omega_Puhser_max;

% Normed rotational speed of the lift motors
omega1234n_est = Omega_speeds(:,2:end)/omega_Liftr_max;

%% Output vector
```

```
y_k = [ p_N, p_E, p_D, ...
        acc_meas_x_body, acc_meas_y_body, acc_meas_z_body, ...
        p, q, r, ...
        phi, theta, psi, ...
        u_a, alpha, beta,...
        u_k1_calb, P_ges,...
        omega_Pn_est, omega1234n_est,...
        u_g, v_g, w_g];
end
```

C. Supplementary data analysis

C.1. Additional data analysis of the commercial transport aircraft flight data

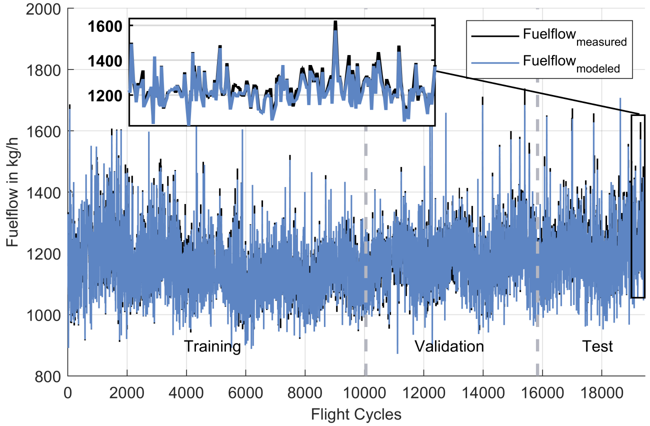


Figure C.1.: Estimation and prediction of the fuel flow compared to the measured fuel flow of the considered A320 aircraft engine using ECM data. Data is split into training, validation, and test data. For the estimation hybrid model 3 is used. [EHH22]

C.2. Comparison of the flight test and simulation data

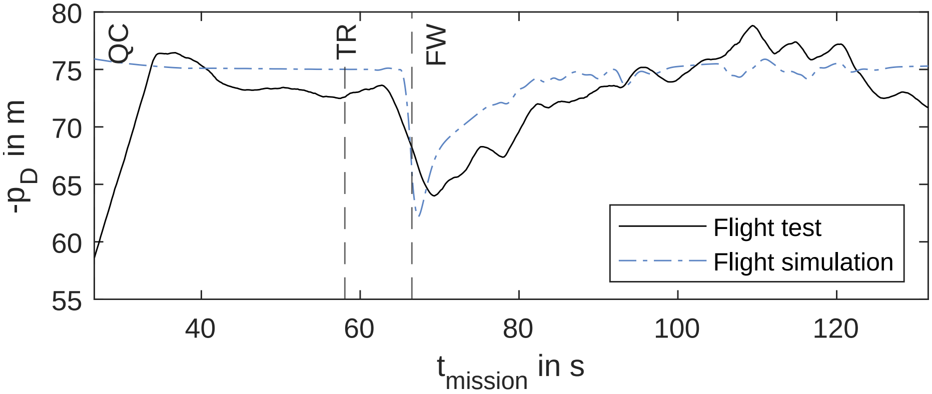


Figure C.2.: Comparison of the flight altitude p_D of the UAS in TR flight, considering an excerpt of flight test and simulation data

C.3. Additional data analysis in TS1

The additional data analysis in TS1 considers the innovation covariance P_{ee} and the output covariance P_{yy} observed in the last bootstrap sample of the training process. A visualization of the measurement residuals and the estimated $\hat{P}_{ee}$ of the first flight in the last bootstrap sample is given in Figure C.4. In addition, the resulting error metrics concerning the measurements, dynamical states, and parameter states are given for consistency analysis. They provide the evaluation of the accuracy using the converged hybrid model with fixed ANN weights. Similar analyses are performed for TS2, TS3, and TS4 on a reduced scale.

The covariance of the innovation $\hat{P}_{ee}$ is introduced in Eq. 2.4 in Subsection 2.2.2. Thus, it directly depends on $\hat{P}_{yy}$ and the noise R_{yy}. Within the TS both $\hat{P}_{ee}$ and $\hat{P}_{yy}$ depend on the flight trajectory and the respective FM, see Figure C.3. However, the effect of the noise R_{yy} on $\hat{P}_{ee}$ complicates its analysis. Therefore, the mean value of $\hat{P}_{ee}$ is considered in the following.

To still enable a distinction according to the flight trajectory, a mean value for the straight flight (Sf) and the curve flight (Cf) are considered. These describe data in the FW flight exclusively. The separation is performed by using the roll angle ϕ. The straight forward flight (Sf) is defined within the limits: $\phi \in [-5\,5]deg$. The curve flight (Cf) lies outside these limits.

The mean value of the estimated $\frac{1}{N}\sum^{N}\hat{P}_{ee}$ and of the actual $\frac{1}{N}\sum^{N}\tilde{P}_{ee}$ are finally compared using $\Delta^*\bar{P}_{ee}$. The actual $\tilde{P}_{ee}$ is assumed by:

$$\tilde{P}_{ee} = \hat{e}\hat{e}^{\mathsf{T}}. \tag{C.1}$$

For comparison, $\Delta^*\bar{P}_{ee}$ is introduced:

$$\Delta^*\bar{P}_{ee} = \frac{\frac{1}{N}\sum^{N}\hat{P}_{ee} - \frac{1}{N}\sum^{N}\tilde{P}_{ee}}{\frac{1}{N}\sum^{N}\tilde{P}_{ee}}. \tag{C.2}$$

The dependence of $\hat{P}_{ee}$ and $\tilde{P}_{ee}$ on the trajectory and their correlation can be seen primarily from RMSE$_{P_{ee}}$ given below.

Equivalent to the assumption made in Eq. C.1 and to Eq. 2.4, the actual output covariance $\tilde{P}_{yy}$ is defined in Eq. C.3 and analyzed in Figure C.3.

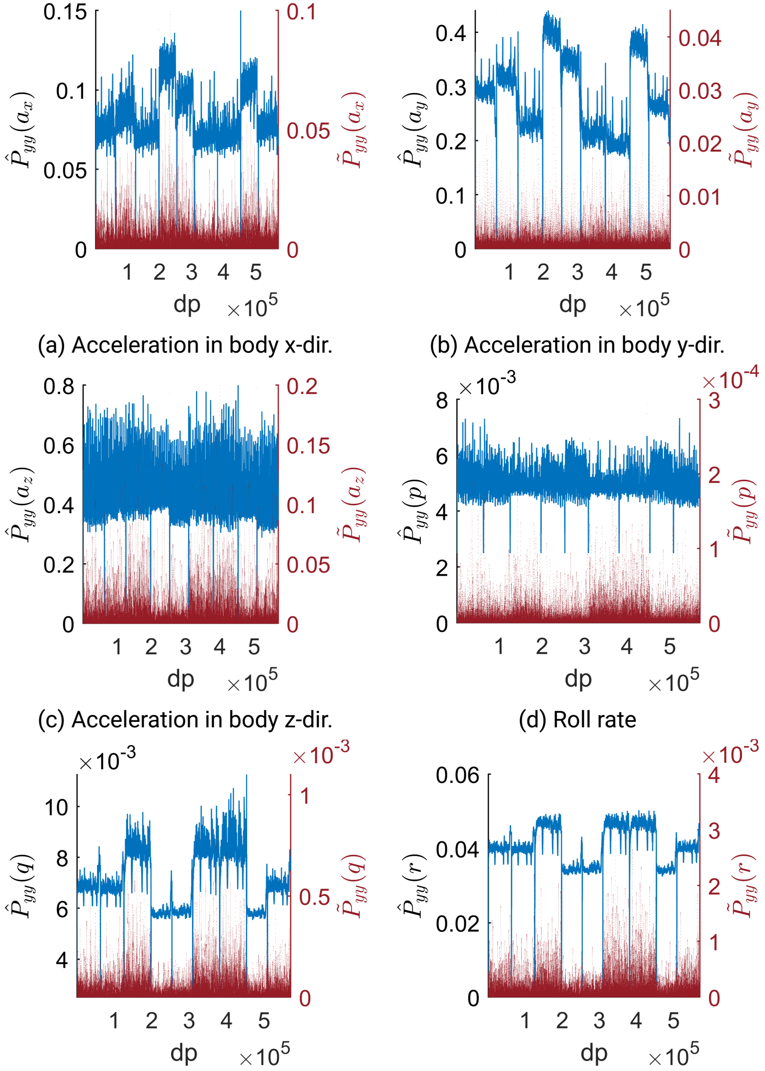
$$\tilde{P}_{yy} = (y_{ideal} - \hat{y})(y_{ideal} - \hat{y})^{\mathsf{T}} \tag{C.3}$$

Table C.1.: Mean main diagonals of P_{ee} in straight FW flight (Sf) in TS1.

Sign	$\frac{1}{N}\sum^{N}\hat{P}_{ee}^{(Sf)}$	$\frac{1}{N}\sum^{N}\tilde{P}_{ee}^{(Sf)}$	$\Delta^{*}\bar{P}_{ee}^{(Sf)}$ in [%]	RMSE$_{P_{ee}}$	Units of P_{ee}
p_N	7.88	5.05	56.2	7.66	m^2
p_E	8.02	5.17	55.1	7.81	m^2
p_D	52.2	31.9	63.9	49.4	m^2
a_x	0.0708	5.0e-3	1320	0.078	$(m/s^2)^2$
a_y	0.285	3.06e-3	9230	0.291	$(m/s^2)^2$
a_z	0.412	7.25e-3	5590	0.408	$(m/s^2)^2$
p	5.30e-3	4.41e-6	120e3	5.3e-3	$(rad/s)^2$
q	7.06e-3	2.34e-5	30e3	7.1e-3	$(rad/s)^2$
r	4.06e-3	1.07e-4	37.7e3	0.0408	$(rad/s)^2$
ϕ	0.0136	0.0101	34.4	0.0148	rad^2
θ	0.0133	0.0102	30.2	0.0148	rad^2
ψ	0.0177	0.0106	66.6	0.0167	rad^2
V_{a_x}	0.491	0.257	91	0.435	$(m/s)^2$
α	1.0e-3	3.05e-4	228	0.0113	rad^2
β	2.69e-3	3.14e-4	756	0.0115	rad^2
ξ_L	3.03e-3	2.47e-3	22.4	3.55e-3	rad^2
ξ_R	3.03e-3	2.50e-3	21.2	3.55e-3	rad^2
η	3.03e-3	2.49e-3	21.5	3.59e-3	rad^2
ζ_L	3.03e-3	2.51e-3	20.5	3.58e-3	rad^2
ζ_R	3.03e-3	2.50e-3	21	3.58e-3	rad^2
P_P	3.03e-3	2.51e-3	20.7	3.58e-3	kW^2
P_{bat}	1.94e-3	1.60e-3	21	2.29e-3	kW^2
ω_{P_n}	1.21e-4	9.98e-5	21.2	1.43e-3	$-$
u_g	0.68	0.315	116	0.584	$(m/s)^2$
v_g	0.801	0.354	126	0.679	$(m/s)^2$
w_g	1.48	1.04	42	1.54	$(m/s)^2$

Table C.2.: Mean main diagonals of P_{ee} in curve flight (Cf) in TS1.

Sign	$\frac{1}{N}\sum^{N} \hat{P}_{ee}^{(Cf)}$	$\frac{1}{N}\sum^{N} \tilde{P}_{ee}^{(Cf)}$	$\Delta^{*} \bar{P}_{ee}^{(Cf)}$ in $[\%]$	$\mathrm{RMSE}_{P_{ee}}$	Units of P_{ee}
p_N	7.92	5.03	57.4	7.71	m^2
p_E	7.99	5.09	56.8	7.76	m^2
p_D	52.4	32.2	62.7	49.3	m^2
a_x	0.0849	5.12e-3	1560	0.109	$(m/s^2)^2$
a_y	0.285	3.10e-3	9090	0.30	$(m/s^2)^2$
a_z	0.45	8.63e-3	5110	0.452	$(m/s^2)^2$
p	5.04e-3	5.39e-6	93.5e3	5.3e-3	$(rad/s)^2$
q	7.09e-3	2.75e-5	25.7e3	7.2e-3	$(rad/s)^2$
r	0.0408	1.06e-4	38.3e3	0.0408	$(rad/s)^2$
ϕ	0.0136	0.0102	33.8	0.0147	rad^2
θ	0.0136	0.0102	33.3	0.0148	rad^2
ψ	0.0174	0.0107	62.4	0.0166	rad^2
V_{a_x}	0.492	0.257	91.4	0.438	$(m/s)^2$
α	6.42e-4	3.06e-4	110	7.22e-3	rad^2
β	2.3e-3	3.13e-4	635	7.49e-3	rad^2
ξ_L	3.03e-3	2.50e-3	21.2	3.58e-3	rad^2
ξ_R	3.03e-3	2.49e-3	21.3	3.58e-3	rad^2
η	3.03e-3	2.50e-3	21.2	3.56e-3	rad^2
ζ_L	3.03e-3	2.49e-3	21.6	3.57e-3	rad^2
ζ_R	3.03e-3	2.50e-3	21	3.58e-3	rad^2
P_P	3.03e-3	2.5e-3	20.9	3.57e-3	kW^2
P_{bat}	1.94e-3	1.6e-3	21	2.27e-3	kW^2
ω_{Pn}	1.21e-4	1.0e-4	20.8	1.43e-4	–
u_g	0.697	0.321	117	0.612	$(m/s)^2$
v_g	0.742	0.338	120	0.644	$(m/s)^2$
w_g	1.51	1.05	43.7	1.55	$(m/s)^2$



(a) Acceleration in body x-dir.

(b) Acceleration in body y-dir.

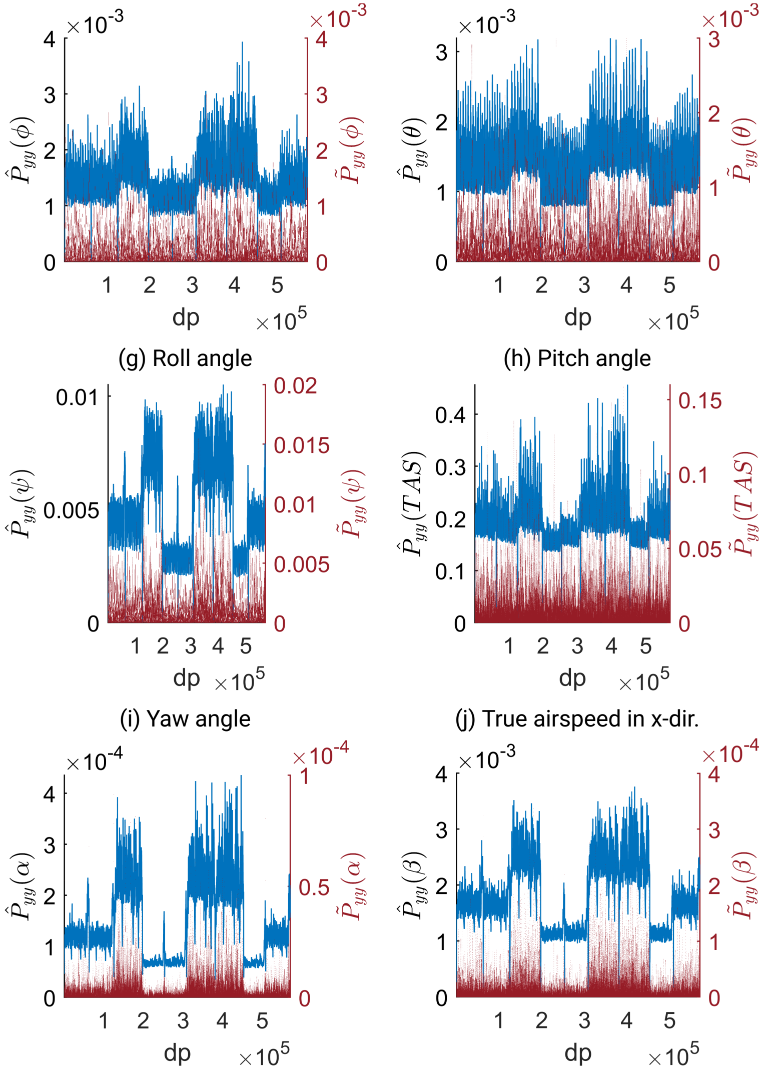
(c) Acceleration in body z-dir.

(d) Roll rate

(e) Pitch rate

(f) Yaw rate

Figure C.3.: Estimated ($\hat{P}_{yy}$) and actual ($\tilde{P}_{yy}$) output covariance recorded in the last bootstrap sample of TS1.



(g) Roll angle

(h) Pitch angle

(i) Yaw angle

(j) True airspeed in x-dir.

(k) Angle Of Attack (AOA)

(l) Side Slip Angle (SSA)

Figure C.3.: Estimated ($\hat{P}_{yy}$) and actual ($\tilde{P}_{yy}$) output covariance recorded in the last bootstrap sample of TS1.

Table C.3.: Error metrics of the hybrid and physical model outputs in TS1.

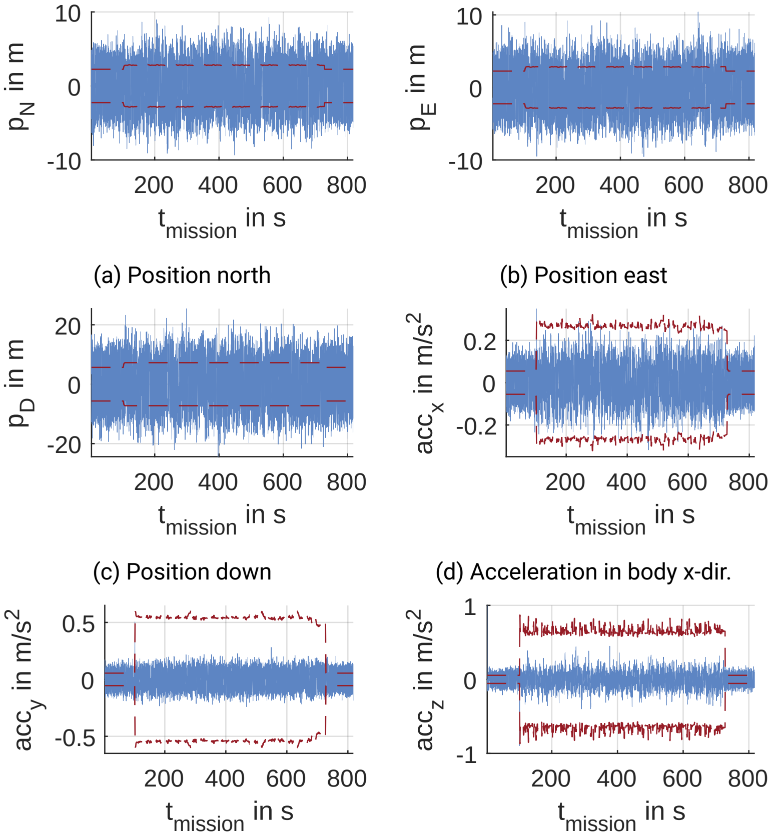
Sign	RMSE$_{y_{hyb}}$	RMSE$_{y_{phy}}$	NRMSE$_{y_{hyb}}$	NRMSE$_{y_{phy}}$	Reference	Unit
p_N	2.242	2.231	0.006	0.006	350	m
p_E	2.263	2.244	0.019	0.019	120	m
p_D	5.619	5.674	0.051	0.052	110	m
a_x	0.070	0.071	0.001	0.001	75	m/s^2
a_y	0.056	0.056	0.006	0.006	10	m/s^2
a_z	0.092	0.093	0.001	0.001	95	m/s^2
p	0.002	0.002	0.002	0.002	1	rad/s
q	0.005	0.005	0.005	0.005	1	rad/s
r	0.010	0.01	0.01	0.01	1	rad/s
ϕ	0.101	0.101	0.101	0.101	1	rad
θ	0.101	0.101	0.101	0.101	1	rad
ψ	0.103	0.103	0.016	0.016	2π	rad
V_{a_x}	0.508	0.507	0.023	0.023	22	m/s
α	0.017	0.017	0.017	0.017	1	rad
β	0.018	0.018	0.018	0.018	1	rad
ξ_L	0.05	0.05	0.05	0.05	1	rad
ξ_R	0.05	0.05	0.05	0.05	1	rad
η	0.05	0.05	0.05	0.05	1	rad
ζ_L	0.05	0.05	0.05	0.05	1	rad
ζ_R	0.05	0.05	0.05	0.05	1	rad
P_P	0.05	0.05	0.05	0.05	1	kW
P_1	0.04	0.04	0.04	0.04	1	kW
P_2	0.04	0.04	0.04	0.04	1	kW
P_3	0.04	0.04	0.04	0.04	1	kW
P_4	0.04	0.04	0.04	0.04	1	kW
P_{bat}	0.04	0.04	0.04	0.04	1	kW
ω_{Pn}	0.01	0.01	0.01	0.01	1	$-$
ω_{1n}	0.01	0.01	0.01	0.01	1	$-$
ω_{2n}	0.01	0.01	0.01	0.01	1	$-$
ω_{3n}	0.01	0.01	0.01	0.01	1	$-$
ω_{4n}	0.01	0.01	0.01	0.01	1	$-$
u_g	0.567	0.566	0.026	0.026	22	m/s
v_g	0.585	0.587	0.073	0.073	8	m/s
w_g	1.023	1.023	0.064	0.064	16	m/s

Table C.4.: Error metrics of the hybrid and physical model dynamical states in TS1.

Sign	$\text{RMSE}_{x_{d_{hyp}}}$	$\text{RMSE}_{x_{d_{phy}}}$	$\text{NRMSE}_{x_{d_{hyp}}}$	$\text{NRMSE}_{x_{d_{phy}}}$	Reference	Unit
p_N	1.013	0.995	0.003	0.003	350	m
p_E	1.058	1.018	0.009	0.008	120	m
p_D	2.567	2.677	0.023	0.024	110	m
u_b	0.084	0.083	0.004	0.004	22	m/s
v_b	0.041	0.04	0.005	0.005	8	m/s
w_b	0.014	0.014	0.001	0.001	16	m/s
p	0.002	0.002	0.002	0.002	1	rad/s
q	0.005	0.005	0.005	0.005	1	rad/s
r	0.01	0.01	0.01	0.01	1	rad/s
ϕ	0.014	0.014	0.014	0.014	1	rad
θ	0.014	0.014	0.014	0.014	1	rad
ψ	0.027	0.026	0.004	0.004	2π	rad
u_w	0.013	0.014	0.013	0.014	1	m/s
v_w	0.012	0.011	0.012	0.011	1	m/s
w_w	0.013	0.012	0.013	0.012	1	m/s
ξ_L	0	0	0	0	1	rad
ξ_R	0	0	0	0	1	rad
η	0	0	0	0	1	rad
ζ_L	0	0	0	0	1	rad
ζ_R	0	0	0	0	1	rad
P_P	0	0	0	0	1	kW
P_1	0	0	0	0	1	kW
P_2	0	0	0	0	1	kW
P_3	0	0	0	0	1	kW
P_4	0	0	0	0	1	kW

Table C.5.: Error metrics of the hybrid and physical model parameter states in TS1.

Sign	$\text{RMSE}_{x_{p_{hyp}}}$	$\text{RMSE}_{x_{p_{phy}}}$	$\text{NRMSE}_{x_{p_{hyp}}}$	$\text{NRMSE}_{x_{p_{phy}}}$	Reference
C_{D_0}	0.0007	0	0.0052	0	0.132
C_{L_0}	0.0022	0	0.0035	0	0.616
C_{M_0}	0	0	0	0	0.0063



(a) Position north

(b) Position east

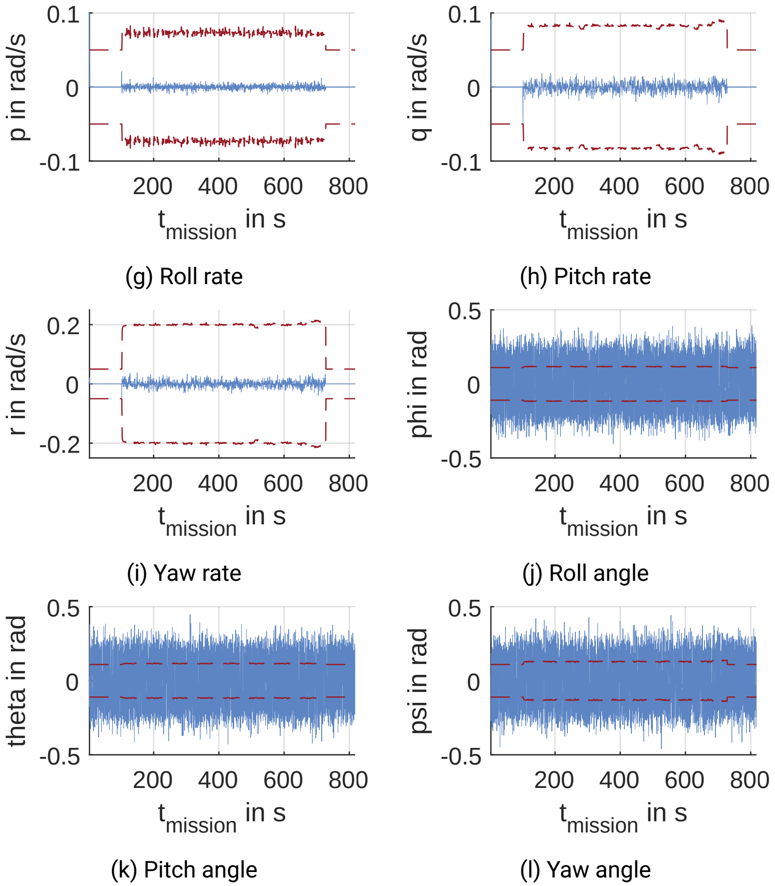
(c) Position down

(d) Acceleration in body x-dir.

(e) Acceleration in body y-dir.

(f) Acceleration in body z-dir.

Figure C.4.: Residuals of measurements and selected states of FM1 in the last bootstrap sample of TS1 with σ_{ee} or σ_{xx} (dashed).



(g) Roll rate

(h) Pitch rate

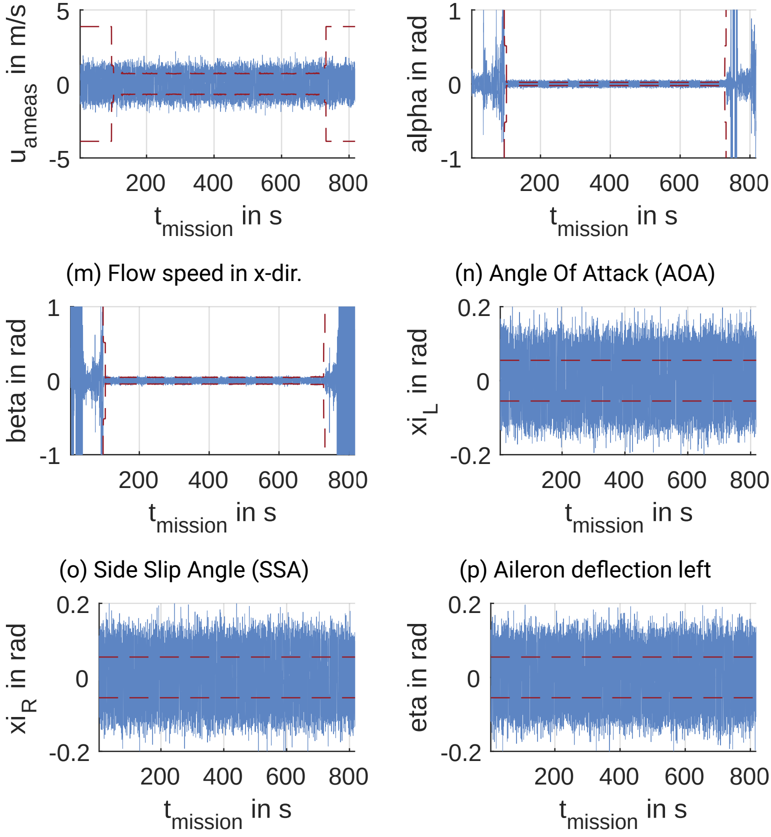
(i) Yaw rate

(j) Roll angle

(k) Pitch angle

(l) Yaw angle

Figure C.4.: Residuals of measurements and selected states of FM1 in the last bootstrap sample of TS1 with σ_{ee} or σ_{xx} (dashed).



(m) Flow speed in x-dir.

(n) Angle Of Attack (AOA)

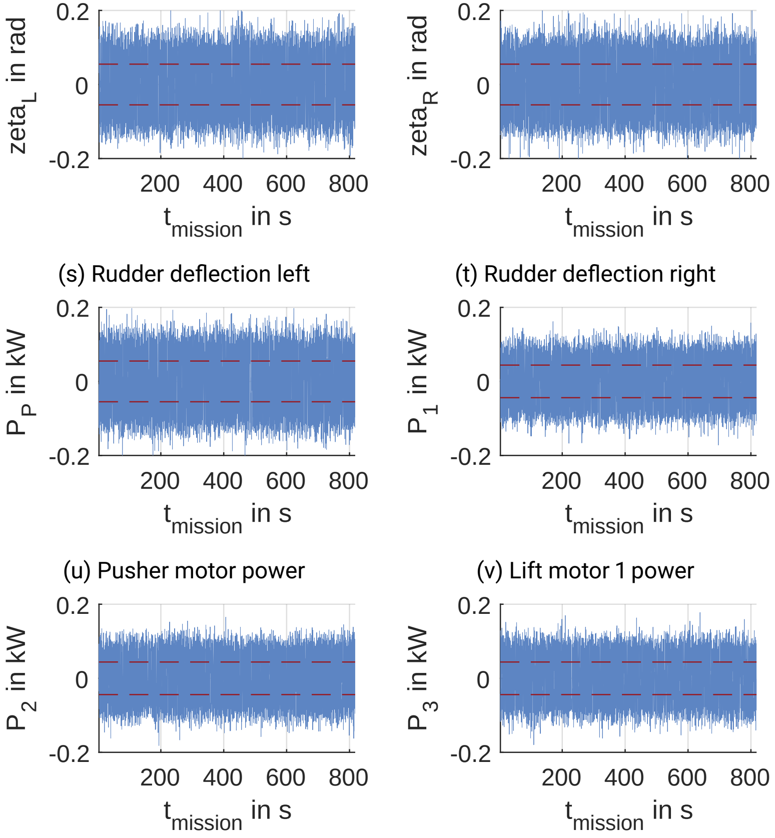
(o) Side Slip Angle (SSA)

(p) Aileron deflection left

(q) Aileron deflection right

(r) Elevator deflection

Figure C.4.: Residuals of measurements and selected states of FM1 in the last bootstrap sample of TS1 with σ_{ee} or σ_{xx} (dashed).



(s) Rudder deflection left

(t) Rudder deflection right

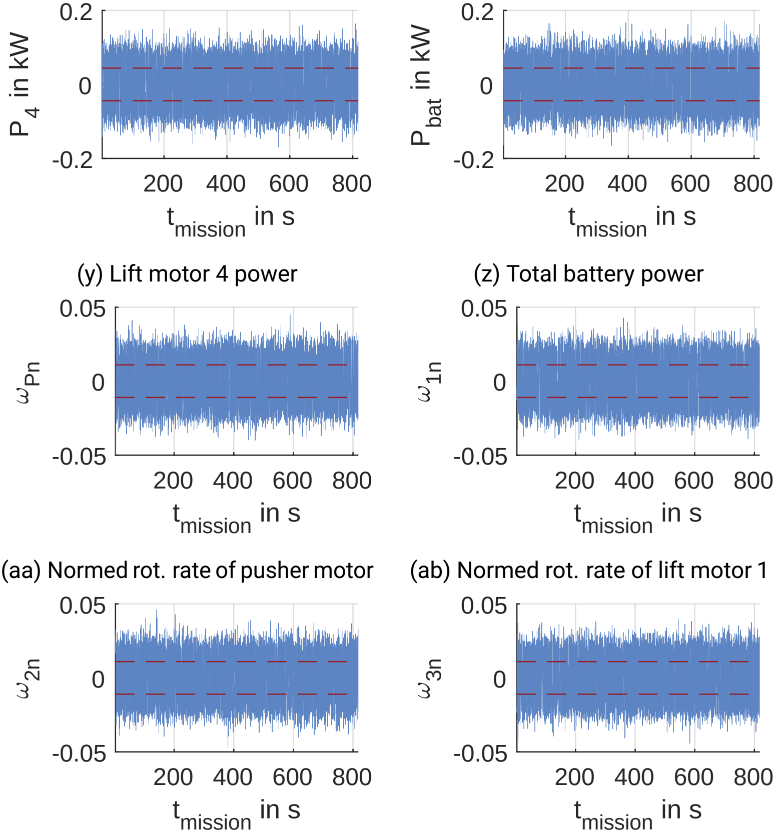
(u) Pusher motor power

(v) Lift motor 1 power

(w) Lift motor 2 power

(x) Lift motor 3 power

Figure C.4.: Residuals of measurements and selected states of FM1 in the last bootstrap sample of TS1 with σ_{ee} or σ_{xx} (dashed).



(y) Lift motor 4 power

(z) Total battery power

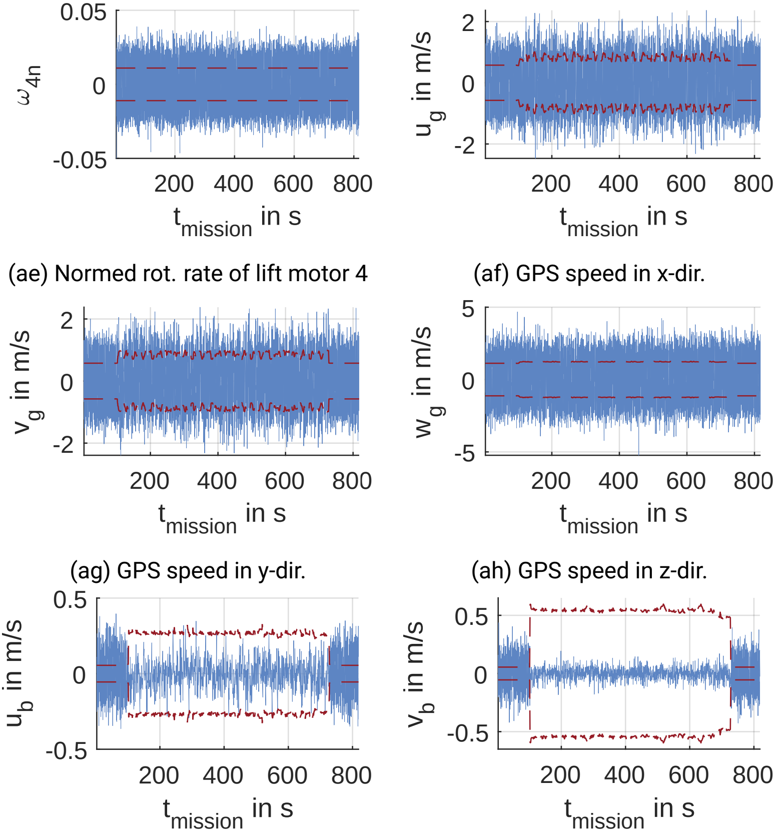
(aa) Normed rot. rate of pusher motor

(ab) Normed rot. rate of lift motor 1

(ac) Normed rot. rate of lift motor 2

(ad) Normed rot. rate of lift motor 3

Figure C.4.: Residuals of measurements and selected states of FM1 in the last bootstrap sample of TS1 with σ_{ee} or σ_{xx} (dashed).



(ae) Normed rot. rate of lift motor 4

(af) GPS speed in x-dir.

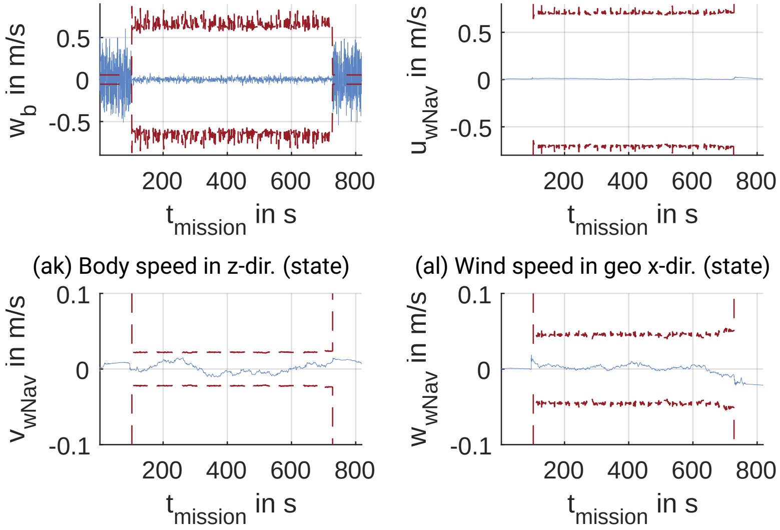
(ag) GPS speed in y-dir.

(ah) GPS speed in z-dir.

(ai) Body speed in x-dir. (state)

(aj) Body speed in y-dir. (state)

Figure C.4.: Residuals of measurements and selected states of FM1 in the last bootstrap sample of TS1 with σ_{ee} or σ_{xx} (dashed).



(ak) Body speed in z-dir. (state)

(al) Wind speed in geo x-dir. (state)

(am) Wind speed in geo y-dir. (state)

(an) Wind speed in geo z-dir. (state)

Figure C.4.: Residuals of measurements and selected states of FM1 in the last bootstrap sample of TS1 with σ_{ee} or σ_{xx} (dashed).

C.4. Additional data analysis in TS2

Table C.6.: Error metrics of the hybrid and physical model outputs in TS2.

Sign	$\text{RMSE}_{y_{hyb}}$	$\text{RMSE}_{y_{phy}}$	$\text{NRMSE}_{y_{hyb}}$	$\text{NRMSE}_{y_{phy}}$	Reference	Unit
p_N	2.252	2.285	0.006	0.007	350	m
p_E	2.254	2.281	0.019	0.019	120	m
p_D	5.646	5.930	0.051	0.054	110	m
a_x	0.071	0.346	0.001	0.005	75	m/s^2
a_y	0.056	0.128	0.006	0.013	10	m/s^2
a_z	0.091	0.462	0.001	0.005	95	m/s^2
p	0.002	0.007	0.002	0.007	1	rad/s
q	0.005	0.059	0.005	0.059	1	rad/s
r	0.010	0.029	0.01	0.029	1	rad/s
ϕ	0.101	0.114	0.101	0.114	1	rad
θ	0.101	0.140	0.101	0.140	1	rad
ψ	0.103	0.111	0.016	0.018	2π	rad
V_{a_x}	0.506	0.562	0.023	0.026	22	m/s
α	0.018	0.022	0.018	0.022	1	rad
β	0.018	0.020	0.018	0.020	1	rad
ξ_L	0.05	0.05	0.05	0.05	1	rad
ξ_R	0.05	0.05	0.05	0.05	1	rad
η	0.05	0.05	0.05	0.05	1	rad
ζ_L	0.05	0.05	0.05	0.05	1	rad
ζ_R	0.05	0.05	0.05	0.05	1	rad
P_P	0.05	0.05	0.05	0.05	1	kW
P_1	0.04	0.04	0.04	0.04	1	kW
P_2	0.04	0.04	0.04	0.04	1	kW
P_3	0.04	0.04	0.04	0.04	1	kW
P_4	0.04	0.04	0.04	0.04	1	kW
P_{bat}	0.04	0.04	0.04	0.04	1	kW
ω_{Pn}	0.01	0.01	0.01	0.01	1	$-$
ω_{1n}	0.01	0.01	0.01	0.01	1	$-$
ω_{2n}	0.01	0.01	0.01	0.01	1	$-$
ω_{3n}	0.01	0.01	0.01	0.01	1	$-$
ω_{4n}	0.01	0.01	0.01	0.01	1	$-$
u_g	0.568	0.82	0.026	0.037	22	m/s
v_g	0.583	0.852	0.073	0.107	8	m/s
w_g	1.020	1.924	0.064	0.120	16	m/s

Table C.7.: Error metrics of the hybrid and physical model dynamical states in TS2.

Sign	RMSE$_{x_{d_{hyp}}}$	RMSE$_{x_{d_{phy}}}$	NRMSE$_{x_{d_{hyp}}}$	NRMSE$_{x_{d_{phy}}}$	Reference	Unit
p_N	1.032	1.108	0.003	0.003	350	m
p_E	1.049	1.111	0.009	0.009	120	m
p_D	2.621	3.173	0.024	0.029	110	m
u_b	0.080	0.259	0.004	0.012	22	m/s
v_b	0.040	0.153	0.005	0.019	8	m/s
w_b	0.012	0.223	0.001	0.014	16	m/s
p	0.002	0.007	0.002	0.007	1	rad/s
q	0.005	0.059	0.005	0.059	1	rad/s
r	0.010	0.029	0.01	0.029	1	rad/s
ϕ	0.014	0.056	0.014	0.056	1	rad
θ	0.014	0.098	0.014	0.098	1	rad
ψ	0.026	0.049	0.004	0.008	2π	rad
u_w	0.010	0.009	0.010	0.009	1	m/s
v_w	0.012	0.050	0.012	0.050	1	m/s
w_w	0.012	0.114	0.012	0.114	1	m/s
ξ_L	0	0	0	0	1	rad
ξ_R	0	0	0	0	1	rad
η	0	0	0	0	1	rad
ζ_L	0	0	0	0	1	rad
ζ_R	0	0	0	0	1	rad
P_P	0	0	0	0	1	kW
P_1	0	0	0	0	1	kW
P_2	0	0	0	0	1	kW
P_3	0	0	0	0	1	kW
P_4	0	0	0	0	1	kW

Table C.8.: Error metrics of the hybrid and physical model parameter states in TS2.

Sign	RMSE$_{x_{p_{hyp}}}$	RMSE$_{x_{p_{phy}}}$	NRMSE$_{x_{p_{hyp}}}$	NRMSE$_{x_{p_{phy}}}$	Reference
C_{D_0}	0.0001	0.0186	0.0008	0.1412	0.132
C_{L_0}	0.0004	0.0867	0.0008	0.1412	0.614
C_{M_0}	0	0	0.0086	0	0.00461

C.5. Additional data analysis in TS3

Table C.9.: Error metrics of the hybrid and physical model outputs in TS3.

Sign	RMSE$_{y_{hyb}}$	RMSE$_{y_{phy}}$	NRMSE$_{y_{hyb}}$	NRMSE$_{y_{phy}}$	Reference	Unit
p_N	2.05	2.047	0.006	0.006	350	m
p_E	2.045	2.043	0.017	0.017	120	m
p_D	5.097	5.073	0.046	0.046	110	m
a_x	3.228	0.429	0.043	0.006	75	m/s^2
a_y	2.486	1.346	0.249	0.135	10	m/s^2
a_z	0.744	0.396	0.008	0.004	95	m/s^2
p	0.065	0.027	0.065	0.027	1	rad/s
q	0.027	0.017	0.027	0.017	1	rad/s
r	0.061	0.018	0.061	0.018	1	rad/s
ϕ	0.338	0.104	0.338	0.104	1	rad
θ	0.294	0.105	0.294	0.105	1	rad
ψ	0.131	0.101	0.021	0.016	2π	rad
V_{a_x}	1.492	1.917	0.068	0.087	22	m/s
α	0.079	0.084	0.079	0.084	1	rad
β	0.118	0.022	0.118	0.022	1	rad
ξ_L	0.05	0.05	0.05	0.05	1	rad
ξ_R	0.05	0.05	0.05	0.05	1	rad
η	0.05	0.05	0.05	0.05	1	rad
ζ_L	0.05	0.05	0.05	0.05	1	rad
ζ_R	0.05	0.05	0.05	0.05	1	rad
P_P	0.05	0.05	0.05	0.05	1	kW
P_1	0.04	0.04	0.04	0.04	1	kW
P_2	0.04	0.04	0.04	0.04	1	kW
P_3	0.04	0.04	0.04	0.04	1	kW
P_4	0.04	0.04	0.04	0.04	1	kW
P_{bat}	0.04	0.04	0.04	0.04	1	kW
ω_{Pn}	0.01	0.01	0.01	0.01	1	$-$
ω_{1n}	0.01	0.01	0.01	0.01	1	$-$
ω_{2n}	0.01	0.01	0.01	0.01	1	$-$
ω_{3n}	0.01	0.01	0.01	0.01	1	$-$
ω_{4n}	0.01	0.01	0.01	0.01	1	$-$
u_g	1.713	1.511	0.078	0.069	22	m/s
v_g	1.553	1.419	0.194	0.177	8	m/s
w_g	3.258	1.206	0.204	0.075	16	m/s

Table C.10.: Error metrics of the hybrid and physical model dynamical states in TS3.

Sign	$\text{RMSE}_{x_{d_{hyp}}}$	$\text{RMSE}_{x_{d_{phy}}}$	$\text{NRMSE}_{x_{d_{hyp}}}$	$\text{NRMSE}_{x_{d_{phy}}}$	Reference	Unit
p_N	0.458	0.436	0.001	0.001	350	m
p_E	0.425	0.417	0.004	0.003	120	m
p_D	0.999	0.836	0.009	0.008	110	m
u_b	1.406	1.849	0.064	0.084	22	m/s
v_b	2.171	0.422	0.271	0.053	8	m/s
w_b	1.467	1.405	0.092	0.088	16	m/s
p	0.065	0.027	0.065	0.027	1	rad/s
q	0.027	0.017	0.027	0.017	1	rad/s
r	0.061	0.018	0.061	0.018	1	rad/s
ϕ	0.323	0.027	0.323	0.027	1	rad
θ	0.277	0.032	0.277	0.032	1	rad
ψ	0.085	0.014	0.014	0.002	2π	rad
u_w	0.434	0.291	0.434	0.291	1	m/s
v_w	0.232	0.244	0.232	0.244	1	m/s
w_w	3.587	1.032	3.587	1.032	1	m/s
ξ_L	0	0	0	0	1	rad
ξ_R	0	0	0	0	1	rad
η	0	0	0	0	1	rad
ζ_L	0	0	0	0	1	rad
ζ_R	0	0	0	0	1	rad
P_P	0	0	0	0	1	kW
P_1	0	0	0	0	1	kW
P_2	0	0	0	0	1	kW
P_3	0	0	0	0	1	kW
P_4	0	0	0	0	1	kW

Table C.11.: Error metrics of the hybrid and physical model parameter states in TS3.

Sign	$\text{RMSE}_{x_{p_{hyp}}}$	$\text{RMSE}_{x_{p_{phy}}}$	$\text{NRMSE}_{x_{p_{hyp}}}$	$\text{NRMSE}_{x_{p_{phy}}}$	Reference
C_D	0.0759	0.0378	0.2168	0.1889	0.20
C_Y	0.0491	0.1129	1.9419	11.2924	0.01
C_L	0.2210	0.2168	0.1793	0.2891	0.75
C_ℓ	0.0092	0.0111	1.0063	1.1070	0.01
C_m	0.0470	0.0191	0.5219	1.9078	0.01
C_n	0.0336	0.0158	0.3323	1.5807	0.01

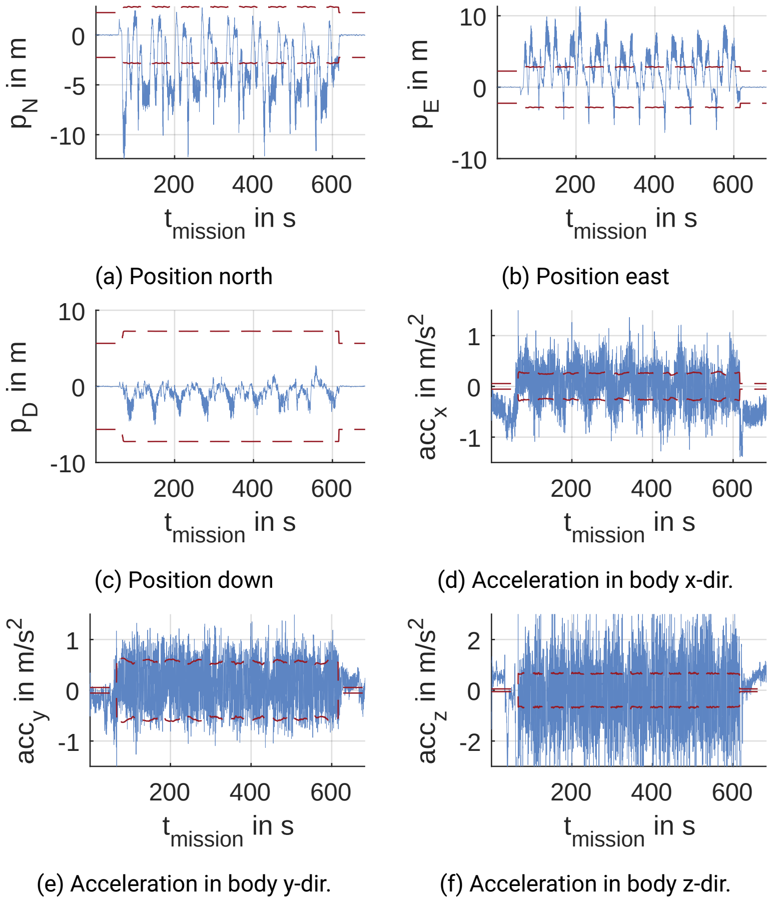
C.6. Additional data analysis in TS4

Table C.12.: Mean main diagonals of P_{ee} in straight FW flight (Sf) in TS4.

Sign	$\frac{1}{N}\sum^{N} \hat{P}_{ee}^{(Sf)}$	$\frac{1}{N}\sum^{N} \tilde{P}_{ee}^{(Sf)}$	$\Delta^* \bar{P}_{ee}^{(Sf)}$ in $[\%]$	$\text{RMSE}_{P_{ee}}$	Units of P_{ee}
p_N	7.99	12.3	-35.3	17.3	m^2
p_E	7.95	8.82	-9.85	10.2	m^2
p_D	52.3	4.88	973	47.9	m^2
a_x	0.078	0.14	-44.1	0.682	$(m/s^2)^2$
a_y	0.292	0.126	131	0.303	$(m/s^2)^2$
a_z	0.413	1.22	-66.1	5.17	$(m/s^2)^2$
p	5.42e-3	0.0226	-76	0.0522	$(rad/s)^2$
q	6.98e-3	3.90e-3	78.9	0.0105	$(rad/s)^2$
r	0.0405	0.0107	279	0.0369	$(rad/s)^2$
ϕ	0.0138	0.023	-40.1	0.0343	rad^2
θ	0.0134	0.0107	25.9	0.0118	rad^2
ψ	0.0178	0.038	-53.1	0.0465	rad^2
V_{a_x}	0.497	0.94	-47.2	1.39	$(m/s)^2$
α	8.99e-4	5.12e-3	-82.4	0.0122	rad^2
β	2.59e-3	4.83e-3	-46.3	0.0141	rad^2
ξ_L	3.03e-3	1.19e-3	155	2.72e-3	rad^2
ξ_R	3.03e-3	8.95e-4	238	2.68e-3	rad^2
η	3.03e-3	3.15e-4	860	2.65e-3	rad^2
ζ_L	3.03e-3	5.87e-4	415	2.62e-3	rad^2
ζ_R	3.03e-3	4.84e-4	525	2.74e-3	rad^2
P_P	3.03e-3	6.10e-3	-50.4	0.0101	kW^2
P_{bat}	1.94e-3	6.95e-3	-72.2	0.0119	kW^2
ω_{P_n}	1.21e-4	7.99e-4	-84.9	1.29e-3	$-$
u_g	0.783	2.74	-71.4	4.94	$(m/s)^2$
v_g	0.746	2.91	-74.4	3.91	$(m/s)^2$
w_g	1.53	1.3	17.1	1.75	$(m/s)^2$

Table C.13.: Mean main diagonals of P_{ee} in curve flight (Cf) in TS4.

Sign	$\frac{1}{N}\sum^{N}\hat{P}_{ee}^{(Cf)}$	$\frac{1}{N}\sum^{N}\tilde{P}_{ee}^{(Cf)}$	$\Delta^*\bar{P}_{ee}^{(Cf)}$ in $[\%]$	RMSE$_{P_{ee}}$	Units of P_{ee}
p_N	7.94	16.6	-52.2	23.1	m^2
p_E	8.02	7.26	10.6	10.3	m^2
p_D	52.5	4.59	1040	48.3	m^2
a_x	0.0797	0.124	-35.7	0.442	$(m/s^2)^2$
a_y	0.32	0.134	138	0.334	$(m/s^2)^2$
a_z	0.429	1.60	-73.1	7.19	$(m/s^2)^2$
p	5.41e-3	0.0317	-82.9	0.0662	$(rad/s)^2$
q	6.62e-3	6.10e-3	8.52	0.0138	$(rad/s)^2$
r	0.0384	0.0131	194	0.0343	$(rad/s)^2$
ϕ	0.0137	0.0318	-57	0.0485	rad^2
θ	0.0136	0.0101	33.9	0.0131	rad^2
ψ	0.0169	0.0299	-43.5	0.0417	rad^2
V_{a_x}	0.496	1.13	-56.2	1.66	$(m/s)^2$
α	6.16e-4	3.90e-3	8.06e-3	-84.2	rad^2
β	2.10e-3	4.27e-3	9.77e-3	-50.8	rad^2
ξ_L	3.03e-3	1.39e-3	118	2.78e-3	rad^2
ξ_R	3.03e-3	1.06e-3	185	2.77e-3	rad^2
η	3.03e-3	3.26e-4	827	2.74e-3	rad^2
ζ_L	3.03e-3	7.0e-4	332	2.57e-3	rad^2
ζ_R	3.03e-3	5.98e-4	406	2.73e-3	rad^2
P_P	3.03e-3	6.37e-3	-52.5	0.011	kW^2
P_{bat}	1.94e-3	7.10e-3	-72.7	0.0127	kW^2
ω_{P_n}	1.21e-4	7.47e-4	-83.8	1.29e-3	$-$
u_g	0.719	4.62	-84.5	7.23	$(m/s)^2$
v_g	0.795	3.13	-74.6	4.84	$(m/s)^2$
w_g	1.58	1.68	-6.17	2.33	$(m/s)^2$



(a) Position north

(b) Position east

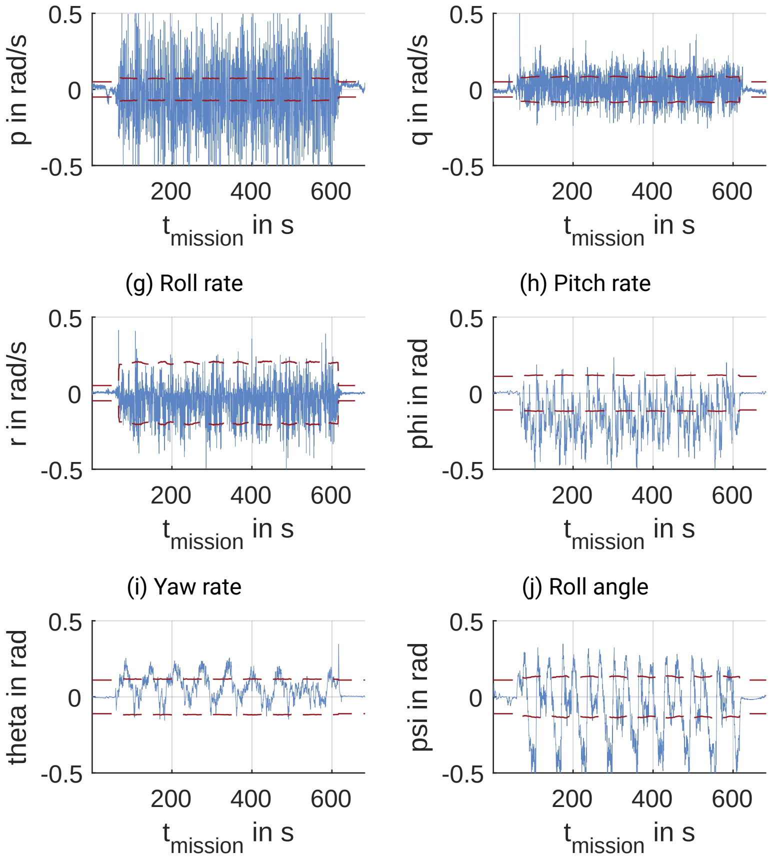
(c) Position down

(d) Acceleration in body x-dir.

(e) Acceleration in body y-dir.

(f) Acceleration in body z-dir.

Figure C.5.: Residuals of measurements of FM1 in the last bootstrap sample of TS4 with σ_{ee} (dashed).



(g) Roll rate

(h) Pitch rate

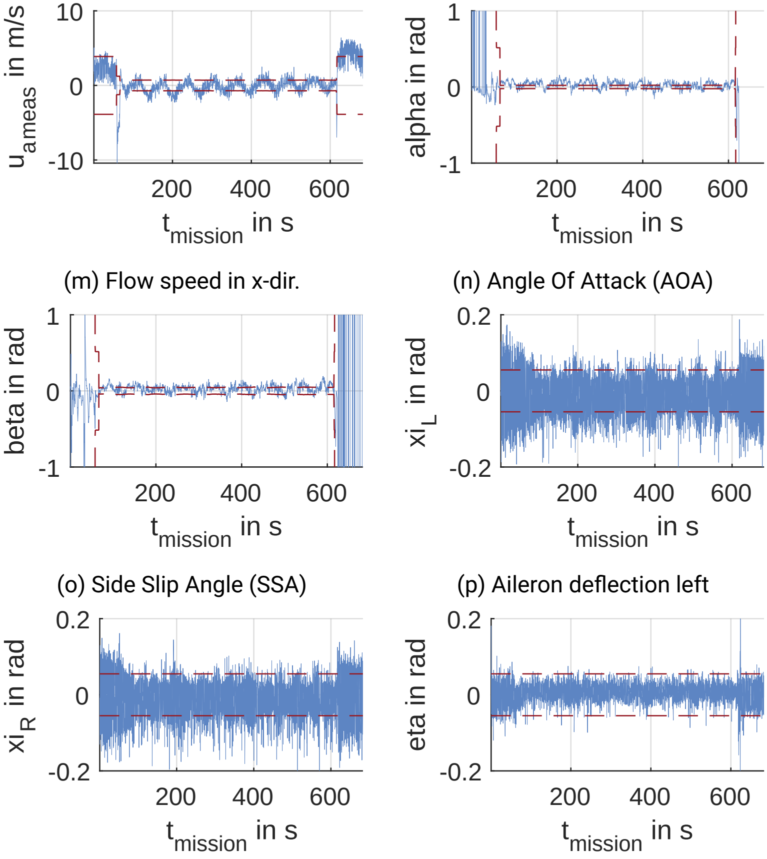
(i) Yaw rate

(j) Roll angle

(k) Pitch angle

(l) Yaw angle

Figure C.5.: Residuals of measurements of FM1 in the last bootstrap sample of TS4 with σ_{ee} (dashed).



(m) Flow speed in x-dir.

(n) Angle Of Attack (AOA)

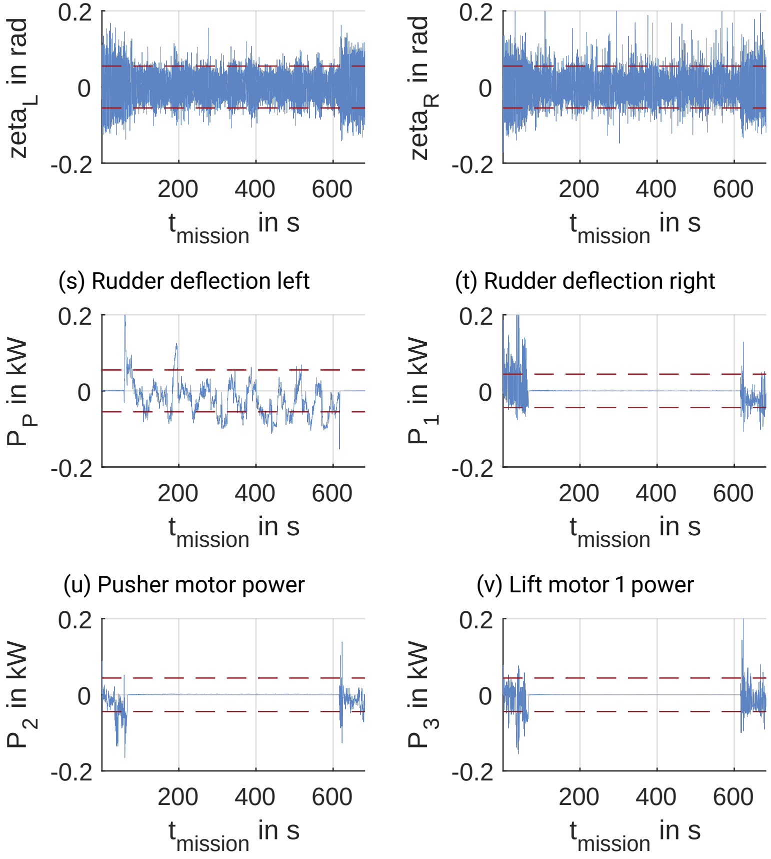
(o) Side Slip Angle (SSA)

(p) Aileron deflection left

(q) Aileron deflection right

(r) Elevator deflection

Figure C.5.: Residuals of measurements of FM1 in the last bootstrap sample of TS4 with σ_{ee} (dashed).



(s) Rudder deflection left

(t) Rudder deflection right

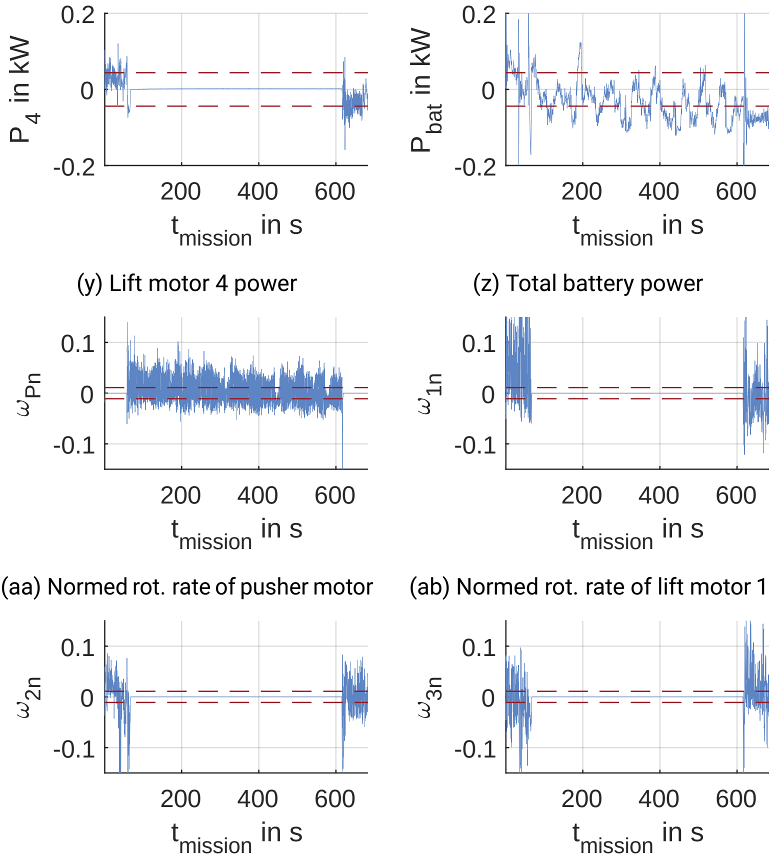
(u) Pusher motor power

(v) Lift motor 1 power

(w) Lift motor 2 power

(x) Lift motor 3 power

Figure C.5.: Residuals of measurements of FM1 in the last bootstrap sample of TS4 with σ_{ee} (dashed).



(y) Lift motor 4 power

(z) Total battery power

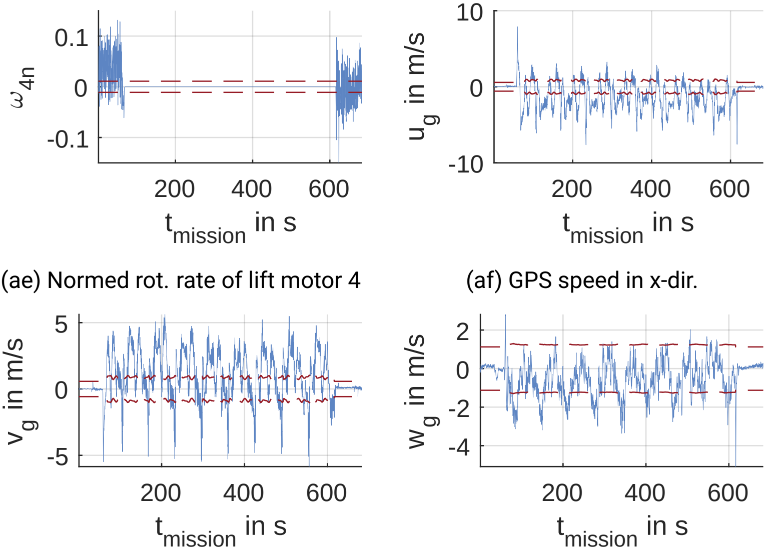
(aa) Normed rot. rate of pusher motor

(ab) Normed rot. rate of lift motor 1

(ac) Normed rot. rate of lift motor 2

(ad) Normed rot. rate of lift motor 3

Figure C.5.: Residuals of measurements of FM1 in the last bootstrap sample of TS4 with σ_{ee} (dashed).



(ae) Normed rot. rate of lift motor 4

(af) GPS speed in x-dir.

(ag) GPS speed in y-dir.

(ah) GPS speed in z-dir.

Figure C.5.: Residuals of measurements of FM1 in the last bootstrap sample of TS4 with σ_{ee} (dashed).