



TECHNISCHE UNIVERSITÄT MÜNCHEN

TUM School of Engineering and Design

Enhanced Predictive Analysis for Flight Incident Probability Using Advanced Uncertainty Quantification Methods

Xiaolong Wang

Vollständiger Abdruck der von der TUM School of Engineering and Design der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktors der Ingenieurwissenschaften

genehmigten Dissertation.

Vorsitzender: Prof. Dr.-Ing. Harald Klein

Prüfer der Dissertation: 1. Prof. Dr.-Ing. Florian Holzapfel
2. Prof. Claudia Czado, Ph.D.

Die Dissertation wurde am 16.12.2021 bei der Technischen Universität München eingereicht und durch die TUM School of Engineering and Design am 11.04.2022 angenommen.

Abstract

Huge amounts of flight data are generated and recorded in the daily operations of civil airlines. These data are analyzed to detect and predict hazards in the Flight Data Monitoring (FDM) program. In addition to that, International Civil Aviation Organization (ICAO) also suggests airlines define their Acceptable Level of Safety Performance (ALoSP) based on their operational data, which can be described by the occurrence probability of an incident/accident. Due to the lack of accident data, it is difficult to directly calculate statistically significant statements on their occurrence probabilities. Therefore, A model-based Predictive Analysis (PA) framework has already been developed to provide a quantitative statement of incident probability from non-accident flight data at the Institute of Flight System Dynamics (FSD). This PA framework combines the operational flight data and the physical-based incident model by using statistical methods.

This thesis introduces advanced uncertainty quantification (UQ) methods to enhance the quality of the PA algorithms in model calibration, contributing factors modeling, sensitivity analysis, and physical modeling. In the PA framework, model calibration is required to ensure that the incident model output can represent the reality of the relevant flight operation statistically. To reduce the execution times of the time-consuming Monte Carlo Simulation (MCS) in the state-of-the-art calibration algorithm, a new calibration framework is proposed based on the polynomial chaos expansion (PCE) method and a frozen sample strategy. The MCS is required only once, and the convergence and efficiency of the calibration algorithm are improved dramatically. In addition, wind variation is critical for some incidents/accidents. Instead of the constant wind speed in the state-of-the-art incident model, wind time series is firstly modeled as the model input in a statistical manner using the Karhunen–Loève (KL) expansion methods with the vine copula approach. The built KL-based wind model allows us to generate the new wind series, which follows the original statistical characteristics and is easily integrated into the PA framework. Moreover, the global sensitivity analysis algorithms, as the post-processing step, are developed to identify the key drivers of incidents. Furthermore, a novel incident model to simulate the evolution of aircraft low energy states during the final approach is built considering flight dynamics and standard operational procedures (SOPs). The enhanced PA framework is implemented on the aircraft low energy incident model to predict the occurrence probability and identify the critical contributing factors.

Zusammenfassung

Im täglichen Betrieb ziviler Luftfahrtunternehmen wird eine große Menge an Flugdaten aufgezeichnet. Diese Daten werden im *Flight Data Monitoring* (FDM) Prozess analysiert, um frühzeitig Gefahren zu erkennen und Risiken vorherzusagen. Darüber hinaus fordert die Internationale Organisation für Zivilluftfahrt (ICAO) von den Luftfahrtunternehmen die Festlegung eines *Acceptable Levels of Safety Performance* (ALoSP), auf dessen Grundlage die Sicherheit des aktuellen Betriebs zu überwachen ist. Aufgrund der geringen Anzahl tatsächlich aufgezeichneter Vor- und Unfälle ist es schwer, deren tatsächliche Eintrittswahrscheinlichkeiten direkt zu berechnen. Deshalb hat der Lehrstuhl für Flugsystemdynamik (FSD) der Technischen Universität München einen modellbasierten Rahmen für die prädiktive Analyse (PA) entwickelt, um valide Vorhersagen von Unfallereigniswahrscheinlichkeiten aus unfallfreien Flugdaten quantitativ zu bestimmen. Dieses System zur prädiktiven Analyse nutzt statistische Methoden, um operative Flugdaten mit physikbasierten Unfallmodellen zu kombinieren.

Die Arbeit nutzt fortschrittliche Methoden der Unsicherheitsquantifizierung (UQ), um die Qualität von PA-Algorithmen bei der Modellkalibrierung, der Modellierung von Einflussfaktoren, der Sensitivitätsanalyse und der physikalischen Modellierung zu verbessern. Im Rahmen der PA ist eine Modellkalibrierung erforderlich, um sicherzustellen, dass die Ergebnisse des Unfallmodells die Realität eines bestimmten Flugbetriebs statistisch repräsentativ abbilden. Um die Ausführungszeit der zeitaufwändigen Monte-Carlo-Simulation (MCS) aktueller Kalibrierungsalgorithmen zu verkürzen, wird eine neue Kalibrierungsmethodik vorgeschlagen, die auf der Polynomial Chaos-Expansion (PCE) und einer Frozen Sample Strategie beruht. Dadurch wird die rechnerintensive MCS nur einmal benötigt, Konvergenz sowie Effizienz des Kalibrierungsalgorithmus werden erheblich verbessert. Für eine realitätsnahe PA sind bei einigen Unfallszenarien Modelle für Windrichtungsänderungen entscheidend. Anstelle der konstanten Windgeschwindigkeit aktueller Unfallmodelle schlägt diese Arbeit ein deutlich realitätsnäheres Windmodell vor. Dieses basiert auf der Karhunen-Loève (KL)-Methode, kombiniert mit Vine-Copula-Verteilungen. Es ermöglicht tatsächliche Wind-Zeitreihendaten als Modellinput zu nutzen. Das entwickelte Windmodell erlaubt es, daraus neue Windsequenzen zu generieren, die den statistischen Eigenschaften der ursprünglichen Messungen folgen und leicht in den PA-Rahmen integriert werden können. Außerdem werden in dieser Arbeit Algorithmen für globale Sensi-

tivitätsanalysen entwickelt, um die wichtigsten Unfallursachen zu ermitteln. Ferner wird ein neues Vorfallsmodell präsentiert, das die Entstehung von Flugzuständen niedriger Energie während des Endanflugs abbildet, wobei sowohl Flugdynamik als auch flzeugtyp- und betreiberspezifische Standardbetriebsverfahren (SOPs) berücksichtigt werden. Die erweiterte PA-Systematik wird auf diese Simulation von Flugzuständen niedriger Energie angewendet, um die Wahrscheinlichkeit eines Unfalls vorherzusagen und die wichtigsten Einflussfaktoren zu erfassen.

Danksagung/Acknowledgments

I would like to express my great gratitude to my supervisor, Florian Holzapfel, for his guidance, smart insights, and immense knowledge. By being influenced by his brilliant thoughts, I have developed an intense interest in how to apply the theory into practice. I also would like to thank Prof. Claudia Czado for the opportunity as my second supervisor, and the many meetings about mathematical discussions. Her rigorous attitudes towards scientific research benefit me a lot.

This dissertation is the result of my work at the Institute of Flight System Dynamics at Technische Universität München. Special thanks to my colleagues in the Flight Safety Group: Ludwig Drees, Lukas Höndorf, Javensius Sembiring, Chong Wang, Phillip Koppitz, Sercin Höndorf, Florian Schwaiger, Lukas Beller, and Stefan Schiele for their wonderful collaboration and patient support.

I also would like to thank my colleagues: Xiang Fang, Dalong Shi, Jun Shi, Guolong Jiang, Jiannan Zhang, Chinmaya Mishra, Bharathkumar Balaji, David Seiferth, Idris E. Putro, and Prof. Jianping Li for many exciting discussions and nice memories. I would like to thank all of my colleagues and friends with whom we have shared a nice time in Munich. I always enjoy memories of the many interesting things.

Finally, I would like to thank my family for their endless love. I bless my grandparent's happiness in heaven. Without their great support, I cannot finish my thesis ever.

Garching bei Munich, June 4, 2022

Xiaolong Wang

Contents

Abstract	i
Kurzfassung	iii
Acknowledgment	v
List of Figures	xi
List of Tables	xv
Acronyms	xvii
Symbols and Indices	xxi
Symbols	xxi
Indices	xxiii
1 Introduction	1
1.1 Motivation and Background	1
1.2 State of the Art	2
1.2.1 Flight Data Monitoring	2
1.2.2 Predictive Analysis Framework	3
1.2.3 Uncertainty Quantification Methods	5
1.2.4 Gaps in Current Predictive Analysis Framework	7
1.3 Objectives	8
1.4 Contributions	8
1.5 Outline	11
2 Operational Flight Data Analysis	13
2.1 Flight Data Recording	13
2.2 QAR Decoding	17
2.3 QAR Data Preprocessing	20
2.3.1 File Splitting	20
2.3.2 Airport and Runway Detection	21
2.3.3 Time Points Detection	22

2.3.4	Flight Data Measurements	23
2.3.5	Flight Path Reconstruction	24
2.4	FDM Events	27
2.5	Physical-based Incident Model	28
3	Mathematical Preliminaries	31
3.1	Introduction	31
3.2	Input Uncertainty Quantification	31
3.2.1	Random Variables and Distributions	31
3.2.2	High Dimensional Dependence Modeling	32
3.2.3	Sampling Methods	40
3.3	Rare Event Probability Estimation	48
3.3.1	Monte Carlo Simulation	48
3.3.2	Importance Sampling	49
3.3.3	Subset Simulation	51
3.4	Global Sensitivity Analysis	70
3.4.1	Correlation-based Sensitivity	70
3.4.2	Regression-based Sensitivity	72
3.4.3	Variance-based Sensitivity	73
3.4.4	Moment-independent Sensitivity	78
3.4.5	Reliability-oriented Sensitivity	79
3.5	Polynomial Chaos Expansion	81
3.5.1	PCE Model	81
3.5.2	Computation of PCE Coefficients	87
3.5.3	Statistical Moments of PCE Output	89
3.5.4	PCE-based Sobol Indices	90
4	Calibration of Contributing Factors for Predictive Analysis	91
4.1	Calibration Strategy	91
4.2	PCE-based Calibration Framework	95
4.2.1	PCE-based Surrogate Model	95
4.2.2	Integration of PCE into the Calibration Framework	96
4.3	Validation of the Calibration Algorithm	99
4.4	Calibration for the Runway Overrun Model	103
4.4.1	Initial PCE Model	105
4.4.2	Calibration of Contributing Factors	107
5	Modeling of Stochastic Wind	111
5.1	Motivation	111
5.2	Stochastic Process Modeling Methods	112
5.2.1	White Noise	113

5.2.2	Markov Process	113
5.2.3	Fourier Transformation-based Spectral Representation	114
5.3	KL-based Stochastic Process Reconstruction and Regeneration	116
5.4	Comparison between the PSD Approach and the KL Expansion Method	120
5.5	Dependence among KL Coefficients	122
5.6	Turbulence Modeling	122
5.6.1	Von Karman Model	123
5.6.2	Modeling of Turbulence Using KL Expansion	124
5.7	KL-based Wind Series Regeneration from the Operational Flight Data	127
5.7.1	Headwind Regeneration	127
5.7.2	Windshear Regeneration	138
5.8	Summary	141
6	Low Energy State Analysis	143
6.1	Motivation	143
6.2	Operational Safety Techniques During the Approach	144
6.2.1	Stabilization Gate	144
6.2.2	Energy Management	148
6.2.3	Late Destabilization	150
6.3	Detection of Low Energy Event During the Final Approach	152
6.3.1	Low Energy Margin	152
6.3.2	Time Margin	153
6.4	Predictive Analysis of Low Energy Event	156
6.4.1	Incident Metric for Low Energy Event	156
6.4.2	Modeling of Low Energy	157
6.4.3	Identify Contributing Factors	166
6.4.4	Occurrence Probability of Low Energy Event	171
6.4.5	Sensitivity Analysis of Contributing Factors	174
6.5	Summary	181
7	Conclusion and Outlook	183
7.1	Conclusion	183
7.2	Outlook	186
A	Nomenclature Principles and Coordinate Frames	I
A.1	Nomenclature Principles	I
A.2	Coordinate Frames and Transformation	III
B	Distribution Families	VII
B.1	One-dimensional Probability Distributions	VII
B.2	Bivariate Copula Families	VIII

C Efficiency of MCMC estimator

XIII

List of Figures

- 1.1 Uncertainty Quantification framework 5
- 1.2 Structure of the dissertation. 11

- 2.1 Black box flight recorders 14
- 2.2 Mandatory parameters of the FDR 15
- 2.3 Flight data recording architecture 15
- 2.4 An overview of on-board data transmission via ED48 FDIMU 16
- 2.5 Data frame 3D structure of the QAR binary data stream. 18
- 2.6 Transformation between a binary data and a decimal value. 19
- 2.7 Data frame information for airspeed. 20
- 2.8 Pressure altitude time series of three flights in a QAR file. 21
- 2.9 Runway detection in Hong Kong airport. 22
- 2.10 Time points calculation of flight phases in a flight. 22
- 2.11 The recorded and smoothed wind speed during the final approach phase
in one flight. 26
- 2.12 Histogram of touchdown distance measurements from 1133 B747-8F flights
in ELLX airport. 27
- 2.13 The stop margin as an incident metric for runway overrun. 28

- 3.1 Pair samples plot in three scales 34
- 3.2 Black box flight recorders 35
- 3.3 Illustration of Sklar theorem 35
- 3.4 The Clayton copula density 36
- 3.5 The graphical structure of vine copula in three dimensions. 37
- 3.6 Inverse transformation sampling 40
- 3.7 Rejection sampling 42
- 3.8 Histograms of samples using the MH random walk sampling 44
- 3.9 Sample trace plots and sample autocorrelation plots 45
- 3.10 The relation of the efficiency measure of the MCMC algorithm, the stan-
dard deviation of the proposal distribution, and the average acceptance
ratio of sample candidates 46
- 3.11 A structure of 3-dimensional vine copula with the construction functions . 47

3.12	Comparison of generated 1000 samples using MCS and IS	51
3.13	Illustration of intermediate events in subset simulation.	51
3.14	Process of the sample generation in each subset level	53
3.15	Illustration of the Markov chains in the subset simulation	56
3.16	The proposal PDF of the conditional sampling	62
3.17	Illustration of the adaptive conditional sampling procedure	64
3.18	Histogram of intermediate failure samples in subset level 15 for the original and adaptive conditional sampling method	66
3.19	Flow chart of subset simulation using the adaptive conditional sampling method considering the dependent inputs	69
3.20	Sample plots of MCS in the runway overrun	71
3.21	Normalized sample plots of MCS in the runway overrun	72
3.22	Quantile-based samples plot of the landing mass and the stop margin . . .	74
3.23	Designed samples to calculate $S_i^{(1)}$ and the $S_i^{(T)}$ in the case of 3 inputs. . .	76
3.24	Designed samples to calculate $S_{ij}^{(2)}$ in the case of 3 inputs.	76
3.25	Estimated conditional PDF of stop margin	78
3.26	Relationship of three moment-independent sensitivity metrics	79
3.27	Histograms of intermediate failure samples of headwind in runway overrun	81
3.28	Classical and hyperbolic truncated scheme of PCE	86
4.1	Illustration of validation	92
4.2	Comparison of simulated and observed outputs	92
4.3	Flow chart of the MCS-based calibration	93
4.4	PCE-based Surrogate Model	96
4.5	Flow chart of PCE-based calibration framework	97
4.6	Output uncertainties in the first order system	100
4.7	Comparison of the original output and the PCE output in the test case .	101
4.8	PCE-based Sobol's indices of contributing factors in the test case	101
4.9	Calibrated contributing factors in the test case	102
4.10	Calibration results in the test case	103
4.11	Comparison of the original output and the PCE output in runway overrun	105
4.12	PCE-based Sobol's indices of contributing factors in runway overrun . . .	106
4.13	Comparison of the simplified model output and the new PCE output in runway overrun	107
4.14	Iteration of the tuned parameters in runway overrun	108
4.15	Calibrated distributions of the contributing factors in runway overrun . .	108
4.16	Calibration results in runway overrun	109
5.1	Predictive Analysis framework with the wind process input	112
5.2	Comparison of the PSD and the eigenvalue in KL expansion for turbulence data in a certain flight condition.	124

5.3	Cumulative variance ratio in KL expansion of the longitudinal turbulence data.	125
5.4	PSDs of the original turbulence data and the KL-based regenerated turbulence data	126
5.5	Comparison between the generated turbulence based on the PSD and the reconstructed turbulence using the KL expansion with different truncated orders.	127
5.6	Headwind in the final approach phase from 849 flights.	128
5.7	Cumulative variance ratio in KL expansion of the headwind.	129
5.8	Eigenfunctions in KL expansion of the headwind.	129
5.9	Reconstruction of three arbitrary headwind series.	130
5.10	Box plot of ξ_k in KL expansion of the headwind.	130
5.11	Empirical marginal CDFs of ξ_k in KL expansion of the headwind.	131
5.12	Estimated marginal PDFs of ξ_k in KL expansion of the headwind.	131
5.13	Empirical normalized pairwise contour plots in the headwind modeling.	133
5.14	Normalized contour plots of bivariate copulas in the vine copula.	134
5.15	Comparison of samples plot between ξ_{10} and ξ_{16} in Z scale.	136
5.16	Empirical CDFs of headwind at a certain altitude.	137
5.17	Estimated PDFs of headwind at a certain altitude.	137
5.18	First four order statistical moments of headwind along with the altitude.	138
5.19	Detection of headwind ramps in one flight.	139
5.20	Mean and standard deviation of the extracted and regenerated headwind ramps.	139
5.21	The estimated PDFs of the change rate of headwind along with the altitude.	141
6.1	FCOM stable approach criteria for an A320 aircraft.	144
6.2	The deviation of headwind $\Delta V_{hws,TD}$ between the METAR information and the encountered wind at touchdown calculated using the operational flight data	146
6.3	The geometry of the glide slope and localizer course.	147
6.4	Aircraft energy boundary during the approach.	149
6.5	Energy circle calculation	150
6.6	Flight parameters time history from 2000 feet above ground to the runway threshold.	151
6.7	Histogram of the energy margin based on 849 flights.	152
6.8	Kinematic speed V_K , aerodynamic speed V_A , and wind speed V_W	153
6.9	Time margin and energy margin in one flight during the final approach.	155
6.10	Histogram of the time margin based on 849 flights.	155
6.11	The runway frame N with three wind components.	159
6.12	Histogram of the maximum F_{avg} factor.	162
6.13	Speed tracking autothrottle system.	162

LIST OF FIGURES

6.14	Simplified autothrottle system block diagram	163
6.15	Comparison of between the raw calibrated airspeed and the simulated calibrated airspeed using the estimated parameters.	164
6.16	Glideslope tracking structure	164
6.17	Simplified inner loop and aircraft dynamics	165
6.18	Time histories of aircraft states during the final approach with the initial deviation in the speed and the altitude, and the gust encountered at 30 seconds.	166
6.19	Distributions of the landing mass during the final approach	167
6.20	Kernel distributions of the landing mass during the final approach	168
6.21	Distributions of the air density at the touchdown point	168
6.22	Distributions of the downwash and the maximum tailwind increment in 100 feet during the final approach phase	169
6.23	Distributions of the headwind and the gust intensity	170
6.24	Estimated PDFs of two wind-related factors using the raw data and the simulated data from the KL-based tailwind model	171
6.25	Histogram of IM based on the MCS with 10000 samples.	172
6.26	Cumulative CDF of the incident metric based on the samples generated in the subset simulation.	173
6.27	Histograms of the model outputs in each subset level.	173
6.28	Proposed σ and acceptable ratios in each subset level.	174
6.29	Correlation-based and regression-based sensitivity indices to IM	174
6.30	Moment-independent sensitivity indices to IM	175
6.31	\mathcal{L}_1 -modified reliability sensitivity indices in each subset level.	176
6.32	Samples plots and histograms of the landing mass in subset simulation.	176
6.33	Samples plots and histograms of the maximum tailwind increment in 100 feet in subset simulation.	177
6.34	Tailwind series along with the altitude in different subset level.	178
6.35	Energy factor series along with the altitude in different subset level.	179
6.36	Comparison of incident probabilities for different mass and flap configuration.	180
6.37	Comparison of incident probabilities for different duration setting.	181
A.1	Overview of rotation angles between frames	V
C.1	States in a Markov chain.	XIV

List of Tables

- 2.1 The evolution of FDR 14
- 2.2 Sync words of subframes 19
- 3.1 Four groups of global sensitivity analysis methods 70
- 3.2 Reliability-oriented Sensitivity Metrics 80
- 3.3 The orthonormal Legendre polynomial basis. 83
- 3.4 The orthonormal Hermite polynomial basis. 84
- 3.5 All the polynomials of z_1, z_2 in the PCE model 87
- 4.1 Contributing factors distributions of the first-order model 99
- 4.2 Statistical moments of the model output in first-order system 100
- 4.3 Contributing factors of runway overrun model. 104
- 5.1 Estimated marginal distributions of ξ_k in KL expansion of the headwind. . 132
- 6.1 Stable approach criteria for B747-8F 145
- 6.2 Speed boundary during the approach 148
- 6.3 Altitude boundary during the approach 149
- 6.4 Estimated parameters in autothrottle system 163
- 6.5 Parameters in the glideslope tracking autopilot system. 165
- 6.6 Contributing factors to low energy events during the final approach 167
- 6.7 Wind shear intensity criteria 170
- A.1 Earth-Centered Inertia (ECI) Frame I III
- A.2 Earth-Centered-Earth-Fixed Frame E III
- A.3 North-East-Down (NED) Frame O III
- A.4 Body-fixed Frame B IV
- A.5 Aerodynamic Frame A IV
- A.6 Kinematic Frame K IV
- A.7 Navigation Frame N V
- B.1 One-dimensional continues distribution VII
- B.2 Bivariate copula families available in VineCopula R package VIII
- B.3 Bivariate copula distribution function. X

Acronyms

ACMS	Aircraft Condition Monitoring System
AGS	Analysis Ground Station
ALoSP	Acceptable Level of Safety Performance
ANOVA	Analysis of Variance
ARMA	Auto Regressive Moving Average
ASHWG	Avionics Systems Harmonization Working Group
ASIAS	Aviation Safety Information Analysis and Sharing
ATC	Air Traffic Control Towers
ATIS	Automatic Terminal Information Service
CAST	Commercial Aviation Safety Team
CDF	Cumulative Distribution Function
CFIT	Controlled Flight into Terrain
CVR	Cockpit Voice Recorder
DAR	Digital Aircraft Condition Monitoring System Recorder
DAU	Data Acquisition Unit
DDM	Depth of Modulation
DFDR	Digital Flight Data Recorder
DFL	Data Frame Layout
EAFR	Enhanced Airborne Flight Recorder
EASA	European Union Aviation Safety Agency
EEM	Equation Error Method
EKF	Extended Kalman Filter
EOFDM	European Operators Flight Data Monitoring
EUROCAE	European Organization for Civil Aviation Equipment
FAA	Federal Aviation Administration
FAF	Final Approach Fix
FCOM	Flight Crew Operating Manual
FDAP	Flight Data Analysis Programmes
FDAU	Flight Data Acquisition Unit
FDIMU	Flight Data Interface Management Unit
FDM	Flight Data Monitoring
FDR	Flight Data Recorder

FMC	Flight Management Computer
FODA	Flight Operation Data Analysis
FOQA	Flight Operations Quality Assurance
FORM	First-order Reliability Methods
FS	Fourier Series
FSD	Institute of Flight System Dynamics
FT	Fourier Transformation
GAJSC	General Aviation Joint Steering Committee
GPS	Global Positioning System
GS	Glideslope
GSA	Global Sensitivity Analysis
HDFS	Hadoop Distributed File System
IATA	International Air Transport Association
ICAO	International Civil Aviation Organization
ILS	Instrument Landing System
IMC	Instrument Meteorological Conditions
IS	Importance Sampling
JSAT	Joint Safety Analysis Team
KL	Karhunen–Loève
LLZ	Localizer
LOC-I	Loss of Control In-flight
LSA	Local Sensitivity Analysis
LSB	Least Significant Bit
MCMC	Markov Chain Monte Carlo
MCP	Mode Control Panel
MCS	Monte Carlo Simulation
MCTOM	Maximum Certificated Take-off Mass
METAR	Meteorological Airport Observation Report
MH	Metropolis-Hasting
MSB	Most Significant Bit
MSE	Mean Square Error
NN	Neural Network
OM	Operational Manual
PCE	Polynomial Chaos Expansion
PCMCIA	Personal Computer Memory Card International Association
PDF	Probability Density Function
QAR	Quick Access Recorder
RS	Runway Safety
RTS	Rauch–Tung–Striebel
SAGA	Safety Analysis for General Aviation

SMM	Safety Management Manual
SMS	Safety Management Systems
SOPs	Standard Operational Procedures
SORM	Second-order Reliability Methods
SPIs	Safety Performance Indicators
SR	Spectral Representation
SRC	Standard Regression Coefficient
SRRC	Standard Rank Regression Coefficient
SSCVR	Solid-State Cockpit Voice Recorder
SSFDR	Solid-State Flight Data Recorder
SuS	Subset Simulation
SVR	Support Vector Regression
TSO	Technical Standard Orders
TUM	Technische Universität München
UA	Unstable Approach
UQ	Uncertainty Quantification
VMC	Visual Meteorological Conditions

Symbols and Indices

Symbols

C_D	Drag coefficient
C_L	Lift coefficient
C	Copula distribution function
E_{factor}	Normalized aircraft energy margin
E_s	Normalized aircraft energy
E	Aircraft energy
E	Intermediate failure events
F^e	Empirical cumulative distribution function
$F_{\mathbf{X}}$	Cumulative distribution function
F	A failure event
F	Force
H_n	Orthonormal Hermite polynomial basis
IM	Incident metric
I_F	Indicator function
J	Cost function
K_I	Integral gain
K_P	Proportional gain
L_n	Orthonormal Legendre polynomial basis
L	Length
$N1_{avg}$	Average reversers fan speed
N_0	Number of seeds in each subset
N_a	Number of chains in each subset
N_c	Number of samples in each chain
N	Number of samples in each subset
P_F	Incident probability
P_r	Probability
P	Static pressure
R	Gas constant
S_i	Sobol's index
T_d	Time delay constant

T_i	Isoprobabilistic transformation
T	Thrust
U	Standard uniform space
V, v	Speed
Var	Varaince
X_i	Contributing factors
X	Original space
Z	Normalized Gaussian space
$\Delta\tau$	Width of the integral interval
Ω	Sample space
Ψ_α	Basis of multivariate orthogonal polynomials
Θ	Unknown parameters
α	Acceptable ratio
α	Angle of attack
β	Angle of sideslip
\mathbf{M}	Transformation matrix
α	Multi-indices of multivariate polynomials
χ_{rwy}	Runway bearing angle
χ	Track angle
δ_F	Flap degrees
$\delta_{GS,DDM}$	Glideslope deviation in DDM
$\delta_{GS,dot}$	Glideslope deviation in Dot
$\delta_{LLZ,DDM}$	Localizer deviation in DDM
$\delta_{LLZ,dot}$	Localizer deviation in Dot
δ_i	Moment-independent index
δ	Coefficient of variance
η_i	Reliability-oriented sensitivity index
γ_m	Intermediate failure threshold in the subset level m
γ	Flight path angle
λ_k	Karhunen–Loève eigenvalues
\mathbb{E}	Expectation
\mathcal{F}_k	Fourier coefficients
\mathcal{N}	Normal/Gaussian distribution
\mathcal{U}	Uniform distribution
μ	Bank angle
μ	Expected value / mean
μ	Roll friction index
μ	Standard deviation
ν	Output measurement noise
ω	One realization in Ω

ω	Input measurement noise
ϕ_k	Karhunen–Loève functions
ϕ	Bank attitude
ψ	Gaussian density function
ψ	Heading
ρ	Linear coefficients
ρ	Air density
τ	Kendall’s tau
θ_{rwy}	Runway slope angle
θ	Pitch attitude
\vec{F}	Forces vector
\vec{M}	Moments vector
ξ	Karhunen–Loève coefficients
ξ	Random candidates sampling from proposed distribution
a	Acceleration
b	Bias
c	Copula density function
d	Number of dimensions
d	Distance
$f_{\mathbf{x}}$	Probability density function
$g(\cdot)$	Model function
g_0	Gravitational acceleration
h	Altitude
k_{α}	Polynomial chaos expansion coefficients
m	Aircraft mass
p_0	Conditional probability
p	Pitch rate
p	Truncated order of univariate polynomials
q	Roll rate
r	Yaw rate
t	Time
u	Input vector
x	State vector
z	Measurement vector

Indices

A	Aerodynamic frame
B	Body-fixed frame
B	Braking
E	Earth-Centered-Earth fixed frame

<i>F</i>	Gravity
<i>I</i>	Earth-Centered inertia frame
<i>K</i>	Kinematic frame
<i>LB</i>	Lower bound
<i>N</i>	Navigation frame
<i>O</i>	North-East-Down frame
<i>P</i>	Propulsion
<i>TD</i>	Touch down point
<i>T</i>	Transposed
<i>UB</i>	Upper bound
<i>W</i>	Wind
<i>cas</i>	Calibrated airspeed
<i>des</i>	Desired
<i>dws</i>	Downwash
<i>gs</i>	Ground speed
<i>hws</i>	Headwind
<i>meas</i>	Measurements
<i>rev</i>	Thrust reverser
<i>sim</i>	Simulated
<i>tws</i>	Tailwind

Chapter 1

Introduction

1.1 Motivation and Background

Safety is always the first concern in aviation. The ultimate goal is to eliminate aircraft accidents and serious incidents. However, human-related activities or systems are not guaranteed to be free of operational errors. Therefore, the safety risk is a dynamic index and should be continuously monitored and mitigated. The International Civil Aviation Organization (ICAO) has provided a systematic approach to manage safety, called Safety Management Systems (SMS), involving the organizational structures, accountabilities, policies, and procedures [1, 2]. The objective of SMS is to provide a structured management framework to mitigate flight risks in operations. Hazard identification and safety risk assessment and mitigation are involved in the safety risk management of SMS. According to the Safety Management Manual (SMM) [3], safety risk assessment means quantifying risks, which is done by analyzing the accident occurrence probability and severity of their consequences.

ICAO SMS defined the safety objectives in terms of an Acceptable Level of Safety Performance (ALoSP) as the minimum level of the safety performance of civil aviation in its state safety program [3]. Furthermore, ALoSP can be described by the probability of an accident or occurrence probability of exceeding the safety performance indicators (SPIs). Airlines are required to set up and assess their safety target according to the ALoSP. The European Commission has defined its safety target as an accident rate of less than one in ten million flights by 2050 [4]. The occurrence probability corresponding to this accident rate is 10^{-7} per flight. Lufthansa has defined its own safety target of less than 10^{-8} per flight [5].

According to the International Air Transport Association (IATA) safety report in 2020 [6], runway safety (RS), controlled flight into terrain (CFIT), and Loss of Control In-flight (LOC-I) are still the top three high-risk occurrence categories. Although the LOC-I accident represented only about 8% of all accidents from 2016 to 2020, it had the highest

percentage of fatal accidents (about 45%) and remains the highest fatality risk (about 65%), accounting for 698 fatalities. Accident reports show that more than half of LOC-I accidents involved the loss of situational awareness, to which the loss of energy awareness is one of the leading precursors [7, 8]. Especially low energy situations during the approach and landing phase bear a high risk, since there is not sufficient time and altitude for the pilot to recover the aircraft. Based on the requirements of ALoSP, quantifying the occurrence probability of low energy events becomes a crucial and challenging problem. The flight data recorders provide a data source for identifying hazards and monitoring the safety performance of the normal operation. The flight safety group at the Institute of Flight System Dynamics (FSD) developed a model-based predictive analysis framework to quantify the occurrence probability of accidents using the operational flight data, which has been implemented in runway overrun [9]. The motivation of this thesis is to apply advanced uncertainty quantification methods to enhance the quality of the Predictive Analysis (PA) framework and implement it for aircraft low energy events.

1.2 State of the Art

This section introduces the development and current activities of flight data monitoring (FDM), followed by the description of the PA framework. Some advanced statistical methods to quantify uncertainties are summarized. Some shortcomings and implementation issues of the current PA algorithms are mentioned in the end.

1.2.1 Flight Data Monitoring

As part of the ICAO SMS process, FDM, also called flight operations quality assurance (FOQA) or flight operation data analysis (FODA), is a proactive and non-punitive program of gathering and analyzing the operational flight data for monitoring the operational safety level. Based on the Commission Regulation (EU) 965/2012, FDM program is mandatory for commercial air transport with a maximum certificated take-off mass (MCTOM) of over 27t for all European Union Aviation Safety Agency (EASA) member states since 29 October 2014 [10]. Many FDM guidances, such as ICAO Flight Data Analysis Programmes Manual (ICAO Doc 10 000) [11], UK Civil Aviation Authority CAP 739 [12], and Federal Aviation Administration (FAA) Advisory Circular 120-82 [13], have been published to share good practices on the implementation of FDM program. Processing of flight data mainly involves exceedance detection, routine measurements, incident investigation, continuing airworthiness and integrated safety analysis.

Nowadays there are several commercial FDM software available, e.g., AirFase [14], Analysis Ground Station (AGS) [15], Aerobytes [16], Austin Digital [17], and so on. The main functions of current FDM software include flight profile visualization, flight animation,

event detection of exceeding the safety-critical threshold, and simple statistical analysis. To help airlines further use and develop the FDM algorithms, the EASA established the European Operators Flight Data Monitoring (EOFDM) [10]. The goal is to share the practical experiences to maximize the safety benefits from flight data analysis among the FDM community, containing airlines, equipment manufacturers, research institutes, and so on. The company Innaxis led a project “SafeClouds” [18], which is supported by many aviation stakeholders. They developed a novel data-driven approach to identify hazards and mitigate aviation risks. In the United States, Aviation Safety Information Analysis and Sharing (ASIAS) program [19], initiated by the FAA, works closely with the Commercial Aviation Safety Team (CAST) and the General Aviation Joint Steering Committee (GAJSC) to monitor known risk, evaluate the effectiveness of deployed mitigations, and detect emerging risk. Another prominent example is the Flight Data Exchange (FDX) service [20], offered by the International Air Transport Association (IATA). The FDX program provides its airline members the deidentified flight data to perform and also a comparative overview to highlight areas of flight safety concern. Many studies are ongoing to mine the benefits from the large set of flight data. In the project “Safety Analysis for General Aviation (SAGA)” [21, 22], researchers applied the Machine Learning algorithms to detect anomalous operations, identify precursors and critical parameters of safety events based on the large set of flight data for general aviation. A Quick Access Recorder (QAR) data analysis software has been developed in [23] for wind shear and turbulence studies. The predictive analysis framework developed at FSD [9, 24] can provide a quantitative statement of an incident probability by combining the operational flight data and the physical models together.

1.2.2 Predictive Analysis Framework

Due to a lack of accident data, the accident probability can not be computed by counting the number of accidents divided by the total number of flights, especially for a new airline that has not had any incident or accident record yet. The PA framework proposed a solution to quantify the incident/accident probability for a specific airline by combining the previous experience (operational flight data) and knowledge (physical model). Although accidents can not be directly observed in daily operation, deviations of the corresponding contributing factors may occur at a high frequency so that they can be measured with statistical significance. Once the relation between the contributing factors and the incident metric can be described by a physical model, uncertainty in the contributing factors can be propagated to the incident metric. The obtained statistical characteristics of the incident metric allow airlines to calculate the occurrence probabilities of accidents and quantify their safety performance level. The PA framework includes seven steps, and a brief introduction is shown here [9, 24]:

Step 1 Define An incident metric (continuous variable) is defined to describe the occurrence of the incident/accident by an inequality constrain. For example, the remaining runway distance can be a metric to describe the runway overrun.

Step 2 Model An incident model is developed to describe the relationship between the causal factors and the incident metric based on the aircraft model, operational procedures, pilot behaviors, and so on.

Step 3 Identify All causal factors, so-called contributing factors, are collected from the FDM data in this step. Some measurements of contributing factors are recorded directly in the flight data recorder. Otherwise, the algebraic calculations and parameter estimation methods are carried out to obtain the other measurements.

Step 4 Cumulate The collection of measurements from many flights shows the uncertainties in contributing factors, which can be described using the probability distribution. The probability distribution and the dependence structure among all contributing factors are identified using the maximum likelihood methods.

Step 5 Calibrate Validation is required to ensure that the model output represents the reality of the relevant flight operation properly before the “Predict” step. The distributions of the incident model output and the corresponding recorded flight measurements are compared in this step. For the minor difference, calibration of contributing factors is implemented without falsifying the measurements of contributing factors. Afterward, the incident model and calibrated contributing factors are used for prediction. If the difference between model output and recorded measurements can not be adjusted to satisfy the tolerated criteria, the incident model should be revised.

Step 6 Revise Advanced statistical methods are applied to identify unknown causal parameters. The incident model is revised by increasing the complexity of the model, or including more causal parameters, or excluding the contributing factors without the significant effect.

Step 7 Predict Uncertainty propagation methods are implemented to obtain the statistical features of the model output and estimate the occurrence probability of the incident.

1.2.3 Uncertainty Quantification Methods

Uncertainty quantification (UQ) is an active research field during the last two decades [25], and the UQ algorithms are widely applied in different research areas, such as structure, materials, electricity, and so on. Nowadays, the UQ plays a significant role in the probabilistic performance assessment of complex engineering systems, especially in the absence of adequate real experimental data [26]. According to the structure of the PA framework, the UQ methods provide theoretical foundations and mathematical tools in the estimation of the incident probability and further uncertainty analyses, such as sensitivity analysis. Figure 1.1 shows the framework of UQ. A literature review in terms of five different objectives of UQ is shown as follows:

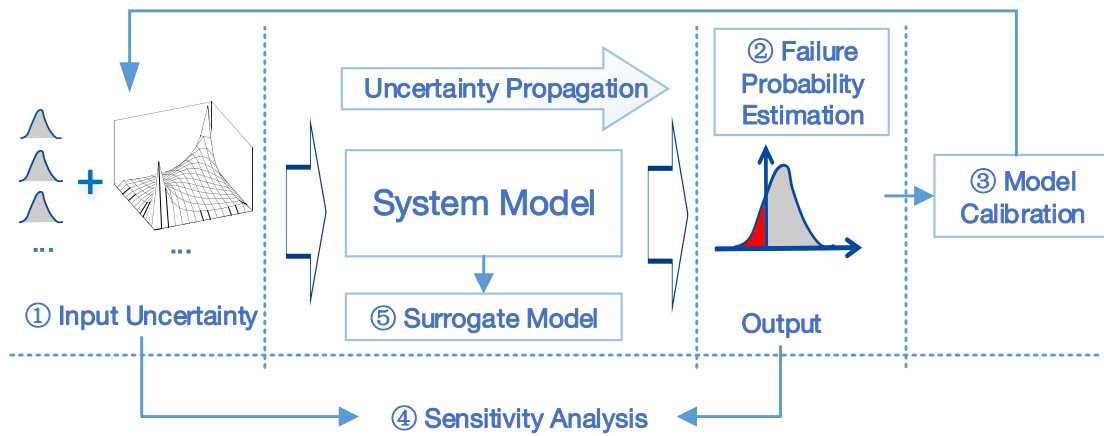


Figure 1.1: *Uncertainty Quantification Framework, modified from [9, 26]*

① Input uncertainty quantification Input variables are usually treated as random variables, which are described using probability distributions. The best-fitting probability distribution function of the measurements can be obtained via the maximum likelihood method. Dependent input variables can be modeled using the multivariate joint probability distribution. However, only multivariate Gaussian or multivariate Student distributions are commonly used. To allow the different marginal distributions, the vine copula approach is proposed in [27, 28] to model the high dimensional dependence via constructing the joint probability distribution using many bivariate copulas.

② Failure probability estimation Estimation of failure probability is not easy, especially when the model evaluation is time-consuming, and the failure is a small probability event. There are two main approaches: approximation-based methods and simulation-based methods. The first-order reliability methods (FORM) and the second-order reli-

bility methods (SORM) approximated the model in the most critical point based on the first/second-order Taylor series of the so-called limit-state functions [29]. However, there is a large error when dealing with high dimensional or highly nonlinear problems using the approximation method. Direct Monte Carlo simulation (MCS) is a classical and robust simulation-based method, independent of the model complexity [30]. Variance reduction techniques such as importance sampling (IS) [31], improve the convergence efficiency of MCS. More advanced methods, such as subset simulation (SuS), are implemented for the small probability event [32]. Paper [33] analyzed the efficiency of different Markov chain Monte Carlo (MCMC) algorithms used in SuS and proposed an adaptive conditional sampling method, which guarantees the optimal acceptance ratio of samples in SuS. To deal with the dependent inputs, paper [34] integrated the vine copula into SuS.

③ **Model calibration** The physical or mathematical model of the system is built based on existing knowledge and observations. It is generally simplified compared to reality. As a necessary step, model validation is usually implemented by comparing the simulation results and observations. The model calibration is to minimize the deviation between the simulated results and measurements. As mentioned in [26], the maximum likelihood method or Bayesian approaches can be applied to adjust the model parameters to obtain a good match between the simulations and measurements. Different from tuning model parameters, [35] minimized the simulation error via calibration of the input factors using the optimization algorithms. Besides, an external error model is proposed in paper [36] for calibration using the statistical approaches.

④ **Sensitivity analysis** Sensitivity analysis is traditionally used in optimization problems to find the steepest gradient of the cost function. In the domain of UQ, sensitivity analysis is applied to measure the influence of different input variables on the output and identify the key driver of output uncertainty [37]. Furthermore, it is also widely applied to reduce the dimensions of the model by ignoring unimportant factors. The local sensitivity analysis (LSA) measures the local response of the output via perturbing the input variables around their nominal ranges, such as the One-at-a-time (OAT) method. In contrast, global sensitivity analysis (GSA), such as variance-based [38] and moment-independent methods [39], describes the output response over the entire range of the input. In addition, the reliability sensitivity indicates the importance of the input factors to failure probability [40].

⑤ **Surrogate model** The surrogate model is used to replace the expensive model and speed up model evaluations [41, 42]. It allows us to achieve the uncertainty of the model output conveniently. Therefore, the surrogate model can be integrated to solve the questions above, like failure probability estimation, model calibration, and sensitivity

analysis. Many surrogate modeling algorithms are available, such as linear and nonlinear regression, polynomial chaos expansion (PCE), support vector regression (SVR), Kriging model (Gaussian process), neural network (NN), and so on. Sufficient samples of input and output are required to guarantee the accuracy of the surrogate models.

1.2.4 Gaps in Current Predictive Analysis Framework

The PA framework provides an advanced strategy to quantify the incident probability. However, there are still some implementation problems demanding a prompt solution for the current development of the PA algorithms. Some questions are listed below.

1. In the “Calibrate” step, [9] first proposed a calibration idea to match the simulation and observations via tuning contributing factors in their nominal ranges, which is subsequently implemented in [35] using the Steepest-Decent optimization algorithm and MATLAB toolbox parameter estimation. Since the calibration process is to solve an optimization problem, MCS is required to compute the cost function in each iteration, which is time-consuming. In addition, the value of the cost function has uncertainties due to the randomness of the samples in MCS. This feature causes the cost function unsmoothed and difficult to converge. An efficient calibration algorithm is required.
2. In the current setup of the PA, each contributing factor remains constant in one evaluation of the incident model. For example, the wind speed is assumed to be constant in the runway overrun incident model [43]. However, the constant wind assumption is not suitable for all incidents, as the variation of wind speed and direction affects the flight performance and handling quality significantly. Therefore, statistical modeling of the wind series is required, and the PA framework should allow the time series as an input.
3. According to the existing seven steps of PA, only the quantitative statement of incident occurrence probability is provided in the results. Sensitivities describe the effect of each contributing factor to the incident metric or the incident probability. Therefore, the sensitivities measure can help airlines to develop better mitigation strategies of incidents/accidents. Sensitivity analysis can be as a post-processing step after the “Predict” step. [44, 45] calculated the sensitivities of contributing factors in the runway overrun case by applying the FORM method. The most likely point of failure used in the FORM is estimated based on the samples in the subset simulation. Since the FORM is the first-order approximation of the limit state function, it will result in considerable errors for the high nonlinear model. More robust and stable algorithms for sensitivity analysis should be investigated to enhance the quality of the quantitative statements.

The PA framework has been only implemented in the runway overrun and veer-off incidents [46]. Other categories of accidents/incidents can be also analyzed using the PA framework. As mentioned in Section 1.1, the aircraft’s low energy state is a safety-critical precursor, especially during the approach and landing. Many energy-based metrics have

been defined for general aviation [47]. Anomaly detection based on energy metrics is implemented in general aviation operations using machine learning algorithms [48]. However, these analyses of energy metrics are only based on the statistical characteristics of the operational flight data. In addition, the researcher in [49] presented abnormal energy risk criteria of large civil airplanes only based on the flight simulations. The operational flight data and the aircraft model are not considered together in the previous research. Furthermore, the operational procedures are not taken into account. Implementation of PA on the aircraft's low energy event can provide a quantitative statement of the occurrence probability and also allows us to identify the key contributing factors for the low energy situation.

1.3 Objectives

The goal of this dissertation is to apply the UQ methods to enhance the quality of the PA algorithms in terms of contributing factors modeling, model calibration, and sensitivity analysis. Furthermore, the enhanced PA algorithms will be implemented for aircraft low energy events. The main objectives are as follows:

- Developing an efficient algorithm to calibrate the contributing factors of the incident model. (Enhancement of the “Calibrate” step)
- Building a wind model that allows us to reproduce the stochastic wind series as an input in the incident model simulations. (Enhancement of statistical modeling of contributing factors)
- Developing sensitivity analysis algorithms to identify the key drivers of the incident metric uncertainty and incident probability. (Development of the post-processing step after the “Predict” step)
- Building a low energy incident model based on flight dynamics and operational procedures, quantifying the occurrence probability, and identifying the critical contributing factors using the operational flight data. (Enhancement of incident modeling)

1.4 Contributions

The following summarizes the contributions of this dissertation:

C1. Calibration framework using PCE in a “frozen sample strategy” with vine copula based dependence modeling

A new calibration framework using a PCE method with a frozen sample strategy is proposed and implemented. The tuned distribution parameters are integrated into the cost function as a penalty term to obtain the overall minimum changes in all contributing factors. The nonlinear dependence structure of contributing factors is considered and integrated into this framework based on vine copula. This approach has been shown to be convergent and greatly improves efficiency because it is not required to rerun the MCS during each iteration of optimization. The proposed calibration algorithm has been validated using a first-order system with three inputs and implemented in the runway overrun incident model.

This contribution will be described in detail in Chapter 4. It has been published in the Scopus listed paper [50] with the title “Calibration of Contributing Factors for Model-Based Predictive Analysis Algorithm using Polynomial Chaos.”

C2. An efficient approach for producing time series with statistical characteristics matching measurements

To appropriately replicate the actual statistical characteristic of the atmospheric motion (wind, gusts, and turbulences) experienced in flights, this dissertation proposes to apply the Karhunen–Loève (KL) expansion method on operational flight data. Compared to the spectral representation (SR) methods, the approach is successfully verified against the well-established von Karman turbulence model. In addition, this approach is also implemented for sensor noise analysis and replication. Results show that the regenerated noise series match the power spectral density of the measurements very well. Furthermore, this statistical modeling approach for producing time series is implemented in simulations to generate noise signals instead of white noise. This contribution will be described in detail in Chapter 5. Together with contributions in C3, it has been published in the Scopus listed paper [51] with the title “Modeling of Stochastic Wind Based on Operational Flight Data Using Karhunen-Loève Expansion Method.”

C3. Dependence construction of KL coefficients for improving the modeling accuracy of time series by utilizing vine copulas

The dissertation introduces a superior description of the statistical characteristics of the KL coefficients by utilizing vine copula to capture high dimensional dependence among the KL coefficients. According to the comparison results of statistical moments, integration of vine copula dependence structure in the KL expansion improves the accuracy of the

constructed stochastic wind model. Furthermore, the headwind during the final approach and the horizontal wind shear ramps detected from operational flights are reconstructed and regenerated using the KL expansion method with vine copula dependence.

This contribution will be described in detail in Chapter 5.

C4. GSA as a post-processing step to enhance the quality of the Predictive Analysis Framework

The original process established for the PA by FSD so far comprised seven steps. This dissertation adds Global Sensitivity Analysis (GSA) as a new step after the “Predict” step to improve the quality of the statements by identifying the key drivers for the predicted incident probability. To achieve that, a broad range of dissimilar GSA approaches have been unified in a common framework and implemented in a general sample-based GSA tool. All the GSA methods are demonstrated for runway overrun and low energy state.

This contribution will be described in detail in Chapter 3 and Chapter 6.

C5. Predictive Analysis of low energy state during the final approach

The low energy event during the final approach due to destabilization is analyzed in detail. According to the stable approach criteria, the energy margin is introduced to detect low energy events. By considering the speed trend and the energy bleed rate, the “time to low energy bound”, called time margin, are also proposed as a new safety metric and applied to a large set of operational flight data. To predict the occurrence probability of low energy events, a novel incident model to simulate the evolution of aircraft low energy states during the final approach is built considering flight dynamics and standard operational procedures (SOPs). Based on the energy margin, the integral of the normalized energy margin in a certain time interval is taken as the incident metric in the PA framework. The expected statistics for the time duration of the aircraft below the low energy bound during the final approach are predicted to assess critical situations. The effects of all contributing factors are analyzed. These properties are computed by stochastic rare-event simulation, using the statistics from recorded operational flight data as initial conditions.

This contribution will be described in detail in Chapter 6. Some results have been published in the Scopus listed paper [52] with the title “Modeling of the Aircraft’s Low Energy State During the Final Approach Phase Using Operational Flight Data.”

1.5 Outline

The structure of this dissertation is summarized in Figure 1.2, and each chapter is briefly described in the following. Chapter 2 introduces the characteristics of the operational flight data, including the evolution of flight recorder, data acquisition, and QAR data decoding. The preprocessing of the operational flight data is also described, which includes the calculation of time points and measurements, and flight path reconstruction. Mathematical preliminaries of rare event probability estimation and sensitivity analysis are outlined in Chapter 3. Chapter 4 describes the strategy of the “Calibrate” step and proposes an efficient calibration framework using PCE methods. Calibration results are subsequently presented using the runway overrun model. Chapter 5 describes the modeling of the stochastic wind process. Headwind with gusts during the final approach is constructed using the KL expansion method, which is used to regenerate the wind series in the incident model simulation. Furthermore, an aircraft low energy incident model based on flight dynamics and operational procedures is proposed in Chapter 6. The occurrence probability of low energy and key contributing factors are estimated and identified. Finally, Chapter 7 concludes this thesis and gives an outlook of future works.

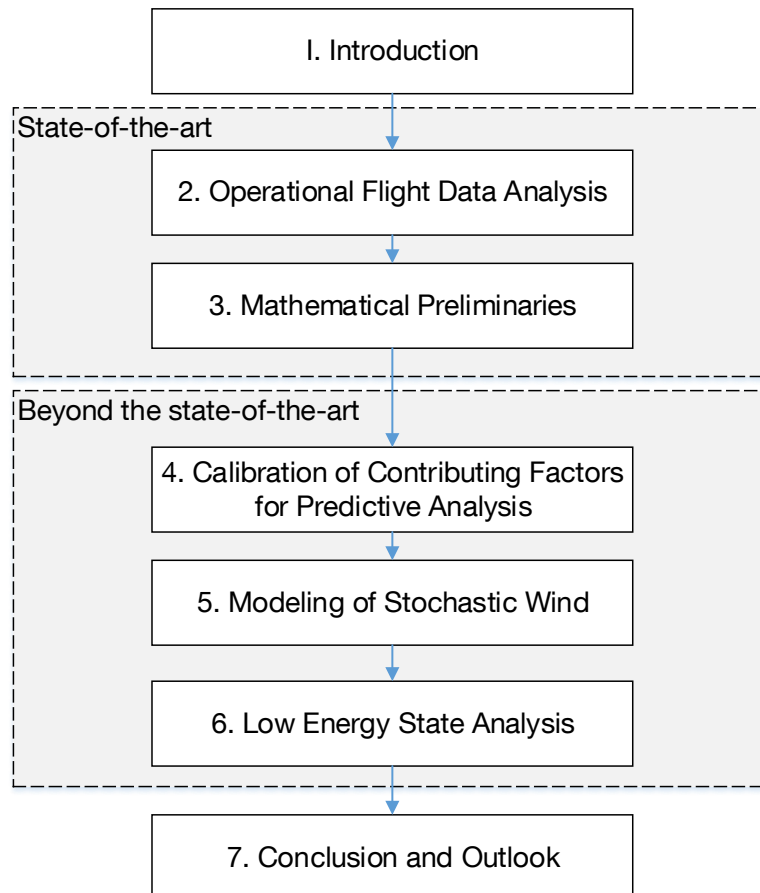


Figure 1.2: *Structure of the dissertation.*

Chapter 2

Operational Flight Data Analysis

The main source of the operational flight data used in this dissertation is the Quick Access Recorder (QAR) data. First, this chapter introduces the evolution of flight data recorders and the aircraft data acquisition structure, followed by an overview of the QAR decoding process. Several necessary preprocessing steps for the decoded QAR data are summarized, which include QAR file splitting, airport and runway detection, time points detection, flight data measurements calculation, and flight path reconstruction. Subsequently, flight data time series and measurements are available for the FDM program. The threshold exceedance detection method used in the FDM program is described, followed by the statistical analysis method of critical measurements. Furthermore, the more advanced physical-based incident model developed at FSD is introduced. The runway overrun model is provided as an example.

2.1 Flight Data Recording

The history of flight data recording can be traced back to World War II. The ‘V-g’ recorder was first installed in military aircraft to collect airspeed and load factor data for structural design improvement [53]. By 1950, the so-called ‘V-g-h’ continuous trace recorders are introduced to consider aircraft height for assessing the structural and aerodynamic implications of gust and loads [54]. In 1957, a combined voice and data recorder was created by Dr. David Warren and his team at Aeronautical Research Laboratory (ARL) [55]. Since the recorded flight data and audio can provide more information about the causes of accidents in addition to the wreckage after accidents, regulatory authorities mandate a Flight Data Recorder (FDR) and a Cockpit Voice Recorder (CVR) into large commercial aircraft for accident investigation in the 1960s. The first type of the FDR only records 5 parameters: heading, altitude, airspeed, vertical acceleration, and time, onto the magnetic metal or photographic film. Recording technology was then developed significantly from the analog signal to the digital signal on the tap, and to the solid-state media. The

Digital FDR (DFDR), Solid-State FDR (SSFDR), and Solid-State CVR (SSCVR) were introduced successively in the 1980s [56]. With the latest technology, Enhanced Airborne Flight Recorder (EAFR) combines the FDR and the CVR into one single device. In summary, the evolution of the FDR in terms of the type and capacity is briefly described in Table 2.1.

Table 2.1: *The evolution of the FDR, source: [57]*

Aircraft type	Service date	FDR type	Number of parameters	FDR data capacity
Boeing 707	1958	Analogue	5	Mechanical limit of about 10 parameters
Airbus 330	1993	DFDR/SSFDR	280	128 wps*
Embraer 170	2004	SSFDR	774	256 wps
Airbus 380	2007	SSFDR	>1000	1024 wps
Boeing 787	2009	EAFR	>1000	Ethernet system

* wps: words per second

Although the FDR and the CVR are referred to as the black box, they are actually in bright orange for searching among the wreckage as shown in Figure 2.1. They are designed to withstand the immense decelerations and temperatures during accidents. In Europe, the European Organization for Civil Aviation Equipment (EUROCAE) specifies the test requirements for the FDR and the CVR in its document ED-112 [58]. Similarly, in the U.S., the FAA listed the test conditions in the documents: Technical Standard Orders (TSO), specifically, TSO-C124b [59] for the FDR and TSO-C123c [60] for the CVR.

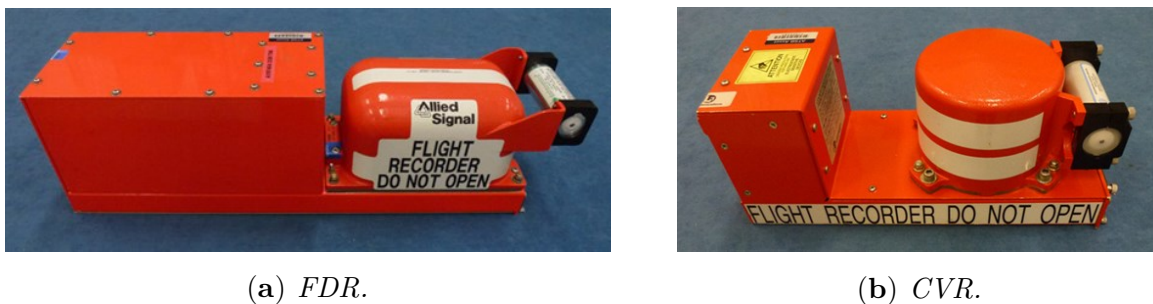


Figure 2.1: *Black box flight recorders, source: [61], Image downloaded on Jan 10, 2021*

Nowadays, for all airplanes of a maximum certificated take-off mass of over 27 ton, the recording time series is requested to contain at least the last 25 hours of aircraft operation to track the aircraft states during the accidents [1]. Furthermore, ICAO regulates the mandatory parameters of the flight recorder for aircraft operators in Appendix 8 'Flight Recorders' of ICAO Annex 6 "Operational of Aircraft" [1], which is adopted by the FAA and EASA. However, the mandatory parameters are not the same for all aircraft. They

depend on the certification date of the aircraft, maximum takeoff mass, and maximum passenger capacity [53]. For example, in the U.S, the set of mandatory parameters based on the manufacturing date is presented in Figure 2.2. Beyond the list of parameters, the authorities also specify the recording parameter properties, such as measurement range, maximum recording interval, accuracy, and recording resolution for the crash-protected FDR.

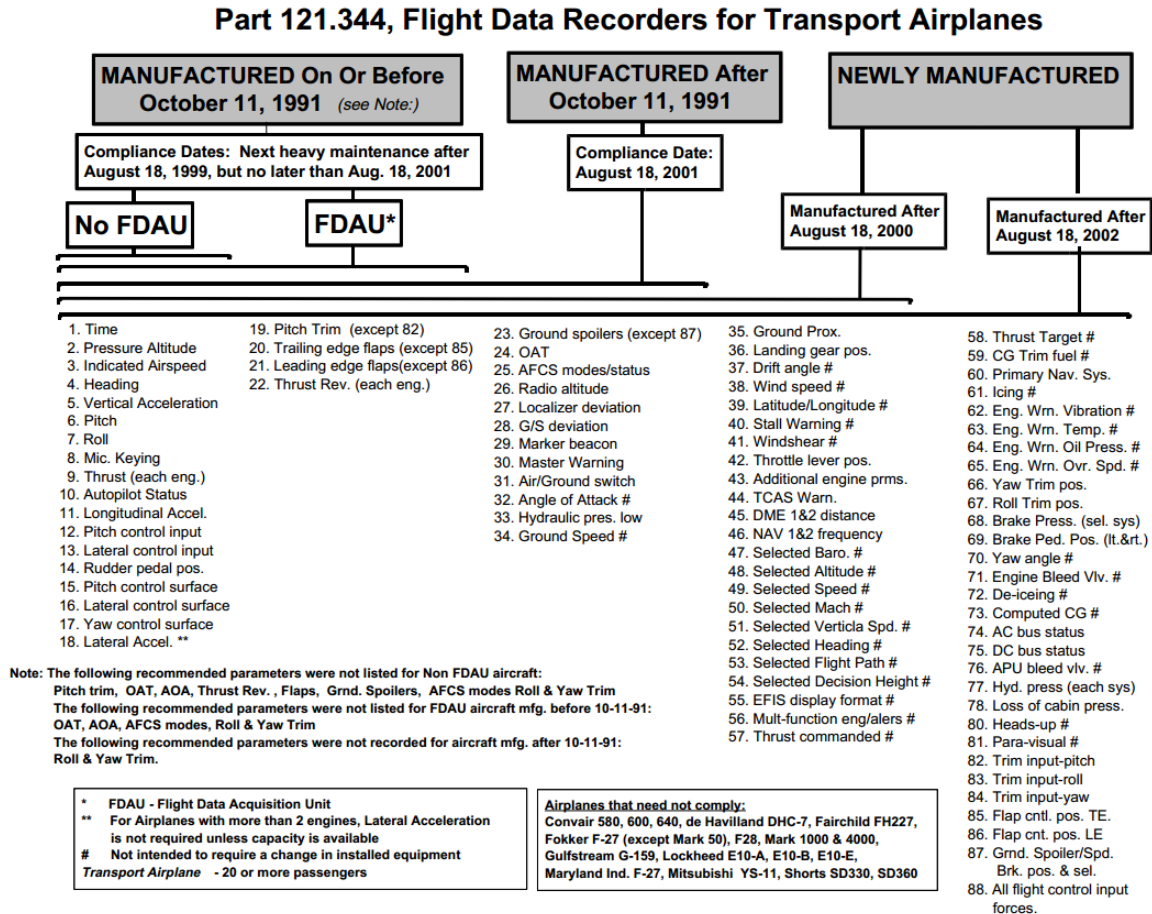


Figure 2.2: Mandatory parameters of the FDR, source: Table 1 of [53]

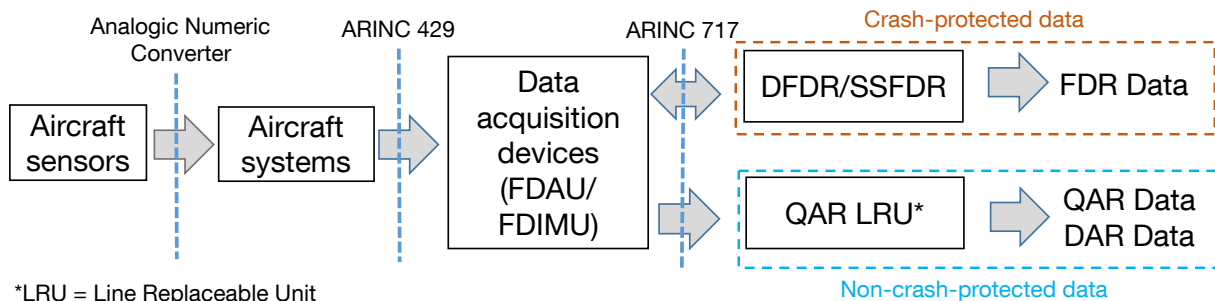


Figure 2.3: Flight data recording architecture, adapted from [62]

Currently, the FDR can record over 3,000 parameters, far more than the mandatory parameters. A data acquisition unit (DAU) is implemented to receive the parameters from all aircraft systems via the relevant ARINC 429 buses and to record the parameters

on the FDR in the ARINC 717 format [63]. Along with the implementation of DFDRs, the current acquisition device is the so-called Flight Data Acquisition Unit (FDAU) for Boeing and Embraer aircraft and Flight Data Interface Management Unit (FDIMU) for Airbus aircraft. The flight data recording architecture is shown in Figure 2.3. It is clearly seen that the data transmission between FDAU/FDIMU and the FDR is bidirectional for verification and synchronization in order to reduce the recording error. This recording chain is crash-protected. Besides the FDR for accident investigation, the flight data analysis processes were encouraged and also requested by authorities. The non-crash-protected QAR is equipped in aircraft for the FDA or FDM programs. Usually, the QAR records the same parameters as the FDR does. Since the data transmission between FDAU/FDIMU and the QAR is unidirectional, the recording errors like synchronization errors might occur. In addition, the FDAU/FDIMU also allows aircraft operators to program and record additional non-mandatory parameters for other purposes, such as maintenance prediction, or fuel monitoring. Through the Aircraft Condition Monitoring System (ACMS), those additional parameters are recorded to other non-crash-protected recorders, called the Digital ACMS Recorder (DAR).

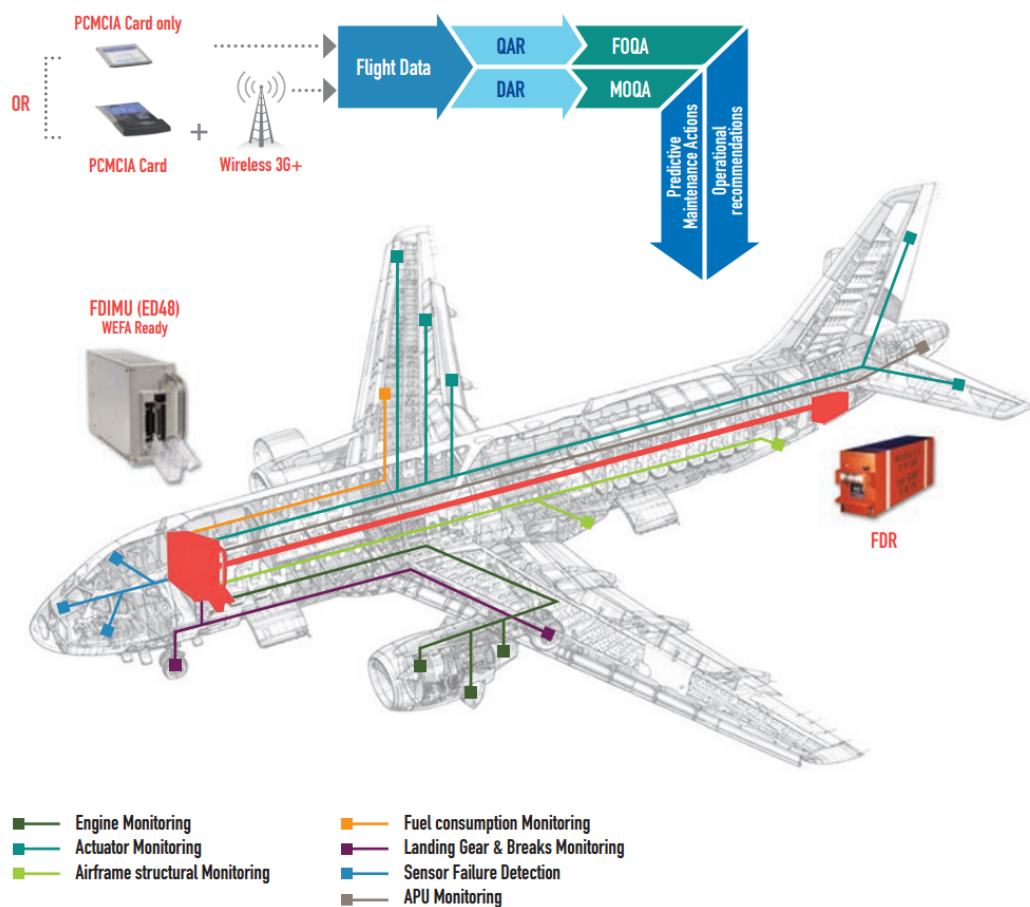


Figure 2.4: An overview of on-board data transmission via ED48 FDIMU, source: [64].

As one example of flight data acquisition and transmission, the ED48 FDIMU for the Airbus A320 family is shown in Figure 2.4. It combines mandatory flight data acquisition and high-performance monitoring capabilities into a single unit. The QAR and the DAR data are stored via the ED48 FDIMU during the flight. As expressed by the name of the QAR, the QAR data is easily accessible. The data is often stored on Personal Computer Memory Card International Association (PCMCIA) media or Secure Digital (SD) memory cards, which are easy to be copied to the flight data analysis software. Furthermore, the so-called ‘Wireless QAR’ (WQAR) allows the aircraft to transmit the data automatically via the cellular network with the suitable infrastructure at the airport. Therefore, the QAR data are quite easy to be copied or transmitted to the airline’s FDM system.

2.2 QAR Decoding

As the QAR data is encoded in binary, it has to be decoded into the engineering units before it is used in the analysis. The QAR decoding process is one of the main functions of the FDM software. Standards of the QAR data format are required for the decoding. As mentioned in the last section, the onboard flight data is recorded based on the ARINC standards. There are several ARINC standards existing for data transmission, encoding, and decoding [65]. ARINC 429 is a predominant data transfer protocol and describes the definition of the physical and electrical interface for bus communication in the aircraft. ARINC 573 provides the data bus standard that defines the acquisition of flight data for recording. As an update of ARINC 573, ARINC 717 is used for QAR decoding for the majority of aircraft types. By utilizing the latest bus technologies, ARINC 767 was developed for QAR decoding of new aircraft such as B787 and A350. And this ARINC 767 can be able to solve the asynchronous and low-frequency recording problems [66]. Owing to the fact that ARINC 717 is still widely used nowadays, an overview of decoding QAR data based on the ARINC 717 standard is given in this section.

The original data in the QAR file is a binary data stream. According to the ARINC 717 format, the structure of the binary data stream can be divided into superframes, frames, subframes, words, and bits [63, 67]. The data frame structure is visualized in Figure 2.5. The bit is the smallest unit, which is 0 or 1. Twelve bits are packed into one word. Depending on the capacity of the QAR and the FDAU, between 64 and 1024 words are combined into a subframe. Each subframe is recorded in 1 second. A single frame contains 4 subframes. Therefore, one frame is recorded in 4 seconds. 16 frames combine a superframe, which becomes a binary data stream by repeating itself subsequently.

The related physical information in four different subframes can be independent, but each frame corresponds to the same information only with different time steps. Therefore, a recorded parameter has to occur once in a frame. It means the minimum sampling frequency is 0.25 Hz. If the parameter occurs once in each subframe, the corresponding

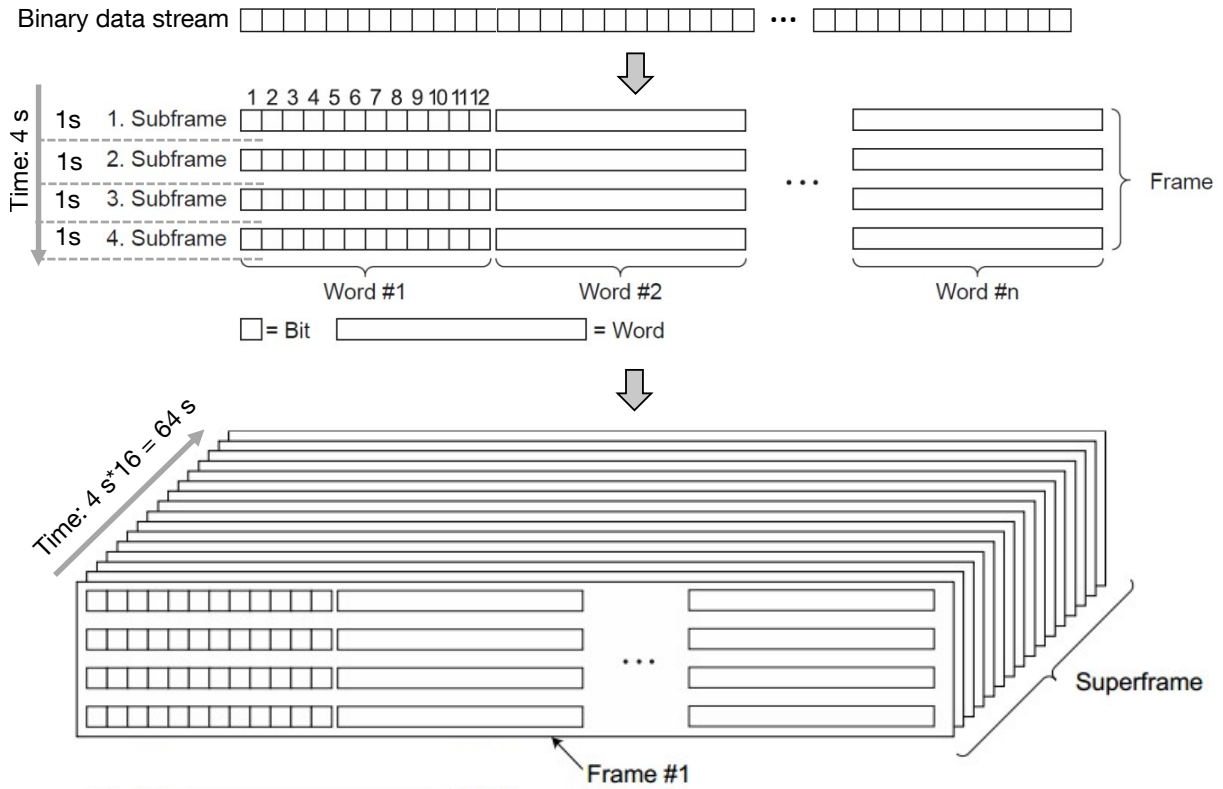


Figure 2.5: Data frame 3D structure of the QAR binary data stream [65].

sampling frequency is 1 Hz. The sampling frequency can be increased by repeating the parameter in each subframe. 8 Hz or 16 Hz is commonly used as the maximum sampling frequency. For some special parameters, such as departure and arrival airport, flight number, or aircraft type, they might change very slowly or do not change during flight. Those parameters can be recorded in terms of the superframe parameter with $1/64 = 0.015625$ Hz. For the superframe parameters, the same position of each frame can store different parameters. To be more precise, the superframe parameter can store 16 parameters with a sampling rate of $1/64$ Hz instead of one parameter recorded in 16 different time steps.

The decoding process is to convert the QAR binary data to the engineering values. Figure 2.6 shows an example of transformation between the binary data and the decimal number. Six bits are used to record this parameter in this case. The corresponding decimal value of the binary stream ‘110101’ is ‘53’. The bit with the lowest position is called the ‘Least Significant Bit’ (LSB). In contrast, the ‘Most Significant Bit’ (MSB) denotes the highest position [68]. LSB is referred to 2^0 in the example, and MSB is equal to 2^5 .

To decode a specific parameter, the location of the corresponding bits in the binary data stream should be known. First, the start position of each subframe needs to be identified. Four sync words are used as markers for four subframes, respectively. They have a unique value without any meaning of the aircraft. These sync words are presented in Table 2.2. By identifying the location of the sync words, the 3D data frame structure of the QAR binary data shown in Figure 2.5 is constructed.

	MSB			LSB		
Binary data	1	1	0	1	0	1
	$2^5 = 32$	$2^4 = 16$	$2^3 = 8$	$2^2 = 4$	$2^1 = 2$	$2^0 = 1$

\downarrow \downarrow \downarrow \downarrow \downarrow \downarrow

$$= (1 * 2^5) + (1 * 2^4) + (0 * 2^3) + (1 * 2^2) + (0 * 2^1) + (1 * 2^0)$$

$$= 32 + 16 + 0 + 4 + 0 + 1 = 53_{10}$$

Figure 2.6: Transformation between a binary data and a decimal value.

Table 2.2: Sync words of subframes, source: [63]

Sync word	Binary	Octal	Decimal	Hexadecimal
Subframe 1	MSB 001001000111 LSB	1107	583	247
Subframe 2	MSB 010110111000 LSB	2670	1464	5B8
Subframe 3	MSB 101001000111 LSB	5107	2631	A47
Subframe 4	MSB 000111011011 LSB	6670	3512	DB8

However, the data can not be decoded yet after the constructed 3D structure, as the recorded QAR data does not provide any information about the parameters, like the name and the data type. Such information is given by the data frame layout (DFL), which provides the necessary information of the parameters as below:

- Location of parameters (words and subframe number)
- Number of bits (including the start and the end bit location)
- Type and method of encoding (Binary coded decimal, discrete, char, linear,...)
- Functions to obtain the engineering values from the recorded binary data.

As an example, the data frame information for airspeed is given in Figure 2.7. According to the specified words (19) and subframes ($ALL(1, 2, 3, 4)$), the locations of the bits for the airspeed parameter in the 3D data frame structure are identified, and the bits can be further converted to decimal values. Obviously, the airspeed is recorded at 1 Hz using 12 bits. Coding a 12-bit binary data is equivalent to a decimal value from 0 to $2^{12} - 1 = 4095$. By the linear conversion, the decimal value from 0 to 4095 is equivalent to the engineering value from 0 knots to 1024 knots. According to the available DFL, all other parameters can be decoded into the engineering value by doing the same procedures. In the end, the time series of the recorded flight parameters are obtained.

Parameters	AIRSPEED
Words	19
Sampling Rate	1 sample per second
Bits	12-1 (12 bits)
Frames	ALL
Subframes	ALL (1,2,3,4)
Units	KNOTS
Minimum	0
Maximum	1024
Parameter Type	<ul style="list-style-type: none"> • Linear • Format: $Y = m * X + b$ <ul style="list-style-type: none"> ○ Input in decimal X ○ Output in engineering unit Y ○ Precision $m = 0.25006$ ○ Offset $b = 0$

Figure 2.7: Data frame information for airspeed [69].

2.3 QAR Data Preprocessing

After decoding the QAR data into engineering values, several data preprocessing steps are commonly required in practice, such as QAR file splitting and airport/runway detection, which are briefly introduced at first. The following subsections discuss the time points and flight data measurements, which are usually used to describe operational behaviors. Furthermore, the flight path reconstruction is also described to improve the QAR data quality.

2.3.1 File Splitting

A QAR file usually records several flights data. It benefits data management and analysis of flights to split the decoded flight time series of a QAR file into individual flights. The pressure altitude recorded in an original QAR file is shown in Figure 2.8. It is clearly seen that this QAR file includes three flights. The accurate split point of two flights can not be distinguished directly only from the altitude data, as the altitude keeps almost constant when the aircraft is on the ground. Theoretically speaking, flight parameters, like ground speed or the engine rotation fan speed N1, can be used to detect the accurate split point, as these parameters become or are close to zero after the aircraft stops completely. However, in practice, the aircraft might stop recording the QAR data before the aircraft or the engine stops completely. The recorded flight data in the QAR might not contain an entire flight. Therefore, a file splitting criterion is required, and it usually is a synthetic

judgment based on several variables, such as ground speed, engine N1, engine fuel flow, and the logical variable like aircraft on the ground [67]. First, the aircraft on ground/air can be distinguished when the ground speed is smaller than 20 m/s , or the recorded signal of aircraft on the ground is positive. Then, the minimum of engine N1 and the engine fuel flow, inside the time range of the aircraft on the ground, is found and the corresponding time point is considered as the split point of two flights. Other parameters, like wheel speed (if recorded), can be also integrated into the split algorithm to improve the robustness and accuracy.

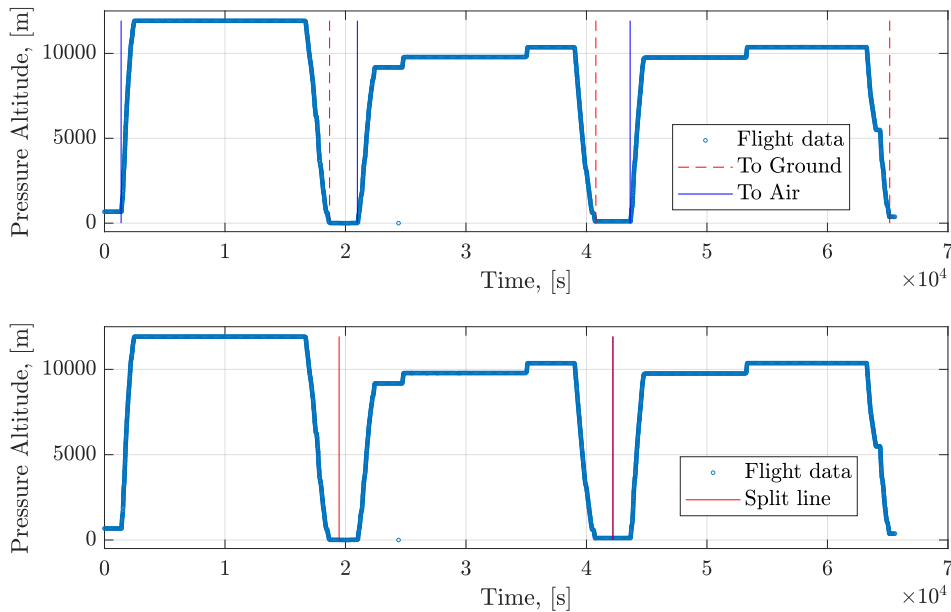


Figure 2.8: Pressure altitude time series of three flights in a QAR file.

2.3.2 Airport and Runway Detection

For an individual flight, the departure/arrival airport and runway of this flight are usually discussed at first for flight data management and further takeoff and landing performance analysis. However, this information might be not recorded directly in the QAR data. The detection algorithm of the departure/arrival airport and runway is requested in practice, which is based on the location parameters recorded in the QAR data. In addition, the world airport database is necessary for the detection to provide the latitude and longitude positions of the runway threshold and the runway heading. The world airport information is available on the website [70]. For example, in the detection of the arrival runway, the touchdown point is detected from the QAR data first. The recorded latitude and longitude of the aircraft at the touchdown point are compared with the world runway

database to find the closest airport and runway. Furthermore, the direction from the runway threshold to the aircraft touchdown point is also computed, which is compared with the runway heading to distinguish the close parallel runways shown in Figure 2.9.

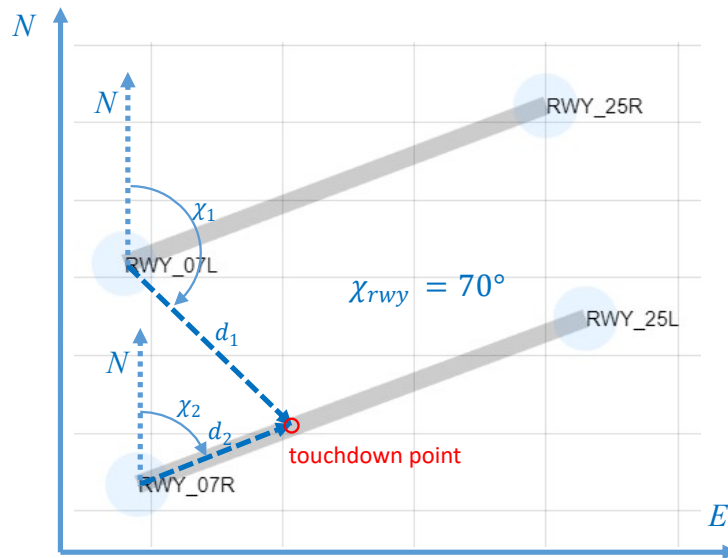


Figure 2.9: Runway detection in Hong Kong airport.

2.3.3 Time Points Detection

The operational requirements in different flight phases are not the same. According to the recorded altitude, speed, configuration, and other variables in the QAR data, time points of starting takeoff, initial climb, climb, cruise, descent, approach, and landing can be derived. Then, the specific flight phase can be extracted based on these time points.

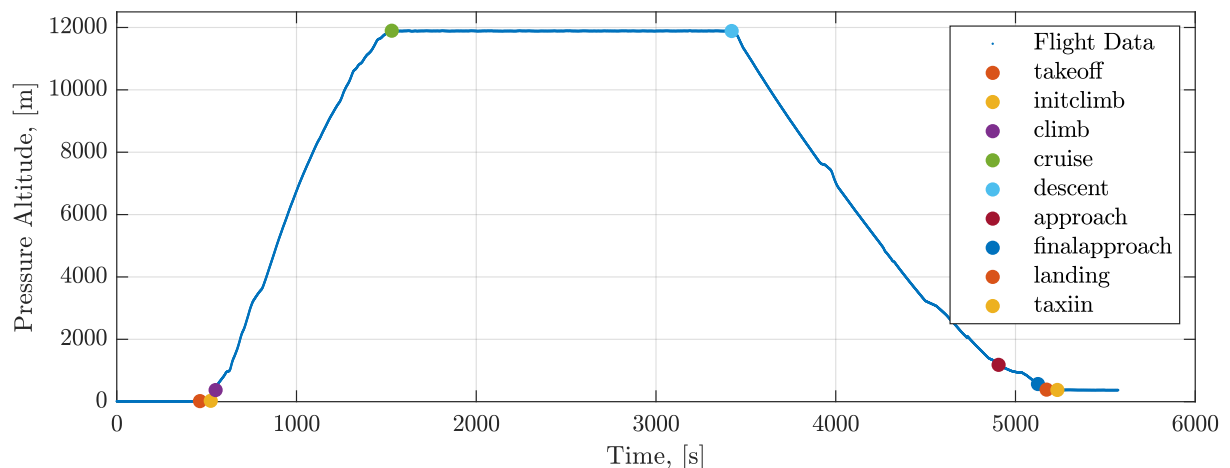


Figure 2.10: Time points calculation of flight phases in a flight.

Beyond the flight phase detection, many time points or their corresponding flight data measurements are very important for flight safety analysis. One example is the touchdown point during the landing phase. The distance from the runway threshold to the touchdown

point, the so-called touchdown distance, is one of the contributing factors of runway overrun. In addition, the attitude at the touchdown point is related to the incidents of wingtip strike or tail strike. Measurements of the load factor increment and the vertical rate of descent increment at the touchdown point indicate if the hard landing event occurs. These mentioned touchdown performances rely on the accurate detection of the touchdown point. Even an error of one second in the touchdown point detection might lead to a 100-meter deviation of the touchdown distance. A considerable variation of the attitude and the load factor also exists in one second. Considering the measurement errors and the low-frequency recording characteristics of the QAR data, it is difficult to determine the touchdown point accurately. Therefore, a high-precise detection algorithm of critical time points, like the touchdown point, is necessary. A model-based touchdown detection algorithm is developed at FSD to detect the touchdown point more precisely [71, 72].

2.3.4 Flight Data Measurements

From the decoded time-series data, flight data measurements are derived to describe operational behaviors [12]. As already mentioned in time points detection, flight data measurements are usually defined as the recorded variable's value at a specific time point. In addition to that, measurements can be also calculated values based on the flight data. According to the complexity of calculation, three types of measurements are considered [9].

Direct Measurements

The simplest type of measurements, the so-called direct measurement, is defined as the recorded time series at a specific time point, such as the ground speed at the touchdown point. As the ground speed is available in QAR data, the measurement of the ground speed can be directly read out once the touchdown point is calculated. Therefore, the main challenge is to calculate the time point accurately. The interpolation or smoothing algorithms might be required due to the low-recording-frequency characteristics of the QAR data.

Algebraic Measurements

Considering a variable, which is not recorded directly in the QAR, it might be obtained by a simple algebraic calculation based on other recorded variables. For example, the air density is usually not provided in the QAR data. Instead, the static pressure and the temperature are commonly recorded directly. The air density can be calculated according to the gas law [73] as follows:

$$\rho = \frac{P}{RT}, \quad (2.1)$$

where P denotes the static pressure, T is the temperature, and R is the gas constant. After the calculation based on Equation (2.1), the air density at a specific time point can be obtained as a measurement.

Another example is the equivalent acceleration during the landing phase. The auto-brake mode is selected in most landing cases of Boeing 747-8F. To describe the manual braking behavior, the equivalent acceleration is proposed [9], which is one of the most important contributing factors of runway overrun, see Section 2.5. Although the actual acceleration varies during the manual braking, the equivalent acceleration can be calculated based on the speed reduction as below:

$$a_{x, equ} = \frac{v_1^2 - v_2^2}{2t}, \quad (2.2)$$

where v_1 denotes the ground speed at the touchdown point. v_2 denotes the ground speed of 80 knots, which is the controllable speed during taxi [74], referred to Section 2.5. t is the duration of the aircraft deceleration from v_1 to v_2 . By using Equation (2.2) in a flight, the measurement of the equivalent acceleration of the landing is obtained.

Parameter Estimation

Unlike the above two types of measurements, the required measurement might not be expressed explicitly based on the available variables in the QAR data, such as the vertical wind speed, the aerodynamic coefficients of the aircraft, etc. These unknown parameters or variables can be estimated using parameter estimation methods based on the maximum likelihood principle. Although these parameter estimation methods have been widely used in the estimation of aerodynamic coefficients from the flight testing data [75], there are many challenges during the implementation using the QAR data. Compared to the flight testing data, the QAR data are recorded in a low frequency and a low resolution. Furthermore, the lack of dedicated maneuver in the operational flights results in lower information contents in the QAR data than the flight testing data. Therefore, there are no parameter estimation techniques provided in the current standard FDM software. A heuristic parameter estimation approach for the QAR data is developed at FSD [76, 77]. To overcome the low information of the QAR data, this approach allows us to process several flights simultaneously. For details, readers are referred to [77].

2.3.5 Flight Path Reconstruction

The recorded variables in the QAR have different recording frequencies with low resolutions. To obtain more accurate aircraft states, like position and speed, the flight path reconstruction or the so-called data compatibility check is implemented to improve the data quality. As the relations among aircraft states are encoded in the aircraft kinematic equations, the flight path reconstruction can utilize these relations to obtain an optimal

estimation of aircraft states without falsifying the recorded data. Extended Kalman Filter (EKF) has been widely used to estimate aircraft state in navigation field [78]. For the offline trajectory reconstruction, the Rauch–Tung–Striebel (RTS) smoother is recommended to get the optimal state estimation [79], which is a combination of a forward EKF and a backward EKF. Landing trajectories reconstruction using the RTS smoother incorporated with instrument landing system deviation information and taxiway locations has been developed by the flight safety group at FSD [71, 80]. The involved aircraft states and measurements are introduced in this section. The general mathematical formulations to describe the aircraft dynamic systems are shown as below:

$$\text{state equation: } \dot{x}(t) = f(x(t), u_m(t) - \omega(t), \Theta), \quad x(t_0) = x_0, \quad (2.3)$$

$$\text{output equation: } z(t) = g(x(t), \Theta) + \nu(t), \quad (2.4)$$

where $f(\cdot)$ denotes the state equation of aircraft motion. $x(t)$ denotes the vector of aircraft states. u_m is the vector of system input measurements with the input noise vector $\omega(t)$. Besides x , Θ is the vector of the extended states, which denotes the unknown parameters. $z(t)$ denotes the vector of aircraft output measurements with the output noise vector $\nu(t)$. The output measurements can be calculated using the output equation $g(\cdot)$ combined with the aircraft states and the extended states. In the landing trajectories reconstruction [80], the involved aircraft states are shown as below:

$$x = [u_B, v_B, w_B, \phi, \theta, \psi, x_N, y_N, z_N]^T, \quad (2.5)$$

where u_B, v_B , and w_B denote three components of the kinematic speed in B frame. ϕ, θ , and ψ denote roll, pitch, and heading angle, respectively. x_N, y_N , and z_N are the three components of the position in the runway N frame. The input measurements contain three accelerations and three angular rates as follows:

$$u_m = [a_x, a_y, a_z, p, q, r]^T. \quad (2.6)$$

Due to the existing biases in the measurements of accelerometers and gyroscopes, three unknown acceleration biases and three unknown angular rate biases are defined as the extended states Θ to estimate. In addition, the biases of the Global Positioning System (GPS) measurements are also included in Θ as below:

$$\Theta = [b_x, b_y, b_z, b_p, b_q, b_r, b_{x,GPS}, b_{y,GPS}]^T. \quad (2.7)$$

Θ are assumed to be constant and will be estimated using the RTS smoother. The output vector $z(t)$ is composed of the measured variables from aircraft sensors as below:

$$z = [V_{gs}, \dot{h}, \chi, \phi, \theta, \psi, x_{GPS}, y_{GPS}, h_{baro,corrected}, h_{RA}, \delta_{LLZ,DDM}, \delta_{GS,DDM}]^T. \quad (2.8)$$

These measurements: ground speed V_{gs} , vertical speed \dot{h} , track angle χ , three attitude angles $[\phi, \theta, \psi]$, GPS positions $[x_{GPS}, y_{GPS}]$, and barometric altitude $h_{baro,corrected}$ are recorded directly and can be also calculated using the x and Θ . In addition, the available

radio altitude h_{RA} combined with the terrain data [81] is integrated into the output equations as well. Furthermore, the localizer deviation $\delta_{LLZ,DDM}$ and the glideslope deviation $\delta_{GS,DDM}$ are utilized when the Instrument Landing System (ILS) signal is available. Once the state equation and the output equation are well defined, the RTS smoother algorithm is implemented to obtain a better estimation of all states by considering the input noise ω and the output noise ν . For the detailed algorithms of the RTS smoother, readers are referred to [80].

Compared to the raw QAR data, the frequencies of the states are homogenized after reconstruction. For the unknown parameters, not only the biases of accelerometers and gyroscopes are estimated, but the unrecorded vertical wind speed can be also obtained. The comparison of the raw and the smoothed wind speed data during the approach phase is shown in Figure 2.11. It shows a good match between the raw and the smoothed horizontal wind speed, which validates that the smoother algorithm works well. The smoothed wind speed data is used in Chapter 5 and Chapter 6. Besides, results of the flight path reconstruction could be applied to improve the accuracy of time point detection and flight data measurements calculation.

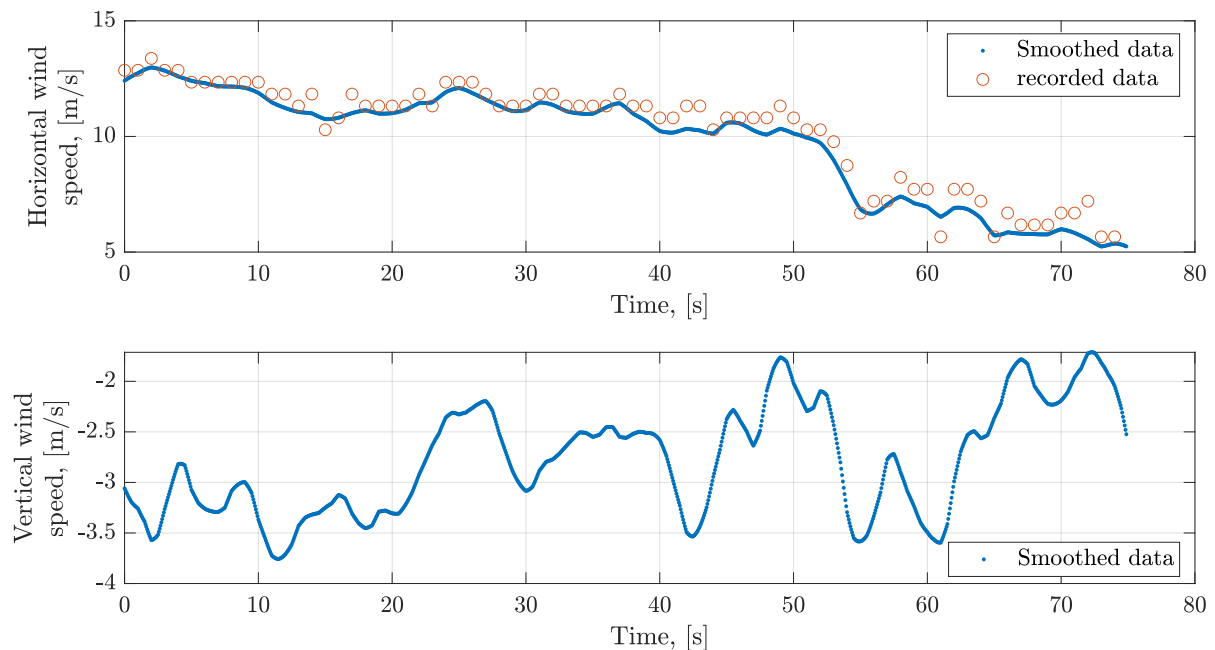


Figure 2.11: *The recorded and smoothed wind speed during the final approach phase in one flight.*

In addition, flight data management is also very important, as a huge amount of confidential QAR data is required to be classified and stored. An IT infrastructure using Hadoop Distributed File System (HDFS), MySQL, and MATLAB, is developed by the Flight Safety working group at FSD. It is not the scope of the thesis, readers are referred to Appendix B of [82].

2.4 FDM Events

After the preprocessing of the QAR data, the obtained time points or flight data measurements can be used directly for the FDM program. As described in the ICAO Manual on Flight Data Analysis Programmes (FDAP) [11], the flight data analysis (FDA) mainly involves three aspects:

“a) capturing and analyzing flight data to determine if the flight deviated from a safe operating envelope; b) identifying trends; c) promoting action to correct potential problems.”

The main objective of FDM is carried out by evaluating deviations of flight data measurements exceeding certain thresholds. The threshold exceeded events, so-called FDM events, are reported in the flight safety report to monitor the abnormal operations. The statistics of the critical measurements related to the FDM events are also analyzed, for example, the analyses and statistical summaries of Boeing 747-400 aircraft operational flight data from 11,066 flights presented in [83]. The analyses enable FAA to reassess existing certification criteria. According to the ATR FDM training report [84], threshold exceeded detection from operational flight data is still the main approach to analyze flight data. Besides, several severity categories for the FDM events, such as low, medium, and high, are defined according to the different thresholds or the exceeding duration time.

A late touchdown is one example of FDM events, which might cause runway overrun. The measurement of touchdown distance is extracted and monitored for this FDM event. The histogram of the touchdown distance obtained from 1133 B747-8F flights is shown in Figure 2.12. The corresponding distribution can be estimated to identify the probability of the touchdown distance exceeding the touchdown zone limit.

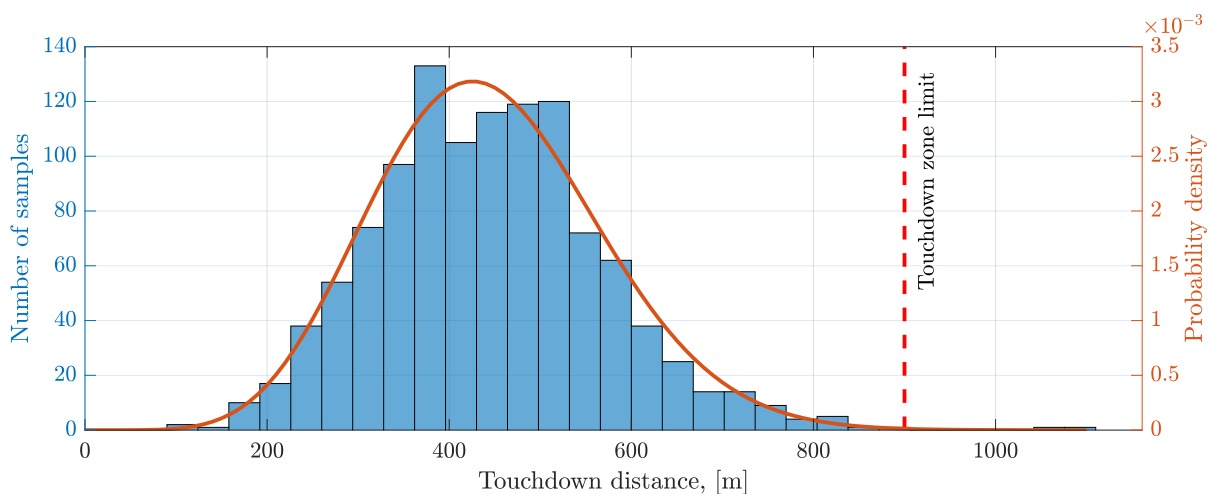


Figure 2.12: Histogram of touchdown distance measurements from 1133 B747-8F flights in ELLX airport.

The touchdown zone limit is a severe threshold for a late touchdown. Other thresholds for low and medium severities can be defined according to the touchdown distance distribution. The touchdown distance is an intermediate state of the aircraft. Before the touchdown, the hidden causal factors contributing to the late touchdown event, such as the flare maneuver, could be analyzed via the aircraft motion. After the touchdown, the aircraft motion can be used to quantify the influence of touchdown distance on the runway overrun. To mitigate potential problems of the FDM event, the physical-based incident model is proposed to analyze the causal factors and also quantify the influence of the FDM precursor on the incidents or accidents, which is discussed in the following section.

2.5 Physical-based Incident Model

The main advantage of the PA framework is that the operational flight data and physical-based incident model are combined in order to assess the operational risk. After the detailed discussion of flight data above, the physical-based incident model is introduced using a runway overrun model in this section.

Before the incident modeling, the incident metric has to be defined firstly in the PA framework. It is a special type of flight data measurement, which can be used to distinguish accident flights from normal flights. The incident metric could be defined based on the incident/accident precursors, FDM events, or SPIs. In addition, the incident metric must be a continuous variable. When the calculated incident metric is closer to the safety threshold, the flight is closer to the incident region. Beyond the simple statistical analysis of the incident metric, the physical-based incident model is built to map the incident metric and its corresponding contributing factors. The operational uncertainties of contributing factors and the incident model are combined. It allows us to estimate the incident probability and also to quantify the influence of all contributing factors on this event. One example of the developed incident model, the runway overrun model [9], is given below.

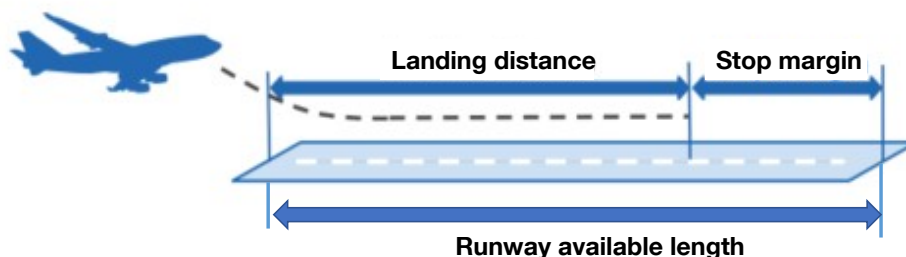


Figure 2.13: *The stop margin as an incident metric for runway overrun, source: Figure 6-2 of [9].*

As shown in Figure 2.13, the stop margin SM is the distance between the end of the runway and the stop position of the aircraft. It is used as the incident metric for runway overrun. The safety threshold is 0 in this case as below:

$$\begin{aligned} \text{Incident metric: } SM &= L_{rwy} - L_{ldg}, \\ \text{Safety threshold: } &0, \end{aligned} \quad (2.9)$$

where L_{rwy} is the runway available length. L_{ldg} is the required landing distance. If the incident metric is less than the safety threshold ($SM < 0$), the runway overrun incident occurs. L_{ldg} is influenced by aircraft states, runway conditions, pilots' behaviors, and so on. L_{ldg} can be split into three phases: touchdown, deceleration, and maximum braking phase, which are modeled respectively.

$$L_{ldg} = d_{td} + d_{80} + d_{stop}. \quad (2.10)$$

In the first phase, the touchdown distance d_{td} , can be identified directly from the QAR data. In the following deceleration phase, the aircraft ground speed decreases to a controllable speed (assumed to be 80 knots [74]) by using braking systems, like ground spoilers and thrust reversers. d_{80} corresponds to the distance from the touchdown point to the position that the ground speed reaches 80 knots. After the deceleration phase, the pilots might keep the controllable speed to the taxiway. To calculate the required landing distance, the maximum braking force is applied in the third phase. The virtual distance d_{stop} denotes the minimum required distance that the aircraft decelerates from 80 knots to a stop. d_{80} and d_{stop} are computed based on the aircraft motion driven by the external forces. The longitudinal acceleration a_x of the aircraft is modeled as below [43]:

$$a_x = \frac{1}{m} (F_P + F_D + F_G + F_B), \quad (2.11)$$

where m is the landing mass. There are four types of external forces. F_P denotes the reverse thrust, and it depends on the time of reverser deployment t_{RevDpl} and the average reverser fan speed $N1_{avg}$ as below:

$$F_P = 0.8 \frac{\rho}{\rho_0} f_T(t_{rev}, N1_{avg}), \quad (2.12)$$

where f_T is the function to calculate the reverse thrust, and it is corrected according to the ratio between the local air density ρ and the standard atmospheric density ρ_0 .

The drag force F_D is calculated as below:

$$F_D = -\frac{1}{2} \rho V_A^2 S_{ref} C_D, \quad (2.13)$$

where S_{ref} is the reference wing area. C_D denotes the drag coefficient. V_A is the true airspeed, and decrease during the landing along with the ground speed. The initial value of V_A can be computed based on its corresponding indicated airspeed V_{IAS} , which is the sum of the target approach speed V_{app} and the measured approach speed deviation ΔV_{app} . V_{app} is influenced by the configuration and the landing mass.

F_G is the longitudinal component of the gravity along the runway, influenced by the runway slope θ_{rwy} as follows:

$$F_G = mg_0 \sin \theta_{Rwy}. \quad (2.14)$$

The braking force F_B driven by the landing gear is the product of the friction index μ and the ground reaction force affected by the lift force L and the gravity:

$$F_B = \mu \left(\frac{1}{2} \rho V_A^2 S_{ref} C_L - mg_0 \cos \theta_{rwy} \right). \quad (2.15)$$

When the braking system is off, the μ is defined as the roll friction index. During the deceleration phase, the Autobrake system is usually applied in most B747-8F flights, in which the deceleration target value is commanded. The μ is controlled to maintain the deceleration target value via the braking pressure. In the case of manual braking, the equivalent acceleration $a_{x, equ}$ is calculated and approximated as the deceleration target value. In the maximum braking phase, the maximum μ is applied based on the runway conditions. In addition to that, the coefficients C_L and C_D are affected by the time of deploying spoilers t_{splr} .

Furthermore, the initial ground speed is influenced by the wind speed V_W and the true airspeed V_A . By integrating Equation (2.11) twice, d_{80} and d_{stop} are obtained. Then, SM can be computed based on Equation (2.9) and Equation (2.10).

In summary, all contributing factors mentioned above are mapped to the SM by the physical-based runway overrun model. Measurements of all contributing factors can be identified from the operational flight data as discussed in Section 2.3.4. With enough flights, the distribution of contributing factors can be estimated, and the detailed results are shown in Table 4.3 of Chapter 4. By combing the built runway overrun model and the obtained distribution of contributing factors, the runway overrun probability can be predicted using advanced statistical methods. In addition, the sensitivity of all contributing factors to the runway overrun can be also quantified.

Chapter 3

Mathematical Preliminaries

3.1 Introduction

As mentioned in Section 1.2.2 and Section 2.5, the incident metric in the Predictive Analysis is defined as a continuous variable. If the incident metric is less than the safety threshold, the incident occurs. According to the aircraft motion, pilot behaviors, and operational procedures, the incident model is built to describe the relationship between the contributing factors and the incident metric. It can be expressed using the mathematical equation: $Y = g(\mathbf{X})$, where $g(\cdot)$ denotes the incident model. The model input \mathbf{X} denotes the contributing factors to the incident. Y is the predefined incident metric. The uncertainties of the contributing factors are obtained from the operational flight data, which can be quantified in terms of a probabilistic manner. Section 3.2 provides the input uncertainty quantification methods. Afterward, the incident probability will be estimated via propagating the uncertainties in the model input to the model output. An overview of the mathematical theory in rare event probability estimation is described in Section 3.3. In order to identify the key factors of the incident and provide mitigation strategies for the incident, the global sensitivity analysis method is provided in Section 3.4. Furthermore, a surrogate model based on the polynomial chaos expansion is introduced to speed up the uncertainty propagation in Section 3.5.

3.2 Input Uncertainty Quantification

3.2.1 Random Variables and Distributions

In daily flight operation, the observations of the flight data measurements vary a lot, for example, the ground speed at the touchdown point. They can be mathematically expressed by a random variable. A random variable $X(\omega)$ is defined as a function that

maps the sample space Ω of a random observation to the real numbers \mathbb{R} [85]. ω is one realization in Ω . The uncertainties of $X(\omega)$ can be described using the probability distribution, which is the probability of the random variable taking certain values. There are continuous and discrete variables. If $X : \Omega \rightarrow \mathbb{R}$ presents a discrete variable, its probability distribution is described by the probability mass function $p_{Xd} : \mathbb{R} \rightarrow [0, 1]$:

$$Pr(X = x_j) = p_{Xd}(x_j) = Pr(\omega \in \Omega : X(\omega) = x_j). \quad (3.1)$$

If $X : \Omega \rightarrow \mathbb{R}$ is a continuous random variable, the probability distribution is described by its probability density function (PDF) $p_X(x) : \mathbb{R} \rightarrow [0, \infty)$:

$$Pr(a \leq X \leq b) = \int_a^b p_X(x) dx. \quad (3.2)$$

Furthermore, the cumulative distribution function (CDF) describes the probability of a random variable X less than or equal to a certain value x for a given probability distribution. The CDF of a discrete random variable is:

$$F_X(x) = \sum_{x_j \leq x} p_{Xd}(x_j). \quad (3.3)$$

For a continuous random variable, the CDF is defined as below:

$$F_X(x) = \int_{-\infty}^x p_X(t) dt. \quad (3.4)$$

There are many distribution families for the discrete and the continuous probability distributions. The best-fitting distribution family and parameters can be estimated based on the observations using the maximum likelihood method. We use the $f_i(x_i)$ and $F_{X_i}(x_i)$ to denote the marginal PDF and CDF with respect to X_i . When d -dimensional random variables are independent distributed, the joint probability density function $f_{\mathbf{X}}(\mathbf{x})$ can be expressed as:

$$f_{\mathbf{X}}(\mathbf{x}) = \prod_{i=1}^d f_i(x_i). \quad (3.5)$$

3.2.2 High Dimensional Dependence Modeling

A joint probability distribution density $f_{\mathbf{X}}(\mathbf{x})$ for dependent random variables does not equal to the product of marginal PDFs. In practice, the multivariate Gaussian distribution with a correlation matrix is widely used. However, they cannot capture the tail dependence. In addition, the joint probability distribution is difficult to express analytically when the marginal PDFs are arbitrary and different to each other. A vine copula approach is proposed in [27, 28] to model the high dimensional dependence. According to the Sklar theorem [86], the d -dimensional joint probability distribution function $F_{\mathbf{X}}(\mathbf{x})$ can be decomposed into a copula distribution function C and d marginal CDFs $F_{X_i}(x_i)$ as below:

$$F(x_1, x_2, \dots, x_d) = C(F_{X_1}(x_1), F_{X_2}(x_2), \dots, F_{X_d}(x_d)). \quad (3.6)$$

Since $F_{X_i}(x_i)$ is the marginal CDF corresponding to the random variable X_i , the observation of $F_{X_i}(x_i)$ should follow the uniform distribution. C describes the dependence structure among the d random variables. The corresponding joint distribution density function is:

$$\begin{aligned} f(x_1, x_2, \dots, x_d) &= \frac{\partial^d F(x_1, x_2, \dots, x_d)}{\partial x_1 \partial x_2 \dots \partial x_d} \\ &= c\left(F_{X_1}(x_1), F_{X_2}(x_2), \dots, F_{X_d}(x_d)\right) \prod_{i=1}^d f_{X_i}(x_i), \end{aligned} \quad (3.7)$$

where c is called copula density function. Compared to the joint probability density function, the c eliminates the impact of marginal distributions. For independent variables, c equals 1 and Equation (3.7) becomes Equation (3.5). In case that all inverse marginal CDFs exist, C can be easily constructed using the joint CDF F and inverse marginal CDFs $F_{X_i}^{-1}(u_i)$ based on Equation (3.6) as below:

$$C(\mathbf{u}) = C(u_1, u_2, \dots, u_d) = F(F_{X_1}^{-1}(u_1), F_{X_2}^{-1}(u_2), \dots, F_{X_d}^{-1}(u_d)). \quad (3.8)$$

Copula Data in Three Scales

In the study of the copula, there are three scales for the random variable [28]: X -scale, U -scale, and Z -scale. The original random variable X_i denotes the data in the X -scale. The pair plots in the X -scale represent varying in the joint probability density of the corresponding two variables, which mixes the dependence and marginal effects. The probability integral transform of the original data based on the marginal CDFs are the copula scale variables, denoted by U as below:

$$U_i = F_{X_i}(X_i), \quad (3.9)$$

where U_i follows the standard uniform distribution $\mathcal{U}(0, 1)$. Furthermore, the normalized scale of variables, Z -scale, is obtained by transforming the U_i into the standard normal space as below:

$$Z_i = \psi^{-1}(U_i), \quad (3.10)$$

where $\psi^{-1}(\cdot)$ is the inverse CDF of the standard normal distribution. Z_i follows the standard normal distribution $\mathcal{N}(0, 1)$. The behavior of dependence is obvious in the Z scale pair plot, which can also be compared with the known bivariate normal variables. A case of pair sample plots in three scales is illustrated in Figure 3.1. X_1 obeys the exponential distribution with the mean of 0.5, and X_2 follows the standard normal distribution. Two variables follow the Gaussian copula with a correlation of 0.7. Small letter x_i , u_i , and z_i , denote the realization of random variable in the X , U , and Z scale, respectively. In the X -scale sample plot, one mostly observes the features of marginal distributions and can not identify the underlying dependence. The feature of dependence in U -scale and Z -scale is more obvious. Furthermore, one can directly know that the two variables are mostly linear correlated from the Z -scale sample plot.

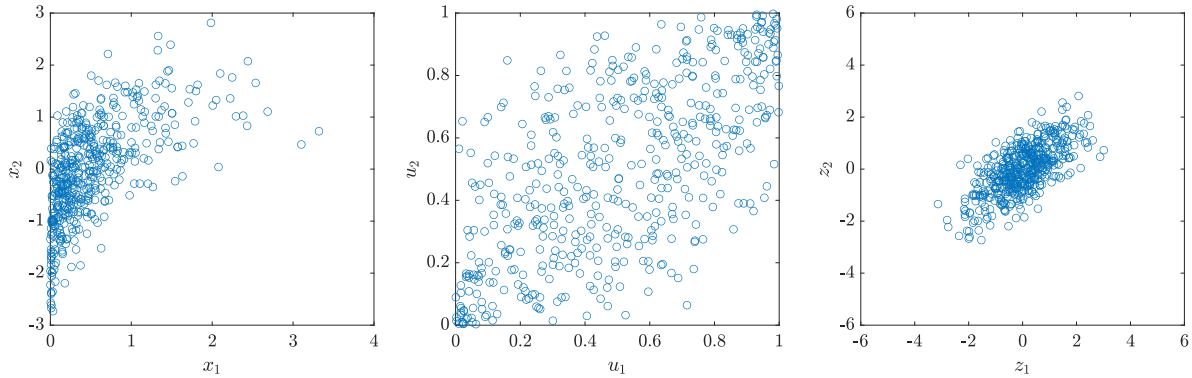


Figure 3.1: *Pair samples plot in three scales, source: Fig 1.7 of [28]*

Bivariate Copula

As an example, a bivariate Gaussian copula is constructed using a two-dimensional joint Gaussian distribution. Assume that the two variables X_1 and X_2 follow the standard Gaussian distribution. The corresponding PDF ϕ and CDF Φ are as follows:

$$\phi(x) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{x^2}{2}\right); \quad \Phi(a) = \int_{-\infty}^a \phi(x) dx. \quad (3.11)$$

The joint probability distribution of X_1 and X_2 follow the multivariate Gaussian distribution with the correlation parameter ρ and the mean vector $(0, 0)$. The joint PDF ϕ_ρ and CDF Φ_ρ are as below:

$$\begin{aligned} \phi_\rho(x_1, x_2) &= \frac{1}{2\pi\sqrt{1-\rho^2}} \exp\left(-\frac{x_1^2 - 2\rho x_1 x_2 + x_2^2}{2(1-\rho^2)}\right); \\ \Phi_\rho(a, b) &= \int_{-\infty}^a \int_{-\infty}^b \phi_\rho(x_1, x_2) dx_2 dx_1. \end{aligned} \quad (3.12)$$

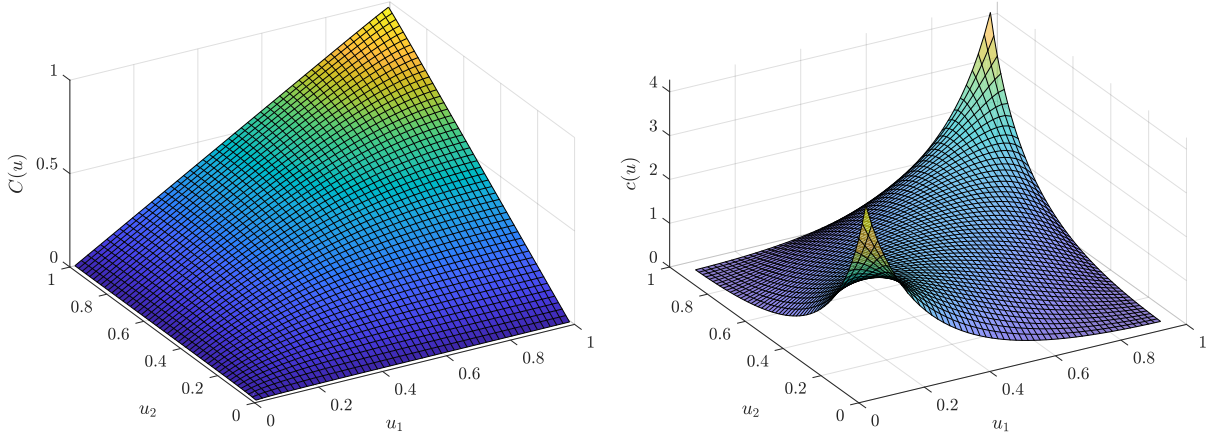
Therefore, the bivariate Gaussian copula is derived based on Equation (3.8) as below:

$$C(\mathbf{u}) = C(u_1, u_2) = \Phi_\rho(\Phi^{-1}(u_1), \Phi^{-1}(u_2)). \quad (3.13)$$

Based on Equation (3.7), Equation (3.11), and Equation (3.12), the bivariate Gaussian copula density is derived as below:

$$\begin{aligned} c(u_1, u_2) &= \frac{\phi_\rho(x_1, x_2)}{\phi(x_1)\phi(x_2)} \\ &= \frac{1}{\sqrt{1-\rho^2}} \exp\left(\frac{2\rho\Phi^{-1}(u_1)\Phi^{-1}(u_2) - \rho^2(\Phi^{-1}(u_1))^2 + \Phi^{-1}(u_2)^2}{2(1-\rho^2)}\right). \end{aligned} \quad (3.14)$$

Figure 3.2 shows the CDF and PDF of the bivariate Gaussian copula. By using the same multivariate Gaussian distribution with $\rho = 0.7$, the relationship between joint density, copula density, and marginal PDFs in Equation (3.7) is illustrated in Figure 3.3. One realization of two variables: $(x_1 = -0.1, x_2 = -0.7)$ is also denoted by the red point in Figure 3.3. It is easy to see the positive correlation between two variables from the feature of the copula density, compared to the joint probability density.



(a) The CDF of the bivariate Gaussian copula. (b) The PDF of the bivariate Gaussian copula.

Figure 3.2: the CDF and PDF of the bivariate Gaussian copula with the correlation coefficient $\rho = 0.7$.

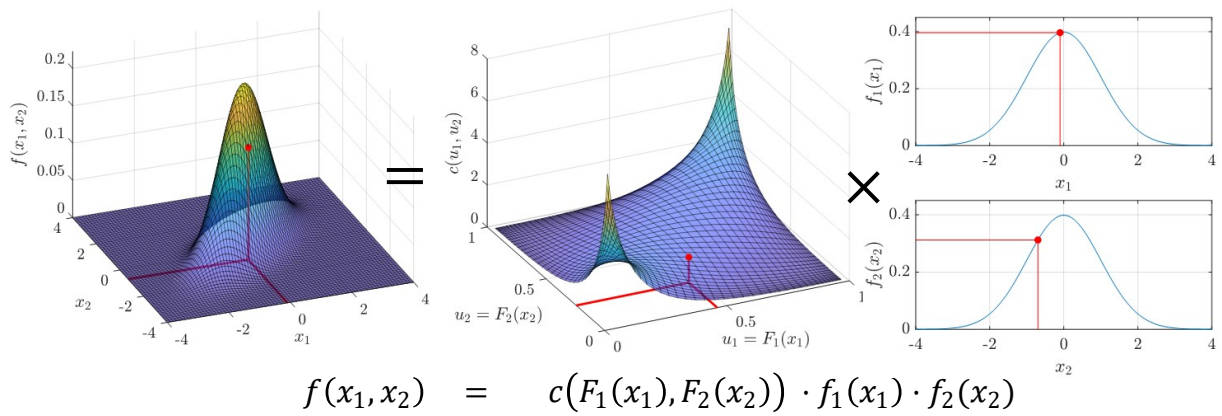
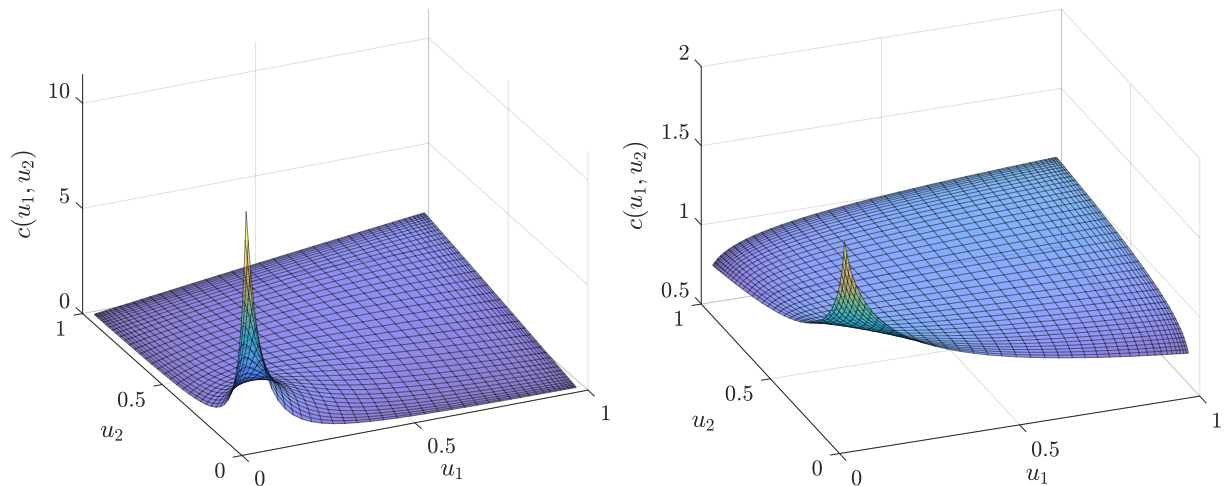


Figure 3.3: Illustration of Sklar theorem, modified from [82]

Besides the Gaussian copula, there are many different bivariate copula distribution families, such as, elliptical copula families, Archimedean copula families, and extreme value copula families [28]. Table B.2 in Appendix B.2 lists all available parametric bivariate copula families in VineCopula R package [87]. As one of parametric bivariate Archimedean copulas with a single parameter, the bivariate Clayton copula is given as an example. The bivariate Clayton copula density function is shown as below [28]:

$$c(u_1, u_2) = \left(u_1^{-\delta} + u_2^{-\delta} - 1\right)^{-\frac{1}{\delta}}, \tag{3.15}$$

where $\delta \in (0, \infty)$. The degree of dependence is controlled by δ . $\delta \rightarrow 0$ corresponds to independence of two variables. The higher the δ is, the stronger the dependence is. Figure 3.4 shows the Clayton copula density with $\delta = 1$ and $\delta = 0.1$, respectively.



(a) The Clayton copula density with $\delta = 1$. (b) The Clayton copula density with $\delta = 0.1$.

Figure 3.4: The Clayton copula density.

Vine Copula

For high dimensional probability, the vine copula approach allows us to construct the high-dimensional density function c using $\frac{d(d-1)}{2}$ bivariate copula density functions by the conditional probability and the Rosenblatt transformation [88]. Therefore, a combination of a vine copula structure and several bivariate copulas can be used to describe the high dimensional dependence. As a demonstration, a joint distribution with three random variables is constructed using three bivariate copula density functions with a vine copula structure. To simplify the notation, the marginal density function f_{X_i} and the marginal distribution function F_{X_i} are replaced by f_i and F_i , respectively.

For two variables X_1 and X_2 , which follow the joint probability density $f(x_1, x_2)$, the conditional probability density $f_{2|1}$ is defined by

$$f_{2|1}(x_2|x_1) = \frac{f(x_1, x_2)}{f_1(x_1)}, \quad (3.16)$$

where $f_1(x_1) > 0$. $f_{2|1}(x_2|x_1)$ describes the distribution of X_2 given that $X_1 = x_1$. By multiplying $f_1(x_1)$ in Equation (3.16), the joint probability density can be rewritten as below:

$$f(x_1, x_2) = f_{2|1}(x_2|x_1)f_1(x_1). \quad (3.17)$$

Based on the definition of the conditional probability density, the joint probability density f of three random variables X_1, X_2, X_3 can be derived as below:

$$f(x_1, x_2, x_3) = f_{3|12}(x_3|x_1, x_2)f_{2|1}(x_2|x_1)f_1(x_1). \quad (3.18)$$

According to the Equation (3.7), joint density functions $f_{2|1}(x_2|x_1)$ and $f_{3|12}(x_3|x_1, x_2)$ can be separated into copula densities and marginal densities:

$$\begin{aligned} f_{2|1}(x_2|x_1) &= \frac{f_{12}(x_1, x_2)}{f_1(x_1)} = c_{12}(F_1(x_1), F_2(x_2))f_2(x_2), \\ f_{3|12}(x_3|x_1, x_2) &= c_{13;2}(F_{1|2}(x_1|x_2), F_{3|2}(x_3|x_2))f_{3|2}(x_3|x_2) \\ &= c_{13;2}(F_{1|2}(x_1|x_2), F_{3|2}(x_3|x_2))c_{23}(F_2(x_2), F_3(x_3))f_3(x_3), \end{aligned} \quad (3.19)$$

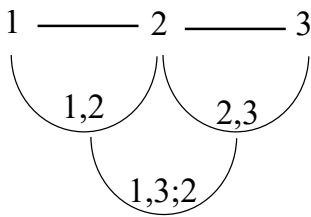
where $c_{ij;k}$ is the conditional copula density between two random variables X_i and X_j for a given X_k . After substitution of $f_{2|1}(x_2|x_1)$ and $f_{3|12}(x_3|x_1, x_2)$ in Equation (3.18), the joint probability density f is represented as the product of three bivariate copula density functions and three marginal probability density functions.

$$\begin{aligned} f(x_1, x_2, x_3) &= c_{13;2}(F_{1|2}(x_1|x_2), F_{3|2}(x_3|x_2)) \\ &\quad \times c_{23}(F_2(x_2), F_3(x_3)) \times c_{12}(F_1(x_1), F_2(x_2)) \\ &\quad \times f_1(x_1) \times f_2(x_2) \times f_3(x_3), \end{aligned} \quad (3.20)$$

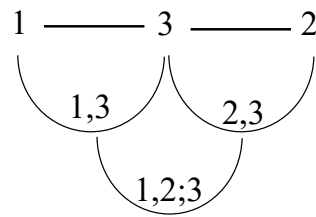
where $F_{1|2}(x_1|x_2)$ and $F_{3|2}(x_3|x_2)$ are conditional distributions. They are computed based on the partial derivative of the copula function [89]:

$$\begin{aligned} F_{i|j}(x_i|x_j) &= \int_{-\infty}^{x_i} f_{i|j}(x_i|x_j)dx_i = \int_{-\infty}^{x_i} \frac{f_{ij}(x_i, x_j)}{f_j(x_j)}dx_i = \int_{-\infty}^{x_i} f_{ij}(x_i, x_j)dx_i \cdot \frac{1}{f_j(x_j)} \\ &= \frac{\partial F_{ij}(x_i, x_j)}{\partial x_j} \cdot \frac{1}{f_j(x_j)} = \frac{\partial F_{ij}(x_i, x_j)}{\partial u_j} \cdot \frac{\partial u_j}{\partial x_j} \cdot \frac{1}{f_j(x_j)} \\ &= \frac{\partial C_{ij}(u_i, u_j)}{\partial u_j} = C_{i|j}(u_i|u_j). \end{aligned} \quad (3.21)$$

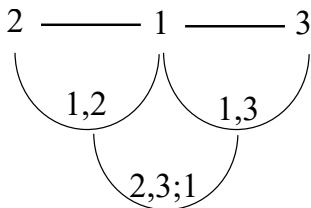
The graphical structure of vine copula is introduced in [90] to help organize the vine structure. Figure 3.5(a) illustrates the vine copula structure of Equation (3.20).



(a) The vine structure in Equation (3.20).



(b) The vine structure in Equation (3.22).



(c) The vine structure in Equation (3.23).

Figure 3.5: The graphical structure of vine copula in three dimensions.

It needs to be mentioned that the construction in Equation (3.20) is not unique. $f(x_1, x_2, x_3)$ can also be constructed in another two ways shown in Equation (3.22) and Equation (3.23). The corresponding vine structures are illustrated in Figure 3.5.

$$\begin{aligned} f(x_1, x_2, x_3) &= c_{12;3}(F_{1|3}(x_1|x_3), F_{2|3}(x_2|x_3)) \\ &\quad \times c_{13}(F_1(x_1), F_3(x_3)) \times c_{23}(F_2(x_2), F_3(x_3)) \\ &\quad \times f_1(x_1) \times f_2(x_2) \times f_3(x_3). \end{aligned} \quad (3.22)$$

$$\begin{aligned} f(x_1, x_2, x_3) &= c_{23;1}(F_{2|1}(x_2|x_1), F_{3|1}(x_3|x_1)) \\ &\quad \times c_{12}(F_1(x_1), F_2(x_2)) \times c_{13}(F_1(x_1), F_3(x_3)) \\ &\quad \times f_1(x_1) \times f_2(x_2) \times f_3(x_3). \end{aligned} \quad (3.23)$$

The three-dimensional construction of vine copula is simple and three construction structure are available. However, the construction structure becomes complex for high dimensional cases, and there are many construction structure candidates. To deal with the high dimensional pair copula construction, the paper [91] proposed a sequential strategy to obtain the most suitable construction structure. Currently, there are C-vine, D-vine, and R-vine construction structures available in the R package ‘rvinetool’ [92].

Copula Estimation

To study the dependence among several random variables using a copula approach, the appropriate marginal distributions should be estimated first. Table B.1 lists well-known one-dimensional distribution families. Based on the maximum likelihood principle, the best-fitting distribution family and the unknown parameters θ are estimated for the given observations of the random variables. As shown in Figure 2.12, the best-fitted distribution of the touchdown distance is obtained. The CDF and PDF of the i_{th} variable are denoted by $F_i(x_i)$ and $f_i(x_i)$ as before. The unknown parameters in the CDF or PDF for the i_{th} variable are denoted by θ_i .

Bivariate Copula Estimation For the two-dimensional dependence, the best-fitting copula and corresponding parameters can be estimated subsequently after the marginal distributions are obtained. Unknown parameters in the copula family are denoted by δ . Observations are the measurements of the two random variables, denoted by $(x_{1,j}, x_{2,j}), j = 1, 2, \dots, n$. n is the number of the measurements. The estimation procedure is shown as below:

1. The so-called pseudo copula data $(\hat{u}_{1,j}, \hat{u}_{2,j})$ are obtained via:

$$(\hat{u}_{1,j}, \hat{u}_{2,j}) = (F_1(x_{1,j}), F_2(x_{2,j})), \quad j = 1, 2, \dots, n. \quad (3.24)$$

2. For a chosen bivariate copula candidate c , the corresponding unknown parameters δ in c are estimated by using the maximum likelihood method. The maximum likelihood estimator δ_{ML} maximizes

$$L(\delta|\mathbf{u}) = \sum_{j=1}^n \log c(\hat{u}_{1,j}, \hat{u}_{2,j}; \delta), \quad j = 1, 2, \dots, n. \quad (3.25)$$

3. Different bivariate copula families in Table B.2 are used in the step 2, while the independent cumulative distribution function mentioned in Table B.3 are also taken as a candidate to distinguish if variables are independent [28]. The best-fitting bivariate copula or the independent case are selected based on the criteria such as Akaike Information Criterion (AIC) and Bayesian Information Criterion (BIC) in which the number of unknown parameters are taken into account besides the maximum likelihood [27].

Vine Copula Estimation When it comes to the high dimensional dependence, the vine copula structure is chosen as mentioned in Section 3.2.2. The three dimensional joint distribution is used as an example to describe the estimation procedures. The vine copula structure of three variables in Equation (3.20) is used. Similar to the estimation procedure of bivariate copula, the pseudo copula data $(\hat{u}_{1,j}, \hat{u}_{2,j}, \hat{u}_{3,j})$ are obtained first. Unknown parameters are denoted by $\boldsymbol{\delta} = (\delta_{12}, \delta_{23}, \delta_{13;2})$, which corresponds to the copula density c_{12} , c_{23} , and $c_{13;2}$ in Equation (3.20), respectively. The unconditioned bivariate copula density c_{12} and c_{23} can be estimated based on the procedure above from the pseudo copula data $(\hat{u}_{1,j}, \hat{u}_{2,j})$ and $(\hat{u}_{2,j}, \hat{u}_{3,j})$. The challenge part is to estimate the conditional bivariate copula $c_{13;2}$. By using the estimated δ_{12} and δ_{23} , the pseudo observations for conditional copula $c_{13;2}$ are computed via:

$$\begin{aligned} \hat{v}_{1|2,j} &= F_{1|2}(x_{1,j}|x_{2,j}, \hat{\delta}_{12}) = C_{1|2}(\hat{u}_{1,j}|\hat{u}_{2,j}, \hat{\delta}_{12}); \\ \hat{v}_{3|2,j} &= F_{3|2}(x_{3,j}|x_{2,j}, \hat{\delta}_{23}) = C_{3|2}(\hat{u}_{3,j}|\hat{u}_{2,j}, \hat{\delta}_{23}), \end{aligned} \quad (3.26)$$

where $C_{1|2}$ and $C_{3|2}$ are computed based on the derivative of copula distribution with respect to the conditioned variable as below:

$$C_{i|j}(u_i|u_j) = \frac{\partial C_{ij}(u_i, u_j)}{\partial u_j} = \frac{\partial}{\partial u_j} \int_0^{u_i} \int_0^{u_j} c_{ij}(u_i, u_j) du_i du_j = \int_0^{u_i} c_{ij}(u_i, u_j) du_i. \quad (3.27)$$

According to the simplified assumption [28], $c_{13;2}(u_1, u_3; u_2) = c_{13;2}(u_1, u_3)$ holds for all conditioning value u_2 . The parameter $\delta_{13;2}$ can be estimated based on the pair data $(\hat{v}_{1|2,j}, \hat{v}_{2|3,j})$. The sequential and maximum likelihood estimation method is applied. The sequential estimation is shown as below:

1. Estimate δ_{12} from $(\hat{u}_{1,j}, \hat{u}_{2,j}), j = 1, 2, \dots, n$ according to the estimation procedure of bivariate copula mentioned above;

2. Estimate δ_{23} from $(\hat{u}_{2,j}, \hat{u}_{3,j}), j = 1, 2, \dots, n$ according to the estimation procedure of bivariate copula mentioned above;
3. Calculate pseudo observations $(\hat{v}_{1|2,j}, \hat{v}_{3|2,j})$ using Equation (3.26) and Equation (3.27);
4. Estimate $\delta_{13;2}$ from $(\hat{v}_{1|2,j}, \hat{v}_{3|2,j}), j = 1, 2, \dots, n$ according to the estimation procedure of bivariate copula mentioned above.

3.2.3 Sampling Methods

As mentioned above, the input uncertainties are quantified based on the observation data via statistical modeling, such as probability distribution. And they will be further propagated through the model. Before evaluating the model, the necessary step is to generate realizations from the estimated probability distribution. Three sampling techniques for the univariate case are mentioned at first, followed by the copula sampling method with respect to the multivariate case.

Inverse Transformation Method

The inverse transformation method is commonly used if the CDF of the random variable is analytically invertible. In this case, the direct sampling method is to transfer the sample from the uniform space to the original X space by the inverse CDF of x :

$$x = F^{-1}(u), \quad (3.28)$$

where u is the generated pseudo random number uniformly distributed in $[0, 1]$. The illustration is shown in Figure 3.6.

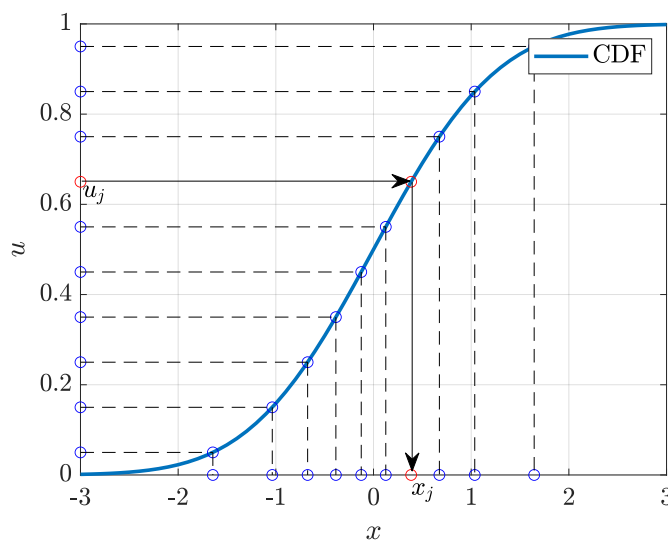


Figure 3.6: *Inverse transformation sampling*

Rejection Sampling

The CDF of x might be not analytically invertible, and the distribution may be only known point-wise. In that condition, the direct inverse transformation sampling does not work. One indirect sampling method is called rejection sampling. For a target distribution $f_X(x)$ that we will sample from, if we can find an arbitrary PDF $\pi(x)$ that satisfies $k\pi(x) \geq f_X(x)$ for all x with $k \geq 1$, the rejection sampling can be applied to generate samples following $f_X(x)$. The $\pi(x)$ is called a proposal PDF to generate candidates at first. The candidates are accepted with a probability of $f_X(x)/k\pi(x)$. Finally, the accepted samples follow the target PDF $f_X(x)$. Algorithm 1 describes the detailed procedure.

Algorithm 1 Rejection sampling's algorithm

- 1: Choose a proposal PDF $\pi(x)$
 - 2: Find the minimum constant k that satisfies $k\pi(x) \geq f_X(x)$ for all x with $k \geq 1$ (see Equation (3.29))
 - 3: Set $j = 1$, and the required number of samples N
 - 4: Generate a sample candidate v from $\pi(x)$ using the inverse CDF transformation mentioned in Section 3.2.3
 - 5: Generate a number u from $\mathcal{U}(0, 1)$
 - 6: **if** $u \leq f_X(v)/k\pi(v)$ **then**
 - 7: $x_j = v$
 - 8: $j = j + 1$
 - 9: **else**
 - 10: Reject v
 - 11: **end if**
 - 12: Repeat step 4 to step 11 until $j \geq N$
-

The constant k mentioned in Algorithm 1 can be calculated by letting k equals the maximum ratio of $f_X(x)$ to $\pi(x)$ as below:

$$k = \max \left(\frac{f_X(x)}{\pi(x)} \right), \text{ for } \pi(x) > 0. \quad (3.29)$$

An example of sampling from a half-normal standard distribution is given to demonstrate the rejection sampling procedure. The PDF of the half normal distribution is shown as below:

$$f_X(x) = \sqrt{\frac{\sigma}{2}} \exp\left(-\frac{x^2}{2}\right), \quad x \geq 0. \quad (3.30)$$

An exponential distribution is chosen as the proposal PDF as below:

$$\pi(x) = \exp(-x). \quad (3.31)$$

To calculate the minimum k to satisfy $k\pi(x) \geq f_X(x)$, the maximum $f_X(x)/\pi(x)$ is calculated. By letting

$$\left(\frac{f_X(x)}{\pi(x)}\right)' = \left(\sqrt{\frac{\sigma}{2}} \exp\left(-\frac{x^2}{2} + x\right)\right)' = 0, \quad (3.32)$$

$x = 1$ is obtained. Therefore,

$$k = \max\left(\frac{f_X(x)}{\pi(x)}\right) = \frac{f_X(1)}{\pi(1)} = 1.315. \quad (3.33)$$

Afterward, samples are generated based on the step 3 to 12 in Algorithm 1. An illustration of the rejected and accepted candidates is shown in Figure 3.7.

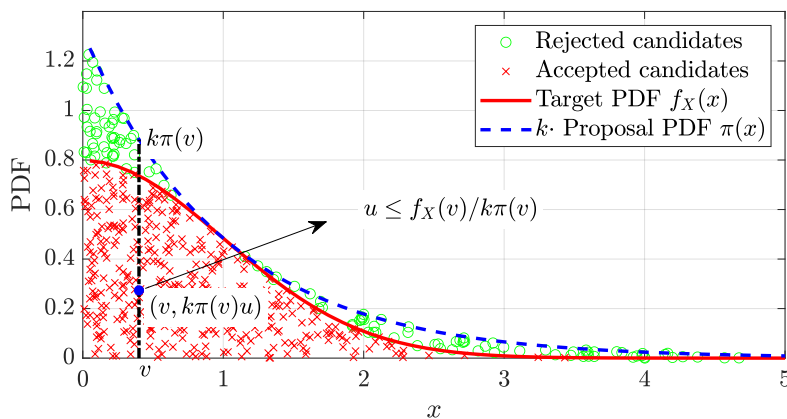


Figure 3.7: *Rejection sampling*

The coordinate of the candidates is $(v, k\pi(v)u)$. The acceptable candidates are located inside the area covered by the target PDF uniformly. A proof that the accepted candidates follow the target distribution is given as follows. Assume that u and v follow the joint PDF $f_{uv}(v, u)$. v follows the proposal PDF $\pi(v)$, and u is uniformly distributed inside the range $(0, k\pi(v))$. $f_{uv}(u, v)$ is computed as below:

$$f_{uv}(u, v) = f_v(v)f_{u|v}(u|v) = \pi(v) \cdot \frac{1}{k\pi(v)} = \frac{1}{k}, \quad 0 \leq u \leq k\pi(v). \quad (3.34)$$

Therefore, the (u, v) follows the uniform distribution in the area: $A = \{(u, v), 0 \leq u \leq k\pi(v)\}$. Following the procedure in Algorithm 1, the acceptable area follows the uniform distribution on $B = \{(u, x), 0 \leq u \leq f_X(x)\}$. The density function (u, x) is:

$$f_{ux}(u, x) = 1, \quad 0 \leq u \leq f_X(x). \quad (3.35)$$

The marginal distribution of $f_{ux}(u, x)$ related to x can be derived as below:

$$\int_0^{f_X(x)} f_{ux}(u, x) du = \int_0^{f_X(x)} 1 du = f_X(x). \quad (3.36)$$

Therefore, the accepted samples follow the target distribution $f_X(x)$.

Markov Chain Monte Carlo

In rejection sampling method, there is no sample generated when the sample candidate is rejected. Therefore, a more efficient sampling technique is widely used, called Markov Chain Monte Carlo (MCMC). Given a Markov chain whose stationary distribution is the target distribution $f_X(x)$, the generated states in the chain will follow $f_X(x)$, but they are not independent [30]. A well-known Metropolis-Hasting (MH) algorithm is described in Algorithm 2 to generate a required Markov chain [93].

Algorithm 2 Metropolis-Hastings algorithm

- 1: Given the current state x_c
 - 2: Generate a sample candidate v from a proposal PDF $q(v|x_c)$ (see Equation (3.37)) using the inverse CDF transformation
 - 3: Generate a sample u from $\mathcal{U}(0, 1)$:
 - 4: Compute the acceptance ratio: $\alpha(x_c, v) = \min\left\{1, \frac{f_X(v)q(x_c|v)}{f_X(x_c)q(v|x_c)}\right\}$
 - 5: **if** $u \leq \alpha(x_c, v)$ **then**
 - 6: $x_{c+1} = v$
 - 7: **else**
 - 8: $x_{c+1} = x_c$
 - 9: **end if**
 - 10: $c = c + 1$
 - 11: Repeat step 2 to step 5 to obtain a sequence of dependent random variables x_c .
-

When the generated candidate v is rejected, the previous state is used as the current state. After a burn-in period [93], the sequence of states in the Markov chain will approximately follow the target distribution $f_X(x)$. A random walk sampler is a simple example of MH-based MCMC sampling algorithms. The proposal PDF in the random walk sampler is:

$$q(v|x) = \frac{1}{\sigma} \psi\left(\frac{v - x_c}{\sigma}\right), \quad (3.37)$$

where x_c is the current state. ψ is the standard normal distribution. σ denotes the standard deviation. Therefore, the proposal state v of a random walk is given by:

$$v = x_c + \xi, \quad (3.38)$$

where the random variable ξ follows the normal distribution $\mathcal{N}(0, \sigma^2)$. As an example, the random walk sampler is used to generate samples from a half-normal standard distribution shown in Equation (3.30). After a burn-in period, the distribution of generated samples will converge to the stationary that samples follows the target distribution. In order to visualize the convergence of the algorithm, the histograms of generated samples using the random walk sampler are shown in Figure 3.8. Three different values of σ : 0.1, 1, and 10 are chosen as the standard deviation of the proposal distribution. Since the burn-in

period in MCMC algorithms exists, four different numbers of samples: 1000, 2000, 5000, and 10000 are chosen to draw here in each column, respectively. Figure 3.8 shows the histogram of 2000 samples in the case of $\sigma = 1$ has converged to the target distribution, while the histograms of samples, in the cases of $\sigma = 0.1$ and $\sigma = 10$, are close to the target distribution until the number of samples goes to 10000. It is intuitively clear that the σ of the proposal distribution influences the convergence speed of the MCMC algorithm.

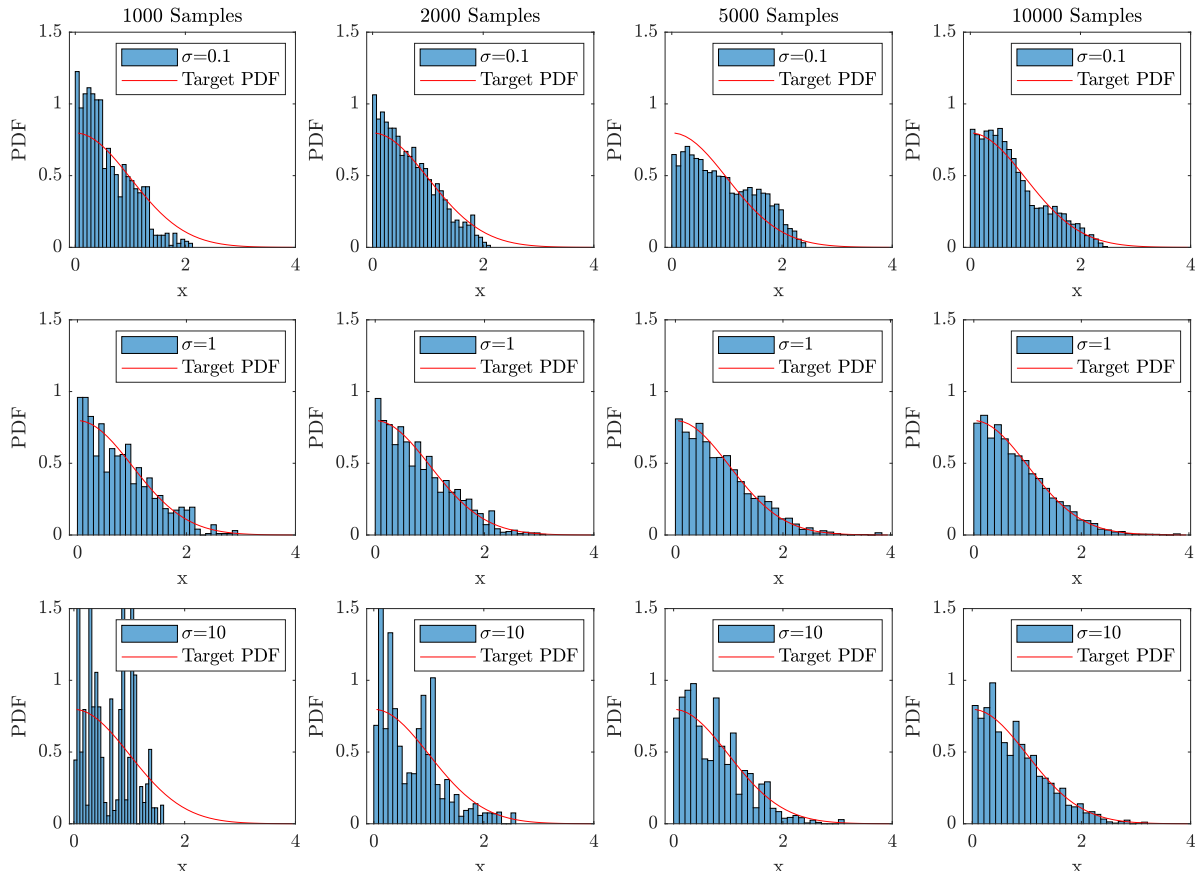


Figure 3.8: Histograms of samples using the MH random walk sampling

The convergence speed of the MCMC algorithm is influenced by the efficiency of the MCMC estimator described in Appendix C. As shown in Equation (C.12), the efficiency of the MCMC estimator depends on the autocorrelation coefficients $\rho(k)$ of samples in the Markov chain. The $\rho(k)$ are calculated as below based on Equation (C.6):

$$\rho(k) = \frac{s_k}{s_0} = \frac{\frac{1}{n} \sum_{i=1}^{N-k} (x_i - \bar{x})(x_{i+k} - \bar{x})}{\frac{1}{n-1} \sum_{i=1}^N (x_i - \bar{x})^2}, \quad (3.39)$$

where s_0 is the sample variance of states, s_k is the sample autocovariance, and \bar{x} is the sample mean of states in the Markov chain. The sample autocorrelations $\rho(k)$ calculated based on 1000 samples are shown in Figure 3.9. The sample autocorrelations in the case of $\sigma = 1$ are lower than the others. It also validates that the σ of the proposal distribution influences the autocorrelation of samples, and further influences the efficiency of the random walk sampler. The corresponding sample trace plots are also drawn in

Figure 3.9. The traces plots show that the MCMC sampler will propose small moves for the smaller σ . These proposed candidates ν have a high correlation with the previous states, and these candidates will be almost accepted since the acceptance ratio α is high due to the continuity of the target PDF $f_X(x)$ and the proposal PDF $q(v|x)$. However, this causes the algorithm to explore the entire state space very slowly and therefore takes a long time to converge to its target distribution. For the larger σ , the MCMC sampler will propose large moves to regions where the acceptance ratio α is small. Many proposed candidates are rejected and the previous states are used for large numbers of iterations. This also leads to high correlations among states in the chain. Hence, the larger σ also causes a slow convergence speed to the stationary distribution.

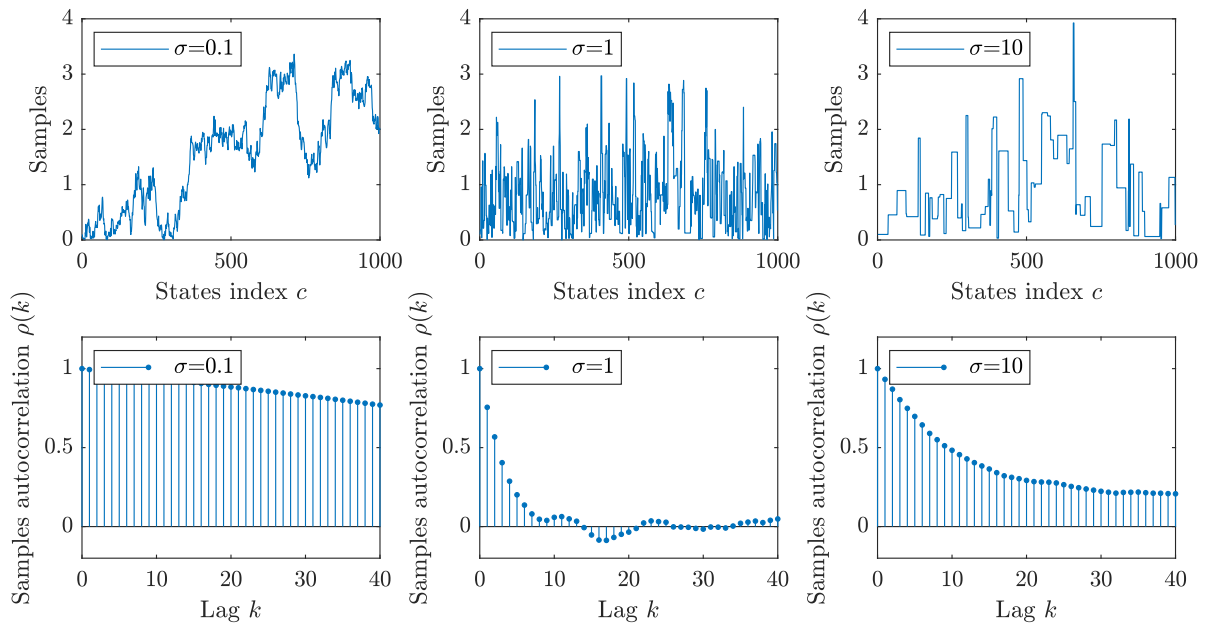


Figure 3.9: *Sample trace plots and sample autocorrelation plots*

Therefore, an optimal value of σ might exist where the MCMC sampler performs optimally. The solution is to select an optimal σ such that the corresponding acceptance ratio α is also optimal since that too small α indicates that there are many repeated samples in the generated Markov chain, while the too large α means that the sample trace of the chain moves very slowly. To visualize the optimal σ and the optimal α , the MCMC sampler is used to generate several chains with 10000 samples by using different proposed σ from 0.1 to 10 with the interval width of 0.1. The measure of MCMC estimator's efficiency eff mentioned in Appendix C is calculated based on Equation (C.12). Furthermore, the average acceptance ratio $\bar{\alpha}$ of samples is also calculated by counting the number of accepted proposal candidates and dividing the total number of samples. Figure 3.10 shows the relation of the MCMC efficiency with the proposal standard deviation and acceptance ratio. Results show that the highest efficiency of the MCMC algorithm is located around the $\sigma = 2$ and $\bar{\alpha} = 0.4$. Noticed that the proposal σ is used to generate samples X , but the autocorrelation $\rho_Y(k)$ is related to the output y_i directly not x_i as shown in

Figure C.1. Therefore, the optimal σ is affected by the specific function $Y = g(X)$ [94]. Instead of σ , researchers have proved that the existing optimal acceptance ratio for the MH-based random walk is 0.44 for one-dimensional cases and 0.23 for multidimensional cases [95, 96]. The optimal acceptance ratio is not influenced by the function $Y = g(X)$. In order to improve the efficiency of the MCMC algorithm, the strategy in practice is to obtain an optimal α of 0.44 for one dimension and 0.23 for multi-dimensions via increasing or decreasing the σ of the proposal distribution. This result will be used later in Section 3.3.3.

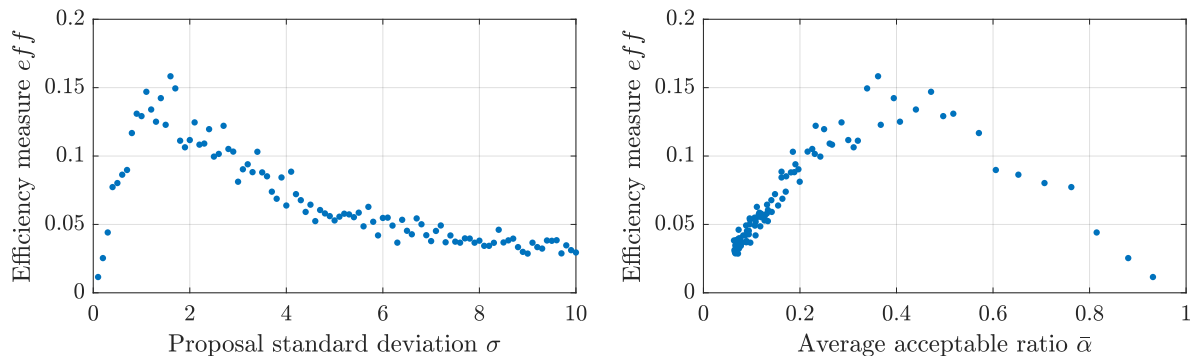


Figure 3.10: *The relation of the efficiency measure of the MCMC algorithm, the standard deviation of the proposal distribution, and the average acceptance ratio of sample candidates*

Copula Sampling

Rosenblatt proposed a transformation in [88] to generate dependent samples from a multivariate distribution via the inverse CDF of the conditional distributions. When the multivariate joint distribution is constructed using the vine copula approach, all bivariate copula functions are available. The dependent samples are generated sequentially using the inverse CDF of the conditional distributions obtained from the bivariate copula functions. The sequential generation procedure is shown in Algorithm 3.

The conditional distributions $C_{d|d-1, \dots, 1}$ are computed using the so-called h -functions and the recursive relationship [28]. For the unconditional bivariate copula function $C_{ij}(u_i, u_j)$, the h -function is defined to denote the conditional distribution $C_{i|j}$ in Equation (3.21):

$$h_{i|j}(u_i|u_j) = C_{i|j}(u_i|u_j) = \frac{\partial C_{ij}(u_i, u_j)}{\partial u_j}. \quad (3.40)$$

Similarly, $C_{j|i}$ is denoted by $h_{j|i}(u_j|u_i)$. Notice that the h functions are directly obtained by calculating the partial derivative of the copula functions, which are known in the vine copula.

Algorithm 3 Generate dependent samples from a vine copula, source: Theorem 6.2 of [28]

- 1: Generate n independent samples $w_{i,j}$ from the $\mathcal{U}[0, 1]$ for each dimension i , respectively. $i = 1, 2, \dots, d, j = 1, 2, \dots, n$;
 - 2: $u_{1,j} = w_{1,j}$;
 - 3: $u_{2,j} = C_{2|1}^{-1}(w_{2,j}|u_{1,j})$;
 - 4: ...
 - 5: $u_{d,j} = C_{d|d-1,\dots,1}^{-1}(w_{d,j}|u_{d,j}, \dots, u_{1,j})$;
 - 6: Collect the dependent samples in U -scale: $\{(u_{1,j}, u_{2,j}, \dots, u_{1,d}), j = 1, 2, \dots, n\}$;
 - 7: Transfer $u_{i,j}$ to $x_{i,j}$ via the inverse marginal CDFs: $x_{i,j} = F_i^{-1}(u_{i,j})$;
 - 8: Collect the dependent samples in X -scale: $\{(x_{1,j}, x_{2,j}, \dots, x_{1,d}), j = 1, 2, \dots, n\}$;
-

For the conditioned copula functions in the vine copula, let $C_{e_a, e_b; D_e}(w_1, w_2)$ denotes the conditioned bivariate copula that describes the dependence between e_a and e_b under the conditioning variables D_e . Then the conditional probability of e_a given e_b under the conditioning variables D_e is described using the notation of h -function as below:

$$h_{e_a|e_b; D_e}(w_1|w_2) = \frac{\partial C_{e_a, e_b; D_e}(w_1, w_2)}{\partial w_2}, \quad (3.41)$$

while the conditional probability of e_b given e_a under the conditioning variables D_e is denoted by $h_{e_b|e_a; D_e}(w_2|w_1)$. All h functions are one-dimensional CDF, and h^{-1} denotes the inverse CDF of h . An example of three-dimensional vine copula is used to illustrate the calculation of $C_{d|d-1,\dots,1}$ in the procedure of Algorithm 3. A structure of 3-dimensional vine copula with the construction functions are redrawn in Figure 3.11 based on Figure 3.5 and Equation (3.23) mentioned before. The input variables in X scale are denoted by (x_1, x_2, x_3) . (u_1, u_2, u_3) is the corresponding variables in U scale. $v_{2|1}$ and $v_{3|1}$ are pseudo observations mentioned in Equation (3.26).

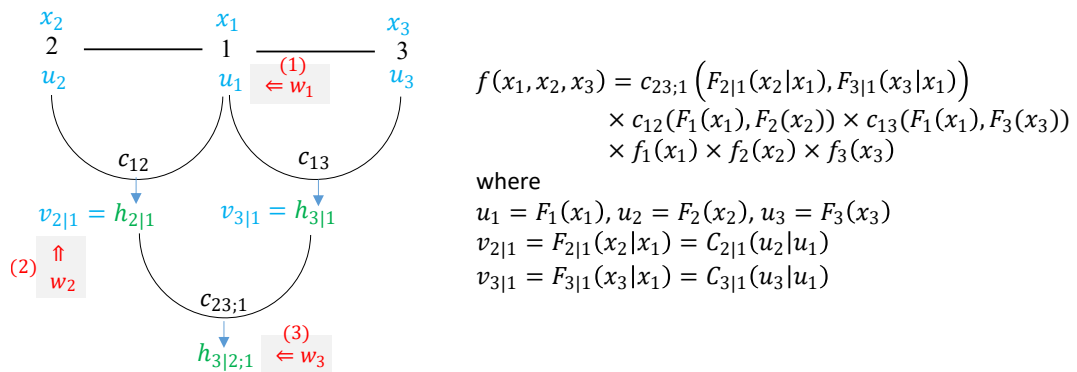


Figure 3.11: A structure of 3-dimensional vine copula with the construction functions

Based on Algorithm 3, the sequential procedure of generating three dependent samples is implementation. w_i are first generated from the $\mathcal{U}[0, 1]$ in the step 1 and will be assigned to different variables in the following steps. The sequential assignment of w_i is also illustrated in Figure 3.11 marked in red color. In the step 2, Let $u_1 = w_1$. In the step 3,

the conditional copula distribution $C_{2|1}$, denoted using $h_{2|1}$, is the one-dimensional CDF of u_2 given $U_1 = u_1$, which is computed from copula function C_{12} based on Equation (3.40). Let $w_2 = h_{2|1}(u_2|u_1)$. Therefore,

$$u_2 = h_{2|1}^{-1}(w_2|u_1) = h_{2|1}^{-1}(w_2|w_1). \quad (3.42)$$

Since the $v_{2|1} = C_{2|1}(u_2|u_1)$, then $v_{2|1} = w_2$. Based on the copula function $C_{23;1}(v_{2|1}, v_{3|1})$, the $h_{3|2;1}(v_{3|1}|v_{2|1})$ is obtained based on Equation (3.41). Let $w_3 = h_{3|2;1}(v_{3|1}|v_{2|1})$, then

$$v_{3|1} = h_{3|2;1}^{-1}(w_3|v_{2|1}) = h_{3|2;1}^{-1}(w_3|w_2). \quad (3.43)$$

Due to the fact that $v_{3|1} = C_{3|1}(u_3|u_1) = h_{3|1}(u_3|u_1)$, then

$$u_3 = h_{3|1}^{-1}(v_{3|1}|u_1) = h_{3|1}^{-1}(h_{3|2;1}^{-1}(w_3|w_2)|w_1). \quad (3.44)$$

In summary, the conditional distributions $C_{2|1}^{-1}$ and $C_{3|1,2}^{-1}$ are computed via h functions in Equation (3.42) and Equation (3.44). After the dependent u_1 , u_2 and u_3 are obtained, x_1 , x_2 and x_3 are obtained via the corresponding inverse marginal CDFs, separately.

3.3 Rare Event Probability Estimation

The incident probability (quantity of interest) can be defined as follows:

$$P_F = Pr(g(\mathbf{x}) < \gamma) = \int_{g(\mathbf{x}) < \gamma} f_{\mathbf{X}}(\mathbf{x}) d\mathbf{x} = \int I_F(\mathbf{x}) f_{\mathbf{X}}(\mathbf{x}) d\mathbf{x} = \mathbb{E}_{f_{\mathbf{X}}}[I_F(\mathbf{x})], \quad (3.45)$$

where γ is the failure threshold, and $I_F(\mathbf{x})$ is the indicator function. $I_F(\mathbf{x})$ is 1 if $g(\mathbf{x}) < \gamma$, and 0 otherwise. Define $Z = I_F(\mathbf{x})$, then Z follows the Bernoulli distribution with the probability P_F . Therefore, the expectation and variance of Z are shown as below:

$$\mathbb{E}_{f_{\mathbf{X}}}[Z] = P_F, \quad Var(Z) = (1 - P_F)P_F. \quad (3.46)$$

Starting from the classical MCS, a more efficient importance sampling method is introduced subsequently. In order to estimate the rare event probability, the advanced subset simulation with several modified MCMC algorithms is described in detail.

3.3.1 Monte Carlo Simulation

According to the Equation (3.45), the integral can be viewed in an expectation manner. Therefore, the P_F can be estimated by means of the statistical averaging. Direct MCS [30] is a classical way to compute the expectation. The estimator of P_F can be expressed as the average of samples:

$$\hat{P}_F = \frac{1}{N} \sum_{i=1}^N I_F(\mathbf{x}_i) = \frac{1}{N} \sum_{i=1}^N z_i, \quad (3.47)$$

where $z_i = I_F(\mathbf{x}_i)$, and they are independent and identically distributed (i.i.d.). The estimator is unbiased as below:

$$\mathbb{E}_{f_{\mathbf{X}}}[\hat{P}_F] = \mathbb{E}_{f_{\mathbf{X}}} \left[\frac{1}{N} \sum_{i=1}^N z_i \right] = \frac{1}{N} \sum_{i=1}^N \mathbb{E}_{f_{\mathbf{X}}}[z_i] = \frac{1}{N} \sum_{i=1}^N \mathbb{E}_{f_{\mathbf{X}}}[Z] = \frac{1}{N} N P_F = P_F. \quad (3.48)$$

To analyze the convergence of the estimator, the variance of the estimator is computed as follows:

$$\text{Var}(\hat{P}_F) = \text{Var} \left(\frac{1}{N} \sum_{i=1}^N z_i \right) = \frac{1}{N^2} \text{Var} \left(\sum_{i=1}^N z_i \right) = \frac{1}{N^2} \sum_{i=1}^N \text{Var}(z_i) = \frac{\text{Var}(Z)}{N}. \quad (3.49)$$

The variance of the estimator decreases along with the increase of the samples. The coefficient of variance (c.o.v.) is used to measure the accuracy of the MCS. The c.o.v. of \hat{P}_F is as below:

$$\delta_{\hat{P}_F} = \frac{\sqrt{\text{Var}(\hat{P}_F)}}{P_F} = \frac{\sqrt{\text{Var}(Z)}}{\sqrt{N P_F}} = \sqrt{\frac{1 - P_F}{N P_F}} \underset{P_F \ll 1}{\approx} \frac{1}{\sqrt{N P_F}}. \quad (3.50)$$

In order to estimate the failure probability $P_F = 10^{-k}$ with the accuracy of $\delta_{\hat{P}_F} = 0.1$, the total $N = 10^{k+2}$ samples are required. Especially in aviation, the failure probability P_F is generally lower than 10^{-7} [5]. In this case, at least 10^9 samples are requested for $\delta_{\hat{P}_F} = 0.1$. The efficiency of MCS to estimate the rare event probability is quite low.

3.3.2 Importance Sampling

There are many modified sampling techniques to improve the efficiency of the direct MCS [30]. Importance sampling (IS) is a fundamental one of variance reduction sampling techniques, which might dramatically reduce the estimator's variance. The basic strategy of IS is sampling from a special proposal PDF $h(\mathbf{x})$ instead of the original input PDF $f_{\mathbf{X}}(\mathbf{x})$. The proposed $h(\mathbf{x})$ allows us to generate more samples close to the failure domain. The failure probability in Equation (3.45) can be computed as below:

$$\begin{aligned} P_F &= \Pr(g(\mathbf{x}) < \gamma) = \int I_F(\mathbf{x}) f_{\mathbf{X}}(\mathbf{x}) d\mathbf{x} = \int I_F(\mathbf{x}) \frac{f_{\mathbf{X}}(\mathbf{x})}{h(\mathbf{x})} h(\mathbf{x}) d\mathbf{x} \\ &= \mathbb{E}_h[I_F(\mathbf{x}) w(\mathbf{x})], \end{aligned} \quad (3.51)$$

where $w(\mathbf{x})$ is called the weight function, and $w(\mathbf{x}) = f_{\mathbf{X}}(\mathbf{x})/h(\mathbf{x})$. K is defined as a random variable that $K = I_F(\mathbf{X})w(\mathbf{X})$. The estimator of P_F using the IS method is computed as follows:

$$\hat{P}_F^{IS} = \frac{1}{N} \sum_{i=1}^N I_F(\mathbf{x}_i) w(\mathbf{x}_i) = \frac{1}{N} \sum_{i=1}^N k_i, \quad (3.52)$$

where $k_i = I_F(\mathbf{x}_i)w(\mathbf{x}_i)$, and \mathbf{x}_i follows the proposal PDF $h(\mathbf{x})$. If the condition $h(\mathbf{x}) > 0$ is satisfied for all $\mathbf{x} \in \{\mathbf{x} | I_F(\mathbf{x})f_{\mathbf{X}}(\mathbf{x}) > 0\}$, the expectation of K is:

$$\mathbb{E}_h[K] = \int I_F(\mathbf{X})w(\mathbf{X})d\mathbf{x} = \int I_F(\mathbf{X})\frac{f_{\mathbf{X}}(\mathbf{x})}{h(\mathbf{x})}h(\mathbf{x})d\mathbf{x} = \int I_F(\mathbf{x})f_{\mathbf{X}}(\mathbf{x})d\mathbf{x} = P_F. \quad (3.53)$$

As mentioned, the k_i is the i.i.d. sample of K . The estimator using IS is also unbiased:

$$\mathbb{E}_h[\hat{P}_F^{IS}] = \mathbb{E}_h \left[\frac{1}{N} \sum_{i=1}^N k_i \right] = \frac{1}{N} \sum_{i=1}^N \mathbb{E}_h[K] = \frac{1}{N} N P_F = P_F. \quad (3.54)$$

Before computing the variance of estimator, the variance of K is as below:

$$\text{Var}(K) = \mathbb{E}_h[K^2] - (\mathbb{E}_h[K])^2 = \int (I_F(\mathbf{X})w(\mathbf{X}))^2 h(\mathbf{x})d\mathbf{x} - P_F^2. \quad (3.55)$$

The variance of estimator can be derived subsequently:

$$\text{Var}(\hat{P}_F^{IS}) = \text{Var} \left(\frac{1}{N} \sum_{i=1}^N k_i \right) = \frac{\text{Var}(K)}{N} = \frac{1}{N} \left(\int (I_F(\mathbf{X})w(\mathbf{X}))^2 h(\mathbf{x})d\mathbf{x} - P_F^2 \right). \quad (3.56)$$

The efficiency of the IS algorithm is improved by reducing the variance of the estimator. To obtain the highest efficiency, the optimal IS density function $h(\mathbf{x})$ is as follows:

$$h(\mathbf{x})_{opt} = \frac{I_F(\mathbf{x})f_{\mathbf{X}}(\mathbf{x})}{P_F}, \quad (3.57)$$

which leads the $\text{Var}(\hat{P}_F^{IS})$ obtains the minimum value 0 as below:

$$\begin{aligned} \text{Var}(\hat{P}_F^{IS}) &= \frac{1}{N} \left(\int (I_F(\mathbf{X})w(\mathbf{X}))^2 h_{opt}(\mathbf{x})d\mathbf{x} - P_F^2 \right) \\ &= \frac{1}{N} \left(\int \frac{(I_F(\mathbf{X})f_{\mathbf{X}}(\mathbf{x}))^2}{h_{opt}(\mathbf{x})} d\mathbf{x} - P_F^2 \right) \\ &= \frac{1}{N} \left(\int I_F(\mathbf{X})f_{\mathbf{X}}(\mathbf{x})P_F d\mathbf{x} - P_F^2 \right) = \frac{1}{N} (P_F \cdot P_F - P_F^2) = 0. \end{aligned} \quad (3.58)$$

However, samples can not be directly generated from the $h_{opt}(\mathbf{x})$. Rejection sampling method can be applied. In practice, the IS method is usually combined with the known samples in the failure domain. For example, the designed point \mathbf{x}^* computed based on the FORM method is located in the boundary of the failure domain [29]. Therefore, the proposed PDF can be defined as:

$$h(\mathbf{x}) = \psi_n(\mathbf{x} - \mathbf{x}^*), \quad (3.59)$$

where ψ_n is a multivariate normal distribution centered at the design point. As illustrated in Figure 3.12, MCS-based samples and IS-based samples are compared for a two-dimensional case. By counting the number of failure samples, the IS method dramatically improves the sampling efficiency in the failure domain.

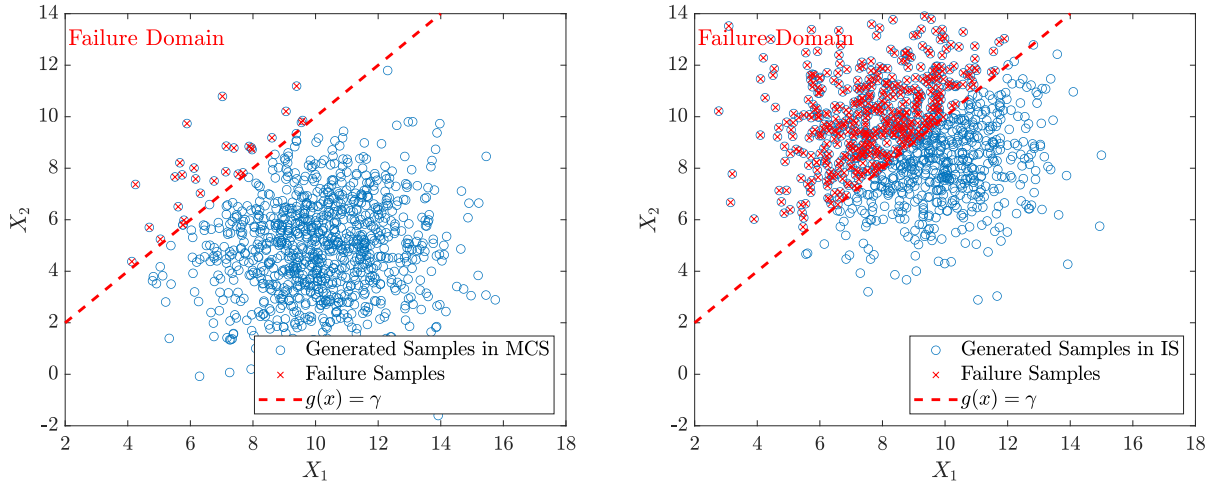


Figure 3.12: Comparison of generated 1000 samples using MCS and IS

3.3.3 Subset Simulation

Since the optimal proposal PDF in IS is difficult to obtain in high-dimensional cases, the IS method can not capture the failure region efficiently anymore [97]. IS is not suitable for the high-dimensional problem with a small failure probability. A subset simulation (SuS) method is proposed by Au[32] to deal with the rare event probability and overcome the curse of dimensionality. The SuS method is also the core algorithm in the ‘Predict’ step of the Predictive Analysis framework. This section introduces the basic strategy at first, followed by several advanced MCMC algorithms which improve the efficiency of SuS. The main idea of SuS is to express a failure event as the intersection of many intermediate events as below:

$$F = \bigcap_{m=1}^M E_m, \quad (3.60)$$

where the event E_m satisfies: $E_0 \supset E_1 \supset \dots \supset E_{M-1} \supset E_M = F$. As shown in Figure 3.13, E_0 denotes the whole available domain: $Pr(E_0) = 1$. M is the number of subset levels.

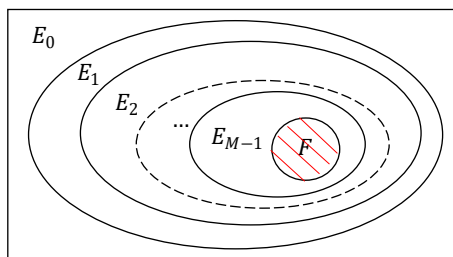


Figure 3.13: Illustration of intermediate events in subset simulation.

In this case, a small failure probability can be expressed as the product of several conditional probabilities as follows:

$$\begin{aligned} P_F &= Pr(F) = Pr\left(\bigcap_{m=1}^M E_m\right) = Pr(E_M|E_{M-1})Pr\left(\bigcap_{m=1}^{M-1} E_m\right) = \dots \\ &= Pr(E_0) \prod_{m=1}^M Pr(E_m|E_{m-1}) = \prod_{m=1}^M Pr(E_m|E_{m-1}), \end{aligned} \quad (3.61)$$

where the intermediate failure event E_m is defined as $\{\mathbf{x}|g(\mathbf{x}) < \gamma_m\}$. γ_m is the intermediate failure threshold. In practice, p_0 is the predefined parameter to denote the conditional probability, and $p_0 \in (0, 1)$. It is used to find the intermediate threshold γ_m , which lets the conditional probability $Pr(E_m|E_{m-1}) = p_0$. $p_0 \in (0, 1)$. The conditional probabilities in each subset level keep the same value p_0 . Until the actual failure domain, γ_m is set to be the actual failure threshold γ . The final conditional probability $Pr(E_M|E_{M-1})$ is computed based on the proportion of the failure samples. Therefore, the estimator of the failure probability using the SuS method is:

$$\hat{P}_F^{SuS} = \prod_{m=1}^M \hat{P}_m = \prod_{m=1}^M \hat{Pr}(E_m|E_{m-1}) \approx p_0^{M-1} \cdot \frac{n_{failure}}{N}, \quad (3.62)$$

where N is the number of samples generated in each subset, and $n_{failure}$ is the number of actual failure samples in the final subset level. Compared to a small probability, the larger conditional probability is easy to estimate. According to the assumption in SuS [32], the model input \mathbf{x} follows the d -dimensional independent standard Gaussian distribution $\psi_d(\mathbf{x})$. For the arbitrary distributed input with dependence, the isoprobabilistic transformation can be implemented [98]. Without loss of generality, \mathbf{x} is considered as independent standard Gaussian distributed variables in this section. Figure 3.14 also illustrates the process of samples generation in the subset simulation. In Figure 3.14(a), the blue circles denote the samples generated from the $\psi_d(\mathbf{x})$ using the MCS. p_0 of samples closer to the failure region are taken as the seeds denoted by the red cross. Based on the seeds, some sampling techniques are implemented to generate new samples inside the region E_1 shown in Figure 3.14(b). The same procedure works for the next subset until achieving the actual failure domain. Finally, all samples in each subset are shown in Figure 3.14(d). The procedure of the SuS algorithm is shown in Algorithm 4.

The main challenge is to generate the samples inside the intermediate failure domain, like samples in Figure 3.14(b), which is executed in the step 9 of Algorithm 4. To be precise, the task is to generate samples from the conditional PDFs $\psi_d(\mathbf{x}|E_m)$, $m = 1, 2, \dots, M-1$, where

$$\psi_d(\mathbf{x}|E_m) = \frac{\psi_d(\mathbf{x})I(g(\mathbf{x}) < \gamma_m)}{Pr(E_m)}. \quad (3.63)$$

Direct sampling is not possible. MCMC techniques provide a solution to generate the samples by constructing a Markov chain [30] (see Section 3.2.3). In SuS, each Markov chain starts from one seed of the previous subset level, and the target distribution $\psi_d(\mathbf{x}|E_m)$ is

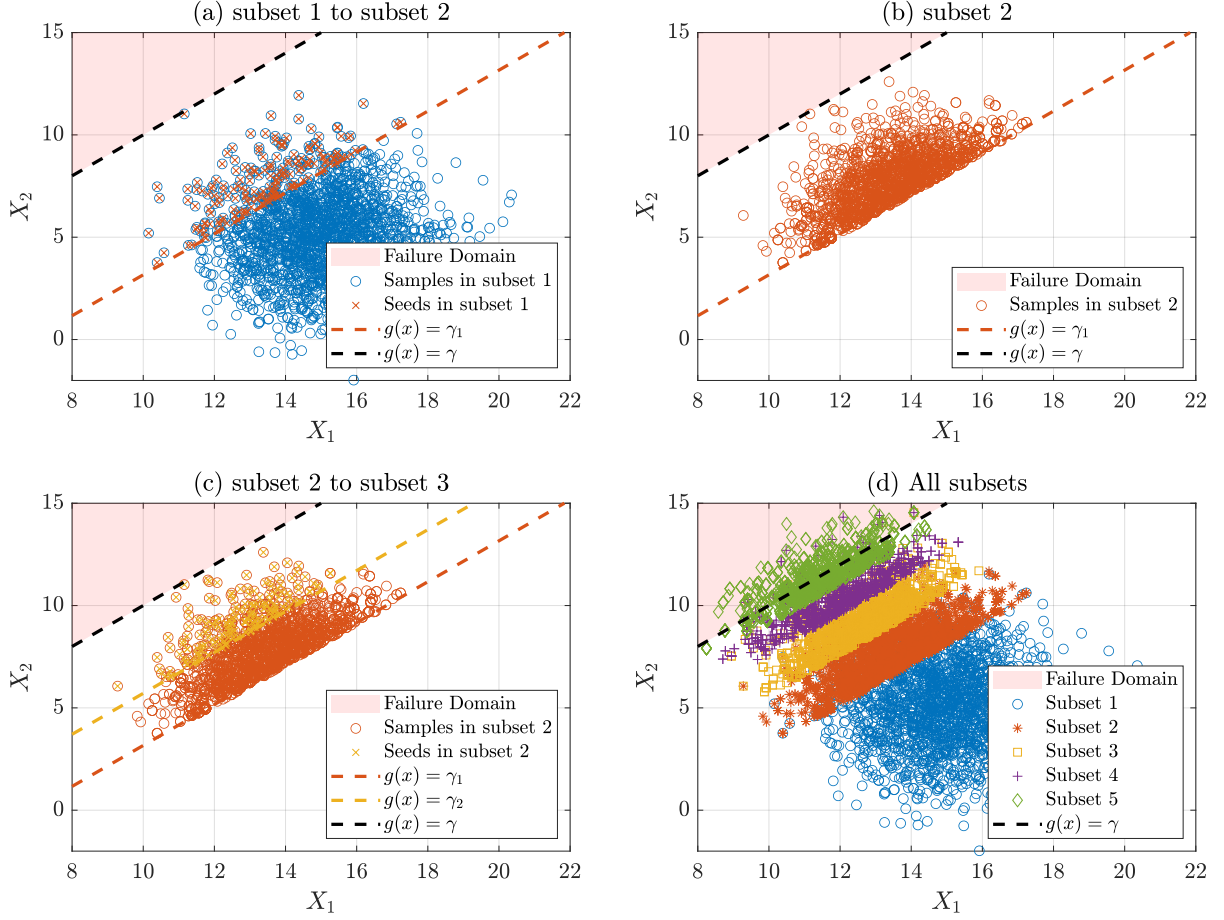


Figure 3.14: Process of the sample generation in each subset level

set as its stationary distribution. Since all seeds $\mathbf{X}_{seeds}^{(m)}$ already follows the distribution of $\psi_d(\mathbf{x}|E_m)$, the generated Markov chain have already reached the stationary distribution. Therefore, the burn-in period of the MCMC algorithm for stationary is not necessary anymore in SuS. Each seed will be taken as the initial state to generate one chain. The proposal distribution needs to be chosen to run the MCMC algorithm. The used proposal distribution and its implementation in SuS will be discussed in detail in the following section. It's worth noting that in Algorithm 4, N_0 is the number of seeds in each subset. Therefore, N_0 is also the number of chains. Furthermore, each chain has $N_c = N/N_0$ samples. It ensures that there is the same number of samples in the next subset. Finally, the probability of the failure is estimated using Equation (3.62).

Implementation of the MCMC algorithm in SuS

The procedure of the Metropolis-Hastings (MH) algorithm [93], as one of the most well-known MCMC method, is already described in Algorithm 2 of Section 3.2.3. When it comes to the step 9 in Algorithm 4 for the SuS, the implementation of the MH algorithm is described below. First, the sample candidates ξ are generated from a proposal PDF

Algorithm 4 Procedure of the SuS algorithm

- 1: Generate N samples of \mathbf{X} : $\{\mathbf{x}_j, j = 1, \dots, N\}$ from $\psi_d(\mathbf{x})$
 - 2: Set $m = 1$
 - 3: Evaluate the model $y_j = g(\mathbf{x}_j)$ and obtain the output $\mathbf{Y} = \{y_j, j = 1, \dots, N\}$
 - 4: Sort the output \mathbf{Y} from the smallest to the largest: $y_{(1)} \leq \dots \leq y_{(N)}$, and select the smallest $N_0 = p_0 \cdot N$ samples as the seeds $\mathbf{X}_{seeds}^{(m)}$.
 - 5: Compute the intermediate threshold $\gamma_m = 0.5(y_{(N_0)} + y_{(N_0+1)})$, where $y_{(N_0)}$ and $y_{(N_0+1)}$ are the N_0^{th} and $(N_0 + 1)^{th}$ value of y sorted from small to large
 - 6: **if** $\gamma_m \leq \gamma$ **then**
 - 7: Reset $\gamma_m = \gamma$, and go to step 12
 - 8: **else**
 - 9: Start from \mathbf{X}_{seeds} , and generate new samples $\{\mathbf{x}_j, j = 1, \dots, N\}$ from $\psi_d(\mathbf{x}|E_m)$ using the MCMC algorithm (discussed below in Algorithm 5)
 - 10: Set $m = m + 1$, and return to step 3
 - 11: **end if**
 - 12: Set $m = M$, and estimate P_F : $\hat{P}_F^{SuS} \approx p_0^{M-1} \cdot \frac{n_{failure}}{N}$
-

$q(\boldsymbol{\xi}|\mathbf{x}_c)$ in Equation (3.66). \mathbf{x}_c denotes the current state of the Markov chain starting from the seeds. In order to generate samples from the conditional distribution $\psi_d(\mathbf{x}|E_m)$, the candidates $\boldsymbol{\xi}$ will be accepted based on the acceptance probability as below:

$$\alpha(\mathbf{x}_c, \boldsymbol{\xi}) = \min \left\{ 1, \frac{\psi_d(\boldsymbol{\xi}|E_m)q(\mathbf{x}_c|\boldsymbol{\xi})}{\psi_d(\mathbf{x}_c|E_m)q(\boldsymbol{\xi}|\mathbf{x}_c)} \right\}. \quad (3.64)$$

As the current state \mathbf{x}_c already follows $\psi_d(\mathbf{x}|E_m)$, then $I(g(\mathbf{x}_c) < \gamma_m) = 1$. By substituting the $\psi_d(\cdot|E_m)$ using Equation (3.63), the acceptance probability can be simplified as below:

$$\begin{aligned} \alpha(\mathbf{x}_c, \boldsymbol{\xi}) &= \min \left\{ 1, \frac{[\psi_d(\boldsymbol{\xi})I(g(\boldsymbol{\xi}) < \gamma_m) / Pr(E_m)]q(\mathbf{x}_c|\boldsymbol{\xi})}{[\psi_d(\mathbf{x}_c)I(g(\mathbf{x}_c) < \gamma_m) / Pr(E_m)]q(\boldsymbol{\xi}|\mathbf{x}_c)} \right\} \\ &= \min \left\{ 1, \frac{\psi_d(\boldsymbol{\xi})q(\mathbf{x}_c|\boldsymbol{\xi})}{\psi_d(\mathbf{x}_c)q(\boldsymbol{\xi}|\mathbf{x}_c)} \right\} I(g(\boldsymbol{\xi}) < \gamma_m). \end{aligned} \quad (3.65)$$

The proposal PDF is chosen as:

$$q(\boldsymbol{\xi}|\mathbf{x}_c) = \phi_d(\boldsymbol{\xi} - \mathbf{x}_c), \quad (3.66)$$

where ϕ_d denotes the d -dimensional independent Gaussian distribution, which is equivalent to ψ_d in mathematics. However, they represent different meanings. ψ_d denotes the distribution of the input, while ϕ_d represents the proposal distribution. Therefore, two different notations are used to distinguish them. Since the proposal PDF $q(\boldsymbol{\xi}|\mathbf{x}_c)$ in Equation (3.66) is symmetric, $q(\boldsymbol{\xi}|\mathbf{x}_c) = q(\mathbf{x}_c|\boldsymbol{\xi})$. The MH algorithm with the symmetric proposal PDF is also called Metropolis algorithm [99]. Thus, the acceptance probability is simplified as:

$$\alpha(\mathbf{x}_c, \boldsymbol{\xi}) = \min \left\{ 1, \frac{\psi_d(\boldsymbol{\xi})}{\psi_d(\mathbf{x}_c)} \right\} I(g(\boldsymbol{\xi}) < \gamma_m). \quad (3.67)$$

Therefore, the acceptance ratio can be separated into two parts. One can generate a sample candidate $\boldsymbol{\xi}$ from the proposal distribution $q(\boldsymbol{\xi}|\mathbf{x}_c)$, which is accepted as the intermediate sample candidate $\boldsymbol{\nu}$ with the acceptance ratio:

$$\alpha^*(\mathbf{x}_c, \boldsymbol{\xi}) = \min \left\{ 1, \frac{\psi_d(\boldsymbol{\xi})}{\psi_d(\mathbf{x}_c)} \right\}. \quad (3.68)$$

It is followed by checking if the $\boldsymbol{\nu}$ is inside the sample space of the event E_m . The accepted intermediate candidate $\boldsymbol{\nu}$ is accepted as the state \mathbf{x}_{c+1} of the Markov chain only if it is located in the intermediate failure domain, which means $I(g(\boldsymbol{\xi}) < \gamma_m) = 1$. Otherwise, the previous state \mathbf{x}_c is used as the current state. The detailed procedure of generating N samples from $\psi_d(\mathbf{x}|E_m)$ using the MCMC algorithm is described in Algorithm 5. The ‘while-loop’ from step 4 to step 19 in Algorithm 5 is used to generate one chain with N_c samples. N_0 chains are constructed, and N samples in total are generated for the subset $m + 1$ in the end of Algorithm 5, which corresponds to the step 9 of Algorithm 4. In summary, the subset simulation is implemented to calculate the failure probability by combining Algorithm 4 and Algorithm 5.

Algorithm 5 Samples generation for subset $m + 1$ using the MCMC algorithm

- 1: Given the seeds $\{\mathbf{x}_k, k = 1, \dots, N_0\}$ from the last subset m , where $\mathbf{x}_k \in \mathbf{X}_{seeds}^{(m)}$
 - 2: **for** $k = 1, 2, \dots, N_0$ **do**
 - 3: $c = 1$ and set the current state $\mathbf{x}_c = \mathbf{x}_k$, where $\mathbf{x}_c = (x_{c,1}, \dots, x_{c,d})$
 - 4: **while** $c \leq N_c$ **do**
 - 5: Generate a candidate $\boldsymbol{\xi}$ from the proposal distribution $q(\mathbf{x}_c|\boldsymbol{\xi})$
 - 6: Generate a sample u from the uniform distributed $U(0, 1)$
 - 7: Compute the acceptance ratio $\alpha^*(\mathbf{x}_c, \boldsymbol{\xi}) = \min \left\{ 1, \frac{\psi_d(\boldsymbol{\xi})}{\psi_d(\mathbf{x}_c)} \right\}$.
 - 8: **if** $u \leq \alpha^*(\mathbf{x}_c, \boldsymbol{\xi})$ **then**
 - 9: $\boldsymbol{\nu} = \boldsymbol{\xi}$
 - 10: **else**
 - 11: $\boldsymbol{\nu} = \mathbf{x}_c$
 - 12: **end if**
 - 13: **if** $I(g(\boldsymbol{\nu}) < \gamma_m) = 1$ ($\Leftrightarrow \boldsymbol{\nu} \in E_m$) **then**
 - 14: $\mathbf{x}_{c+1} = \boldsymbol{\nu}$
 - 15: **else**
 - 16: $\mathbf{x}_{c+1} = \mathbf{x}_c$
 - 17: **end if**
 - 18: $c = c + 1$
 - 19: **end while**
 - 20: Store the samples $\{\mathbf{x}_c, c = 1, \dots, N_c\}$ in the k th Markov chain
 - 21: **end for**
 - 22: Gather all samples from N_0 chains, and obtain N samples in total
-

Coefficients of variance of the SuS estimator

Once the failure probability \hat{P}_F^{SuS} is obtained using the SuS, the accuracy of the \hat{P}_F^{SuS} should be of concern. The c.o.v. of the SuS estimator allows us to have an assessment of the quality of the estimated failure probability based on the available information after one subset simulation run. According to the analysis of efficiencies of the MCS estimator and the MCMC estimator in Appendix C, the c.o.v of \hat{P}_F^{SuS} is analyzed in this section by assessing the accuracy of each subset estimator in Equation (3.62) separately [100].

MCS estimator \hat{P}_1 Samples in the first subset are obtained by the direct MCS. According to Equation (3.50), the c.o.v. of $\hat{P}_1 = \hat{Pr}(E_1) = p_0$ is:

$$\delta_1 = \sqrt{\frac{1 - P_1}{NP_1}} = \sqrt{\frac{1 - p_0}{Np_0}}. \quad (3.69)$$

Conditional probability estimator \hat{P}_m ($2 \leq m \leq M$) From the subset level 2, the MCMC algorithm is used to generate samples following the target PDF. The illustration of Markov chains in the subset simulation is shown in Figure 3.15.

Each Markov chain in subset level m is generated based on the seeds, which are used in the previous subset level $m - 1$. Therefore, the different Markov chains in subset level m might be correlated due to the possible dependence among the seeds in subset level $m - 1$. As shown in Figure 3.15, the two chains in the subset level $m = 2$ or $m = 3$ are independent, while the two chains in the subset level $m = 4$ are dependent since their seeds are dependent.

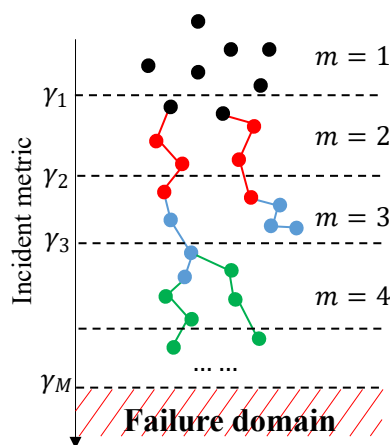


Figure 3.15: Illustration of the Markov chains in the subset simulation

Let $I_{jl}^{(m)}(\mathbf{x})$ denotes the indicator function for the intermediate failure domain E_m . \mathbf{x} here is the samples located in E_{m-1} . For simplifying the notation, $I_{jl}^{(m)}$ is used later. j denotes the j th Markov chain and there are N_0 chains in each subset as mentioned

before. l denotes the l th sample in the j th Markov chain, and $l = 1, 2, \dots, N/N_0$. $I_{jk}^{(m)} = 1$ if $g(\mathbf{x}) < \gamma_m$, and 0 otherwise. Therefore, the estimator of P_m is as below:

$$\hat{P}_m = \frac{1}{N} \sum_{j=1}^{N_0} \sum_{l=1}^{N/N_0} I_{jl}^{(m)}. \quad (3.70)$$

During the process of the SuS, γ_m is calculated to allow \hat{P}_m equal the predefined p_0 for $m = 1, 2, \dots, M - 1$. In the final subset, $\hat{P}_M = n_{failure}/N$ as mentioned. However, the variance of the conditional probability $\hat{P}_m = Pr(E_m|E_{m-1})$ can be calculated based on Equation (3.70) as below:

$$Var(\hat{P}_m) = Var \left[\frac{1}{N} \sum_{j=1}^{N_0} \sum_{l=1}^{N/N_0} I_{jl}^{(m)} \right] = \frac{1}{N^2} Var \left[\sum_{j=1}^{N_0} \sum_{l=1}^{N/N_0} I_{jl}^{(m)} \right] \quad (3.71)$$

All chains in one subset are probabilistically equivalent. As mentioned above, samples in different chains may be dependent. It is assumed for simplicity that they are uncorrelated through the indicator function $I_{jl}^{(m)}$, such that $Cov(I_{jl}^{(m)}, I_{j'l}^{(m)}) = 0$, j and j' denote two different chains. By neglecting the dependence among different chains in one subset, the $Var(\hat{P}_m)$ is derived further based on the derivation of variance of the MCMC algorithm in Equation (C.9) as below:

$$\begin{aligned} Var(\hat{P}_m) &= \frac{1}{N^2} \left(\sum_{j=1}^{N_0} \sum_{l=1}^{N/N_0} Var[I_{jl}^{(m)}] + \sum_{j=1}^{N_0} \sum_{l=1}^{N/N_0} \sum_{k=1, k \neq l}^{N/N_0} Cov(I_{jl}^{(m)}, I_{jk}^{(m)}) \right) \\ &= \frac{1}{N^2} \left(N Var[I_{jl}^{(m)}] + 2 \sum_{j=1}^{N_0} \sum_{k=1}^{N/N_0-1} \left(\frac{N}{N_0} - k \right) Var[I_{jl}^{(m)}] \rho_m(k) \right) \\ &= \frac{1}{N^2} \left(N Var[I_{jl}^{(m)}] + 2 \frac{N}{N_0} \sum_{j=1}^{N_0} \sum_{k=1}^{N/N_0-1} \left(1 - \frac{kN_0}{N} \right) Var[I_{jl}^{(m)}] \rho_m(k) \right) \\ &= \frac{Var[I_{jl}^{(m)}]}{N} \left(1 + 2 \sum_{k=1}^{N/N_0-1} \left(1 - \frac{kN_0}{N} \right) \rho_m(k) \right). \end{aligned} \quad (3.72)$$

Since $I_{jl}^{(m)}$ follows the Bernoulli distribution with the probability P_m . Therefore,

$$Var[I_{jl}^{(m)}] = P_m(1 - P_m). \quad (3.73)$$

The c.o.v. of the estimator \hat{P}_m is thus obtained as below [100]:

$$\delta_m = \frac{Var[\hat{P}_m]}{P_m} = \sqrt{\frac{1 - P_m}{NP_m} \left(1 + 2 \sum_{k=1}^{N/N_0-1} \left(1 - \frac{kN_0}{N} \right) \rho_m(k) \right)} = \sqrt{\frac{1 - P_m}{NP_m} (1 + \lambda_m)}. \quad (3.74)$$

where

$$\lambda_m = 2 \sum_{k=1}^{N/N_0-1} \left(1 - \frac{kN_0}{N} \right) \rho_m(k). \quad (3.75)$$

Since the $\lambda_m > 0$, the c.o.v. of the conditional probability estimation is larger than that of the direct MCS estimator. As the samples autocorrelation $\rho_m(k)$ in each Markov chain decreases, the λ_m will decrease, and the accuracy of estimator will increase. Based on Equation (3.39), $\rho_m(k)$ is calculated based on the samples:

$$\rho_m(k) = \frac{s_k^{(m)}}{s_0^{(m)}}, \quad (3.76)$$

where $s_0^{(m)}$ is the variance of $I_{jl}^{(m)}$. Therefore,

$$s_0^{(m)} = \text{Var}[I_{jl}^{(m)}] = P_m(1 - P_m). \quad (3.77)$$

$s_k^{(m)}$ is the sample covariance. It is estimated by using the average of sample covariances in all chains at subset level m as below:

$$\begin{aligned} s_k^{(m)} &\approx \bar{s}_k^{(m)} = \frac{1}{N - kN_0} \sum_{j=1}^{N_0} \sum_{l=1}^{N/N_0-k} ((I_{jl} - E[I_{jl}])(I_{j,l+k} - E[I_{j,l+k}])) \\ &= \frac{1}{N - kN_0} \sum_{j=1}^{N_0} \sum_{l=1}^{N/N_0-k} (I_{jl}I_{j,l+k} - \hat{P}_m I_{jl} - \hat{P}_m I_{j,l+k} + \hat{P}_m^2) \\ &= \left(\frac{1}{N - kN_0} \sum_{j=1}^{N_0} \sum_{l=1}^{N/N_0-k} I_{jl}I_{j,l+k} \right) - \hat{P}_m E[I_{jl}] - \hat{P}_m E[I_{j,l+k}] + \hat{P}_m^2 \\ &= \left(\frac{1}{N - kN_0} \sum_{j=1}^{N_0} \sum_{l=1}^{N/N_0-k} I_{jl}I_{j,l+k} \right) - \hat{P}_m^2 \end{aligned} \quad (3.78)$$

Failure probability estimator \hat{P}_F^{SuS} The statistical properties of the \hat{P}_F^{SuS} is analyzed in [32, 100]. The simplified derivation of the c.o.v. of \hat{P}_F^{SuS} is shown below. Instead of \hat{P}_F^{SuS} , \hat{P}_F is used for simplicity later. The square of the c.o.v. of \hat{P}_F is as below:

$$\begin{aligned} \delta_{\hat{P}_F^{SuS}}^2 &= \frac{\text{Var}[\hat{P}_F]}{P_F^2} = \frac{E[\hat{P}_F - P_F]^2}{P_F^2} \\ &= E \left[\frac{\hat{P}_F - P_F}{P_F} \right]^2 = E \left[\frac{\prod_{m=1}^M \hat{P}_m}{\prod_{m=1}^M P_m} - 1 \right]^2 \\ &= E \left[\prod_{m=1}^M \frac{\hat{P}_m}{P_m} - 1 \right]^2 \end{aligned} \quad (3.79)$$

To simplify the equation above using the c.o.v. of all subset estimators, the paper [100] proposed to define the variable Z_m as

$$Z_m = (\hat{P}_m - P_m) / \sqrt{\text{Var}[\hat{P}_m]}. \quad (3.80)$$

Obviously, $E[Z_m] = 0$, and $E[Z_m^2] = 1$, since

$$E[\hat{P}_m - P_m] = P_m - P_m = 0; \quad E[(\hat{P}_m - P_m)^2] = \text{Var}[\hat{P}_m]. \quad (3.81)$$

Therefore, Equation (3.79) is further derived as below:

$$\begin{aligned}
 \delta_{\hat{P}_F^{SuS}}^2 &= E \left[\prod_{m=1}^M \left(\frac{\hat{P}_m - P_m}{P_m} + 1 \right) - 1 \right]^2 \\
 &= E \left[\prod_{m=1}^M \left(\frac{\hat{P}_m - P_m}{\sqrt{\text{Var}[\hat{P}_m]} P_m} \sqrt{\text{Var}[\hat{P}_m]} + 1 \right) - 1 \right]^2 \\
 &= E \left[\prod_{m=1}^M (Z_m \delta_m + 1) - 1 \right]^2,
 \end{aligned} \tag{3.82}$$

where

$$\begin{aligned}
 \prod_{m=1}^M (Z_m \delta_m + 1) - 1 &= \sum_{m=1}^M \delta_m Z_m + \sum_{m_1 > m_2} \delta_{m_1} \delta_{m_2} Z_{m_1} Z_{m_2} \\
 &+ \sum_{m_1 > m_2 > m_3} \delta_{m_1} \delta_{m_2} \delta_{m_3} Z_{m_1} Z_{m_2} Z_{m_3} + \cdots + \prod_{m=1}^M \delta_m Z_m.
 \end{aligned} \tag{3.83}$$

Since $\delta_m \rightarrow 0$ in the order of $1/\sqrt{N}$ denoted by $O(1/\sqrt{N})$ based on Equation (3.74), the first term in Equation (3.83) is $O(1/\sqrt{N})$ while the remaining sums of higher order products approximate 0 in a higher order than $O(1/\sqrt{N})$ that is denoted by $o(1/\sqrt{N})$. Therefore, Equation (3.82) is further derived:

$$\begin{aligned}
 \delta_{\hat{P}_F^{SuS}}^2 &= E \left[\prod_{m=1}^M (Z_m \delta_m + 1) - 1 \right]^2 = E \left[\sum_{m=1}^M \delta_m Z_m + o(1/\sqrt{N}) \right]^2 \\
 &= E \left[\left(\sum_{m_1=1}^M \delta_{m_1} Z_{m_1} \right) \left(\sum_{m_2=1}^M \delta_{m_2} Z_{m_2} \right) + o(1/N) \right] \\
 &= \sum_{m_1=1}^M \sum_{m_2=1}^M \delta_{m_1} \delta_{m_2} E[Z_{m_1} Z_{m_2}] + o(1/N),
 \end{aligned} \tag{3.84}$$

where $E[Z_m^2] = 1$ as mentioned before. Based on the Cauchy-Schwarz inequality, $E[Z_{m_1} Z_{m_2}]$ satisfies [100]:

$$E[Z_{m_1} Z_{m_2}] \leq \sqrt{E[Z_{m_1}^2] E[Z_{m_2}^2]} = 1. \tag{3.85}$$

Therefore,

$$\begin{aligned}
 \delta_{\hat{P}_F^{SuS}}^2 &= \sum_{m_1=1}^M \sum_{m_2=1}^M \delta_{m_1} \delta_{m_2} E[Z_{m_1} Z_{m_2}] + o(1/N) \\
 &\leq \sum_{m_1=1}^M \sum_{m_2=1}^M \delta_{m_1} \delta_{m_2} + o(1/N).
 \end{aligned} \tag{3.86}$$

As illustrated in Figure 3.15, the conditional probability estimators $\{\hat{P}_m\}$ are correlated due to the fact that the samples, used to calculate the \hat{P}_m , are used to start the Markov chains in the $m + 1$ subset level to calculate \hat{P}_{m+1} . Based on the definition of Z_m in Equation (3.80), $\{Z_m\}$ are correlated due to the correlated $\{\hat{P}_m\}$. Noted that the upper bound in Equation (3.86) corresponds to the case that \hat{P}_{m+1} are fully correlated, since

$E[Z_{m_1}Z_{m_2}] = 1$ for the fully correlated $\{Z_m\}$. If all \hat{P}_{m+1} are uncorrelated, $\{Z_m\}$ are thus uncorrelated. Then $E[Z_{m_1}Z_{m_2}] = 0$ for $m_1 \neq m_2$. In this case, the c.o.v. of \hat{P}_F^{SuS} is as below by ignoring higher order terms of $o(1/N)$:

$$\delta_{\hat{P}_F^{SuS}}^2 = \sum_{m=1}^M \delta_m^2 E[Z_m^2] = \sum_{m=1}^M \delta_m^2. \quad (3.87)$$

The exact correlations among the conditional probability estimator $\{\hat{P}_m\}$ are difficult to calculate, while $\delta_{\hat{P}_F^{SuS}}^2$ is well approximated by $\sum_{m=1}^M \delta_m^2$ in Equation (3.87) in practice [100]. And $\sum_{m_1=1}^M \sum_{m_2=1}^M \delta_{m_1} \delta_{m_2}$ in Equation (3.86) is too conservative but it can be taken as the upper bound of $\delta_{\hat{P}_F^{SuS}}^2$ as a reference.

In summary, the accuracy of the SuS estimator is highly influenced by the δ_m in each subset level. Furthermore, the δ_m depends on the sample autocorrelations in each Markov chain due to the properties of the MCMC algorithm. The procedures and properties of the MCMC algorithms are already discussed in Section 3.2.3. However, there are still some implementation issues of the MCMC algorithm specifically for the subset simulation. Several variants of the MCMC algorithms are discussed for the improvement of the SuS estimator's accuracy in the following context.

Variants of the MCMC algorithms in SuS

① Component-wise Metropolis-Hastings algorithm

In practice, the average acceptance rate of generated candidates using the MH algorithm decreases fast when the dimension of the input grows [101]. This is due to the fact that the term $\psi_d(\boldsymbol{\xi})/\psi_d(\mathbf{x}_c)$ of the acceptance ratio α^* in Equation (3.68) drops along with the increase of d . To overcome the low acceptance ratio in high dimensional problems, Au proposed a component-wise MH algorithm in [100]. Instead of sampling from a d -dimensional proposal PDF $q(\boldsymbol{\xi}|\mathbf{x}_c)$, each component ξ_i of the candidate $\boldsymbol{\xi}$ is generated from the one-dimensional PDF:

$$q_i(\xi_i|x_{c,i}) = \phi_i(\xi_i - x_{c,i}), \quad (3.88)$$

where ξ_i and $x_{c,i}$ denote the i th component of $\boldsymbol{\xi}$ and \mathbf{x} , respectively. ϕ_i is the standard Gaussian distribution density function. Furthermore, each component ξ_i of the candidate $\boldsymbol{\xi}$ is accepted or rejected separately. This allows us to disassemble $\psi_d(\boldsymbol{\xi})/\psi_d(\mathbf{x}_c)$ into one-dimensional $\psi_i(\xi_i)/\psi_i(x_{c,i})$ and to avoid the low acceptance ratio in high dimensions. The detailed procedure of generating N samples from $\psi_d(\mathbf{x}|E_m)$ using the component-wise MH algorithm is described in Algorithm 6. Compared to Algorithm 5, the ‘while-loop’ from step 4 to 22 in Algorithm 6 replaces the ‘while-loop’ from step 4 to 19 in Algorithm 5.

Algorithm 6 Samples generation for subset $m + 1$ using the component-wise MH algorithm

```

1: Given the seeds  $\{\mathbf{x}_k, k = 1, \dots, N_0\}$  from the last subset  $m$ , where  $\mathbf{x}_k \in \mathbf{X}_{seeds}^{(m)}$ 
2: for  $k = 1, 2, \dots, N_0$  do
3:    $c = 1$  and set the current state  $\mathbf{x}_c = \mathbf{x}_k$ , where  $\mathbf{x}_c = \{x_{c,1}, \dots, x_{c,d}\}$ 
4:   while  $c \leq N_c$  do
5:     for  $i = 1, 2, \dots, d$  do
6:       Generate a candidate  $\xi_i$  from the proposal distribution  $q_i(\xi_i|x_{c,i})$ 
7:       Generate a sample  $u$  from the uniform distributed  $U(0, 1)$ 
8:       Compute the acceptance ratio  $\alpha_i(x_{c,i}, \xi_i) = \min\left\{1, \frac{\psi_i(\xi_i)}{\psi_i(x_{c,i})}\right\}$ 
9:       if  $u \leq \alpha_i(x_{c,i}, \xi_i)$  then
10:          $v_i = \xi_i$ 
11:       else
12:          $v_i = x_{c,i}$ 
13:       end if
14:     end for
15:     Obtain  $\mathbf{v} = (v_1, \dots, v_d)$ 
16:     if  $I(g(\mathbf{v}) < \gamma_m) = 1$  ( $\Leftrightarrow \mathbf{v} \in E_m$ ) then
17:        $\mathbf{x}_{c+1} = \mathbf{v}$ 
18:     else
19:        $\mathbf{x}_{c+1} = \mathbf{x}_c$ 
20:     end if
21:      $c = c + 1$ 
22:   end while
23:   Store the samples  $\{\mathbf{x}_c, c = 1, \dots, N_c\}$  in the  $k$ th Markov chain
24: end for
25: Gather all samples from  $N_0$  chains, and obtain  $N$  samples in total

```

② Conditional Sampling

In the previous section, the component-wise MH algorithm is discussed to ensure a high acceptance probability α^* in high dimensions. A new variant of the MCMC algorithm called conditional sampling method is shown to guarantee that the first part of the candidate acceptance probability α^* in Equation (3.68) always equals 1 by choosing a special proposal distribution. This approach is presented by Iason in paper [33]. A similar idea is also proposed by Au [102], which is called the Limiting algorithm. As mentioned, the current sample depends on its previous sample in a Markov chain. One can directly propose a candidate $\boldsymbol{\xi}$ correlated with the previous state \mathbf{x}_c during the Markov chain generation. The new proposal PDF is given by a d -dimensional Gaussian distribution with the mean

of $\rho \mathbf{x}_c$ and the variance of $(1 - \rho^2)\mathbf{I}$ as below:

$$q(\boldsymbol{\xi}|\mathbf{x}_c) = \phi_d(\boldsymbol{\xi} - \rho \mathbf{x}_c; (1 - \rho^2)\mathbf{I}), \quad (3.89)$$

where ρ is the correlation coefficient between the previous state and the current candidate. ρ is predefined as a constant value from 0 to 1, e.g. $\rho = 0.8$. The proposal PDF of the conditional sampling for one dimension is illustrated in Figure 3.16. Noticed that the non-standard Gaussian distributed x_i has been transferred into the standard Gaussian Z space before. The proposed PDF of the conditional sampling is more close to the distribution of x_i and its variance is less compared to the random walker sampling.

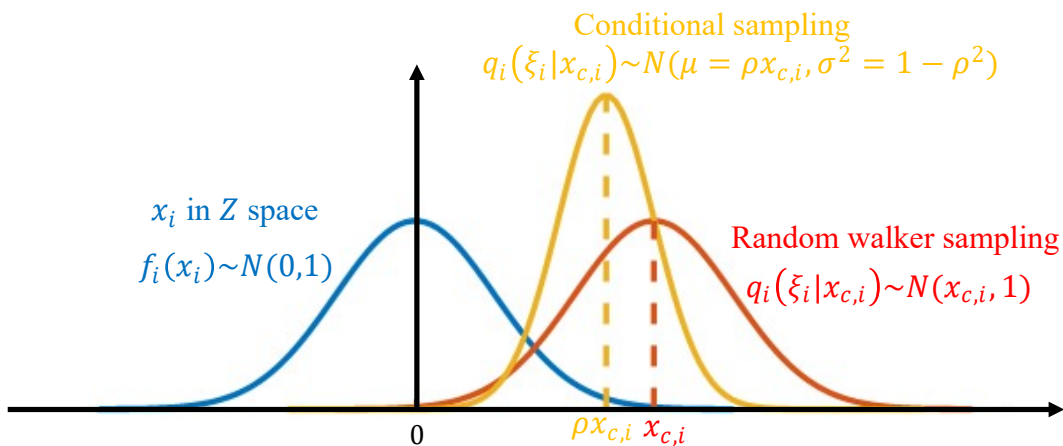


Figure 3.16: The proposal PDF of the conditional sampling

The acceptance probability in Equation (3.65) can be further derived as:

$$\begin{aligned} \alpha(\mathbf{x}_c, \boldsymbol{\xi}) &= \min \left\{ 1, \frac{\psi_d(\boldsymbol{\xi})q(\mathbf{x}_c|\boldsymbol{\xi})}{\psi_d(\mathbf{x}_c)q(\boldsymbol{\xi}|\mathbf{x}_c)} \right\} I(g(\boldsymbol{\xi}) < \gamma_m) \\ &= \min \left\{ 1, \frac{\psi_d(\boldsymbol{\xi})\phi_d(\mathbf{x}_c - \rho\boldsymbol{\xi}; (1 - \rho^2)\mathbf{I})}{\psi_d(\mathbf{x}_c)\phi_d(\boldsymbol{\xi} - \rho\mathbf{x}_c; (1 - \rho^2)\mathbf{I})} \right\} I(g(\boldsymbol{\xi}) < \gamma_m) \\ &= I(g(\boldsymbol{\xi}) < \gamma_m). \end{aligned} \quad (3.90)$$

Both ψ_d and ϕ_d are d -dimensional standard Gaussian distribution and they are equivalent. The numerator and the denominator inside the min operation can be canceled out since

$$\psi_d(\boldsymbol{\xi})\phi_d(\mathbf{x}_c - \rho\boldsymbol{\xi}; (1 - \rho^2)\mathbf{I}) = \psi_d(\mathbf{x}_c)\phi_d(\boldsymbol{\xi} - \rho\mathbf{x}_c; (1 - \rho^2)\mathbf{I}). \quad (3.91)$$

Equation (3.91) above can be proved by introducing a 2d-dimensional Gaussian random vector \mathbf{U} with the joint PDF $\psi_{2d}(\mathbf{u}; \Sigma)$ [33]. \mathbf{U} has the zero mean vector with the covariance matrix:

$$\Sigma = \begin{bmatrix} \mathbf{I} & \rho\mathbf{I} \\ \rho\mathbf{I} & \mathbf{I} \end{bmatrix}. \quad (3.92)$$

Let $\mathbf{U} = [\mathbf{U}_1; \mathbf{U}_2]$ and both \mathbf{U}_1 and \mathbf{U}_2 follow the d -dimensional standard Gaussian distribution. Thus, \mathbf{U}_1 and \mathbf{U}_2 have the marginal PDF $\psi_d(\cdot)$. Then the conditional PDF of \mathbf{U}_1 given \mathbf{U}_2 is $\psi_d(\mathbf{u}_1 - \rho\mathbf{u}_2; (1 - \rho^2)\mathbf{I})$ and the conditional PDF of \mathbf{U}_2 given \mathbf{U}_1 is

$\psi_d(\mathbf{u}_2 - \rho\mathbf{u}_1; (1 - \rho^2)\mathbf{I})$. According to the Bayes rule, the conditional PDFs are shown as below:

$$\psi_d(\mathbf{u}_1 - \rho\mathbf{u}_2; (1 - \rho^2)\mathbf{I}) = \frac{\psi_{2d}(\mathbf{u}; \Sigma)}{\psi_d(\mathbf{u}_2)}; \quad \psi_d(\mathbf{u}_2 - \rho\mathbf{u}_1; (1 - \rho^2)\mathbf{I}) = \frac{\psi_{2d}(\mathbf{u}; \Sigma)}{\psi_d(\mathbf{u}_1)} \quad (3.93)$$

By multiplying the denominator in both sides of the equation above, the joint PDF of \mathbf{U} is expressed as below:

$$\psi_{2d}(\mathbf{u}; \Sigma) = \psi_d(\mathbf{u}_1)\psi_d(\mathbf{u}_2 - \rho\mathbf{u}_1; (1 - \rho^2)\mathbf{I}) = \psi_d(\mathbf{u}_2)\psi_d(\mathbf{u}_1 - \rho\mathbf{u}_2; (1 - \rho^2)\mathbf{I}). \quad (3.94)$$

while \mathbf{u}_1 and \mathbf{u}_2 correspond to \mathbf{x}_c and $\boldsymbol{\xi}$, respectively. The above shows Equation (3.91) is satisfied. Therefore, the min operation in Equation (3.90) always equals 1. The only acceptance condition becomes if the generated candidate belongs to the sample space of the event E_m . And the acceptance ratio does not depend on the dimensions anymore. Furthermore, the component-wise strategy can also work with the conditional sampling method. The procedure of obtaining \mathbf{v} from step 5 to step 15 in Algorithm 6 can be replaced by directly sampling from the new proposal distribution $\phi_d(\boldsymbol{\xi} - \rho\mathbf{x}_c; (1 - \rho^2)\mathbf{I})$. The procedure of the component-wise conditional sampling method is described as below:

Algorithm 7 The component-wise conditional sampling algorithm

- 1: Following the step 4 in Algorithm 6
 - 2: **for** $i = 1, 2, \dots, d$ **do**
 - 3: Generate a candidate ξ_i from the standard normal distribution $N(0, 1)$
 - 4: $v_i = \rho x_{c,i} + \xi_i \cdot \sqrt{1 - \rho^2}$
 - 5: **end for**
 - 6: Obtain $\mathbf{v} = (v_1, \dots, v_d)$
 - 7: Go to step 16 in Algorithm 6
-

According to the MCMC properties mentioned in Section 3.2.3, the autocorrelation between samples determine the efficiency of the MCMC. It is worth noting that a small predefined correlation parameter ρ between the previous state \mathbf{x}_c and the current candidate state $\boldsymbol{\xi}$ does not imply a small autocorrelation of the final samples in the Markov chain, since a small ρ will cause many rejected candidates. The previous state will be repeated as the current state, which causes a high correlation among the final samples. In contrast, a large ρ will lead the candidate to be very close to the previous state. Most candidates will be accepted but with a high correlation. The parameter ρ needs to be defined before run the component-wise conditional sampling algorithm. ρ can be set by trial and error. To adjust ρ automatically, the adaptive conditional sampling is presented in the following.

③ Adaptive Conditional Sampling

Based on the conditional sampling algorithm, the adaptive conditional sampling is presented in this section to improve the accuracy of the SuS by adjusting ρ automatically. Noticed that $\sqrt{1 - \rho^2}$ is the standard deviation σ of the proposal PDF. Therefore, the pre-defined correlation parameter ρ and the standard deviation σ correspond each other. As mentioned in Section 3.2.3, the optimal acceptance ratio α is proposed for the efficiency of the MCMC algorithm instead of the optimal σ of the proposal PDF. Furthermore, the available range of the input variable becomes smaller as the subset level increases considering the procedure of the SuS. This is due to the reduction of the intermediate failure domain with the increase of the subset level. Most of the candidates are rejected due to the large σ of the proposal PDF defined at the first subset level, corresponding to a small ρ . An adaptive conditional sampling method is proposed by Iason [33] to reduce the autocorrelations by achieving the optimal acceptance ratio. ρ mentioned above is same for each dimension. However, the component-wise conditional sampling algorithm allows us to set different value of ρ for different dimension of samples. ρ_i denotes the defined correlation parameter for i th dimension of samples, while σ_i denotes the standard deviation of the proposal PDF for i th dimension. The σ_i , corresponding to the parameter ρ_i , is continually modified to obtain the target acceptance ratio. The adjusting strategy is illustrated in Figure 3.17 and described in the following context.

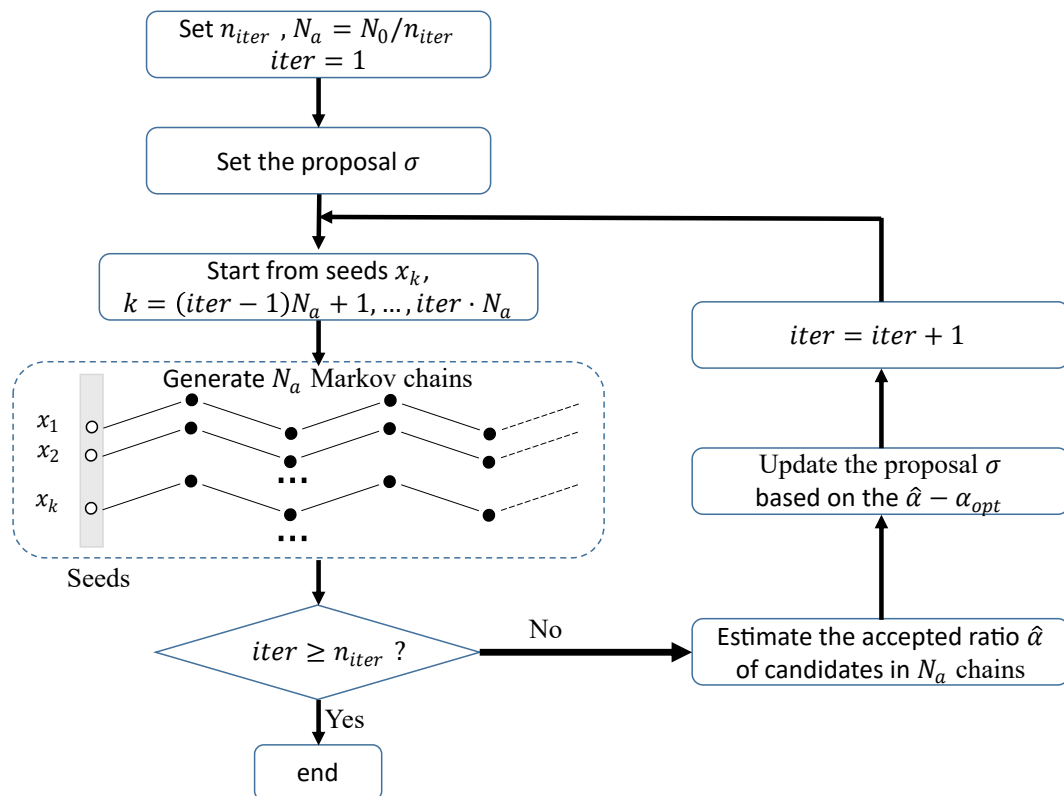


Figure 3.17: Illustration of the adaptive conditional sampling procedure

Instead of generating N_0 chains at once in each subset level, N_0 chains are generated separately in n_{iter} iterations [33]. First, $N_a = N_0/n_{iter}$ seeds $\{\mathbf{x}_j, j = 1, \dots, N_a\}$ are chosen randomly from the seeds $\mathbf{X}_{seeds}^{(m)}$. After that, N_a chains are generated using the conditional sampling algorithm with the initial σ_i , which is set based on the estimated standard deviation of all seeds \mathbf{x}_j . The average acceptance ratio of N_a chains is estimated by:

$$\hat{\alpha}_{iter} = \frac{1}{N_a} \sum_{j=1}^{N_a} E[\alpha(\boldsymbol{\xi}_{jl})], \quad (3.95)$$

where $E[\alpha(\boldsymbol{\xi}_{jl})]$ is the average acceptance ratio of sample candidates $\{\boldsymbol{\xi}_{jl}\}$ in the j th chain. l denote the order of sample candidates in a chain, and $l = 1, 2, \dots, N/N_0 - 1$. $E[\alpha(\boldsymbol{\xi}_{jl})]$ is calculated via counting the number of accepted candidates. Before generating the N_a chains in the next iteration, the scaling factor λ_{iter} is introduced to adjust the current $\sigma_{i,iter}$ in order to let the $\hat{\alpha}_{iter}$ close to the optimal acceptance ratio α_{opt} . One strategy of updating λ_{iter} is proposed by [33] as below:

$$\log \lambda_{iter+1} = \log \lambda_{iter} + k_{iter}[\hat{\alpha}_{iter} - \alpha_{opt}], \quad (3.96)$$

where the subscript $iter$ denotes the order of the iteration. α_{opt} is set as 0.44, which is the optimal acceptance ratio of the MH random walker sampler in one dimension. k_{iter} is used to reduce the variation of the λ_{iter} along with the increase of the iterations. It increases the convergence of the algorithm to obtain the optimal standard deviation. k_{iter} is set to $1/\sqrt{iter}$ here. The σ_i of the proposal PDF, which is used to generate another N_a chains in the next iteration, will be updated as follows [33]:

$$\sigma_{i,iter+1} = \min(\lambda_{iter+1}\sigma_{i,iter}, 1). \quad (3.97)$$

The updated σ_i cannot be larger than the standard deviation of the input variable, which is 1 for the standard normal distribution. If the estimated acceptance ratio $\hat{\alpha}_{iter}$ is less than 0.44, the updated λ_{iter+1} becomes smaller via Equation (3.96). This leads to a smaller standard deviation $\sigma_{iter+1,i}$ via Equation (3.97), and the corresponding ρ_i increases. The increased ρ_i will allow us to generate the candidates close to the previous state, which will increase the acceptance ratio in the next iteration of N_a chains generation. The opposite measures are taken when $\hat{\alpha}_{iter}$ is larger than 0.44. The acceptance ratio will be around 0.44 after several iterations. Finally, a total of N samples following $\psi_d(\mathbf{x}|E_m)$ are generated. The whole procedure is shown in Algorithm 8.

Furthermore, this adaptive conditional sampling algorithm also avoids the discrete distributed features of the failure samples mentioned in the paper [103]. As an example, the SuS algorithm is implemented in the runway overrun case mentioned in Section 2.5. The original conditional sampling algorithm and the adaptive conditional sampling algorithm are applied separately. Other settings of the SuS are the same. The number of samples in each subset N equals 5000, and the conditional probability p_0 is set as 0.1. Figure 3.18 shows two histograms of headwind samples in the subset level 15.

Algorithm 8 Samples generation for subset $m + 1$ using the adaptive component-wise conditional sampling algorithm, adapted from [33]

- 1: Given the seeds $\{\mathbf{x}_k, k = 1, \dots, N_0\}$ from the last subset m , where $\mathbf{x}_k \in \mathbf{X}_{seeds}^{(m)}$
- 2: Compute the standard deviation $\hat{\sigma}_i$ of seeds \mathbf{x}_k for each dimension.
- 3: Set the initial standard deviation of the proposal density function $\sigma_{iter,i} = \hat{\sigma}_i, i = 1, 2, \dots, d$. and $iter = 1$.
- 4: Permute the seeds $\{\mathbf{x}_k, k = 1, \dots, N_0\}$ randomly
- 5: **for** $iter = 1, 2, \dots, N_0/N_a$ **do**
- 6: Compute the correlation coefficients $\rho_i = \sqrt{1 - \sigma_{iter,i}^2}$
- 7: Start from the seeds $\{\mathbf{x}_k, k = (iter - 1)N_a + 1, \dots, iter \cdot N_a\}$, and generate N_a chains with N_c samples each chain from $\psi_d(\mathbf{x}|E_m)$ using the conditional sampling method with ρ_i according to Algorithm 6 and Algorithm 7
- 8: Compute the average acceptance ratio \hat{a}_{iter} of the generated N_a chains
- 9: Compute the scaling parameter λ_{iter+1}
- 10: Update the standard deviation $\sigma_{iter+1,i}$
- 11: **end for**
- 12: Collect all samples $N = N_a \cdot N_c \cdot iter$

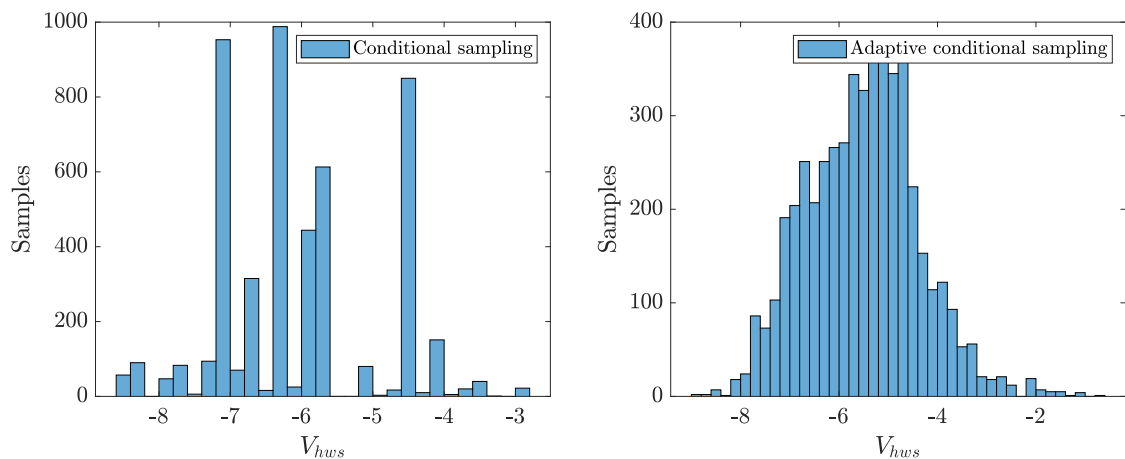


Figure 3.18: Histogram of intermediate failure samples in subset level 15 for the original and adaptive conditional sampling method

Compared to the continuous distribution obtained by the adaptive conditional sampling, the histogram obtained by the original conditional sampling shows a discrete feature. This is due to the low acceptance ratio in the high subset level when using the original conditional sampling with a fixed σ . The low acceptance ratio means that most candidates are rejected, which leads to a high number of repeated samples. However, the adaptive conditional sampling method avoid this by adjusting σ continually and it is recommended when the high subset level is required in practice.

Dependent Sampling in SuS

All sampling methods mentioned above are applied for the independent input random variables. When the contributing factors of the original model are dependent, the isoprobabilistic transformation can be applied to transfer the original dependent random variables to the independent standard normal distributed random variables. Höndorf proposed an approach to integrate vine copula into the SuS for the realization of the high dimensional isoprobabilistic transformation [34]. The transformation proposed in [34] allows us to add or remove the dependence structure, which is described as below:

$$\mathbf{X}^{dep} \xrightleftharpoons[F_i^{-1}]{F_i} \mathbf{U}^{dep} \xrightleftharpoons[add\ C]{remove\ C} \mathbf{U}^{ind} \xrightleftharpoons[\phi_i]{\phi_i^{-1}} \mathbf{Z}^{ind}, \quad (3.98)$$

where the superscript *dep* indicates variables are dependent, and the *ind* denotes the independent variables. Let X_i and Z_i denote each component of \mathbf{X}^{dep} and \mathbf{Z}^{ind} , respectively. F_i and F_i^{-1} are the estimated marginal CDF and its corresponding inverse CDF with respect to X_i . ϕ_i and ϕ_i^{-1} denote the CDF and the inverse CDF of Z_i . The variables U_i^{dep} and U_i^{ind} denote the component of \mathbf{U}^{dep} and \mathbf{U}^{ind} , and they are the cumulative results of the corresponding samples X_i and Z_i , respectively. Therefore, U_i^{dep} and U_i^{ind} are uniformly distributed on $[0, 1]$. C denotes the dependent structure among all variables. As mentioned in Section 3.2.2, C is constructed based on the vine structure and the bivariate copula distribution functions, which are estimated based on the measurements of \mathbf{X}^{dep} before the transformation. The procedure of generating the dependent samples \mathbf{U}^{dep} from the independent samples \mathbf{U}^{ind} is already described in Algorithm 3 of Section 3.2.3. The inverse procedure also works to remove the dependence. To simplify the notation, \mathbf{T} and \mathbf{T}^{-1} are used to denote the transformation between the dependent samples and the independent samples as below:

$$\mathbf{Z}^{ind} = \mathbf{T}(\mathbf{X}^{dep}), \quad \mathbf{X}^{dep} = \mathbf{T}^{-1}(\mathbf{Z}^{ind}). \quad (3.99)$$

Noticed that the Z scale data mentioned in Section 3.2.2 is different from the independent samples \mathbf{Z}^{ind} . Based on Equation (3.9,3.10), the dependence of Z scale copula data is same with the original samples \mathbf{X} . In order to distinguish the Z scale data with independent samples \mathbf{Z}^{ind} , Z scale data is denoted by \mathbf{Z}^{dep} . The transformation is shown as below:

$$\mathbf{X}^{dep} \xrightleftharpoons[F_i^{-1}]{F_i} \mathbf{U}^{dep} \xrightleftharpoons[\phi_i]{\phi_i^{-1}} \mathbf{Z}^{dep}. \quad (3.100)$$

Based on the Höndorf's idea in [34] and the MCMC methods mentioned above, the modified complete procedure of the subset simulation with the adaptive conditional sampling and dependent input variables is shown in Figure 3.19. X space denotes the original space of variables with dependence, while Z space is the independent standard Gaussian space here. All \mathbf{X} denotes the original variables and \mathbf{Z} denotes the independent variables in the independent standard Gaussian Z space without mentioning superscripts of *dep* and *ind* for simplification. The three required transformation between independent samples and dependent samples, marked by blue boxes in Figure 3.19, is described as below:

- (1) It is used to transfer the dependent seeds $\mathbf{X}_{seeds}^{(m)}$ for the subset level m in the original space to the independent samples $\mathbf{Z}_{seeds}^{(m)}$ in Z space, since the generated candidates using the MCMC algorithm start from those seeds in Z space.
- (2) It is used to transfer the candidate \mathbf{v}_Z produced in Z space to \mathbf{v}_X in X space for the following original model evaluation, which happens inside a Markov chain generation.
- (3) The third transformation is to transfer N samples of \mathbf{Z} obtained in Markov chains to \mathbf{X} in X space together, which is used for the evaluation of the original model.

Statements about the implementation of subset simulation for readers:

- For any arbitrary distributed inputs, the isoprobabilistic transformation is implemented to transfer the original inputs from X space into independent standard Gaussian Z space before generating states using the MCMC algorithm. Likewise, the samples in Z space should be transferred back into X space before evaluating the physical model.
- If the model inputs are dependent, the dependence needs to be removed during the transformation from X space and Z space, and to be added during the transformation from Z space and X space.
- An acceptance ratio of 0.44 in one-dimensional random walker sampling is optimal for the highest efficiency, which is utilized in the adaptive conditional sampling.
- The adaptive conditional sampling method is recommended in the subset simulation in practice to avoid too high rejection in the high subset level.
- The coefficients of variance of the estimated failure probability using subset simulation is well approximated by $\sum_{m=1}^M \delta_m^2$ in practice, while $\sum_{m_1=1}^M \sum_{m_2=1}^M \delta_{m_1} \delta_{m_2}$ is too conservative but can be taken as the upper bound.

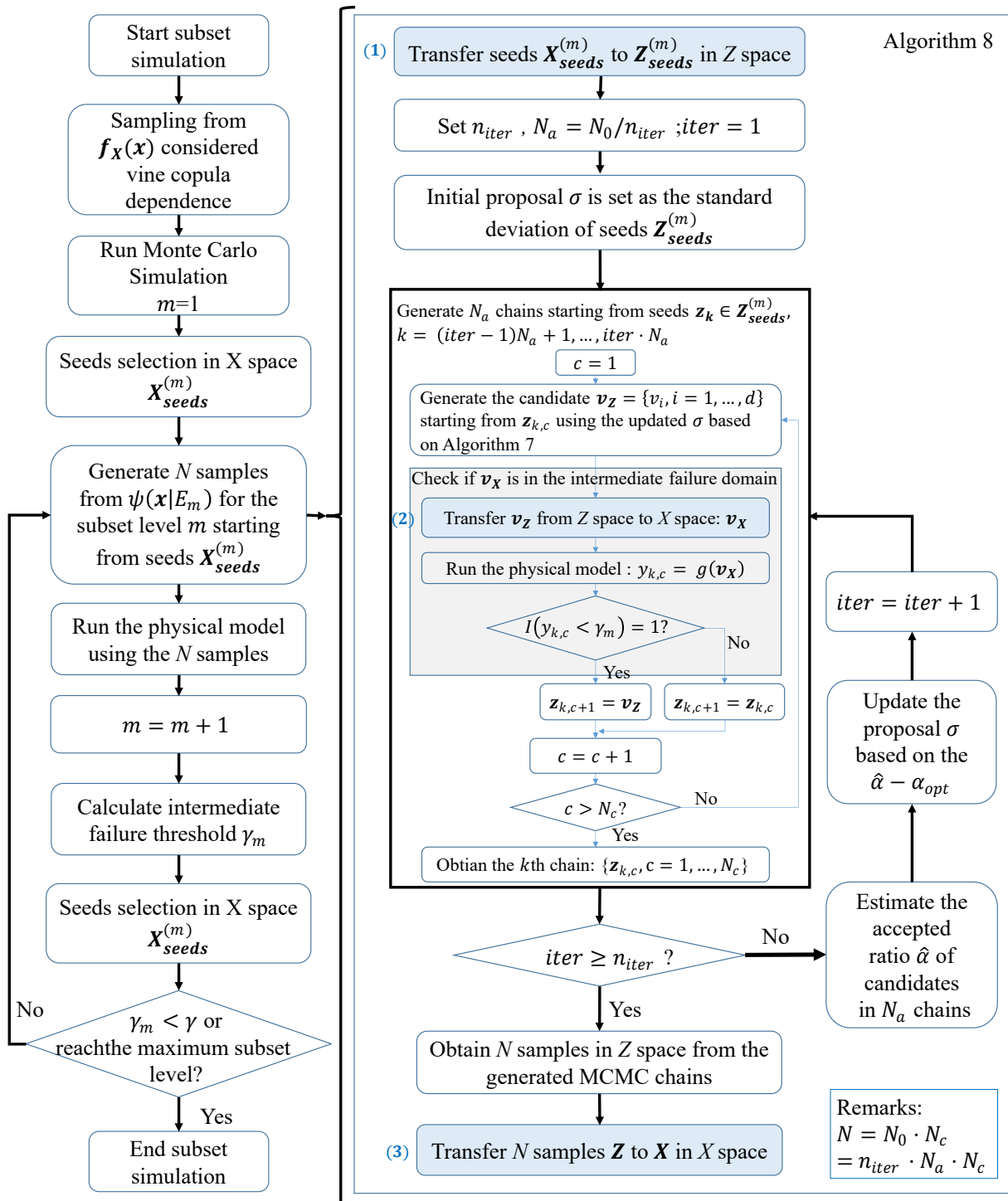


Figure 3.19: Flow chart of subset simulation using the adaptive conditional sampling method considering the dependent inputs, adapted from Fig.4 of [98]

3.4 Global Sensitivity Analysis

Sensitivity analysis is widely used in optimization problems to find the steepest gradient of the cost function, which is called local sensitivity analysis (LSA). In contrast, global sensitivity analysis (GSA), also called uncertainty importance analysis, is to describe output uncertainty over the entire range of the input variables and identify the most critical and essential contributing factors to the output uncertainty. GSA can be as a post-processing step to enhance the quality of the Predictive Analysis framework. Furthermore, GSA allows us to reduce the uncertainty of the output, or failure probability by taking actions on some specific contributing factors. In terms of different interests, four groups of sample-based GSA methods listed in Table 3.1 are described in this section.

Table 3.1: *Four groups of global sensitivity analysis methods*

	Methods	Interpretation
1	Correlation-based	correlations between input and output.
2	Regression-based	sensitivity based on the regression model.
3	Variance-based	sensitivity to the output variance.
4	Moment-independent	sensitivity to the output distribution.
5	Reliability-oriented	sensitivity to the failure probability.

3.4.1 Correlation-based Sensitivity

The correlation coefficients are a statistical dependence measure that indicates the linear relationship between two variables around their expectations. They vary from -1 to 1. A positive value indicates that the variable increases as the other variable increases, and vice versa. The bigger the absolute value of correlation coefficients is, the stronger the relationship of two variables are. For example, in the case of runway overrun, the sample plots of the headwind and the landing mass with the model output stop margin are shown in Figure 3.20. The sample plot shows that the stop margin becomes less along with the decrease of the headwind and the increase of the landing mass. Intuitively, the correlation coefficients between model output and each contributing factor provide a sensitivity measure of each input to the model output. Three kinds of correlation coefficients: Pearson correlation, Spearman correlation, and Kendall's tau, are introduced here.

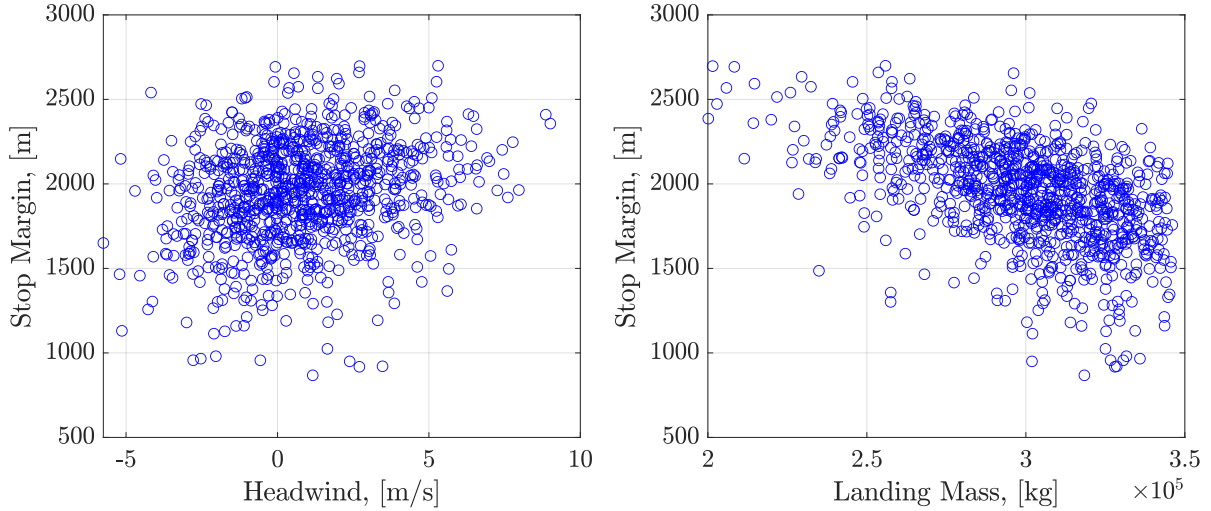


Figure 3.20: Sample plots of MCS in the runway overrun

Pearson Correlation ρ

The Pearson correlation coefficient is also called linear correlation coefficient. The correlation-based sensitivity can be computed using:

$$\rho_i = \rho(X_i, Y) = \frac{E[(X_i - \mu_i)(Y_i - \mu_Y)]}{\sigma_i \sigma_Y}, \quad (3.101)$$

where μ_i and μ_Y are the expectation of the corresponding variable X_i and Y . σ_i and σ_Y are the related standard deviation. Normalized X_i and Y are computed as below:

$$\tilde{X}_i = \frac{X_i - \mu_i}{\sigma_i}, \quad \tilde{Y} = \frac{Y - \mu_Y}{\sigma_Y}. \quad (3.102)$$

The normalized sample plots of headwind and landing mass with stop margin are shown in Figure 3.21. It shows that headwind has a positive effect on stop margin, and landing mass has a negative effect. According to the slope of the red dash line in Figure 3.21, the influence of landing mass is larger than the headwind.

Spearman Correlation ρ_s

For the nonlinear but monotonic relationship, the correlation coefficients can be calculated based on the rank number, which is called Spearman correlation. The raw data can be transferred into the uniform space (U -scale) via the empirical CDF F_i^e as below:

$$X_i^U = F_i^e(X_i), \quad Y^U = F_Y^e(Y), \quad (3.103)$$

Where the F_i^e and F_Y^e are the empirical CDFs of X_i and Y , respectively. Therefore, the Spearman coefficient do not depend on the marginal distributions. The Spearman correlation is computed using:

$$\rho_i^s = \rho_s(X_i, Y) = \rho(F_i^e(X_i), F_Y^e(Y)) = \rho(X_i^U, Y^U). \quad (3.104)$$

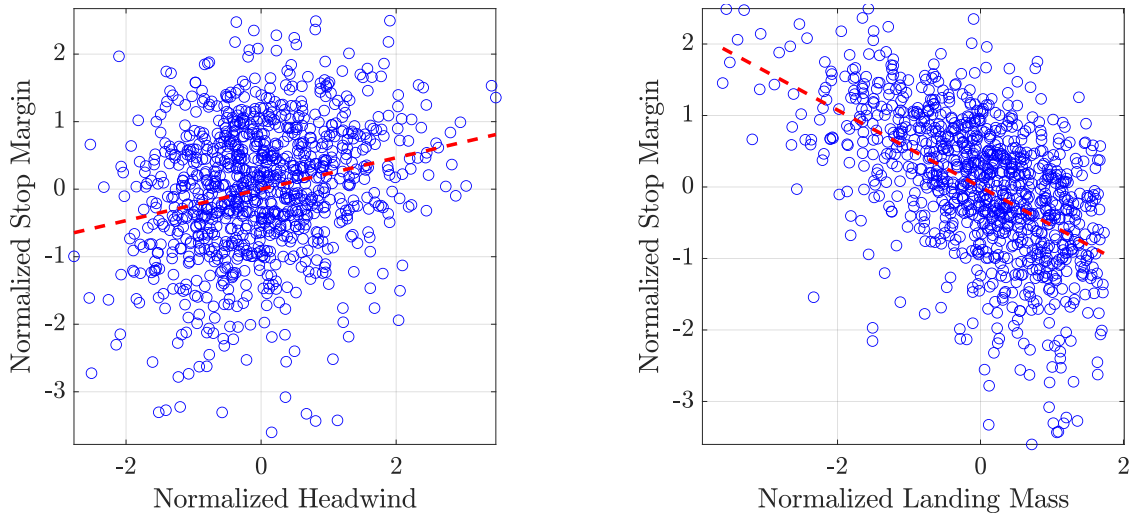


Figure 3.21: Normalized sample plots of MCS in the runway overrun

Kendall's tau τ

Kendall's tau is another rank-based correlation coefficient, and it can be used to measure the nonlinear and non-monotonic correlation. It is defined as the concordance probability minus the discordance probability between two variables. According to the definition in [28], the Kendall's tau of two continue variables X_i and Y is defined based on the two independent pairs of variables $(X_{i,1}, Y_1)$ and $(X_{i,2}, Y_2)$ shown as below:

$$\tau_i = Pr[(X_{i,1} - X_{i,2}) \cdot (Y_1 - Y_2) > 0] - Pr[(X_{i,1} - X_{i,2}) \cdot (Y_1 - Y_2) < 0], \quad (3.105)$$

where $Pr[(X_{i,1} - X_{i,2}) \cdot (Y_1 - Y_2) > 0]$ is the probability of concordance, which indicates the positive correlation between X_i and Y . In contrast, $Pr[(X_{i,1} - X_{i,2}) \cdot (Y_1 - Y_2) < 0]$ is the probability of discordance, which represents the negative correlation between X_i and Y . For the two independent random variables with continuous distributions, the Kendall's tau is 0.

3.4.2 Regression-based Sensitivity

Apart from the correlations, the standard regression coefficient (SRC) is another sensitivity measure. This approach simplifies the model as a linear model. Furthermore, It decomposes the total variance of the output into the individual effect of each contributing factor. Therefore, it only works for independent input variables. The original model $g(\mathbf{X})$ is approximated by a linear regression:

$$Y = g(\mathbf{X}) \approx \beta_0 + \sum_{i=1}^d \beta_i X_i, \quad (3.106)$$

where β_i is the coefficient, which is computed based on the lease-square method as below:

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{Y}, \quad (3.107)$$

where \mathbf{X} and \mathbf{Y} are samples array of input and output, respectively. According to Equation (3.106), the variance of the output are estimated by:

$$\hat{\sigma}_Y^2 = \sum_{i=1}^d \beta_i^2 \sigma_{X_i}^2, \quad (3.108)$$

where σ_{X_i} and $\hat{\sigma}_Y$ are the standard deviations of the input X_i and output Y , respectively. The SRC index is defined as:

$$SRC_i = \frac{\hat{\beta}_i \sigma_i}{\sigma_Y}. \quad (3.109)$$

To deal with the nonlinear but monotonic relations, the samples array of input and output are transferred into the uniform space via empirical CDFs, denoted by \mathbf{X}^U and \mathbf{Y}^U . The standard rank regression coefficients (SRRC) are computed as below:

$$\hat{\gamma} = ((\mathbf{X}^U)^T \mathbf{X}^U)^{-1} (\mathbf{X}^U)^T \mathbf{Y}^U, \quad (3.110)$$

$$SRRC_i = \frac{\hat{\gamma}_i \sigma_i^U}{\sigma_Y^U} = \hat{\gamma}_i. \quad (3.111)$$

where σ_i^U and σ_Y^U denote the standard deviations of the input X_i^U and output Y^U , respectively. Since both X_i^U and Y^U are uniformly distributed and follow $\mathcal{U}(0, 1)$, $\sigma_i^U = \sigma_Y^U$ and they are canceled out in above equation.

3.4.3 Variance-based Sensitivity

Variance, as an important measure of uncertainties, is also used to calculate global sensitivity. Assuming the original model can be represented by some additive terms, the total variance of the model output can be described using the sum of the variance of each term. The variance decomposition is only valid for independent inputs, and it is usually referred to as the analysis of variance (ANOVA). Based on the law of total variance [104], the variance can be decomposed as follows:

$$Var(Y) = \mathbb{E}_{X_i}[Var(Y|X_i)] + Var_{X_i}(\mathbb{E}[Y|X_i]), \quad (3.112)$$

where $Var(Y)$ is the total variance of the model output Y , and $\mathbb{E}_{X_i}[Var(Y|X_i)]$ is the conditional expected value of $Var(Y)$ given X_i . $Var_{X_i}(\mathbb{E}[Y|X_i])$ can be interpreted as the variance reduction of the output Y conditioned on a fixed input X_i . Instead of conditioning on the specific value of X_i , a class of samples is used to compute the conditional expectation and the conditional variance in practice. An example is given for visualization of the conditional expectation and variance. The range of landing mass is split into 10 classes $C_j, j = 1, \dots, 10$ based on the quantile lines of landing mass samples as shown in Figure 3.22. The variation of 10 estimations of $\mathbb{E}[Y|X_i \in C_j]$ in Figure 3.22 indicates that the stop margin is affected by the class C_j that X_i belongs to. Therefore, $Var_{X_i}(\mathbb{E}[Y|X_i])$ can be defined as a sensitivity measure to describe the influence of this input to the output.

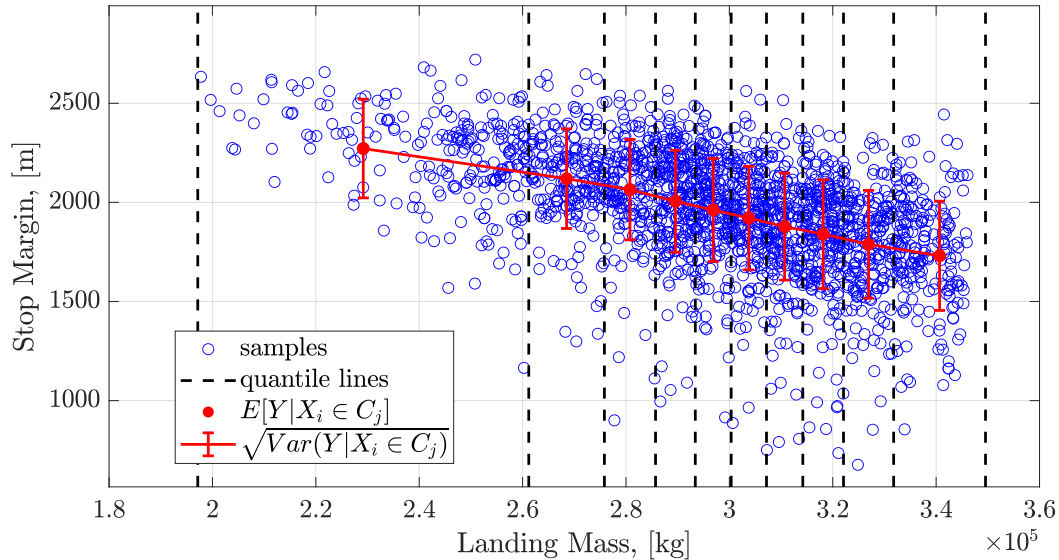


Figure 3.22: *Quantile-based samples plot of the landing mass and the stop margin*

As mentioned in Section 3.4.2, SRC is also one approach of variance decomposition, but only valid for the linear model. A complex Sobol's variance decomposition approach is applied for the nonlinear model. The normalized $Var_{X_i}(\mathbb{E}[Y|X_i])$ is taken as the sensitivity indices, called the Sobol's indices. The definition and computation of the Sobol's indices are shown in the following sections.

Sobol's Indices

The original model $g(\mathbf{X})$ is decomposed into effect functions with different dimensions via the Sobol's decomposition as below [105, 106]:

$$y = g(x_1, \dots, x_d) = g_0 + \sum_{i=1}^d g_i(x_i) + \sum_{1 \leq i < j \leq d} g_{ij}(x_i, x_j) + \dots + g_{1,2,\dots,d}(x_1, x_2, \dots, x_d), \quad (3.113)$$

with two conditions hold. The first is that g_0 is the expectation of the model output: $g_0 = \mathbb{E}[g(\mathbf{X})]$. The second condition is that the integral of the functions g_{i_1,\dots,i_s} with respect to their own variables are null, i.e.,

$$\int_{D_{i_k}} g_{i_1,\dots,i_s}(x_{i_1}, \dots, x_{i_k}, \dots, x_{i_s}) f_i(x_{i_k}) dx_{i_k} = 0, \quad x_{i_k} \in D_{i_k} \quad \text{and} \quad 1 \leq k \leq s, \quad (3.114)$$

where $f_i(x_{i_k})$ denotes the PDF with respect to x_{i_k} . All terms of the decomposition can be calculated using the recursive equations. Obviously, g_0 is computed as below:

$$g_0 = \int g(\mathbf{x}) f_{\mathbf{x}}(\mathbf{x}) d\mathbf{x}, \quad (3.115)$$

where $f_{\mathbf{x}}(\mathbf{x})$ is the joint PDF of all variables \mathbf{x} . Let $\mathbf{x}_{\sim i}$ denote the \mathbf{x} without x_i . By integrating Equation (3.113) with respect to $\mathbf{x}_{\sim i}$, g_i is obtained based on the second condition in Equation (3.114):

$$\int g(\mathbf{x}) f_{\mathbf{x}_{\sim i}}(\mathbf{x}_{\sim i}) d\mathbf{x}_{\sim i} = g_0 + g_i, \quad (3.116)$$

where $f_{\mathbf{x}_{\sim i}}(\mathbf{x}_{\sim i})$ denotes the joint PDF of variables \mathbf{x} without x_i . In addition, the left part of the equation above equals $\mathbb{E}[Y|X_i]$ based on the definition of expectation, since

$$\begin{aligned}\mathbb{E}[Y|X_i] &= \int g(\mathbf{x}) f_{\mathbf{x}|x_i}(\mathbf{x}|x_i) d\mathbf{x}_{\sim i} = \int g(\mathbf{x}) \frac{f_{\mathbf{x}}(\mathbf{x})}{f_{x_i}(x_i)} d\mathbf{x}_{\sim i} \\ &= \int g(\mathbf{x}) \frac{\prod_{j=1}^d f(x_j)}{f(x_i)} d\mathbf{x}_{\sim i} = \int g(\mathbf{x}) f_{\mathbf{x}_{\sim i}}(\mathbf{x}_{\sim i}) d\mathbf{x}_{\sim i}.\end{aligned}\quad (3.117)$$

Therefore,

$$g_i = \int g(\mathbf{x}) f_{\mathbf{x}_{\sim i}}(\mathbf{x}_{\sim i}) d\mathbf{x}_{\sim i} - g_0 = \mathbb{E}[Y|X_i] - g_0. \quad (3.118)$$

Let $\mathbf{x}_{\sim ij}$ denote the \mathbf{x} without x_i and x_j . By integrating Equation (3.113) with respect to $\mathbf{x}_{\sim ij}$, g_{ij} is explicitly expressed as below:

$$\begin{aligned}g_{ij} &= \int \dots \int g(\mathbf{x}) f_{\mathbf{x}_{\sim ij}}(\mathbf{x}_{\sim ij}) d\mathbf{x}_{\sim ij} - g_0 - g_i(x_i) - g_j(x_j) \\ &= \mathbb{E}[Y|X_i, X_j] - g_0 - g_i - g_j.\end{aligned}\quad (3.119)$$

Computing the variance of Equation (3.113) in both sides, the variance of the $g(\mathbf{x})$ is written down:

$$Var(g(\mathbf{x})) = \sum_i^d V_i + \sum_i \sum_j V_{ij} + \dots + V_{1,2,\dots,d}, \quad (3.120)$$

where V_i and V_{ij} denote the first and second order variance, respectively. They are computed as follows:

$$V_i = Var(g_i) = Var(\mathbb{E}[Y|X_i]) - Var(g_0) = Var(\mathbb{E}[Y|X_i]). \quad (3.121)$$

$$\begin{aligned}V_{ij} &= Var(g_{ij}) = Var(\mathbb{E}[Y|X_i X_j]) - Var(g_0) - Var(g_i) - Var(g_j) \\ &= Var(\mathbb{E}[Y|X_i X_j]) - V_i - V_j.\end{aligned}\quad (3.122)$$

The sensitivity indices are defined as the ratio of each term's variance to the total variance:

$$1 = \sum_i^d S_i^{(1)} + \sum_i \sum_j S_{ij}^{(2)} + \sum_i \sum_j \sum_k S_{ijk}^{(3)} + \dots + S_{1,2,\dots,d}^{(d)}. \quad (3.123)$$

$S_i^{(1)}$ denotes the first order Sobol's index, which describes the main effect of one input and is computed as below:

$$S_i^{(1)} = \frac{V_i}{Var(Y)} = \frac{Var(\mathbb{E}[Y|X_i])}{Var(Y)}. \quad (3.124)$$

The second order Sobol's index $S_{ij}^{(2)}$ indicates the interaction effect of two inputs as follows:

$$S_{ij}^{(2)} = \frac{V_{ij}}{Var(Y)} = \frac{Var(\mathbb{E}[Y|X_i X_j]) - V_i - V_j}{Var(Y)} = \frac{Var(\mathbb{E}[Y|X_i X_j])}{Var(Y)} - S_i^{(1)} - S_j^{(1)}. \quad (3.125)$$

Furthermore, the total order Sobol' index of X_i includes the main effect of the X_i and the interaction effect of X_i with other inputs. It is computed via:

$$S_i^T = 1 - S_{\sim i} = 1 - \frac{Var(\mathbb{E}[Y|X_{\sim i}])}{Var(Y)}. \quad (3.126)$$

The variances described in Equations (3.124-3.126) can be computed by the Monte Carlo simulation (MCS). The algorithm is shown in the next section.

Monte Carlo-based Estimation of Sobol's Indices

Let \mathbf{X}_v denotes the conditioning variables. The calculation of the variance of conditional expectation $Var(\mathbb{E}[Y|\mathbf{X}_v])$ is the challenge part for Sobol's indices. Suppose that the unconditioned and conditional random variables Y and $Y|\mathbf{X}_v$ are square-integrable. Janon [107] proved that the variance of expectation is equivalent to the covariance as below:

$$Var(\mathbb{E}[Y|\mathbf{X}_v]) = Cov(Y, Y|\mathbf{X}_v), \tag{3.127}$$

which is referred to the estimator of the so-called closed index S_v proposed by Saltelli and Homma [108]. The S_v is computed using the samples via:

$$S_v = \frac{Cov(Y, Y|\mathbf{X}_v)}{Var(Y)} = \frac{Cov(\mathbf{Y}, \mathbf{Y}_v)}{Var(\mathbf{Y})} = \frac{\frac{1}{N} \sum y_j y_j^v - (\frac{1}{N} \sum y_j)(\frac{1}{N} \sum y_j^v)}{\frac{1}{N} \sum y_j^2 - (\frac{1}{N} \sum y_j)^2}, \tag{3.128}$$

where \mathbf{Y} and \mathbf{Y}_v denote the unconditioned output samples array and conditional output samples array. $y_j \in \mathbf{Y}$, and $y_j^v \in \mathbf{Y}_v$ with $j = 1, \dots, N$. The first or higher order Sobol's indices are computed based on the S_v by selecting the conditioning variables \mathbf{X}_v .

Sufficient samples are required for the convergence of the MCS estimator \hat{S}_v . In order to improve the efficiency of the algorithm, a special experiment design of sample array is proposed in [108] to reuse the evaluation results and reduce the total number of evaluations. The sample array design for the first order Sobol' indices calculation is demonstrated using a case of 3-dimensional input.

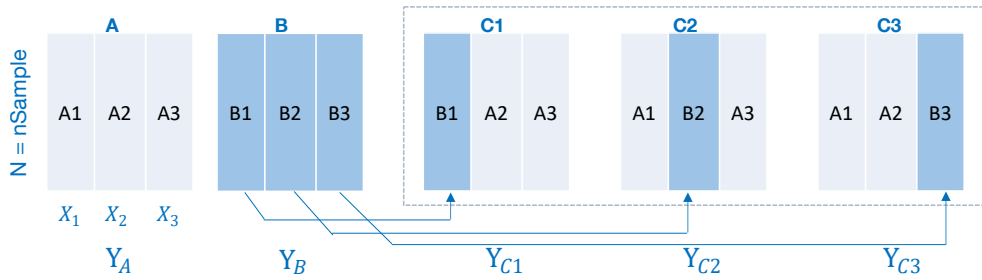


Figure 3.23: Designed samples to calculate $S_i^{(1)}$ and the $S_i^{(T)}$ in the case of 3 inputs.

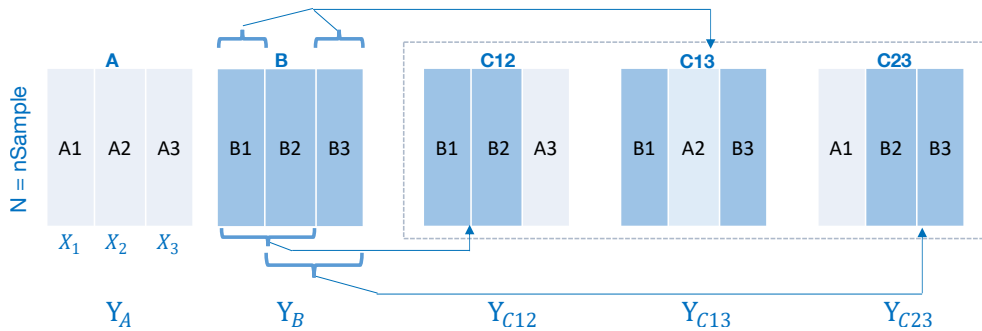


Figure 3.24: Designed samples to calculate $S_{ij}^{(2)}$ in the case of 3 inputs.

As shown in Figure 3.23, \mathbf{A} and \mathbf{B} denote the sample array of the input, while $\mathbf{Y}_\mathbf{A}$ and $\mathbf{Y}_\mathbf{B}$ are the corresponding output, respectively. The sample array \mathbf{C}_1 , \mathbf{C}_2 and \mathbf{C}_3 are obtained by exchanging the components of the samples in \mathbf{A} and \mathbf{B} . Thus, $\mathbf{Y}_\mathbf{B}$ and $\mathbf{Y}_{\mathbf{C}_1}$ can be used to indicate the unconditioned output \mathbf{Y} and conditional output $\mathbf{Y}|X_1$, respectively. Based on Equation (3.128), the first order Sobol index S_i can be computed using $\mathbf{Y}_\mathbf{B}$ and $\mathbf{Y}_{\mathbf{C}_1}$. A similar strategy is also demonstrated for the second order Sobol index in Figure 3.24. The procedure for Sobol's indices calculation is described in Algorithm 9.

Algorithm 9 Procedure of calculating Sobol's indices

- 1: Set the number of the samples N
 - 2: Generate $2N$ samples from $f_{\mathbf{X}}(\mathbf{x})$; Obtain samples matrix \mathbf{A} and \mathbf{B}
 - 3: **if** compute $S_i^{(1)}$ or S_i^T **then**
 - 4: Exchange each column of \mathbf{A} using the corresponding column of \mathbf{B} as shown in Figure 3.23; Obtain the samples matrices: $\mathbf{C}_1, \dots, \mathbf{C}_d$
 - 5: Evaluate the model using samples of \mathbf{A} , \mathbf{B} , $\mathbf{C}_1, \dots, \mathbf{C}_d$ and obtain the output response $\mathbf{Y}_\mathbf{A}$, $\mathbf{Y}_\mathbf{B}$, $\mathbf{Y}_{\mathbf{C}_1}, \dots, \mathbf{Y}_{\mathbf{C}_d}$, respectively
 - 6: $S_i^{(1)} = Cov(\mathbf{Y}_\mathbf{B}, \mathbf{Y}_{\mathbf{C}_i}) / Var(\mathbf{Y}_\mathbf{B})$ using Equation (3.128)
 - 7: $S_i^T = 1 - S_{\sim i}^{(1)} = 1 - Cov(\mathbf{Y}_\mathbf{A}, \mathbf{Y}_{\mathbf{C}_i}) / Var(\mathbf{Y}_\mathbf{A})$ using Equation (3.128)
 - 8: **if** compute $S_{ij}^{(2)}$ **then**
 - 9: Exchange each two columns of \mathbf{A} using the corresponding columns of \mathbf{B} as shown in Figure 3.24; Obtain $d(d-1)/2$ samples matrix: $\mathbf{C}_{1,2}, \dots, \mathbf{C}_{d-1,d}$
 - 10: Evaluate the model using samples of $\mathbf{C}_{1,2}, \dots, \mathbf{C}_{d-1,d}$ and obtain the output response $\mathbf{Y}_{\mathbf{C}_{1,2}}, \dots, \mathbf{Y}_{\mathbf{C}_{d-1,d}}$
 - 11: $S_{ij}^{(2)} = Cov(\mathbf{Y}_\mathbf{B}, \mathbf{Y}_{\mathbf{C}_{ij}}) / Var(\mathbf{Y}_\mathbf{A}) - S_i^{(1)} - S_j^{(1)}$ using Equation (3.128)
 - 12: **end if**
 - 13: **end if**
-

Based on the procedure in Algorithm 9, the number of evaluations to calculate the $S_i^{(1)}$ is $(d+2) \times N$, and S_i^T can be subsequently obtained without more evaluations. To calculate the k_{th} order Sobol's indices, the number of required evaluations is $\binom{d}{k} \times N$. The first two Sobol's indices are widely used in practice. Therefore, the total number of evaluations to calculate $S_i^{(1)}$, S_i^T , and $S_{ij}^{(2)}$ is

$$(d+2) \times N + \binom{d}{2} \times N = \frac{d(d+1)+4}{2} \times N, \quad (3.129)$$

where N is the predefined number of samples in sample matrix \mathbf{A} , and d denotes the dimensions.

3.4.4 Moment-independent Sensitivity

Instead of only considering the variance of the conditional expectation $\mathbb{E}[Y|\mathbf{X}_v]$, the distribution changes of the conditional output $Y|\mathbf{X}_v$ can be also measured to evaluate the sensitivities of contributing factors. Based on the sample plots in Figure 3.22, the kernel estimated conditional PDFs in each class C_j are shown in Figure 3.25. The conditional PDF of stop margin changes along with the landing mass.

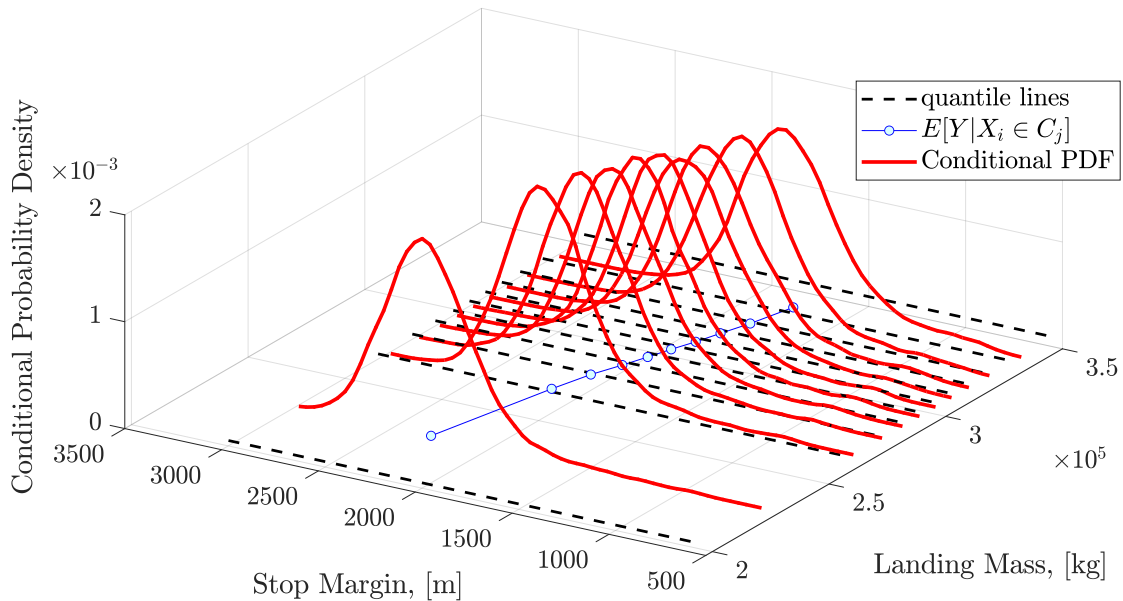


Figure 3.25: *Estimated conditional PDF of stop margin*

The moment-independent sensitivity indices, so-called Borgonovo indices [37], measure the expected shift of the probability distribution of the output when one of the input is constant. The deviation between the original distribution of the output and the conditional distribution of the output given a fixed X_i measures how the input influence the output distribution. The Borgonovo indices are defined as the average \mathcal{L}_1 norm of conditional PDF deviation:

$$\delta_i = \frac{1}{2} \mathbb{E}_{X_i} \left[\int_{D_y} |f_Y(y) - f_{Y|X_i}(y)| dy \right] = \frac{1}{2} \int_{D_{x_i}} \left[\int_{D_y} |f_Y(y) - f_{Y|X_i}(y)| dy \right] f_i(x_i) dx_i, \quad (3.130)$$

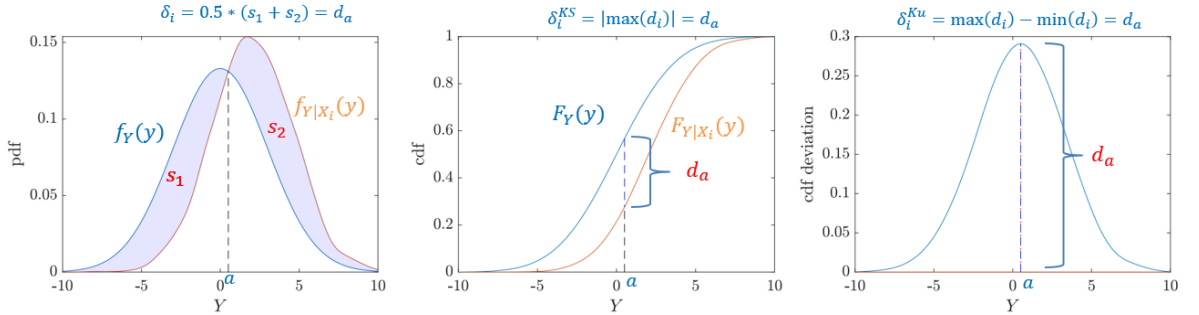
where D_{x_i} and D_y are the range of the X_i and Y , respectively. The deviation between the two PDFs is in the range of $[0, 2]$. By normalizing the deviation using $1/2$, $0 \leq \delta_i \leq 1$. In addition, other metrics to measure the difference between two distributions can also be used. For example, sensitivities based on the Kolmogorov-Smirnov distance are defined as the average \mathcal{L}_∞ norm of conditional PDF deviation as below:

$$\delta_i^{KS} = \mathbb{E}_{X_i} \left[\max_y |F_Y(y) - F_{Y|X_i}(y)| \right]. \quad (3.131)$$

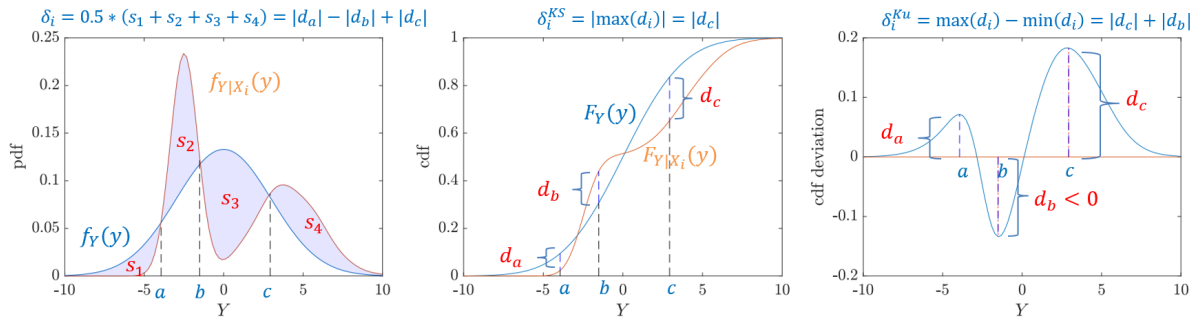
Furthermore, another sensitivities based on the Kuiper distance are defined as the average range of the conditional CDF deviation by:

$$\delta_i^{Ku} = \mathbb{E}_{X_i} \left[\max_y \left(F_Y(y) - F_{Y|X_i}(y) \right) - \min_y \left(F_Y(y) - F_{Y|X_i}(y) \right) \right], \quad (3.132)$$

where δ_i, δ_i^{KS} , and $\delta_i^{Ku} \in [0, 1]$. $\delta_i = 0, \delta_i^{KS} = 0$, or $\delta_i^{Ku} = 0$ all imply that X_i and Y are independent. According to the intersection points of the original PDF and conditional PDF of the output, the relationship of three sensitivity metrics is illustrated in Figure 3.26. The original distribution is assumed to be a unimodal distribution in the demonstration. When the conditional distribution is an unimodal distribution shown in Figure 3.26(a), only one intersection between two PDFs. Therefore, the three sensitivities are equivalent each other: $\delta_i = \delta_i^{KS} = \delta_i^{Ku}$. For a multimodal conditional distribution, the three metrics are not equivalent anymore. However, δ_i^{KS} is always less than δ_i^{Ku} for multiple intersection cases.



(a) One intersection point between the original PDF and the conditional PDF



(b) Multiple intersection points between the original PDF and the conditional PDF.

Figure 3.26: Relationship of three moment-independent sensitivity metrics

3.4.5 Reliability-oriented Sensitivity

All global sensitivity methods mentioned above are applied to measure the sensitivity to the model output. Since the incident failure probability is our concern, a sensitivity of contributing factors to the failure probability is worth studying. The reliability-oriented sensitivity metrics are defined to measure the averaging shift of the failure probability

for a given X_i . P_F is the probability of failure, and $P_{F|X_i}$ denotes the conditional failure probability for a given X_i . Based on the definition of \mathcal{L}_1 norm, the metric is shown as below:

$$\mathcal{L}_1(\Delta P_{F|X_i}) : \eta_i^P = \mathbb{E}_{X_i} [|P_F - P_{F|X_i}|] = \int_{D_{x_i}} |P_F - P_{F|X_i}| f_{X_i}(x_i) dx_i, \quad (3.133)$$

where $f_{X_i}(x_i)$ indicates the original distribution of X_i . D_{x_i} denotes the range of X_i . According to the Bayesian rule, $P_{F|X_i}$ can be rewritten as below [109]:

$$\begin{aligned} P_{F|X_i} &= Pr(F|X_i = x_i) = \frac{Pr(\{X_i = x_i\} \cap F)}{Pr(X_i = x_i)} = \frac{Pr(X_i = x_i|F)Pr(F)}{Pr(X_i = x_i)} \\ &= \frac{f_{X_i|F}(x_i)dx_i P_F}{f_{X_i}(x_i)dx_i} = \frac{f_{X_i|F}(x_i)P_F}{f_{X_i}(x_i)}, \end{aligned} \quad (3.134)$$

where $f_{X_i|F}(x_i)$ denotes the failure-conditioned distribution. The \mathcal{L}_1 -based sensitivity index η_i^P in Equation (3.133) can be further derived by utilizing Equation (3.134) as follows:

$$\eta_i^P = \int_{D_{x_i}} \left| P_F - \frac{f_{X_i|F}(x_i)P_F}{f_{X_i}(x_i)} \right| f_{X_i}(x_i) dx_i = P_F \int_{D_{x_i}} |f_{X_i}(x_i) - f_{X_i|F}(x_i)| dx_i. \quad (3.135)$$

Some researchers [110] also proposed to use the $\eta_i^P/(2P_F)$ as the sensitivity index as below:

$$\eta_i^f = \frac{\eta_i^P}{2P_F} = \frac{1}{2} \int_{D_{x_i}} |f_{X_i}(x_i) - f_{X_i|F}(x_i)| dx_i, \quad (3.136)$$

where this index describes the deviation between the original PDF of X_i and failure-conditioned PDF of $X_i|F$. Since the absolute deviation of two PDFs is from 0 to 2, the index η_i^f has a property that $\eta_i^f \in [0, 1]$. The mentioned \mathcal{L}_1 -based reliability sensitivity indices and the modified \mathcal{L}_1 -based indices are summarized in Table 3.2.

Table 3.2: *Reliability-oriented Sensitivity Metrics*

Measures	Definition	Required estimation
1 $\mathcal{L}_1(\Delta P_{F X_i})$	$\eta_i^P = \mathbb{E}_{X_i} [P_F - P_{F X_i}]$	$P_F, f_{X_i}(x_i) - f_{X_i F}(x_i) $
2 \mathcal{L}_1 -modified	$\eta_i^f = \frac{1}{2} \int_{D_{x_i}} f_{X_i}(x_i) - f_{X_i F}(x_i) dx_i$	$ f_{X_i}(x_i) - f_{X_i F}(x_i) $

So far, $f_{X_i|F}(x_i)$ is required for the calculation of two mentioned sensitivity indices. According to Section 3.3, failure samples can be obtained based on SuS. Therefore, the density function $f_{X_i|F}(x_i)$ can be directly estimated based on the obtained failure samples. Kernel distribution is utilized to allow the estimated PDF to match the histogram of failure samples better. The subtraction between $f_{X_i}(x_i)$ and $f_{X_i|F}(x_i)$ can be computed easily based on the estimated Kernel distributions. \mathcal{L}_1 -modified index η_i^f is used in the analysis of reliability sensitivity with the property $0 \leq \eta_i^f \leq 1$. Furthermore, the intermediate failure samples are also obtained after SuS. For example, histograms of headwind samples at different subset levels are shown in Figure 3.27.

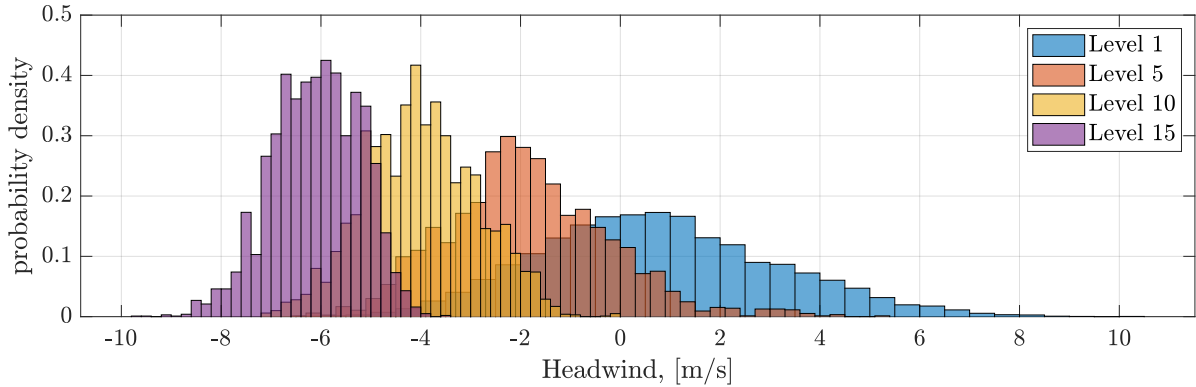


Figure 3.27: Histograms of intermediate failure samples of headwind in runway overrun

Based on the intermediate event E_m and intermediate failure samples in SuS, we can also measure the trend of the sensitivity along with the decrease of the intermediate threshold γ_m . Based on Equation (3.136), the intermediate indices are computed by:

$$\eta_i^f(\gamma_m) = \frac{1}{2} \int_{D_{x_i}} |f_{X_i}(x_i) - f_{X_i|E_m}(x_i)| dx_i, \quad (3.137)$$

where $f_{X_i|E_m}(x_i)$ is the conditioned PDF and estimated based on the intermediate samples. This trend of sensitivities illustrates how the distribution of contributing factors vary when generated samples are closing to the failure domain.

3.5 Polynomial Chaos Expansion

As one of the surrogate model, Polynomial Chaos Expansion (PCE) is used to speed up the uncertainty propagation by replacing costly function calls in the MCS using calls of the surrogate model, especially for the computationally expensive model. The PCE model is applied in ‘Calibrate’ step of Predictive Analysis framework, which is discussed in detail in Chapter 4. This section introduces the mathematical fundamentals of PCE.

3.5.1 PCE Model

For any square-integrable model $Y = g(\mathbf{X})$, the PCE model can represent the original model by a sum of a series of orthonormal polynomials as below [42]:

$$Y = \sum_{\alpha} k_{\alpha} \Psi_{\alpha}(\mathbf{Z}), \quad (3.138)$$

where \mathbf{Z} is the independent random variables following the distribution PDF $f_{\mathbf{Z}}(\mathbf{z})$. k_{α} denotes the PCE coefficients, and $\Psi_{\alpha}(\mathbf{Z})$ is the basis of multivariate orthogonal polynomials with respect to $f_{\mathbf{Z}}(\mathbf{z})$. Furthermore, a mapping relation is required to transfer the original input variables \mathbf{X} to the independent variables \mathbf{Z} . \mathbf{X} follows the distribution

with the PDF $f_{\mathbf{X}}(\mathbf{x})$. For the independent input variables \mathbf{X} , the mapping relation is described using the simple algebra calculation if the relation between \mathbf{X} and \mathbf{Z} is obvious and described easily, such as Equation (3.158) in the given example. In case that the relation between \mathbf{X} and \mathbf{Z} might be not easy to describe explicitly or the \mathbf{X} might be dependent, the isoprobabilistic transformation \mathbf{T} mentioned in Equation (3.98) and Equation (3.99) can be applied. Compared to the transformation \mathbf{T} used in the subset simulation, each component of \mathbf{Z} in the PCE model can follow any distribution, not only the standard Gaussian distribution. In the following context, the univariate orthogonal polynomials are derived first, subsequently extended to multivariate orthogonal polynomials.

Univariate Orthogonal Polynomials

The inner product of two univariate orthogonal polynomials with respect to the corresponding marginal density $f_Z(z)$ is defined as [41]:

$$\langle P_j(z), P_k(z) \rangle = \int_{D_z} P_j(z)P_k(z)f_Z(z)dz = \gamma_j\delta_{jk}, \quad z \in D_z, \quad (3.139)$$

where P_j and P_k denote the monic polynomials like $P_n(z) = z^n + a_{n-1}z^{n-1} + \dots + a_1z + a_0$. $a_{(\cdot)}$ is the polynomial coefficients of the $P_n(z)$. δ_{jk} is the Kronecker delta function. δ_{jk} is 1, if $j = k$, and 0 otherwise. γ_j is a constant value. A family of monic orthogonal polynomials $P_n(z)$ can be built based on a 3-term recurrence relation [42]:

$$P_{n+1}(z) = (z - \alpha_n)P_n(z) - \beta_n P_{n-1}(z), \quad (3.140)$$

where α_n and β_n are computed by

$$\alpha_n = \frac{\langle zP_n(z), P_n(z) \rangle}{\langle P_n(z), P_n(z) \rangle}, \quad \beta_n = \frac{\langle P_n(z), P_n(z) \rangle}{\langle P_{n-1}(z), P_{n-1}(z) \rangle}. \quad (3.141)$$

Normally, P_0 is set to 1 and P_1 is set to z . The constructed polynomials $P_n(z)$ are orthogonal but not orthonormal, since

$$\|P_n\|^2 = \int_{D_Z} P_n^2 f_Z(z)dz = \gamma_n \neq 1. \quad (3.142)$$

Therefore, the orthonormal polynomials are defined by

$$\Psi_n(z) = \frac{P_n(z)}{\|P_n\|} = \frac{P_n(z)}{\sqrt{\gamma_n}}. \quad (3.143)$$

Obviously, $\beta_n = \gamma_n/\gamma_{n-1}$. By substituting $P(z)$ in Equation (3.140) using $\Psi(z)$ in Equation (3.143), the corresponding 3-term recurrence relation becomes:

$$\begin{aligned} \Psi_{n+1}(z)\sqrt{\gamma_{n+1}} &= (z - \alpha_n)\Psi_n(z)\sqrt{\gamma_n} - \beta_n\Psi_{n-1}(z)\sqrt{\gamma_{n-1}} \\ \Rightarrow \Psi_{n+1}(z)\frac{\sqrt{\gamma_{n+1}}}{\sqrt{\gamma_n}} &= (z - \alpha_n)\Psi_n(z) - \frac{\gamma_n}{\gamma_{n-1}}\Psi_{n-1}(z)\frac{\sqrt{\gamma_{n-1}}}{\sqrt{\gamma_n}} \\ \Rightarrow \sqrt{\beta_{n+1}}\Psi_{n+1}(z) &= (z - \alpha_n)\Psi_n(z) - \sqrt{\beta_n}\Psi_{n-1}(z). \end{aligned} \quad (3.144)$$

Based on Equation (3.141), $\alpha_n = \langle z\Psi_n(z), \Psi_n(z) \rangle$ and $\beta_n = \gamma_n/\gamma_{n-1}$. Therefore, γ_n is requested to compute β_n . Due to the orthogonality of $\Psi_n(z)$,

$$\mathbb{E}[\Psi_j(Z)\Psi_k(Z)] = \int_{D_z} \Psi_j(z)\Psi_k(z)f_Z(z)dz = \delta_{jk}. \quad (3.145)$$

The first term of polynomials is usually set to 1. Therefore,

$$\mathbb{E}[\Psi_j(Z)] = \begin{cases} 1, & \text{when } j = 0 \\ 0, & \text{when } j \neq 0 \end{cases} \quad \text{and} \quad \mathbb{E}[\Psi_j(Z)\Psi_k(Z)] = \begin{cases} 1, & \text{when } j = k \\ 0, & \text{when } j \neq k \end{cases}. \quad (3.146)$$

Based on the distribution family of Z , different orthogonal polynomials have been defined [41]. The well-known Legendre and Hermite polynomials are illustrated here.

Legendre polynomials When Z follows the uniform distribution $\mathcal{U}[-1, 1]$, the non-monic Legendre polynomials are shown by the 3-term recurrence relation:

$$\begin{aligned} L_0(z) &= 1; \quad L_1(z) = z; \\ (n+1)L_{n+1}(z) &= (2n+1)zL_n(z) - nL_{n-1}(z). \end{aligned} \quad (3.147)$$

The first 5 terms of the orthonormal Legendre polynomials are listed in Table 3.3. The norm of n -th Legendre polynomials is as below:

$$\|L_n\|^2 = \int_{-1}^1 L_n^2(z) \frac{1}{2} dz = \frac{1}{2n+1}. \quad (3.148)$$

Therefore, the orthonormal Legendre polynomials are $\tilde{L}_n = \sqrt{2n+1}L_n$.

Table 3.3: *The orthonormal Legendre polynomial basis, source: [42]*

n	L_n	$\ L_n\ ^2$	\tilde{L}_n
0	1	1	1
1	z	1/3	$\sqrt{3}L_1$
2	$\frac{1}{2}(3z^2 - 1)$	1/5	$\sqrt{5}L_2$
3	$\frac{1}{2}(5z^3 - 3z)$	1/7	$\sqrt{7}L_3$
4	$\frac{1}{8}(35z^4 - 30z^2 + 3)$	1/9	$\sqrt{9}L_4$
5	$\frac{1}{8}(63z^5 - 70z^3 + 15z)$	1/11	$\sqrt{11}L_5$

Hermite polynomials When Z follows the standard normal distribution $\mathcal{N}(0, 1)$, the non-monic Hermite polynomials are shown by the 3-term recurrence relation:

$$\begin{aligned} H_0(z) &= 1; \quad H_1(z) = z; \\ H_{n+1}(z) &= zH_n(z) - nH_{n-1}(z). \end{aligned} \quad (3.149)$$

The first 5 terms of the orthonormal Hermite polynomials are listed in the Table 3.4. The norm of n -th Hermite polynomial can be obtained by

$$\|H_n\|^2 = \int_{-\infty}^{+\infty} H_n^2(z) \frac{1}{\sqrt{2\pi}} e^{-\frac{z^2}{2}} dz = n!. \quad (3.150)$$

The orthonormal Hermite polynomials are $\widetilde{H}_n = H_n/\sqrt{n!}$.

Table 3.4: *The orthonormal Hermite polynomial basis, source: [42]*

n	H_n	$\ H_n\ ^2$	\widetilde{H}_n
0	1	1	H_0
1	z	1	H_1
2	$z^2 - 1$	2	$H_2/\sqrt{2}$
3	$z^3 - 3z$	6	$H_3/\sqrt{6}$
4	$z^4 - 6z^2 + 3$	24	$H_4/\sqrt{24}$
5	$z^5 - 10z^3 + 15z$	120	$H_5/\sqrt{120}$

The selection of the polynomial basis depends on the original distribution of input variables \mathbf{X} . For example, the Hermite polynomial basis is usually chosen for Gaussian distributed variables or variables without bounds. The Legendre polynomial basis is selected for uniform distributed variables or variables with bounds. In addition, Laguerre and Jacobi polynomials are also widely used. Furthermore, the univariate orthonormal polynomials $\Psi_n(z)$ can be also constructed with respect to an arbitrary probability distribution $f(z)$, which is called the arbitrary polynomial basis [42, 111]. For the arbitrary polynomial basis, $f(z)$ is same as the original input distribution $f(x)$. Therefore, the transformation \mathbf{T} between X and Z becomes simple. After the distribution of Z is chosen, the orthonormal polynomials $\Psi_n(z)$ with respect to the distribution $f(z)$ are calculated according to the recurrence Equation (3.144).

Multivariate Orthogonal Polynomials

Univariate orthogonal polynomials are defined for each input variable Z_i individually. To deal with multivariate inputs, multivariate orthogonal polynomials are constructed as below [42]:

$$\Psi_{\boldsymbol{\alpha}}(\mathbf{Z}) = \prod_{i=1}^d \Psi_{\alpha_i}^{(i)}(Z_i), \quad (3.151)$$

where $\Psi_{\alpha_i}^{(i)}(Z_i)$ denotes the univariate orthonormal polynomials. α_i ($\alpha_i = \{0, 1, \dots, p, \dots\}$) is the order of the univariate polynomial bases for each variable Z_i . In practice, the truncated order p is used for each univariate polynomials. Therefore, there are $p + 1$ polynomial bases for each variable. $\boldsymbol{\alpha} = \{\alpha_1, \alpha_1, \dots, \alpha_d\}$ is called multi-indices. It is a

combination of each univariate order α_i . For example, $\boldsymbol{\alpha}$ can be a tensor product of all α_i . Therefore, the total number of polynomial bases is

$$P = (p + 1)^d, \quad (3.152)$$

where P denotes the total number of polynomial bases for the multivariate orthogonal polynomials. Therefore, there are P corresponding coefficients required to compute. Evaluations of the original model should be equal or larger than P for the computation of PCE coefficients $k_{\boldsymbol{\alpha}}$. The required number of evaluations grows rapidly along with the increasing dimensions d of input.

To avoid the curse of dimensionality, the classical truncated PCE is proposed in [41, 112]. Instead of truncating each α_i , only the total degree of $|\boldsymbol{\alpha}|$ is truncated to p as below:

$$|\boldsymbol{\alpha}| = \sum_{i=1}^d \alpha_i \leq p. \quad (3.153)$$

The total number P can be computed by

$$P = \frac{(d + p)!}{d! p!}. \quad (3.154)$$

To further reduce the polynomial bases, a hyperbolic PCE by truncating the q -norm of $\boldsymbol{\alpha}$ proposed in [113] is as follows:

$$\|\boldsymbol{\alpha}\|_q = \left(\sum_{i=1}^d \alpha_i^q \right)^{1/q} \leq p, \quad 0 < q < 1. \quad (3.155)$$

When $q = 1$, the hyperbolic PCE is equivalent to the classical PCE. When $q \rightarrow 0$, the interaction terms among different variables are reduced. An example of the hyperbolic truncation with two dimensional input is illustrated in Figure 3.28. The black solid line denotes $\|\boldsymbol{\alpha}\|_{0.5} = 7$ for the setting $q = 0.5$ and $p = 7$. All points in the area lower than the black solid line satisfy $\|\boldsymbol{\alpha}\|_{0.5} \leq 7$, which are drawn using blue circles. As an example, the combination of $\alpha_1 = 2$ and $\alpha_2 = 1$ denoted by the point $A(2, 1)$ in Figure 3.28 should be included, since

$$\|\boldsymbol{\alpha}\|_{0.5} = (\alpha_1^{0.5} + \alpha_2^{0.5})^{1/0.5} = (\sqrt{2} + \sqrt{1})^2 \approx 5.8284 < 7. \quad (3.156)$$

For consistency with the application in Chapter 4, the truncated PCE model mentioned in Equation (3.138) can be rewritten as below:

$$Y = \sum_{j=0}^{P-1} k_j \Psi_j(\mathbf{Z}) = \sum_{j=0}^{P-1} k_j \Psi_j(\mathbf{T}(\mathbf{X})). \quad (3.157)$$

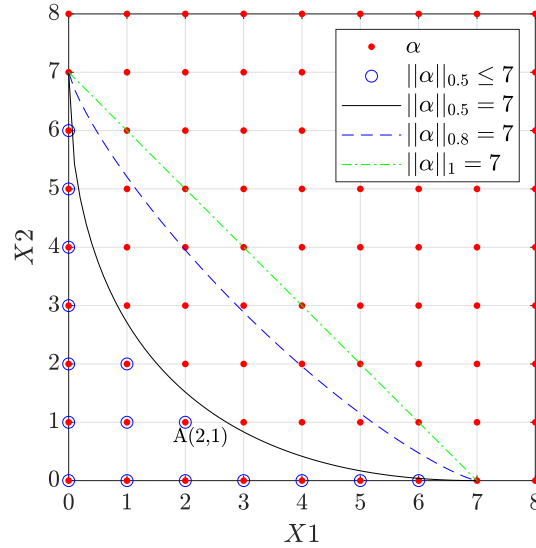


Figure 3.28: Classical and hyperbolic truncated scheme of PCE with $p = 7, q = 0.5, 0.8$ and 1 , adapted from [112].

Example for Multivariate Orthogonal Polynomials

An example of the multivariate polynomials construction in 2 dimensions is given. The original model is $Y = g(\mathbf{X}) = g(x_1, x_2)$ with two independent variables, while $x_1 \sim \mathcal{N}(\mu, \sigma)$ and $x_2 \sim \mathcal{U}[a, b]$. To construct a PCE model, the two polynomial bases are chosen first based on the distributions of \mathbf{X} . The Hermite polynomials $\tilde{H}_n(z_1)$ are used for the dimension of x_1 , while the Legendre polynomials $\tilde{L}_n(z_2)$ are used for the dimension of x_2 . The mapping relation between \mathbf{X} and \mathbf{Z} is as below:

$$\begin{aligned} x_1 &= \mu + \sigma z_1; & z_1 &\sim \mathcal{N}(\mu, \sigma) \\ x_2 &= \frac{a+b}{2} + \frac{b-a}{2} z_2; & z_2 &\sim \mathcal{U}[-1, 1] \end{aligned} \quad (3.158)$$

Based on Equation (3.151), the multivariate polynomials is constructed as below:

$$\Psi_{\alpha}(\mathbf{Z}) = \Psi_{\alpha_1, \alpha_2}(z_1, z_2) = \tilde{H}_{\alpha_1}(z_1) \cdot \tilde{L}_{\alpha_2}(z_2), \quad (3.159)$$

The truncation order p is set to 3, and $|\alpha| \leq 3$. This indicates that all the polynomials of z_1, z_2 that the total degree is less than or equal to 3 are considered in the PCE model. The univariate polynomials of \tilde{H}_{α_1} and \tilde{L}_{α_2} are shown in Table 3.4 and Table 3.3. All terms of the multivariate polynomials are shown in Table 3.5.

Table 3.5: All the polynomials of z_1, z_2 in the PCE model

j	$\alpha = [\alpha_1, \alpha_2]$	\widetilde{H}_{α_1}	\widetilde{L}_{α_2}	$\Psi_{\alpha} \equiv \Psi_j$
0	[0,0]	1	1	$\Psi_0 = 1$
1	[1,0]	z_1	1	$\Psi_1 = z_1 \cdot 1$
2	[0,1]	1	$\sqrt{3}z_2$	$\Psi_2 = 1 \cdot \sqrt{3}z_2$
3	[2,0]	$\frac{1}{\sqrt{2}}(z_1^2 - 1)$	1	$\Psi_3 = \frac{1}{\sqrt{2}}(z_1^2 - 1) \cdot 1$
4	[1,1]	z_1	$\sqrt{3}z_2$	$\Psi_4 = z_1 \cdot \sqrt{3}z_2$
5	[0,2]	1	$\frac{\sqrt{5}}{2}(3z_2^2 - 1)$	$\Psi_5 = 1 \cdot \frac{\sqrt{5}}{2}(3z_2^2 - 1)$
6	[3,0]	$\frac{1}{\sqrt{6}}(z_1^3 - 3z_1)$	1	$\Psi_6 = \frac{1}{\sqrt{6}}(z_1^3 - 3z_1) \cdot 1$
7	[2,1]	$\frac{1}{\sqrt{2}}(z_1^2 - 1)$	$\sqrt{3}z_2$	$\Psi_7 = \frac{1}{\sqrt{2}}(z_1^2 - 1) \cdot \sqrt{3}z_2$
8	[1,2]	z_1	$\frac{\sqrt{5}}{2}(3z_2^2 - 1)$	$\Psi_8 = z_1 \cdot \frac{\sqrt{5}}{2}(3z_2^2 - 1)$
9	[0,3]	1	$\frac{\sqrt{7}}{2}(5z_2^3 - 3z_2)$	$\Psi_9 = 1 \cdot \frac{\sqrt{7}}{2}(5z_2^3 - 3z_2)$

By summing up all the polynomials $\{\Psi_j\}$ in Table 3.5 with the corresponding coefficients, the structure of the PCE model is obtained as below:

$$\begin{aligned}
\hat{Y} &= g_{PCE}(x_1, x_2) = \sum_{j=0}^{P-1} k_j \Psi_j(\mathbf{Z}) \\
&= k_0 + k_1 \cdot z_1 + k_2 \cdot \sqrt{3}z_2 + k_3 \cdot \frac{1}{\sqrt{2}}(z_1^2 - 1) \\
&\quad + k_4 \cdot z_1 \cdot \sqrt{3}z_2 + k_5 \cdot \frac{\sqrt{5}}{2}(3z_2^2 - 1) + k_6 \cdot \frac{1}{\sqrt{6}}(z_1^3 - 3z_1) \\
&\quad + k_7 \cdot \frac{1}{\sqrt{2}}(z_1^2 - 1) \cdot \sqrt{3}z_2 + k_8 \cdot z_1 \cdot \frac{\sqrt{5}}{2}(3z_2^2 - 1) + k_9 \cdot \frac{\sqrt{7}}{2}(5z_2^3 - 3z_2).
\end{aligned} \tag{3.160}$$

After the structure of the PCE model is constructed, the only remaining thing is to calculate the corresponding coefficients $\{k_j\}$.

3.5.2 Computation of PCE Coefficients

Two well-known methods: quadrature-based projection and Least-square minimization, are available to calculate the PCE coefficients. Their analytical solutions are identical [114]. The least-square minimization method is used in this dissertation, and the detailed algorithm is introduced in this section. Based on the PCE-based surrogate model, the residual of the surrogate modeling is computed by [42]:

$$\epsilon(\mathbf{X}) = g(\mathbf{X}) - \sum_{j=0}^{P-1} k_j \Psi_j(\mathbf{T}(\mathbf{X})). \tag{3.161}$$

The unknown coefficients k_j are estimated by minimizing the mean square residuals:

$$\begin{aligned}\hat{\mathbf{K}} &= \arg \min \mathbb{E}[\epsilon^2] = \arg \min \mathbb{E} \left[\left(g(\mathbf{X}) - \sum_{j=0}^{P-1} k_j \Psi_j(\mathbf{T}(\mathbf{X})) \right)^2 \right] \\ &= \arg \min \frac{1}{n} \sum_{l=1}^n \left(g(\mathbf{x}_l) - \sum_{j=0}^{P-1} k_j \Psi_j(\mathbf{T}(\mathbf{x}_l)) \right)^2,\end{aligned}\tag{3.162}$$

where $\mathbf{K} = \{k_0, k_1, \dots, k_{P-1}\}$. \mathbf{x}_l denotes the l th sample of \mathbf{X} . Let \mathbf{Y} denote the output response vector and Ψ denotes the value of polynomials for the given \mathbf{X} as below:

$$\mathbf{Y} = \{g(\mathbf{x}_1), g(\mathbf{x}_2), \dots, g(\mathbf{x}_n)\}^T, \quad \text{size} : n \times 1;\tag{3.163}$$

$$\Psi = \{\Psi_{lj}\}_{n \times P}, \quad l = 1, 2, \dots, n, \quad \text{size} : n \times P.\tag{3.164}$$

Therefore, the mean square ϵ is:

$$\begin{aligned}\mathbb{E}[\epsilon^2] &= (\mathbf{Y} - \mathbf{K}\Psi)^T (\mathbf{Y} - \mathbf{K}\Psi) \\ &= \mathbf{Y}^T \mathbf{Y} - 2\mathbf{K}^T \Psi^T \mathbf{Y} + \mathbf{K}^T (\Psi^T \Psi) \mathbf{K}.\end{aligned}\tag{3.165}$$

Let the derivative of the mean square residual with respect to \mathbf{K} equals 0:

$$\frac{\partial \mathbb{E}[\epsilon^2]}{\partial \mathbf{K}^T} = -2\Psi^T \mathbf{Y} + 2(\Psi^T \Psi) \mathbf{K} = 0.\tag{3.166}$$

Therefore, the estimated \mathbf{K} is obtained:

$$\hat{\mathbf{K}} = (\Psi^T \Psi)^{-1} \Psi^T \mathbf{Y}.\tag{3.167}$$

The procedure of computing PCE coefficients using the least-square minimization method is described in Algorithm 10. In the step 1, the used polynomial basis or distribution families followed by \mathbf{Z} are chosen according to the original distributions of \mathbf{X} as mentioned in Section 3.5.1. In the step 2, the p and the $|\boldsymbol{\alpha}|$ are set by trial and error. Noticed that the selection of the truncated order p not only depends on the complexity of the original model but also influenced by the transformation relationship between X space and Z space. Furthermore, the computational ability limits the maximum value of p . In the step 3, the construction of the univariate polynomial basis is based on the 3-term recurrence relation in Equation (3.144). In the step 4, $3P$ evaluations of the original model are applied, since the number of samples should be larger than the number of polynomial bases to solve the least-squares problem. From the step 5 to the step 8, the PCE coefficients are obtained based on the least-square minimization solution after evaluating the model. Before using the obtained PCE model, the validation step is implemented in the step 9. By running the MCS though the original model and the PCE model, the corresponding output responses are compared to validate if the PCE model can represent the original model. If there is a large deviation in the output responses, the distribution type for Z_i , the p , and the $|\boldsymbol{\alpha}|$ can be adjusted by trial and error.

Algorithm 10 Calculate PCE coefficients using the least-square minimization approach

- 1: Select the distribution type for each variable Z_i according to $f_{X_i}(x_i)$.
 - 2: Set the truncated order p and $|\boldsymbol{\alpha}|$ by trial and error
 - 3: Construct the univariate polynomial basis $\Psi_{\alpha_i}^{(i)}(Z_i)$ based on Equation (3.144).
 - 4: Generate n samples of \mathbf{X} from $f_{\mathbf{X}}(\mathbf{x})$. $n = 3P$, and $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}^T$.
 - 5: Evaluate the original model $Y = g(\mathbf{X})$, and obtain $\mathbf{Y} = \{g(\mathbf{x}_1), g(\mathbf{x}_2), \dots, g(\mathbf{x}_n)\}^T$.
 - 6: Transfer \mathbf{X} from space X to Z : $\mathbf{Z} = \mathbf{T}(\mathbf{X}) = \{\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_n\}^T$ using Equation (3.99).
 - 7: Calculate the value of polynomials $\boldsymbol{\Psi} = \{\Psi_{lj}\}_{n \times P}$:
 $\Psi_{lj} = \prod_{i=1}^d \Psi_{\alpha_i}^{(i)}(z_l)$, $l = 1, 2, \dots, n$. $j = 1, 2, \dots, P$.
 - 8: Solve the PCE coefficients: $\hat{\mathbf{K}} = (\boldsymbol{\Psi}^T \boldsymbol{\Psi})^{-1} \boldsymbol{\Psi}^T \mathbf{Y}$
 - 9: Validate the PCE model results using MCS
-

3.5.3 Statistical Moments of PCE Output

A good property of PCE is that the low-order statistical moments of the model output are encoded in the PCE coefficients. After the PCE-based surrogate model is obtained, we can directly compute the statistical moments without further evaluations of the original model. According to the properties mentioned in Equation (3.146), the mean and the variance of the model output Y are computed as below:

Mean

$$\hat{\mu}_Y = \mathbb{E} \left[\sum_{j=0}^{P-1} k_j \Psi_j(\mathbf{Z}) \right] = \sum_{j=0}^{P-1} (k_j \mathbb{E}[\Psi_j(\mathbf{Z})]) = k_0; \quad (3.168)$$

Variance

$$\hat{\sigma}_Y^2 = \mathbb{E}[(Y^{PCE} - \hat{\mu}_Y)^2] = \mathbb{E} \left[\left(\sum_{j=1}^{P-1} k_j \Psi_j(\mathbf{Z}) \right)^2 \right] = \sum_{j=1}^{P-1} (k_j^2 \mathbb{E}[\Psi_j^2(\mathbf{Z})]) = \sum_{j=1}^{P-1} k_j^2, \quad (3.169)$$

where Y_{PCE} denotes the obtained model output Y from the PCE model. For the calculation of higher order moments, the expectation of the polynomials is required to be computed. The skewness coefficient δ_Y and the kurtosis coefficient κ_Y are computed as below:

Skewness coefficient δ_Y

$$\begin{aligned} \hat{\delta}_Y &= \mathbb{E} \left[\left(\frac{(Y^{PCE} - \hat{\mu}_Y)}{\sigma_Y} \right)^3 \right] = \frac{\mathbb{E}[(Y^{PCE} - \hat{\mu}_Y)^3]}{\sigma_Y^3} = \frac{1}{\sigma_Y^3} \mathbb{E} \left[\left(\sum_{\alpha} k_{\alpha} \Psi_{\alpha}(\mathbf{Z}) \right)^3 \right] \\ &= \frac{1}{\sigma_Y^3} \sum_{\alpha} \sum_{\beta} \sum_{\gamma} k_{\alpha} k_{\beta} k_{\gamma} \mathbb{E}[\Psi_{\alpha}(\mathbf{Z}) \Psi_{\beta}(\mathbf{Z}) \Psi_{\gamma}(\mathbf{Z})]; \end{aligned} \quad (3.170)$$

Kurtosis coefficient κ_Y

$$\begin{aligned}\hat{\kappa}_Y &= \mathbb{E} \left[\left(\frac{(Y^{PCE} - \hat{\mu}_Y)}{\sigma_Y} \right)^4 \right] = \frac{\mathbb{E}[(Y^{PCE} - \hat{\mu}_Y)^4]}{\sigma_Y^4} = \frac{1}{\sigma_Y^4} \mathbb{E} \left[\left(\sum_{\alpha} k_{\alpha} \Psi_{\alpha}(\mathbf{Z}) \right)^4 \right] \\ &= \frac{1}{\sigma_Y^4} \sum_{\alpha} \sum_{\beta} \sum_{\gamma} \sum_{\delta} k_{\alpha} k_{\beta} k_{\gamma} k_{\delta} \mathbb{E}[\Psi_{\alpha}(\mathbf{Z}) \Psi_{\beta}(\mathbf{Z}) \Psi_{\gamma}(\mathbf{Z}) \Psi_{\delta}(\mathbf{Z})].\end{aligned}\quad (3.171)$$

3.5.4 PCE-based Sobol Indices

As mentioned in Section 3.4.3, the computation of Sobol's indices based on MCS is time-consuming. Sudret in paper [115] proposed a PCE-based algorithm to compute Sobol's indices. Since the Sobol decomposition in Equation (3.113) is unique, the PCE model exactly provides the decomposition solution. A decomposed function containing all elements of \mathbf{X}_v can be described using the polynomial expansions:

$$g_v(\mathbf{x}_v) = \sum_{\alpha \in D_v} k_{\alpha} \Psi_{\alpha}(\mathbf{Z}), \quad (3.172)$$

where D_v denotes all combinations that contains at least one variable of \mathbf{X}_v . According to the PCE property in statistical moments, the variance of the function g_v is computed as below:

$$Var(g_v(\mathbf{x})) = Var \left(\sum_{\alpha \in D_v} k_{\alpha} \Psi_{\alpha}(\mathbf{Z}) \right) = \sum_{\alpha \in D_v; |\alpha| \neq 0} k_{\alpha}^2. \quad (3.173)$$

Noticed that $Var(g_v(\mathbf{x})) = Var(\mathbb{E}[Y|\mathbf{X}_v])$, which can be proved by integrating Equation (3.113) with respect to $\mathbf{x}_{\sim v}$, like computing g_i and g_{ij} in Equation (3.118, 3.119). Therefore, the closed Sobol's index S_v in Equation (3.128) can be computed directly using the PCE coefficients by

$$S_v = \frac{Var(\mathbb{E}[Y|\mathbf{X}_v])}{Var(Y)} = \frac{Var(g_v(\mathbf{x}))}{Var(g(\mathbf{x}))} = \frac{\sum_{\alpha \in D_v; |\alpha| \neq 0} k_{\alpha}^2}{\sum_{\alpha; |\alpha| \neq 0} k_{\alpha}^2}, \quad (3.174)$$

Other order Sobol's indices are easily computed afterwards using Equation (3.124) and Equation (3.125).

Chapter 4

Calibration of Contributing Factors for Predictive Analysis

Model calibration is a necessary step of adjusting the model parameters to obtain a good match between the model prediction and the real measurements. In Predictive Analysis Framework, the ‘Calibrate’ step is required to ensure that the incident model output can represent the reality of the relevant flight operation in a statistical manner. Instead of tuning the model parameters, the calibration of contributing factors is first proposed in [9] to avoid the abuse of the ‘Identify’ step and minimize the distribution fitting error introduced in the ‘Cumulate’ step. To improve the convergence and efficiency of the calibration algorithm, a new calibration framework using a PCE method with a frozen sample strategy is proposed in this chapter. Major results of this chapter have been published in [50].

In order to better describe the idea of the ‘Calibrate’ step, the runway overrun model mentioned in Section 2.5 is used for demonstration. The basic calibration strategy using Monte Carlo simulation (MCS) is explained in Section 4.1, followed by a new calibration framework using PCE in Section 4.2. In Section 4.3, a first-order system is used to validate the new calibration algorithm. At last, the calibration results of runway overrun are shown in Section 4.4.

4.1 Calibration Strategy

Model calibration and validation is a necessary process to ensure that the built model predicts reality. Generally, the built model is validated by comparing model output and observations and calibrated by minimizing the deviation between the simulated model output and real observations via adjusting model parameters. As shown in [26], the maximum likelihood methods or Bayesian approaches are applied for calibration. However,

this process is involved in the ‘Identify’ step of the Predictive Analysis Framework. The unknown parameters are estimated to have a good match between model output and measurements. Readers may refer to the dissertation [77] for the ‘Identify’ step. Considering the objectives of the Predictive Analysis, the occurrence probability of the incident is required to match the real operations. Due to the lack of incidents data, the direct comparison of the incident probability is impossible. Alternatively, the incident model output should represent the same statistical feature with the routine operations. The illustration of validation proposed by [9] is shown in Figure 4.1.

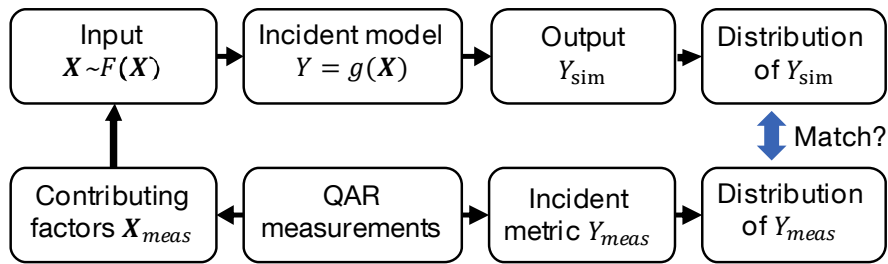


Figure 4.1: Illustration of validation, source: Fig.1 of [50]

If and only if the probability distribution of model simulation results is identical to the distribution of the recorded data from the QAR, the incident model with the corresponding contributing factors can be used to predict the incident probability. However, this is often not the case in reality, for example, the runway overrun case. There is no stop margin (incident model output) recorded directly in the QAR. Instead, the model output d_{80} , i.e., the distance from touchdown to 80 knots, is computed directly from the recorded ground speed and position data. Therefore, the simulated and observed distances d_{80} is utilized for validation. Distributions of the simulated d_{80} in MCS and the observed d_{80} in the QAR are shown in Figure 4.2. It shows that the simulated output distribution is not identical to the distribution of the observations. The calibration of the incident model and contributing factors might reduce the deviation.

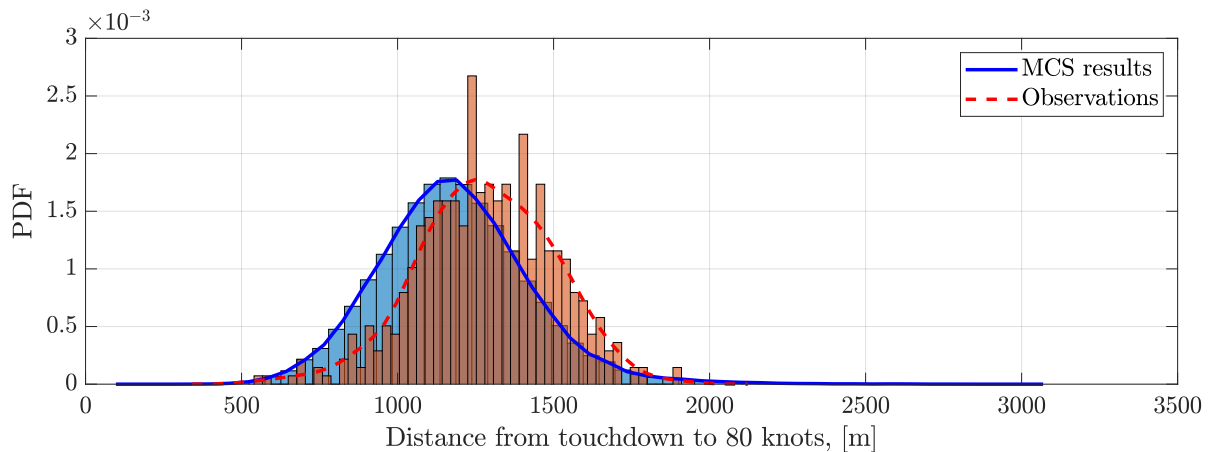


Figure 4.2: Comparison of simulated and observed outputs

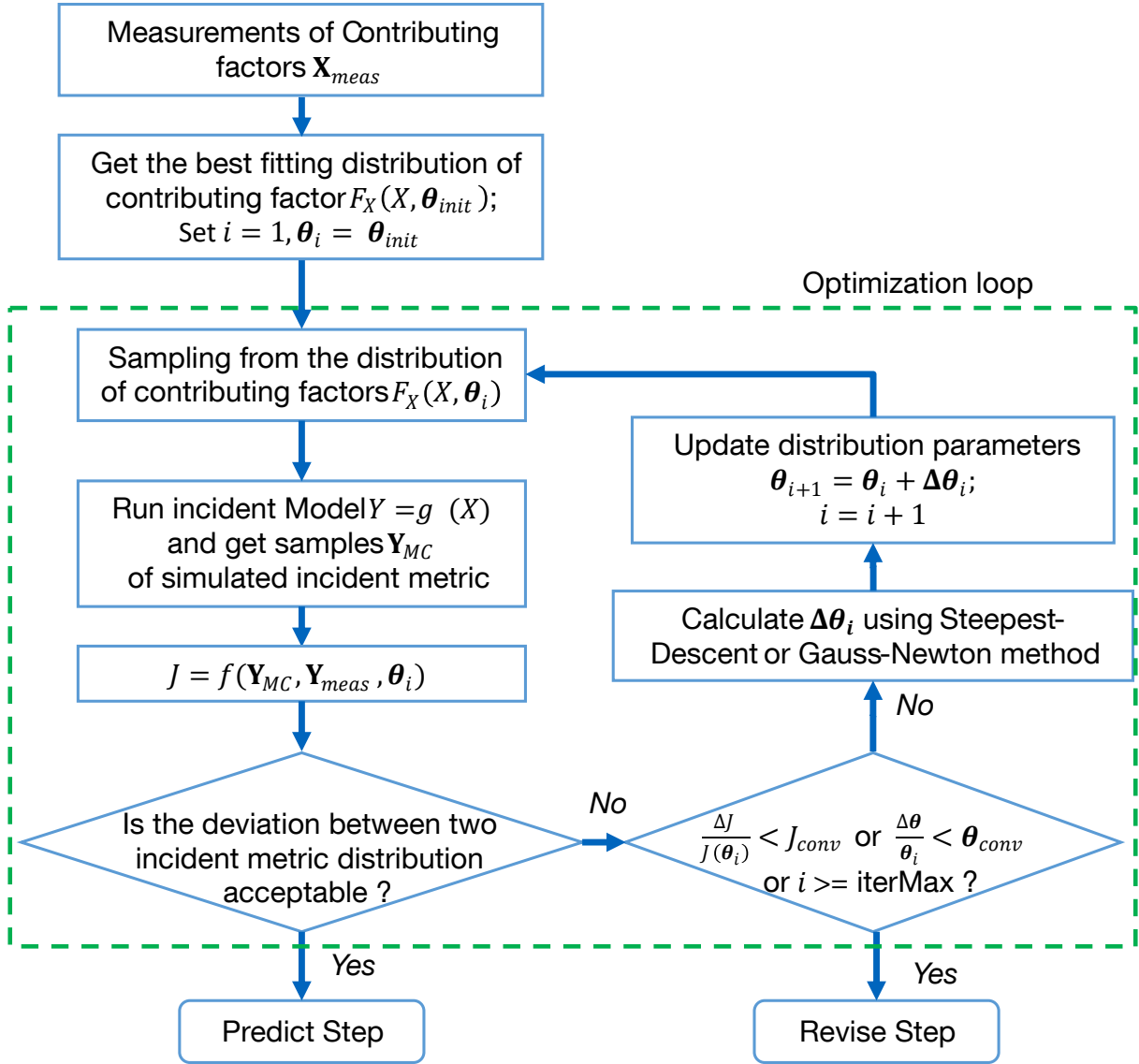


Figure 4.3: Flow chart of the MCS-based calibration

Generally, the model calibration is implemented via tuning the model parameters. In the ‘Identify’ step of the Predictive Analysis framework, the unknown model parameters have been estimated to match the simulation output to the observations in each flight. In addition, the ‘Cumulate’ step introduces a fitting error in the distributions of contributing factors. The error will be subsequently propagated to the output during the MCS. Furthermore, there is also a distribution-fitting error for the model output. Therefore, the calibration strategy is to obtain an optimum reproduction for the real operational features by balancing the distribution errors in both contributing factors and the incident metric. Without falsifying the observations of contributing factors, only a tiny modification in the distributions of contributing factors is allowed to tune. The calibration procedure based on the MCS is described in Figure 4.3. The X denotes the model input. Y denotes the model output, which is the distance d_{80} in the runway overrun case. The subscript *meas*

indicates that the model output is obtained from the observation directly. Alternatively, the subscript MC denotes the model output from the MCS results. $\boldsymbol{\theta}$ is the distribution parameters of all contributing factors.

The cost function J computes the deviation between simulated and recorded output distributions. In the previous work [116], the author used the integrated quadratic distance between two output CDFs as the cost function:

$$J = \int_{y_{min}}^{y_{max}} (F_{meas}(y) - F_{sim}(y))^2 dy, \quad (4.1)$$

where $F_{meas}(y)$ and $F_{sim}(y)$ denote the CDFs of the measured output Y_{meas} and the simulated output Y_{sim} , respectively. In order to balance the distribution-fitting errors between contributing factors and the incident metric, a penalty term of input fitting error, is integrated into the cost function as below:

$$\begin{aligned} J &= \int_{y_{min}}^{y_{max}} (F_{meas}(y) - F_{sim}(y))^2 dy + J_X(\boldsymbol{\theta}, X_{meas}) \\ &= \int_{y_{min}}^{y_{max}} (F_{meas}(y) - F_{sim}(y))^2 dy + \sum_{i=1}^d \left[\int_{D_{x_i}} (F_{X_i}(x_i|\boldsymbol{\theta}^{X_i}) - F_{X_i}^e(x_i))^2 dx_i \right], \end{aligned} \quad (4.2)$$

where $J_X(\boldsymbol{\theta}, X_{meas})$ denotes the input fitting errors influenced by the distribution parameters and the measured data of input variables. D_{x_i} denotes the range of X_i . F_{X_i} denotes the estimated CDF function of input X_i with the distribution parameter $\boldsymbol{\theta}^{X_i}$, while $F_{X_i}^e$ is the empirical CDF of X_i based on the measurements $X_{i,meas}$. J_X is defined as the sum of the integrated quadratic distance between the two CDFs F_{X_i} and $F_{X_i}^e$. Since the accurate distribution function of the model output might be difficult to obtain, the statistical moments of the model output can be also used instead. The modified cost function is shown as below:

$$J = (M_{Y_{sim}} - M_{Y_{meas}})^T (M_{Y_{sim}} - M_{Y_{meas}}) + J_X, \quad (4.3)$$

where M_Y denotes the statistical moments of the model output Y . $M_Y = [\mu_Y, \sigma_Y, \delta_Y, \kappa_Y]$, where μ_Y , σ_Y , δ_Y and κ_Y are the mean, standard deviation, skewness and kurtosis coefficients of Y , respectively. The input fitting errors J_x are also replaced by comparing the statistical moments of the estimated distribution and the original measurements. Only mean value of contributing factors are adjusted in this thesis. The modified cost function is shown as below:

$$J = (M_{Y_{sim}} - M_{Y_{meas}})^T W_1 (M_{Y_{sim}} - M_{Y_{meas}}) + (\boldsymbol{\mu}_X(\boldsymbol{\theta}) - \boldsymbol{\mu}_{X_{meas}})^T W_2 (\boldsymbol{\mu}_X(\boldsymbol{\theta}) - \boldsymbol{\mu}_{X_{meas}}), \quad (4.4)$$

where $\boldsymbol{\mu}_X$ denotes the vector of all mean values of the estimated distributions of all contributing factors. $\boldsymbol{\mu}_{X_{meas}}$ is the vector of the mean value of original measurements for all contribution factors. Weight matrices W_1 and W_2 are applied for scaling the different distributions as below:

$$W_1 = \text{diag}\left(\frac{10}{\sigma_Y^2}, \frac{5}{\sigma_Y^2}, 1, 1\right), \quad W_2 = \frac{1}{d} * \text{diag}\left(\frac{1}{\sigma_1^2}, \frac{1}{\sigma_2^2}, \dots, \frac{1}{\sigma_d^2}\right), \quad (4.5)$$

where σ_i^2 denotes the variance of the X_i . $\boldsymbol{\theta}$ is adjusted to minimize the cost function in Equation (4.4). Nonlinear programming methods, such as the Steepest-Descent method and the Gauss-Newton method [117], can be used to solve the optimization problem. The optimization loop stops when the cost function does not decrease, the $\Delta\boldsymbol{\theta}$ becomes very small, or the number of iterations reaches the maximum iterations *iterMax*. If the deviation between the simulated and the observed distributions of the incident metric is acceptable, the optimization loop also stops, and the calibrated contributing factors will be used in the following ‘Predict’ step. Otherwise, the model should be revised, for example, by including more crucial contributing factors or excluding the unimportant factors.

4.2 PCE-based Calibration Framework

For the calibration framework demonstrated in Figure 4.3, the MCS is embedded in the optimization loop. Due to the fact that the MCS requires a large number of evaluations to obtain convergent results, and it is required in each iteration of the optimization, this framework is time-consuming. Since the samples are randomly generated from the input distributions, the cost function varies even for the same distribution parameters $\boldsymbol{\theta}$. Therefore, the cost function is difficult to converge when using the above MCS-based calibration. As mentioned in Section 3.5.3, the statistical moments of the PCE model output are encoded in the PCE coefficients. By utilizing this convenient property, this section proposes a new PCE-based calibration framework with a frozen sample strategy.

4.2.1 PCE-based Surrogate Model

The mathematical fundamentals of the PCE model are already introduced in Section 3.5. First, the PCE-based surrogate model is constructed based on the polynomial bases and computed coefficients. Then, the PCE model can be evaluated instead of the original model. A relationship between the PCE-based surrogate model and the original model is illustrated in Figure 4.4.

The original model is represented by a sum of the series of polynomial bases. If the original inputs are arbitrarily distributed, an isoprobabilistic transformation \mathbf{T} is required to transfer samples \mathbf{X} to \mathbf{Z} as below:

$$\mathbf{Z} = \mathbf{T}(\mathbf{X}), \quad (4.6)$$

where the inverse transformation is denoted by \mathbf{T}^{-1} , and $\mathbf{X} = \mathbf{T}^{-1}(\mathbf{Z})$. The detailed procedure of the isoprobabilistic transformation \mathbf{T} is already described in Equation (3.98) and Equation (3.99). The transformation \mathbf{T} is mainly obtained based on marginal CDFs $F_{X_i}(X_i)$ and $\phi_i(Z_i)$ and their corresponding inverse CDFs, while the dependence C is also

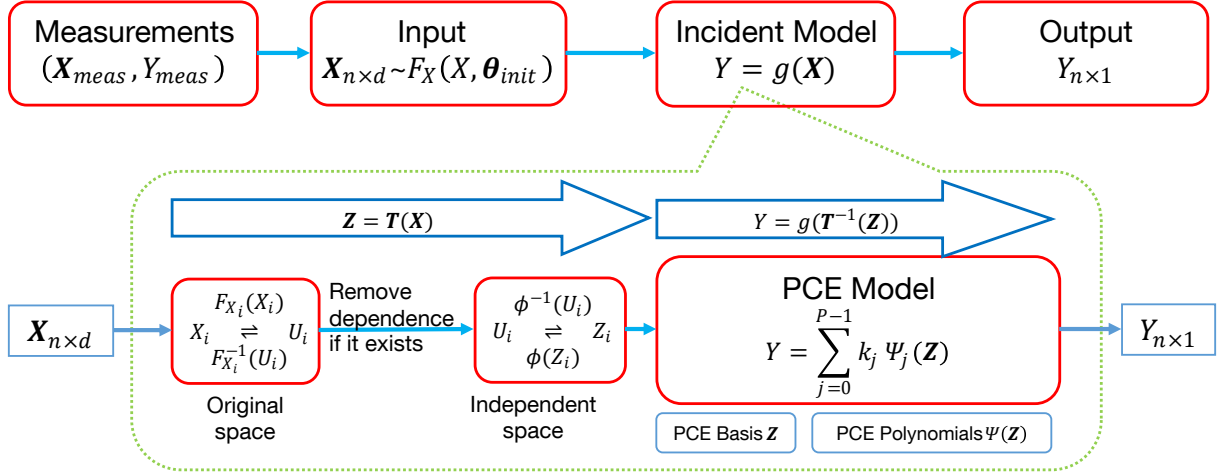


Figure 4.4: PCE-based Surrogate Model

required for dependent variables. $F_{X_i}(X_i)$ is estimated based on the measurements of X_i , while $\phi_i(Z_i)$ is selected based on the selected polynomial basis families. The dependence of \mathbf{X} is removed during the transformation from \mathbf{X} to \mathbf{Z} if it exists [34, 42]. After the transformation, the value of the polynomials $\Psi_j(\mathbf{Z})$ is subsequently computed based on the value of samples \mathbf{Z} . Then the output Y is obtained by summing up all polynomials. As shown in Figure 4.4, the PCE model does not only surrogate the original model, but also represent the inverse transformation \mathbf{T}^{-1} as follows:

$$Y = g(\mathbf{X}) = g(\mathbf{T}^{-1}(\mathbf{Z})) = G(\mathbf{Z}) \approx \sum_{j=0}^{P-1} k_j \Psi_j(\mathbf{Z}). \quad (4.7)$$

where $G(\cdot)$ denotes the relation $g(\mathbf{T}^{-1}(\cdot))$. Instead of the original model $g(\cdot)$, the PCE model uses a series of polynomial expansions to surrogate $G(\cdot)$. It means that not only the original model $g(\cdot)$, but also the inverse transformation \mathbf{T}^{-1} is surrogated. Obviously, the dependence among the input variables increases the complexity of $G(\cdot)$. A high order PCE model might be required for dependent input variables to obtain the same accuracy for the same model $g(\cdot)$ with independent input variables.

4.2.2 Integration of PCE into the Calibration Framework

A PCE-based calibration framework with frozen samples is shown in Figure 4.5. In the beginning, an initial PCE model is computed based on the random samples $\mathbf{X}_{n \times d}$ and the corresponding output $\mathbf{Y}_{n \times 1}$ using the Algorithm 10. Furthermore, the MCS is required to validate if the obtained PCE model can surrogate the original model properly. If the output distributions from the obtained PCE model and the original model match very well, the obtained polynomial bases can be used for the further calibration procedure. Otherwise, more polynomial bases might be required in the PCE model to improve the surrogate accuracy. In the following optimization loop, the polynomial bases of the PCE

model will remain the same. However, the PCE coefficients will be updated in each iteration step of the optimization. Since the statistical moments of the PCE model output are encoded in the PCE coefficients, the cost function is defined by comparing the statistical moments mentioned in Equation (4.4). The Gauss-Newton optimization algorithm [117] is applied to update θ as below:

$$\Delta\theta = - \left(\frac{\partial^2 J}{\partial \theta^2} \right)^{-1} \frac{\partial J}{\partial \theta}, \quad (4.8)$$

$$\theta_{i+1} = \theta_i + \Delta\theta.$$

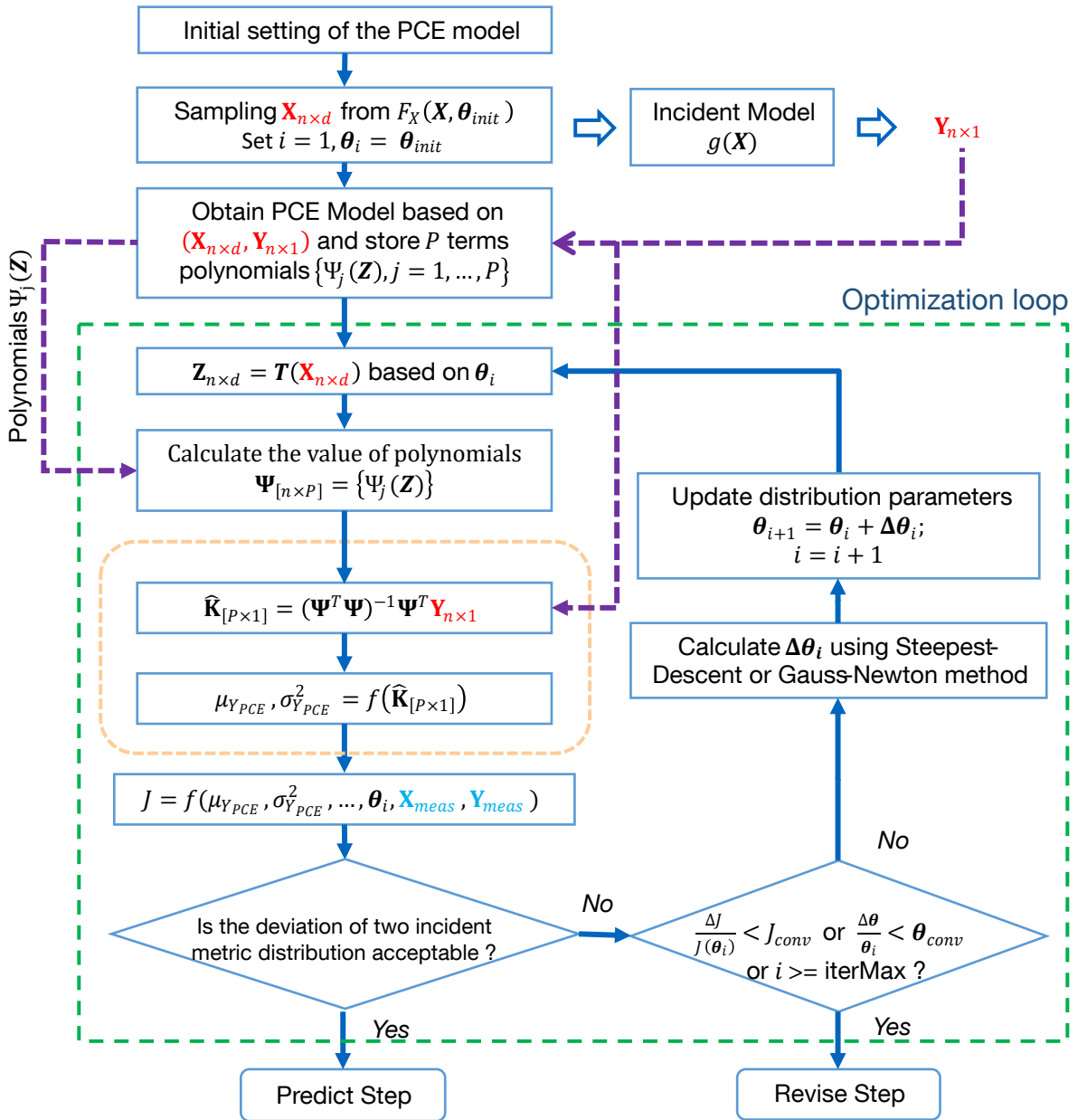


Figure 4.5: Flow chart of PCE-based calibration framework

Compared to the MCS-based calibration framework, the samples $\mathbf{X}_{n \times d}$ and $\mathbf{Y}_{n \times 1}$ are set as the designed samples in the PCE-based calibration. They will be reused during the optimization process. Although samples $\mathbf{X}_{n \times d}$ are frozen, the description parameters of input distributions $\boldsymbol{\theta}$ are updated in each iteration. Furthermore, the samples $\mathbf{Z}_{n \times d}$ will be updated correspondingly. Subsequently, the new PCE coefficients are calculated to minimize the deviation in the statistical moments between the PCE model output and the observed output. By freezing the designed samples, the MCS in each iteration step is not necessary anymore. The original incident model is only evaluated in the first step. Therefore, the PCE-based calibration framework dramatically improves the efficiency of the ‘Calibrate’ step. The procedure of the PCE-based calibration algorithm is described in detail in Algorithm 11.

Algorithm 11 The procedure of the PCE-based calibration algorithm

- 1: Initial setting of the PCE model: the truncated scheme p and $|\alpha|_q$
 - 2: Store the P terms polynomial bases $\{\Psi_j(\mathbf{Z}), j = 1, \dots, P\}$
 - 3: Generate n samples \mathbf{X} from $\mathbf{F}(\mathbf{X}|\boldsymbol{\theta}_{init})$, $n = 3P$
 - 4: Evaluate the model $Y = g(\mathbf{X})$, obtain the output \mathbf{Y}
 - 5: Compute the PCE coefficients using Algorithm 10
 - 6: Run MCS in the original and the PCE model using the same inputs, and obtain two output distributions with respect to $Y_{original}$ and Y_{PCE}
 - 7: **if** two output distributions from MCS do not match **then**
 - 8: Go back to step 1, and add more polynomial bases
 - 9: **end if**
 - 10: Set the iteration number $i = 1$, $\boldsymbol{\theta}_i = \boldsymbol{\theta}_{init}$
 - 11: Starting the optimization loop
 - 12: Transfer samples \mathbf{X} to \mathbf{Z} : $\mathbf{Z} = \mathbf{T}(\mathbf{X})$ based on the current value of $\boldsymbol{\theta}_i$
 - 13: Calculate the value of polynomials $\boldsymbol{\Psi} = \{\Psi_j(\mathbf{Z})\}$
 - 14: Solve the new PCE coefficients $\hat{\mathbf{K}} = (\boldsymbol{\Psi}^T \boldsymbol{\Psi})^{-1} \boldsymbol{\Psi}^T \mathbf{Y}$
 - 15: Calculate the statistical moments $M_{Y_{PCE}}$
 - 16: Calculate the cost function $J = f(M_{Y_{PCE}}, \boldsymbol{\theta}_i, X_{meas}, Y_{meas})$ using Equation (4.4)
 - 17: **if** Deviation between simulated and measured output is acceptable **then**
 - 18: **Stop:** return $\boldsymbol{\theta} = \boldsymbol{\theta}_i$ and go to the Predict step
 - 19: **else if** $\frac{\Delta J}{J} \leq J_{conv}$ or $\frac{\Delta \boldsymbol{\theta}}{\boldsymbol{\theta}} \leq \theta_{conv}$ or $i \geq iterMax$ **then**
 - 20: **Stop:** go to the Revise step
 - 21: **else**
 - 22: Calculate the gradient $\mathcal{G} = \partial J / \partial \boldsymbol{\theta}$ and the Hessian matrix $\mathcal{F} = \partial^2 J / \partial \boldsymbol{\theta}^2$
 - 23: Calculate $\Delta \boldsymbol{\theta} = -\mathcal{F}^{-1} \mathcal{G}$
 - 24: Update $\boldsymbol{\theta}_{i+1} = \boldsymbol{\theta}_i + \Delta \boldsymbol{\theta}$, $i = i + 1$
 - 25: **Return** to Step 12
 - 26: **end if**
 - 27: Run the MCS using the calibrated contributing factors and the original model
 - 28: Validate results by comparing two distributions of the simulated and observed outputs
- Remarks: J_{conv} , θ_{conv} , and $iterMax$ are the stop criteria for the optimization.
-

4.3 Validation of the Calibration Algorithm

Before using the algorithm in the runway overrun model, a first-order system is used as a test case in this section for validation. The system is also used as a test model for validation in [35] and [50]. The system shows a proportional transition behavior with a time delay. The model equation is as below:

$$y(t) = y_0 + \frac{1}{T} \int_0^t K u(\tau) - y(\tau) d\tau, \quad (4.9)$$

where the system input is a time series, denoted by u . A step signal of u is chosen. There are three uncertainty parameters: the initial value y_0 of y , the proportional gain K , and the time constant T . They are defined as three contributing factors. The model output Y is the value of y at a certain time t_s of the transition period. The uncertainties of the output Y driven by the three contributing factors are analyzed. The generalized model equation could be described as below:

$$Y = g(\mathbf{X}) = g(K, T, y_0). \quad (4.10)$$

For the step response of the first-order system, the model output Y can be computed analytically by

$$Y = y_0 + K(1 - e^{-t_s/T}). \quad (4.11)$$

Since the uncertainties in the contributing factors are not always Gaussian distributed. Three different distributions: ‘Lognormal’, ‘GEV’, and ‘Normal’ are chosen for K , T , and y_0 shown in Table 4.1. The values of these three contributing factors are also limited to a certain interval.

Table 4.1: *Contributing factors distributions of the first-order model, adapted from Table 2 of [50]*

\mathbf{X}	Symbols	Distribution	Bounds	Parameters	Shifted parameters
X_1	K	lognormal	[6,14]	$[\mu_1, \sigma_1] = [2.2, 0.2]$	$\mu_1 = 2.15$
X_2	T	GEV	[6,10]	$[k_2, \mu_2, \sigma_2] = [-0.2, 7, 1]$	$\mu_2 = 8$
X_3	y_0	Normal	[0,3]	$[\mu_3, \sigma_3] = [2.0, 0.1]$	$\mu_3 = 1.9$

By setting $t_s = 3.5$ and applying the distribution parameters in Table 4.1, the output distribution is obtained based on the MCS. As shown in Figure 4.6, the distribution of y at $t = 3.5$ is set as the target output distribution.

A shifted distribution parameters of contributing factors is shown in Table 4.1. They are chosen as the initial distribution parameters. The MCS also allows us to obtain the initial output distribution by using the shifted distributions. The first three statistical moments of the target and initial output are compared in Table 4.2.

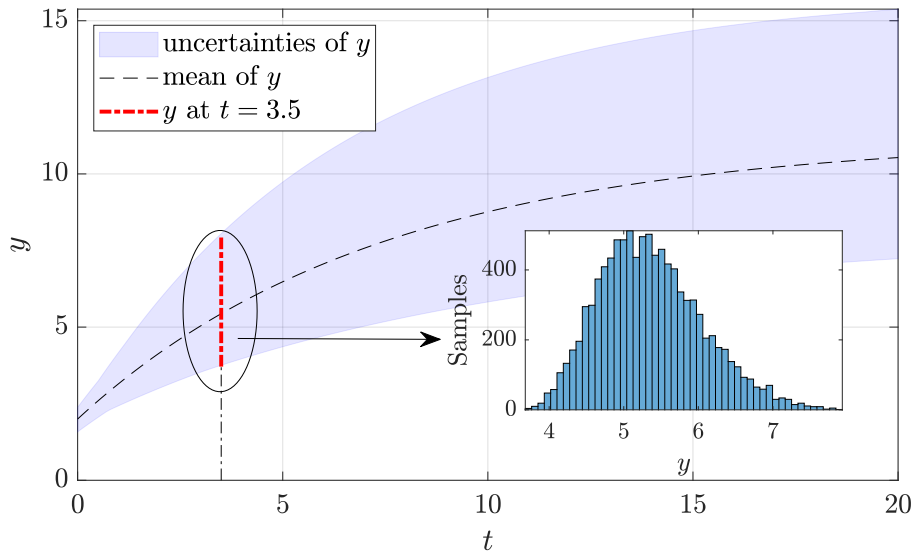


Figure 4.6: *Output uncertainties in the first order system*

Table 4.2: *Statistical moments of the model output, source: Table 3 of [50]*

Y	μ_Y	σ_Y	δ_Y
Target output	5.4435	0.7121	0.4832
Initial output	4.9767	0.6293	0.6227
Calibrated output	5.4357	0.7062	0.5068

The PCE-based calibration algorithm can be implemented to minimize the model output deviation by tuning contributing factors' distribution parameters. The first step is to obtain an initial PCE model. Since the input distributions are arbitrary with truncated boundaries, the arbitrary polynomial basis [42, 111] is applied. By trial and error, the truncated scheme with $p = 5$ and $q = 1$ is selected. Therefore, $P = (p + d)!/p!d! = 56$ polynomial terms compose the PCE model. In order to avoid the over-fitting issue, $10P = 560$ samples are used to calculate the PCE coefficients. MCS with the 10^5 samples are implemented in the original model and the obtained PCE model to check the surrogate accuracy. Samples plot between the original model output and the PCE output is shown using the blue cross in the left figure of Figure 4.7. The red line denotes the 45° positive diagonal line. The samples are almost located at the red line, which indicates that the PCE model has the same output as the original model. Furthermore, the CDFs and PDFs of the original and the PCE output match very well shown in Figure 4.7. It shows that the obtained PCE model could surrogate the original model with high accuracy.

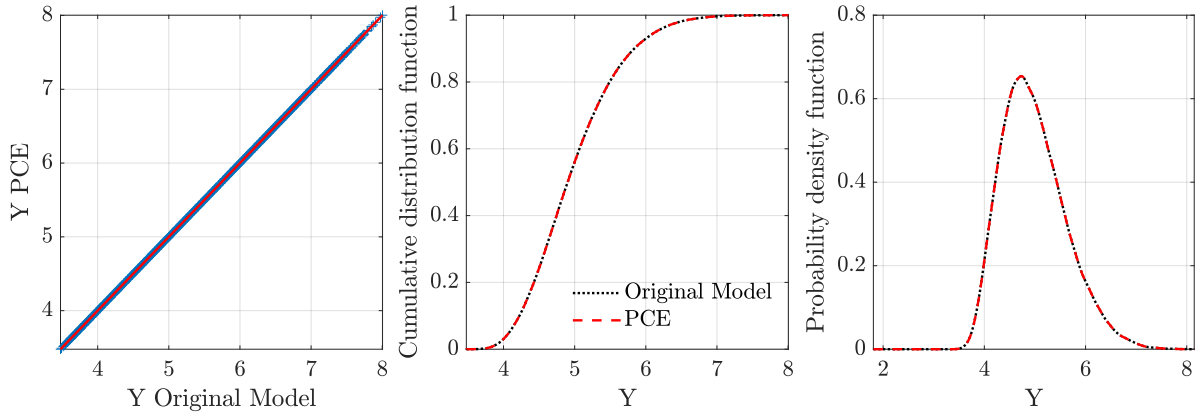


Figure 4.7: Comparison of the original output and the PCE output in the test case

Before the calibration algorithm, the sensitivities of contributing factors to output variance, so-called Sobol's indices, are computed easily based on the obtained PCE coefficients. By applying Equation (3.174), the Sobol's indices are shown in Figure 4.8. It allows us to obtain a quantitative importance measure for each contributing factor.

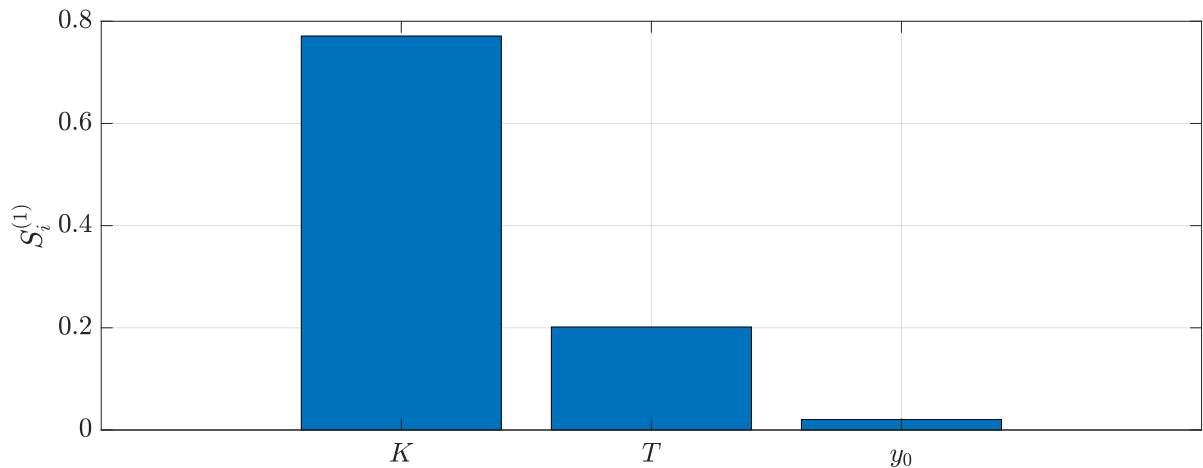
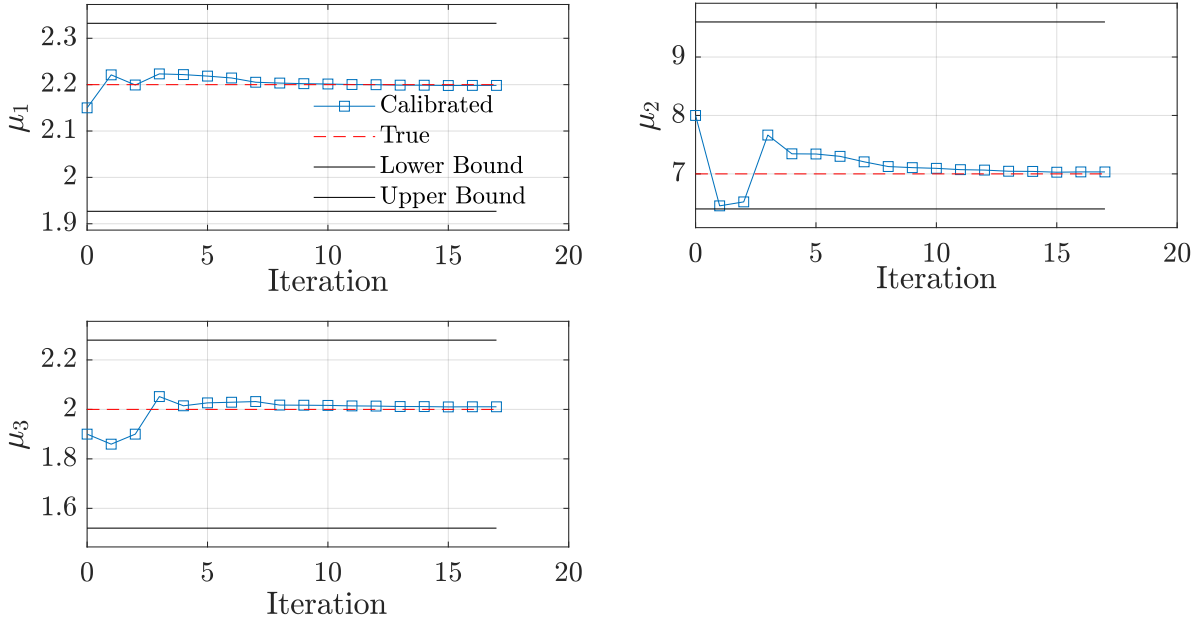
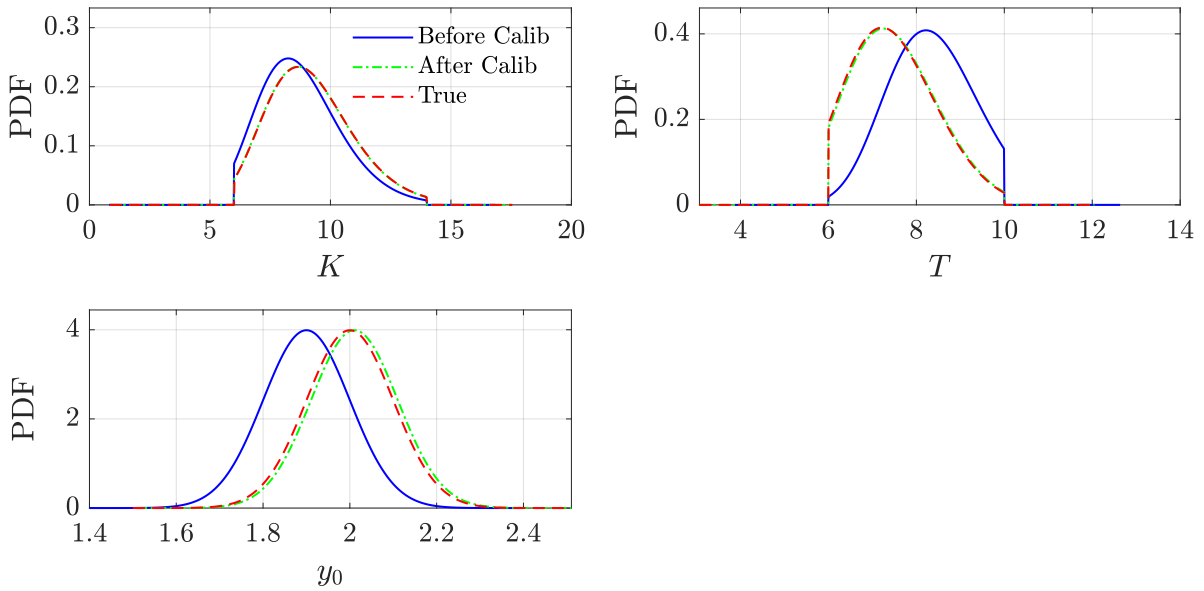


Figure 4.8: PCE-based Sobol's indices of contributing factors in the test case

Since there is no fitting error about the target distribution in this case, the cost function of the calibration is based on Equation (4.3) with the first three order statistical moments $M_Y = [\mu_Y, \sigma_Y, \delta_Y]$. The tuned parameters are μ_1 , μ_2 , and μ_3 . The shift of 20% standard deviation in the mean value of tuned parameters is chosen as the upper and lower boundaries. After the calibration, the iteration process of the tuned parameters is shown in Figure 4.9(a). The iteration process in Figure 4.9(a) shows that the tuned parameters converge to the true value. Figure 4.9(b) illustrates the calibrated distributions of the contributing factors. The calibrated contributing factors obey the true distributions.



(a) Iteration of tuned parameters in the test case



(b) Calibrated distributions of the contributing factors in the test case

Figure 4.9: Calibrated contributing factors in the test case

MCS is applied by using the calibrated contributing factors to validate the output results. As shown in Figure 4.10, the kernel-fitted output PDFs are compared. The ‘Target’ denotes the true value with the true contributing factors. The ‘Propagate Before Calib-Original’ indicates the simulated output with the shifted contributing factors, which has a deviation with the ‘Target’ results. During the calibration process, the designed samples are frozen, and they do not obey the calibrated distribution. Therefore, the output PDFs obtained using the ‘Designed Samples’ is close to the ‘Propagate Before Calib-Original.’ Furthermore, the calibrated contributing factors are simulated through the PCE model and the original model. Both simulated output PDFs match the ‘Target’ PDF very well. The calibrated first three statistical moments are also listed in Table 4.2. In summary, the PCE-based calibration algorithm works very well in this first-order system.

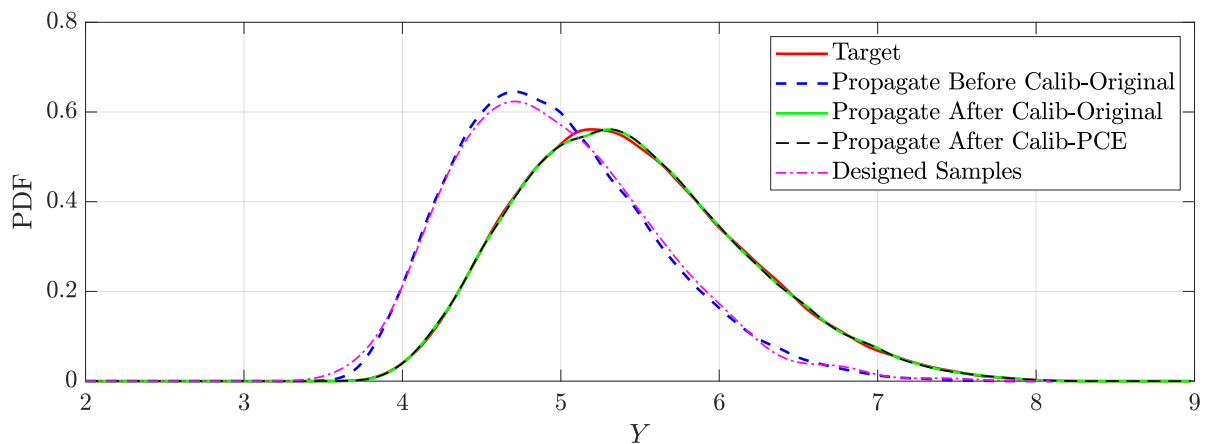


Figure 4.10: Calibration results in the test case

4.4 Calibration for the Runway Overrun Model

The proposed PCE-based calibration framework is implemented in the runway overrun. The detailed runway overrun model and all contributing factors have been introduced in Section 2.5. The distance from touchdown to 80 *knots*, denoted by d_{80} , is used as the model output for calibration. The model input contains all contributing factors X_1 to X_{10} in Table 4.3. One observation of \mathbf{X} is identified from the QAR data of a single flight. A batch of measurements for each contributing factor is obtained by gathering many flights. The ‘Cumulate’ step is carried out to estimate the best-fitted distributions for all contributing factors. The upper and lower bounds of all contributing factors are set based on the operational limits. Table 4.3 shows the ‘Cumulate’ results for runway overrun case from 552 flights of Boeing 747-8F with a full flap configuration. All those flights landed on the same runway. The estimated distributions are used as the input of the runway overrun model. An initial PCE-based surrogate model is described at first, followed by the discussion of the calibration results.

Table 4.3: Contributing factors of runway overrun model, source: Table 1 of [50]

X	Symbols	Meaning	Unit	Distribution	Parameters	Bounds
X_1	ρ	Air density	$[kg/m^3]$	Lognormal	$[\mu_1, \sigma_1] = [0.1755, 0.0261]$	$[0.9, 1.5]$
X_2	V_{hws}	Headwind speed	$[m/s]$	GEV	$[k_2, \sigma_2, \mu_2] = [-0.1753, 2.2049, -0.2243]$	$[-33, 33]$
X_3	m	Landing mass	$[kg]$	Logistic	$[\mu_3, \sigma_3] = [3.0399e5, 1.8864e4]$	$[197131, 346091]$
X_4	ΔV_{app}	Approach speed deviation	$[m/s]$	t Location-Scale	$[\mu_4, \sigma_4, \nu_4] = [-3.3728, 1.5115, 14.1691]$	$[-10, 15]$
X_5	t_{splr}	Time of deploying spoilers	$[s]$	t Location-Scale	$[\mu_5, \sigma_5, \nu_5] = [3.3944, 0.2668, 4.0864]$	$[0, 8]$
X_6	$t_{startBRK}$	Time of starting brake	$[s]$	GEV	$[k_6, \sigma_6, \mu_6] = [0.3974, 1.4242, 1.5981]$	$[-5, 130]$
X_7	t_{RevDpl}	Time of deploying reversers	$[s]$	Log-Logistic	$[\mu_7, \sigma_7] = [1.1775, 0.1425]$	$[0, 26]$
X_8	$t_{duraBRK}$	Duration of braking	$[s]$	t Location-Scale	$[\mu_8, \sigma_8, \nu_8] = [34.7409, 12.8798, 3.6911]$	$[0, 100]$
X_9	$N1_{avg}$	Average reverser fan speed	$[-]$	GEV	$[k_9, \sigma_9, \mu_9] = [0.8927, 0.0357, 0.2654]$	$[0.15, 1]$
X_{10}	$a_{x, equ}$	Equivalent acceleration	$[m/s^2]$	t Location-Scale	$[\mu_{10}, \sigma_{10}, \nu_{10}] = [-1.8366, 0.1458, 2.6290]$	$[-4, 1]$
X_{11}	x_{TD}	Touchdown distance	$[m]$	Gamma	$[a, b] = [14.1377, 31.7250]$	$[-100, 3000]$

4.4.1 Initial PCE Model

According to the calibration procedure, the initial PCE model is firstly obtained to surrogate the original runway overrun model accurately. As the obtained input distributions are arbitrary with truncated boundaries, the complexity of transformation \mathbf{T} from X space to Z space is high when the regular polynomial bases like Legendre polynomial bases are used. Alternatively, the arbitrary polynomial bases [42] are implemented for each input, respectively. For the same truncated scheme, the arbitrary polynomial bases will better represent the input distributions than the traditional Hermite or Legendre bases. The monic orthonormal polynomials $\Psi_{\alpha}(\mathbf{Z})$ are calculated based on the 3-term recurrence relations in Equation (3.144).

The classical PCE is applied at first with the truncated order $p = 6$. Based on the definition, the total degree $|\alpha|$ should be equal or less than 6. Therefore, there are $(10+6)!/10!6! = 8008$ polynomial bases in total. Considering the computational capacity, the total number of bases is too large. In order to further reduce the number of polynomial bases P , the hyperbolic PCE with $q_{norm} = 0.5$ is applied. For this truncated scheme, there are 196 polynomial bases in total. The PCE coefficients are computed using the least-square method in Algorithm 10. To validate the accuracy of the obtained PCE model, 10^5 input samples are generated to run the MCS using the obtained PCE model and the original model, respectively. The corresponding 10^5 output samples are obtained. The output CDFs and PDFs are estimated subsequently.

Figure 4.11 shows the comparison in output sample plots and distributions. Most samples are almost located in the positive diagonal line. However, there is a deviation between the original model and the PCE model using the current truncated scheme. Nevertheless, two probability distributions of the model output match very well. The obtained PCE allows us to represent the statistical behavior of the model output with high accuracy. Due to the limit of the computational capacity and efficiency, the truncated scheme $p = 6$, $q_{norm} = 0.5$ is used.

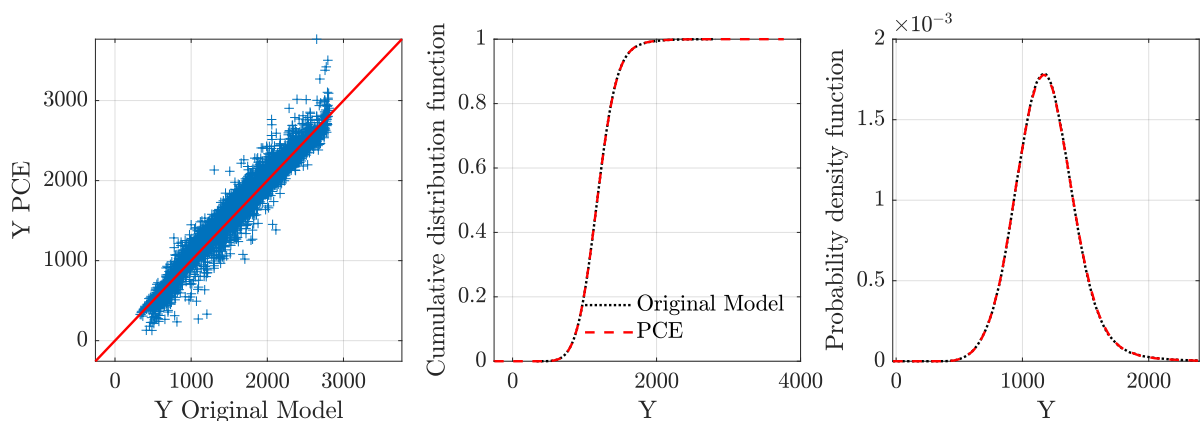


Figure 4.11: Comparison of the original output and the PCE output in runway overrun

Before the calibration, the PCE-based Sobol indices are computed and visualized in Figure 4.12. The top six important factors are

1. X_3 : Landing mass m
2. X_{10} : Equivalent acceleration $a_{x, equ}$
3. X_6 : Time to start braking $t_{startBRK}$
4. X_2 : Headwind V_{hws}
5. X_4 : Approach speed deviation ΔV_{app}
6. X_1 : Air density ρ

The remaining four factors X_5, X_7, X_8, X_9 influence less, which is due to the auto-braking system applied in the runway overrun model. The drag force caused by the spoilers, the reverse thrust, and the braking force compromise to obtain the equivalent target acceleration. Therefore, a minor shift in the mean of the remaining four factors can not introduce a large deviation in the output. Therefore, only the six influential contributing factors will be calibrated.

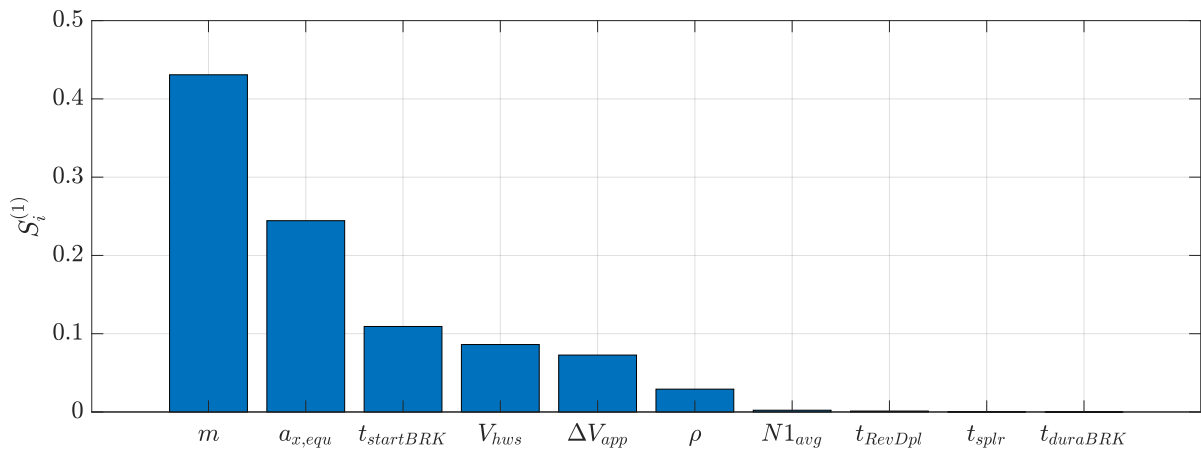


Figure 4.12: PCE-based Sobol's indices of contributing factors in runway overrun

Although the PCE-based calibration algorithm can be implemented using the current PCE model, a modification is applied to improve the accuracy of the PCE model further. Since the contributing factors X_5, X_7, X_8, X_9 have no significant influence on the output uncertainties, the model input dimensions reduce to 6 during the calibration procedure. The mean value of X_5, X_7, X_8, X_9 are taken instead of random values. The simplified model with reduced dimensions reduces the complexity of the original model. Furthermore, the simplified model allows us to increase the truncated order of the PCE model. $p = 7$, $q_{norm} = 0.5$ is used. MCS with 10^5 samples is implemented to validate the obtained PCE model shown in Figure 4.13.

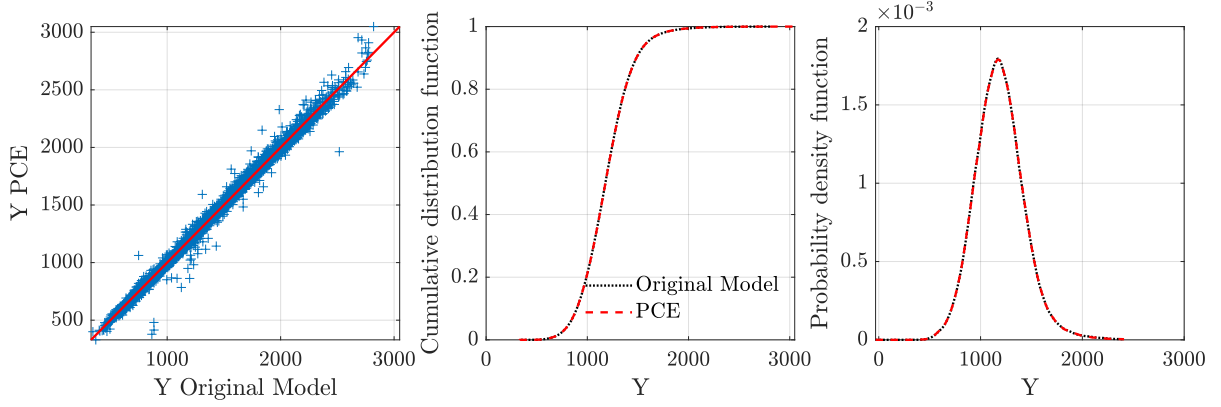


Figure 4.13: Comparison of the simplified model output and the new PCE output in runway overrun

There is only a small error in output between the new PCE model and the simplified model. Compared to Figure 4.11, the new PCE model has a better match with the simplified model output. In addition, the corresponding CDFs and PDFs match very well. The new PCE model is used in the following calibration.

4.4.2 Calibration of Contributing Factors

The distribution location parameter μ of six contributing factors is set as the tuned parameter $\theta = \{\mu_1, \mu_2, \mu_3, \mu_4, \mu_6, \mu_{10}\}$. The calibrated tolerance of the mean value is bounded based on the 20% of the corresponding standard deviation around the expectation value. The upper and lower bounds are transferred from the mean to the location parameters according to the distribution function. The mean and standard deviation of output are used in the cost function. To balance the fitting error of the input and output, the weighted functions are set as follows:

$$W_1 = \text{diag}\left(\frac{10}{\sigma_Y^2}, \frac{5}{\sigma_Y^2}\right), \quad (4.12)$$

and

$$W_2 = \frac{1}{6} \cdot \text{diag}\left(\frac{1}{\sigma_1^2}, \frac{1}{\sigma_2^2}, \frac{1}{\sigma_3^2}, \frac{1}{\sigma_4^2}, \frac{1}{\sigma_6^2}, \frac{1}{\sigma_{10}^2}\right). \quad (4.13)$$

The constant values 10 and 5 are set to increase the weight of the output distribution deviation. Contributing factors are calibrated using the Algorithm 11. Figure 4.14 shows the iteration process of the tuned distribution parameters. The comparison between the original and the calibrated input distributions is shown in Figure 4.15.

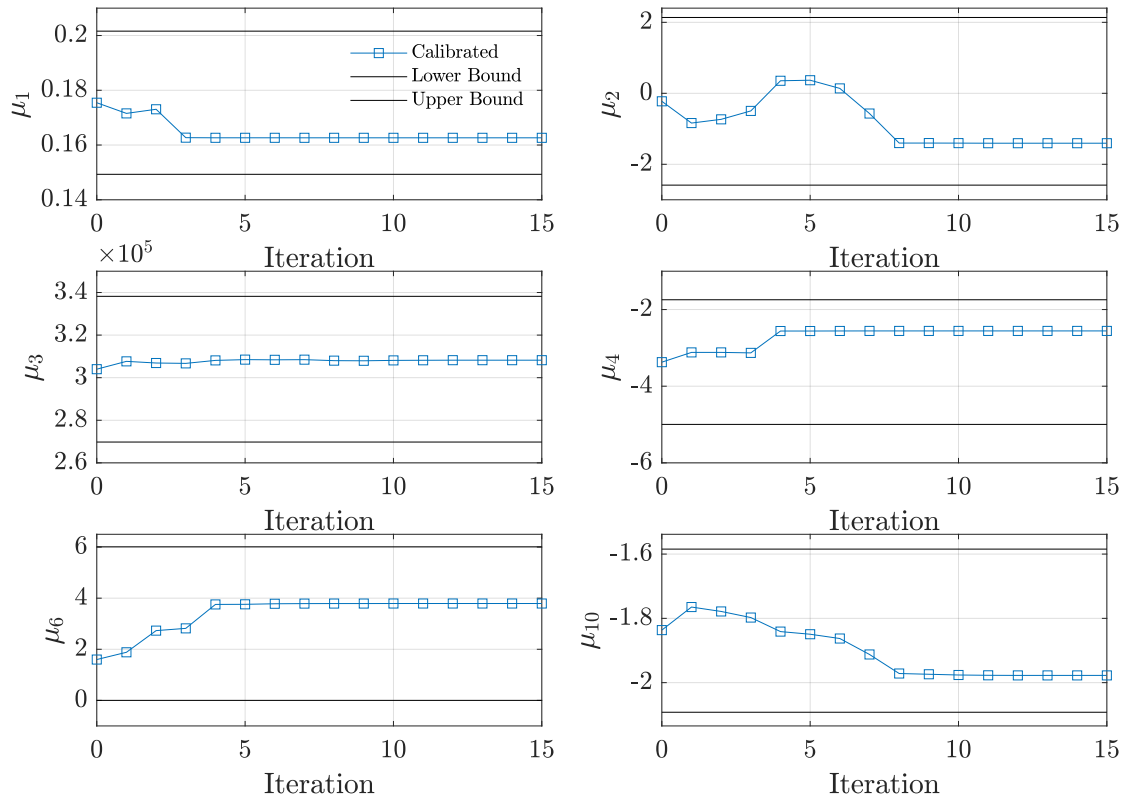


Figure 4.14: Iteration of the tuned parameters in runway overrun

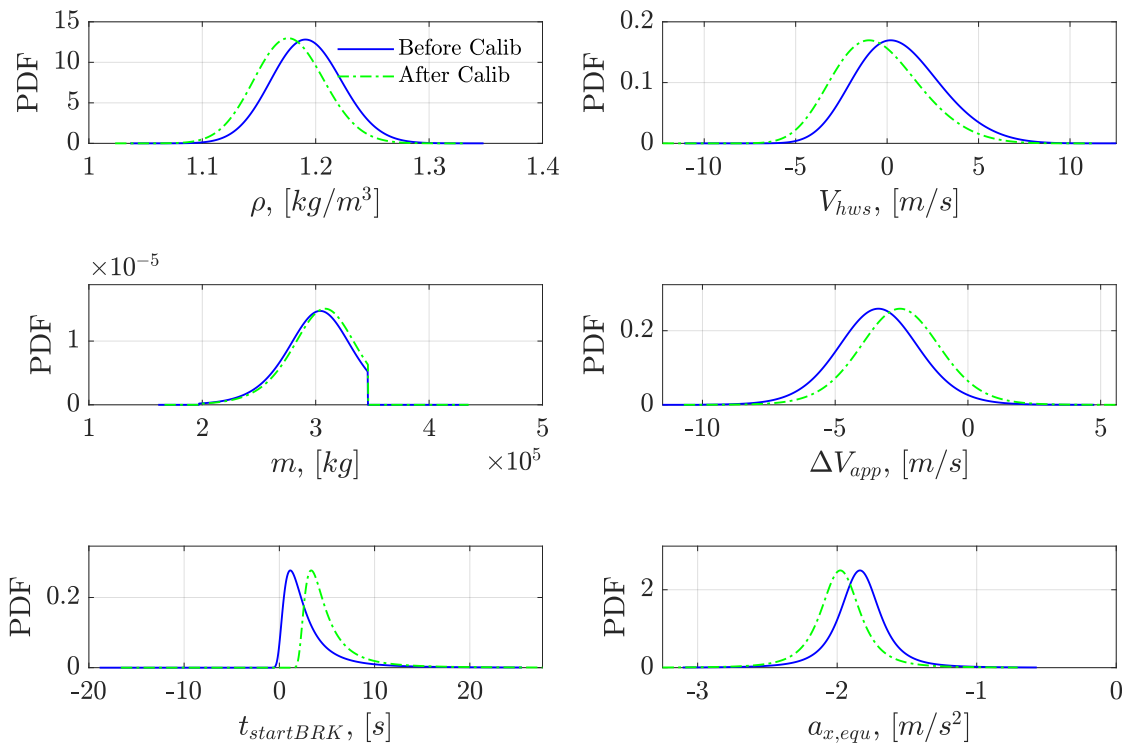


Figure 4.15: Calibrated distributions of the contributing factors in runway overrun

In order to validate the calibration results, the MCS is implemented using the calibrated contribution factors through the original model. Figure 4.16 shows the kernel-fitted output PDFs. The output distribution using the calibrated contributing factors and the original model is close to the observed distributions. In summary, the minor shift of the contributing factors allows the model output to match the observed output distribution. The calibrated contributing factors and the incident model can be used in the ‘Predict’ step.

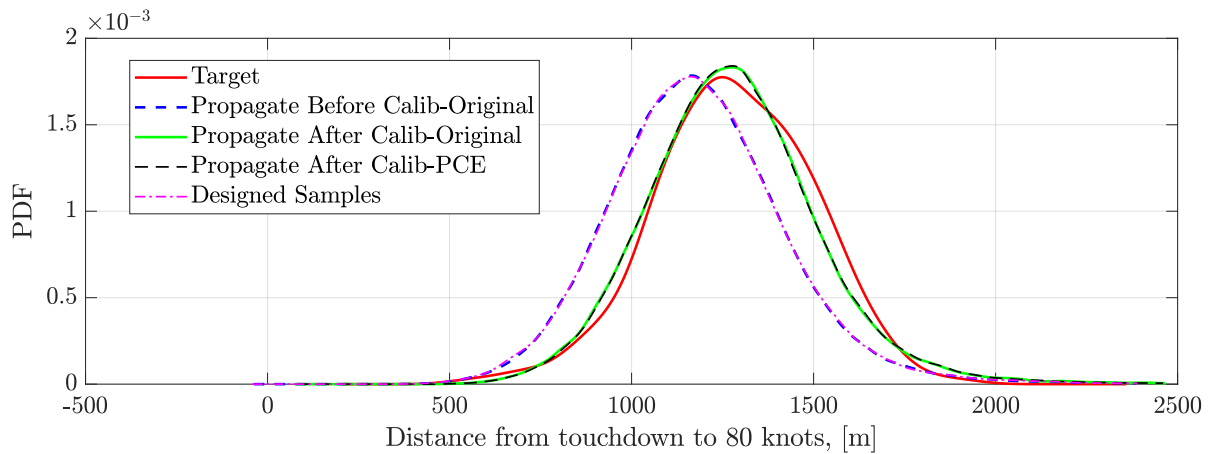


Figure 4.16: Calibration results in runway overrun

Chapter 5

Modeling of Stochastic Wind

Wind speed is a safety-critical factor for the operational flight data. A wind series generator is required to quantify the wind effect in the Predictive Analysis framework. In this chapter, a new solution is proposed to model the stochastic wind from the operational flight data based on the Karhunen–Loève (KL) expansion methods. The KL-based wind model allows us to generate the wind series, following the original statistical characteristics. The main contents of this chapter have been published in [51].

This chapter is organized as follows: the motivation of the stochastic wind modeling is introduced first in Section 5.1. Section 5.2 describes the classical stochastic process modeling methods. KL-based stochastic process reconstruction and regeneration methods are proposed in Section 5.3, followed by the comparison between the PSD approach and KL expansion in Section 5.4. The integration of vine copula into KL expansion is recommended to capture the dependence among KL coefficients in Section 5.5. In Section 5.6, the von Karman turbulence model is applied to compare the power spectral density and KL expansion coefficients. The results of KL-based wind modeling from the operational flight data are discussed in Section 5.7. A short summary is given in Section 5.8.

5.1 Motivation

In the flight dynamics and performance analysis, the discrete gust model, low-level wind shear model, and von Karman or Dryden turbulence model are widely used [118–120]. EASA also summarizes the mean wind, wind shear, and turbulence model for airworthiness assessment [121]. However, the features of wind are affected by the local terrain and climate. There is a considerable difference in the wind characteristics for different airports. These mentioned wind models do not describe the local wind features and they are not appropriate for quantifying the operational risk in the Predictive Analysis framework mentioned in Section 1.2.2. Nevertheless, as described in Chapter 2, QAR records

the real encountered wind series data for each flight. Hundreds of Flights landing on the same runway can provide a local wind series database, which allows us to analyze the local wind features statistically.

In the application of Predictive Analysis approach, wind speed is already considered as one of the contributing factors. For example, in the runway overrun model described in Section 2.5, the wind speed at the touchdown point is extracted from the QAR data. According to the measurements recorded in the QAR, one can estimate the distribution of the wind speed. Samples of wind speed generated randomly from the obtained distribution are used as the model input, and the wind speed keeps constant in one simulation of the incident model. However, the constant wind assumption is not appropriate for all accidents since the wind variation has a significant effect on flight safety. Instead of the constant wind, wind series should be taken into account in the incident modeling. The classical turbulence and wind shear model provide the wind series, but they can not describe wind's local statistical characteristics for the specific airport. Therefore, the aim of this chapter is to provide a wind series model in order to generate a wind process, which has the same statistical features as the encountered wind data. Furthermore, the generated wind process should be suitable in the MCS of the incident model. The Predictive Analysis framework with the wind process input is illustrated in Figure 5.1.

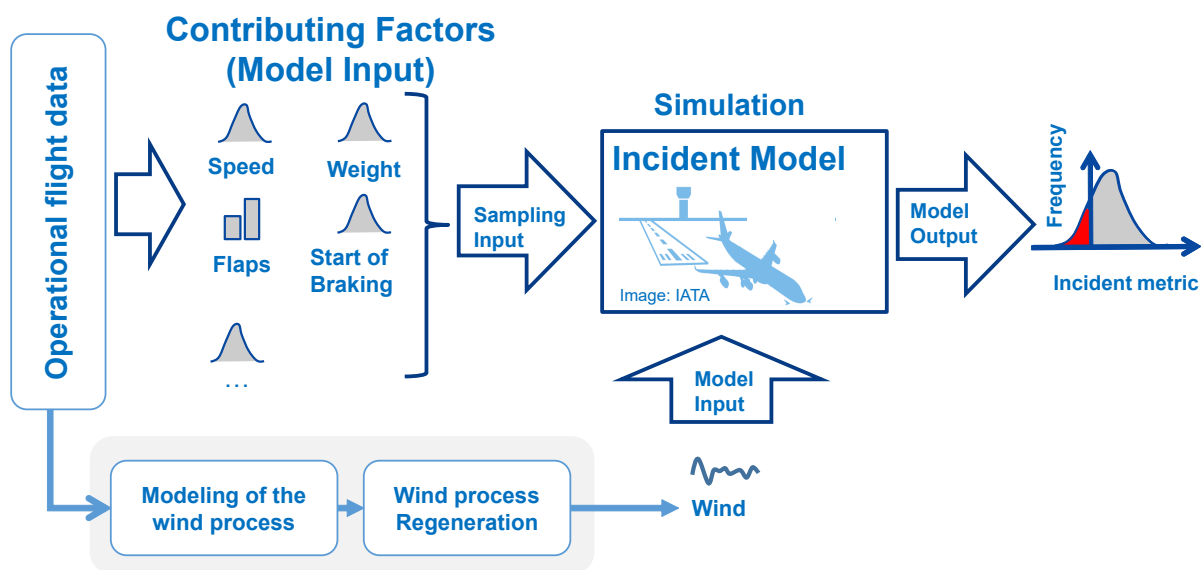


Figure 5.1: Predictive Analysis framework with the wind process input, modified from [9]

5.2 Stochastic Process Modeling Methods

Wind series can be taken as a stochastic process. Let $X(t) = \{X(t_1), X(t_2), \dots, X(t_i), \dots\}$ denote a stochastic process, and $\mathbb{E}[|X(t_i)|^2] < \infty$, $t_i \in T$. Each component $X(t_i)$ at one time point t_i is a random variable. Correlations between these random variables are usu-

ally applied to analyze the stochastic process. The correlation relationship is described using the autocorrelation and autocovariance functions as below:

$$\begin{aligned} R(t, s) &= \mathbb{E}[X(t)X(s)], \\ Cov(t, s) &= \mathbb{E}[(X(t) - \mu_t)(X(s) - \mu_s)] = R(t, s) - \mu_t\mu_s, \end{aligned} \quad (5.1)$$

where $t, s \in T$. μ_t and μ_s are the mean of the process $X(t)$ at given time t and s , respectively. The process $X(t)$ is called weakly stationary process if the mean μ does not change with the t , and $R(t, s)$ only depends on the time lag between t and s as below:

$$\begin{aligned} \mathbb{E}[X(t)] &= \mathbb{E}[X(s)] = \mu, \\ R(t, s) &= R(t + D, s + D), \quad \forall D \in T. \end{aligned} \quad (5.2)$$

The time lag between t and s is denoted by the $\tau = t - s$. Therefore, $R(t, s) = R(\tau)$. There are many approaches to further analyze the stochastic process with different assumptions on the autocorrelation. This section introduces several widely used methods for wind modeling briefly.

5.2.1 White Noise

White noise is a random process composed of a sequence of uncorrelated random variables [122]. These variables follow the same Gaussian distribution with zero mean and a finite variance. The sensor noise is usually assumed to be white noise.

5.2.2 Markov Process

According to the Markov properties, the current state only depends on the previous state [123]. Since there is no direct physical relation for the wind speed, an Estimation-Before-Modeling (EBM) approach is proposed in [124] for the parameter estimation. By applying the EBM approach, a third-order Gauss-Markov process is used to describe the dynamics of the wind as below [71, 125]:

$$\begin{pmatrix} \dot{x} \\ \ddot{x} \\ \dddot{x} \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} x \\ \dot{x} \\ \ddot{x} \end{pmatrix} + \begin{pmatrix} w_1 \\ w_2 \\ w_3 \end{pmatrix}, \quad (5.3)$$

where x denotes the wind speed, and w_1 , w_2 , and w_3 are the zero-mean Gaussian white noise. The first-order, second-order and third-order derivatives of x are driven by w_1 , w_2 , w_3 . The x , \dot{x} , \ddot{x} will also follow the Gaussian distribution through the linear transformation. In addition, the current state x only depends on the previous state. Instead of the Markov process, the auto-regressive-moving-average (ARMA) approach can also be used to describe the stochastic process, which allows the current state x depending on the several

previous states [126]. However, the Markov process and ARMA approach only work on the stationary process. In order to construct a realization of the wind time series, one or several random numbers are required to be generated from the white noise at each time point. These random variables for each time point will dramatically increase dimensions of the model input.

5.2.3 Fourier Transformation-based Spectral Representation

Both the Markov process and the ARMA model analyze the stochastic process in the time domain directly. In contrast, the Fourier transform-based spectral representation (FTSR) is often implemented to analyze the stochastic process in the frequency domain. The Fourier series (FS) expansion and Fourier transformation (FT) are introduced first for the periodic signal, and extended to the aperiodic signal. The power spectral density (PSD) obtained by the FT of autocorrelation function is used to describe the stochastic process.

Fourier Transformation

Assume that $f(t)$ is a periodic function and T_0 is the period. If $f(t)$ satisfies Dirichlet's conditions, i.e., $f(t)$ is integrable, its Fourier series expansions are as below [122]:

$$f(t) = \frac{a_0}{2} + \sum_{n=1}^{\infty} (a_n \cos(n\omega_0 t) + b_n \sin(n\omega_0 t)), \quad n = 1, 2, \dots, \quad (5.4)$$

where $\omega_0 = 2\pi/T_0$ and it denotes the angular frequency. The coefficients $a_0, a_n,$ and b_n are calculated as follows:

$$\begin{aligned} a_0 &= \frac{2}{T_0} \int_0^{T_0} f(t) dt, & a_n &= \frac{2}{T_0} \int_0^{T_0} f(t) \cos(n\omega_0 t) dt, \\ b_n &= \frac{2}{T_0} \int_0^{T_0} f(t) \sin(n\omega_0 t) dt. \end{aligned} \quad (5.5)$$

Fourier series expansions can be also written in the exponential form as below:

$$f(t) = \sum_{n=-\infty}^{\infty} F_n e^{in\omega_0 t}, \quad (5.6)$$

where F_n is the Fourier coefficients and calculated via

$$F_n = F(n\omega_0) = \frac{1}{T_0} \int_0^{T_0} f(t) e^{-in\omega_0 t} dt. \quad (5.7)$$

Based on Euler's formula $e^{ix} = \cos x + i \sin x$, the two expansions Equation (5.4) and Equation (5.6) are equivalent each other. F_n is a complex function and is written as

$$F_n = |F_n| e^{i\psi_n}, \quad (5.8)$$

where $|F_n|$ and ψ_n are the amplitude and phase of the signal $f(t)$ respectively. They are commonly used in the frequency spectrum analysis. Then $f(t)$ can be written as below:

$$\begin{aligned} f(t) &= \sum_{n=-\infty}^{\infty} |F_n| e^{i\psi_n} e^{in\omega_0 t} = F_0 + \sum_{n=1}^{\infty} 2|F_n| e^{i\psi_n} e^{in\omega_0 t} \\ &= F_0 + \sum_{n=1}^{\infty} 2|F_n| \cos(n\omega_0 t + \psi_n). \end{aligned} \quad (5.9)$$

It is worth noting that the frequency spectrum of the periodic signal is discrete. For the aperiodic signal with the finite time T_0 , the signal is assumed as a periodic signal with the period T_0 . When $T_0 \rightarrow \infty$, we have

$$\begin{aligned} \omega_0 &= 2\pi/T_0 \rightarrow 0, \\ n\omega_0 &\rightarrow \omega, \\ \Delta(\omega) &\rightarrow d\omega, \\ F(n\omega_0)/\omega_0 &\rightarrow F(\omega)/(2\pi). \end{aligned} \quad (5.10)$$

Therefore, the Fourier transformation and inverse Fourier transformation of the aperiodic signal are as below [122]:

$$F(\omega) = \int_{-\infty}^{\infty} f(t) e^{-i\omega t} dt, \quad (5.11)$$

$$f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(\omega) e^{i\omega t} d\omega. \quad (5.12)$$

Power Spectral Density

The PSD function $P(\omega)$ can be calculated based on the Fourier transformation of the autocorrelation function $R(t + \tau, t)$ [127] as below:

$$P(\omega) = \int_{-\infty}^{\infty} R(t + \tau, t) e^{-i\omega\tau} d\tau, \quad (5.13)$$

where $R(t + \tau, t) = R(\tau)$ for the stationary process. $R(\tau)$ can also be computed via the inverse Fourier transformation as follows:

$$R(\tau) = \frac{1}{2\pi} \int_{-\infty}^{\infty} P(\omega) e^{i\omega\tau} d\omega. \quad (5.14)$$

The PSD of the white noise is a constant value. In aviation, the classical von Karman or Dryden turbulence model describes the wind by using the PSD [128]. In addition, one-sided PSD is commonly used in practice since the real signal does not have the negative frequency. The one-sided PSD $S(\omega)$ is obtained as below:

$$\begin{aligned} S(\omega) &= 2P(\omega), \text{ when } \omega > 0 \\ S(0) &= P(0). \end{aligned} \quad (5.15)$$

5.3 KL-based Stochastic Process Reconstruction and Regeneration

All stochastic process modeling methods mentioned above can only be applied to describe the stationary process. In order to represent the non-stationary process and achieve a more realistic quantitative statement of the operational situation, a high fidelity model to describe the encountered wind series is required. Without any assumption about the stochastic process, a Karhunen–Loève (KL) expansion approach is applied to describe the observed non-stationary stochastic phenomena, for example, seismic ground motion in [129, 130]. This approach is also called functional principal component analysis, which has a good performance in dimension reduction.

The definition of eigenfunction and integral operator is given first. Let Q is a linear operator on a function space. Then an eigenfunction of the operator Q is a non-zero function ϕ in that space such that the application of Q on ϕ equals ϕ multiplied by a scalar factor as below [131]:

$$Q\phi = \lambda\phi, \quad (5.16)$$

where λ is a constant called the eigenvalue of the operator Q . As one type of the operator, a linear integral operator \tilde{Q} , also called an integral transformation, maps a function f from its original function space into another function space via integration:

$$f \rightarrow \tilde{Q}f : \tilde{Q}f(t) = \int_D K(t, s)f(s)ds, \quad (5.17)$$

where D is a given measurable set that $t, s \in D$. $K(t, s)$ is a continuous function and called the kernel of an integral operator. For example, the Fourier transformation in Equation (5.11) is one of integral operator.

Let $X(t)$ denote a continuous process at the given time interval $D = [0, T]$ with a zero mean at any time $t \in D$. It means that $E[X(t_i)] = 0$ for all $t_i \in D$. If $X(t)$ is integrable: $E[X^2] < \infty$, the linear operator \tilde{Q} in Equation (5.17) can be defined by setting the autocorrelation function $R(t, s)$ of the process $X(t)$ as the kernel $K(t, s)$. Based on the Mercer's theorem [135], there exists a set of eigenfunctions $\phi_k(t)$ and eigenvalues $\lambda_k \geq 0$ for the operator \tilde{Q} , and $R(t, s)$ can be decomposed as below:

$$R(t, s) = \sum_{k=1}^{\infty} \lambda_k \phi_k(t) \phi_k(s). \quad (5.18)$$

The eigenfunctions $\phi_k(t)$ and the eigenvalues λ_k of \tilde{Q} satisfy

$$\tilde{Q}\phi_k(t) = \int_D R(t, s)\phi_k(s)ds = \lambda_k\phi_k(t), \quad t, s \in D. \quad (5.19)$$

where the eigenfunctions $\phi_k(t)$ are orthonormal:

$$\langle \phi_k(t), \phi_l(t) \rangle = \int_D \phi_k(t)\phi_l(t)dt = \delta_{kl}, \quad k, l \geq 1. \quad (5.20)$$

where δ_{kl} denotes the Kronecker delta function. $\delta_{kl} = 1$, if $k = l$. Otherwise, $\delta_{kl} = 0$. The continuous process $X(t)$ can be expressed using a series of KL expansions [134]:

$$X(t) = \sum_{k=1}^{\infty} \sqrt{\lambda_k} \xi_k \phi_k(t). \quad (5.21)$$

where the scalar value λ_k and the continuous function $\phi_k(t)$ are the eigenvalue and the eigenfunction of the operator \tilde{Q} . ξ_k is the KL coefficient, and it is given by the projection of $X(t)$ on the k th eigenfunction $\phi_k(t)$:

$$\xi_k = \frac{1}{\sqrt{\lambda_k}} \int_D X(t) \phi_k(t) dt. \quad (5.22)$$

According to the properties that eigenfunctions ϕ_k are orthonormal and $X(t)$ is a zero mean process, ξ_k holds that [133]:

$$E[\xi_k] = 0, \quad Var[\xi_k] = 1, \quad E[\xi_k \xi_l] = \delta_{kl}. \quad (5.23)$$

Furthermore, the total variance of the finite zero-mean process $X(t)$ is described as the integral of the variance at each time point over the whole time interval D as follows [129]:

$$\begin{aligned} \int_D Var[X(t)] dt &= \int_D E[X^2(t)] dt = \int_D E \left[\left(\sum_{k=1}^{\infty} \sqrt{\lambda_k} \xi_k \phi_k(t_i) \right)^2 \right] dt \\ &= \int_D E \left[\sum_{k=1}^{\infty} \lambda_k (\xi_k)^2 (\phi_k(t_i))^2 \right] dt = \sum_{k=1}^{\infty} \lambda_k E[(\xi_k)^2] \int_D (\phi_k(t_i))^2 dt = \sum_{k=1}^{\infty} \lambda_k. \end{aligned} \quad (5.24)$$

The integrated variance of the process can be measured by the sum of all λ_k .

Since the measured data of the stochastic process is discrete, the discrete form of the functions used in the KL expansion is described. For a non-zero mean process $Y(t)$ measured at time points $\mathbf{T} = [t_1, t_2, \dots, t_N]$. $Y(t) = \{Y(t_1), Y(t_2), \dots, Y(t_N)\}$ denotes a discrete stochastic process with N samples. L series in total are obtained by collecting a batch of measured series data. The collected data is written in a matrix \mathbf{Y} as below:

$$\mathbf{Y} = [\mathbf{Y}(t_1), \mathbf{Y}(t_2), \dots, \mathbf{Y}(t_N)] = \begin{bmatrix} Y^1(t_1) & Y^1(t_2) & \dots & Y^1(t_N) \\ Y^2(t_1) & Y^2(t_2) & \dots & Y^2(t_N) \\ \vdots & \vdots & \ddots & \vdots \\ Y^L(t_1) & Y^L(t_2) & \dots & Y^L(t_N) \end{bmatrix}_{[L \times N]} \quad (5.25)$$

Each row of the matrix \mathbf{Y} denotes one realization of the stochastic process. Each column $\mathbf{Y}(t_i)$ denotes L realizations of $Y(t)$ at a certain time t_i : $\mathbf{Y}(t_i) = [Y^1(t_i), Y^2(t_i), \dots, Y^L(t_i)]^T$. The mean value of all realizations at a certain time t_i is computed as below:

$$\mu_{Y(t_i)} = E[\mathbf{Y}(t_i)] = \frac{1}{L} \sum_{l=1}^L Y^l(t_i). \quad (5.26)$$

Therefore, the mean of the process over time is written as below:

$$\boldsymbol{\mu}_{\mathbf{Y}} = E[\mathbf{Y}(t)] = [E[\mathbf{Y}(t_1)], E[\mathbf{Y}(t_2)], \dots, E[\mathbf{Y}(t_N)]] \quad (5.27)$$

By subtracting the $\boldsymbol{\mu}_Y$ from the process $Y(t)$, the zero-mean stochastic process is obtained, denoted by $X(t)$, and $\mathbf{X}(t_i) = \mathbf{Y}(t_i) - E[\mathbf{Y}(t_i)]$ for all $t_i \in \mathbf{T}$. Thence, the covariance of the stochastic process $Y(t)$ at $t = t_i$ and $t = t_j$ is shown as below:

$$\text{Cov}(t_i, t_j) = E[(\mathbf{Y}(t_i) - E[\mathbf{Y}(t_i)])^T (\mathbf{Y}(t_j) - E[\mathbf{Y}(t_j)])] = E[(\mathbf{X}(t_i))^T (\mathbf{X}(t_j))], \quad (5.28)$$

which is also the autocorrelation $R_X(t_i, t_j)$ of the process $X(t)$ at $t = t_i$ and $t = t_j$.

$$\begin{aligned} R_X(t_i, t_j) &= E[(\mathbf{X}(t_i))^T (\mathbf{X}(t_j))] = \frac{1}{L} (\mathbf{X}(t_i))^T (\mathbf{X}(t_j)) \\ &= \frac{1}{L} [X^1(t_i), X^2(t_i), \dots, X^L(t_i)] \cdot \begin{bmatrix} X^1(t_j) \\ X^2(t_j) \\ \vdots \\ X^L(t_j) \end{bmatrix} = \frac{1}{L} \sum_{l=1}^L X^l(t_i) X^l(t_j). \end{aligned} \quad (5.29)$$

Thence, the autocorrelation matrix \mathbf{R}_X is shown as below:

$$\mathbf{R}_X = \begin{bmatrix} R_X(t_1, t_1) & R_X(t_1, t_2) & \dots & R_X(t_1, t_N) \\ R_X(t_2, t_1) & R_X(t_2, t_2) & \dots & R_X(t_2, t_N) \\ \vdots & \vdots & \ddots & \vdots \\ R_X(t_N, t_1) & R_X(t_N, t_2) & \dots & R_X(t_N, t_N) \end{bmatrix} = \frac{1}{L} \mathbf{X}^T \mathbf{X} \quad (5.30)$$

The matrix \mathbf{R}_X is decomposed based on its eigenvalues and eigenvectors as below:

$$\begin{aligned} \mathbf{R}_X &= \mathbf{V} \boldsymbol{\Lambda} \mathbf{V}^T = [\boldsymbol{\phi}_1, \boldsymbol{\phi}_2, \dots, \boldsymbol{\phi}_N] \begin{bmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \dots & \lambda_N \end{bmatrix} \begin{bmatrix} \boldsymbol{\phi}_1 \\ \boldsymbol{\phi}_2 \\ \vdots \\ \boldsymbol{\phi}_N \end{bmatrix} \\ &= \begin{bmatrix} \phi_1(t_1) & \phi_2(t_1) & \dots & \phi_N(t_1) \\ \phi_1(t_2) & \phi_2(t_2) & & \phi_N(t_2) \\ \vdots & & \ddots & \vdots \\ \phi_1(t_N) & \phi_2(t_N) & \dots & \phi_N(t_N) \end{bmatrix} \begin{bmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \dots & \lambda_N \end{bmatrix} \begin{bmatrix} \phi_1(t_1) & \phi_1(t_2) & \dots & \phi_1(t_N) \\ \phi_2(t_1) & \phi_2(t_2) & & \phi_2(t_N) \\ \vdots & & \ddots & \vdots \\ \phi_N(t_1) & \phi_N(t_2) & \dots & \phi_N(t_N) \end{bmatrix} \end{aligned} \quad (5.31)$$

where $\boldsymbol{\Lambda}$ is the eigenvalue matrix, and $\mathbf{V} = [\boldsymbol{\phi}_1, \boldsymbol{\phi}_2, \dots, \boldsymbol{\phi}_N]$ is the eigenvector matrix. The k th eigenvector $\boldsymbol{\phi}_k$ is a column vector, which is composed of all values of the continuous function $\phi_k(t)$ at all time points $t_i \in \mathbf{T}$ as below:

$$\boldsymbol{\phi}_k = [\phi_k(t_1), \phi_k(t_2), \dots, \phi_k(t_N)]^T. \quad (5.32)$$

Therefore, one element of the matrix \mathbf{R}_X is computed using the eigenvector and the eigenvalue as follows:

$$R_X(t_i, t_j) = [\phi_1(t_i)\lambda_1, \phi_2(t_i)\lambda_2, \dots, \phi_N(t_i)\lambda_N] \begin{bmatrix} \phi_1(t_j) \\ \phi_2(t_j) \\ \vdots \\ \phi_N(t_j) \end{bmatrix} = \sum_{k=1}^N \lambda_k \phi_k(t_i) \phi_k(t_j), \quad (5.33)$$

which has N terms compared to Equation (5.18) for the continuous stochastic process. The KL expansion of the discrete stochastic process $X(t)$ at a certain time $t_i \in \mathbf{T} = \{t_1, t_2, \dots, t_N\}$ is:

$$X(t_i) = \sum_{k=1}^N \sqrt{\lambda_k} \xi_k \phi_k(t_i). \quad (5.34)$$

Let the superscript l denote the l th realization. Then, $X^l(t_i)$ denotes l th realization of $X(t_i)$. Therefore, the corresponding discrete version of calculating the random variable ξ_k for $X^l(t_i)$ is shown as below:

$$\xi_k^l = \frac{1}{\sqrt{\lambda_k}} \sum_{i=1}^N X^l(t_i) \phi_k(t_i). \quad (5.35)$$

where ξ_k^l denotes the coefficient of the process $\{X^l(t), t \in \mathbf{T}\}$ projected on the k th eigenvector $\{\phi_k(t), t \in \mathbf{T}\}$. The matrix form of the obtained ξ_k^l from L series is shown as below:

$$\begin{aligned} \boldsymbol{\xi}_{[L \times N]} &= \begin{bmatrix} \xi_1^1 & \xi_2^1 & \dots & \xi_N^1 \\ \xi_1^2 & \xi_2^2 & \dots & \xi_N^2 \\ \vdots & & \ddots & \vdots \\ \xi_1^L & \xi_2^L & \dots & \xi_N^L \end{bmatrix} = \mathbf{X}_{[L \times N]} \begin{bmatrix} 1/\sqrt{\lambda_1} & 0 & \dots & 0 \\ 0 & 1/\sqrt{\lambda_2} & & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \dots & 1/\sqrt{\lambda_N} \end{bmatrix}_{[N \times N]} \mathbf{V}_{[N \times N]} \\ &= \begin{bmatrix} X^1(t_1) & X^1(t_2) & \dots & X^1(t_N) \\ X^2(t_1) & X^2(t_2) & \dots & X^2(t_N) \\ \vdots & \vdots & \ddots & \vdots \\ X^L(t_1) & X^L(t_2) & \dots & X^L(t_N) \end{bmatrix}_{[L \times N]} \begin{bmatrix} 1/\sqrt{\lambda_1} & 0 & \dots & 0 \\ 0 & 1/\sqrt{\lambda_2} & & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \dots & 1/\sqrt{\lambda_N} \end{bmatrix}_{[N \times N]} \\ &\quad \begin{bmatrix} \phi_1(t_1) & \phi_2(t_1) & \dots & \phi_N(t_1) \\ \phi_1(t_2) & \phi_2(t_2) & & \phi_N(t_2) \\ \vdots & & \ddots & \vdots \\ \phi_1(t_N) & \phi_2(t_N) & \dots & \phi_N(t_N) \end{bmatrix}_{[N \times N]} \end{aligned} \quad (5.36)$$

The k th column of the obtained $\boldsymbol{\xi}$ denotes L realizations of the corresponding random variable ξ_k . Furthermore, the marginal distributions of ξ_k can be estimated based on the computed realizations $\{\xi_k^l, l = 1, 2, \dots, L\}$ using the maximum likelihood methods. As mentioned in Equation (5.23), the ξ_k are distributed with zero mean and unit variance. Moreover, the KL coefficients ξ_k are linearly uncorrelated since $E[\xi_k \xi_l] = \delta_{kl}$. However, there might be nonlinear dependence existing among the ξ_k for the non-Gaussian stochastic process. The estimated distribution of ξ_k allows us to generate new samples of ξ_k , which is further used to construct new time series of the stochastic process.

In practice, the KL expansion in Equation (5.34) is truncated at a certain order K . A ratio of the achieved variance to the total variance is called cumulative variance ratio, which is used to assess the accuracy of the KL reconstruction:

$$\eta = \frac{\sum_{k=1}^K \lambda_k}{\sum_{k=1}^N \lambda_k}. \quad (5.37)$$

$\epsilon = 1 - \eta$ denotes the relative error of the reconstruction. In order to reduce the relative error ϵ with a truncated order, λ_k is sorted in the order of descent: $\lambda_1 \geq \lambda_2 \geq \dots \geq 0$. Therefore, the most dominant terms are selected. The calculated realizations of ξ_k allow us to reconstruct the original stochastic process using K coefficients as below:

$$Y(t) \approx \boldsymbol{\mu}_{Y(t_i)} + \sum_{k=1}^K \sqrt{\lambda_k} \xi_k \phi_k(t), t \in \{t_1, t_2, \dots, t_N\}. \quad (5.38)$$

By generating a new sample of $\boldsymbol{\xi}' = [\xi'_1, \xi'_2, \dots, \xi'_K]$, the new stochastic process is produced using the Equation (5.38). The new discrete process \mathbf{Y} is shown in a matrix form:

$$\mathbf{Y}_{[1 \times N]} = (\boldsymbol{\mu}_{\mathbf{Y}})_{[1 \times N]} + \boldsymbol{\xi}'_{[1 \times K]} \begin{bmatrix} \sqrt{\lambda_1} & 0 & \dots & 0 \\ 0 & \sqrt{\lambda_2} & & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \dots & \sqrt{\lambda_K} \end{bmatrix}_{[K \times K]} (\mathbf{V}^T)_{[K \times N]}. \quad (5.39)$$

The detailed algorithm of the KL-based stochastic process reconstruction and regeneration is described in Algorithm 12.

5.4 Comparison between the PSD Approach and the KL Expansion Method

The KL expansions approach is one of the spectral representation methods, but it is not in the frequency domain. Some researchers concluded that the PSD and the KL expansion approaches coincide for the weakly stationary process [136, 137]. The relationship between the KL expansion and the PSD method is derived and summarized in the section.

As mentioned in Section 5.2.3, a zero-mean stationary stochastic process $X(t)$ within a finite time interval $t \in [0, T]$ can be expressed using the Fourier series expansion as below:

$$X(t) = \sum_{k=-\infty}^{\infty} \mathcal{F}_k e^{ik\Delta\omega t} = \sum_{k=-\infty}^{\infty} |\mathcal{F}_k| e^{i\psi_k} e^{i\omega_k t}, \quad (5.40)$$

where \mathcal{F}_k denotes the Fourier coefficient, which is calculated using Equation 5.7. ω_k is the angular frequency: $\omega_k = k\Delta\omega = k \cdot 2\pi/T$. $|\mathcal{F}_k|$ denotes the frequency amplitude. ψ_k is the corresponding phase angle. The average power of $X(t)$ in the time window $[0, T]$ is defined as

$$P = \frac{1}{T} \int_0^T X^2(t) dt = \frac{1}{T} \int_0^T \left| \sum_{k=-\infty}^{\infty} \mathcal{F}_k e^{i\omega_k t} \right|^2 dt = \int_0^T \left| \sum_{k=-\infty}^{\infty} \mathcal{F}_k \frac{e^{i\omega_k t}}{\sqrt{T}} \right|^2 dt, \quad (5.41)$$

Algorithm 12 Procedure of the KL-based stochastic process reconstruction and regeneration, source: Algorithm 1 of [51]

- 1: Collect L series of the discrete stochastic process $Y(t), t = \{t_1, t_2, \dots, t_N\}$ in a matrix form: $\mathbf{Y} = [\mathbf{Y}(t_1), \mathbf{Y}(t_2), \dots, \mathbf{Y}(t_N)]$, where $\mathbf{Y}(t_i) = [Y^1(t_i), Y^2(t_i), \dots, Y^L(t_i)]^T$.
 - 2: Obtain the realizations of the zero-mean process $X(t): \mathbf{X} = [\mathbf{X}(t_1), \mathbf{X}(t_2), \dots, \mathbf{X}(t_N)]$ where $\mathbf{X}(t_i) = \mathbf{Y}(t_i) - \mu_{Y(t_i)}$
 - 3: Calculate the autocorrelation matrix \mathbf{R}_X of $X(t)$ using Equation (5.30)
 - 4: Obtain the eigenvectors ϕ_k and eigenvalues λ_k via the eigendecomposition of \mathbf{R}_X using Equation (5.31)
 - 5: Calculate the cumulative variance ratio η using Equation (5.37) and set the truncated order K
 - 6: Calculate the coefficients ξ_k , where $k = 1, 2, \dots, K$ using Equation (5.36)
 - 7: **if** Reconstruct the original process **then**
 - 8: Calculate the Y_{rec} using the computed ξ_k based on the Equation (5.38)
 - 9: **end if**
 - 10: **if** Regenerate the new process **then**
 - 11: Estimate the distribution of ξ_k
 - 12: Generate one sample $\xi' = [\xi'_1, \xi'_2, \dots, \xi'_K]$ from the estimated distribution
 - 13: Calculate the Y_{new} using the sample ξ' based on the Equation (5.38)
 - 14: **end if**
-

where the exponential terms $e^{i\omega_k t}/\sqrt{T}$ in Equation (5.41) are orthogonal:

$$\left\langle \frac{e^{i\omega_k t}}{\sqrt{T}}, \frac{e^{i\omega_l t}}{\sqrt{T}} \right\rangle = \int_0^T \frac{e^{i\omega_k t}}{\sqrt{T}} \frac{e^{-i\omega_l t}}{\sqrt{T}} dt = \delta_{kl}. \quad (5.42)$$

Therefore, the average power P can be rewritten as below:

$$P = \left\langle \sum_{k=-\infty}^{\infty} \mathcal{F}_k \frac{e^{i\omega_k t}}{\sqrt{T}}, \sum_{k=-\infty}^{\infty} \mathcal{F}_k \frac{e^{i\omega_k t}}{\sqrt{T}} \right\rangle = \sum_{k=-\infty}^{\infty} |\mathcal{F}_k|^2 \left\langle \frac{e^{i\omega_k t}}{\sqrt{T}}, \frac{e^{i\omega_k t}}{\sqrt{T}} \right\rangle = \sum_{k=-\infty}^{\infty} |\mathcal{F}_k|^2, \quad (5.43)$$

Let $P_k = |\mathcal{F}_k|^2$ denotes the average power for the specific ω_k . $|\mathcal{F}_k|$ can be substituted by $\sqrt{P_k}$ in Equation (5.40):

$$X(t) = \sum_{k=-\infty}^{\infty} \sqrt{P_k} e^{i\psi_k} e^{i\omega_k t} = \sum_{k=-\infty}^{\infty} \sqrt{P_k T} e^{i\psi_k} \frac{e^{i\omega_k t}}{\sqrt{T}}. \quad (5.44)$$

Furthermore, $R(\tau)$ in Equation (5.14) can be rewritten using the Fourier series expansion:

$$R(\tau) = \sum_{k=-\infty}^{\infty} P_k e^{i\omega_k \tau} = \sum_{k=-\infty}^{\infty} P_k e^{i\omega_k(t-s)} = \sum_{k=-\infty}^{\infty} P_k T \left\langle \frac{e^{i\omega_k t}}{\sqrt{T}}, \frac{e^{i\omega_k s}}{\sqrt{T}} \right\rangle, \quad (5.45)$$

where $t, s \in T$. According to Equation (5.10), the $P(\omega)$ of an infinite stationary process can be computed by taking the limit of $P_k T$ when $T \rightarrow \infty$ as below:

$$P(\omega) = \lim_{\Delta\omega_k \rightarrow 0} \frac{2\pi P_k}{\Delta\omega_k} = \lim_{T \rightarrow \infty} P_k T, \text{ for } \omega_k \in [\omega_{k-1}, \omega_k], \quad (5.46)$$

where $\Delta\omega_k = \omega_k - \omega_{k-1}$. Compare the Fourier series expansion in Equation (5.44) with the KL expansion in Equation (5.21), and the autocorrelation equation in Equation (5.45) with that in Equation (5.18), respectively. The $P_k T$ and $e^{i\omega_k t}/\sqrt{T}$ obtained in Fourier series expansions are the solution of the Equation (5.19). Furthermore, the $P_k T$ and $e^{i\omega_k t}/\sqrt{T}$ corresponds to the eigenvalue λ_k and the eigenfunction $\phi_k(t)$, respectively. It shows that the Fourier series expansion (in PSD approach) and the KL expansion coincide in the reconstruction of the finite stationary stochastic process. Noticed that $P(\omega)$ of the infinite stationary stochastic process is a continues function with two sides: the positive frequency $\omega > 0$ and the negative frequency $\omega < 0$. Since the KL expansion is used in the finite process, the eigenvalue λ_k in KL expansion is always discrete, and the corresponding angular frequency ω_k is always larger than 0.

5.5 Dependence among KL Coefficients

As mentioned in Equation (5.23), the KL coefficients ξ_k might not be independent if the process is non-Gaussian and ξ_k is also non-Gaussian distributed. The dependence among the KL coefficients should be analyzed to construct the actual process better. Poirion [132] first used several most dependent pairs of variables to capture the bivariate dependence of ξ_k . Later, the Gaussian kernel estimator is applied to describe the dependence among the KL coefficients in paper [129]. The joint distribution of $\{\xi_1, \xi_2, \dots\}$ is modeled as a multivariate Gaussian mixture [138]. Furthermore, the truncated order in KL expansion might be set as 20 or more to achieve the accurate reconstruction. It is difficult to describe the dependence using a high-dimensional joint distribution density. As mentioned in Section 3.2.2, the vine copula approach can reconstruct the high-dimensional dependence function using several bivariate copula densities. The advanced vine copula approach is applied to capture the high-dimensional dependence among ξ_k in this thesis. The procedure of using the vine copula approach in the KL expansion is discussed in detail in Section 5.7.

5.6 Turbulence Modeling

The well-known von Karman turbulence model in aviation describes the turbulence using the PSD function, which is compared to the KL-based turbulence model in this section. The turbulence series are simulated from the von Karman model, further used to construct the KL expansion model. The regenerated turbulence series using the KL expansion model is compared with the original turbulence series in the time domain and the frequency domain.

5.6.1 Von Karman Model

Assuming an aircraft flies at a certain speed V through a frozen turbulence field, the PSD functions of the von Karman turbulence model in three directions are as follows [128]:

$$\begin{aligned}\Phi_u(\Omega) &= \sigma_u^2 \frac{2L_u}{\pi} \frac{1}{[1 + (1.339L_u\Omega)^2]^{5/6}}, & (\text{Longitudinal}) \\ \Phi_v(\Omega) &= \sigma_v^2 \frac{L_v}{\pi} \frac{1 + \frac{8}{3}(1.339L_v\Omega)^2}{[1 + (1.339L_v\Omega)^2]^{11/6}}, & (\text{Lateral}) \\ \Phi_w(\Omega) &= \sigma_w^2 \frac{L_w}{\pi} \frac{1 + \frac{8}{3}(1.339L_w\Omega)^2}{[1 + (1.339L_w\Omega)^2]^{11/6}}, & (\text{Vertical})\end{aligned}\quad (5.47)$$

where u, v, w denote the turbulence speed in three directions: longitudinal, lateral, and vertical. $L_u, L_v,$ and L_w (in feet) are the corresponding length scale of the turbulence. $\sigma_u, \sigma_v,$ and σ_w refer to the turbulence intensity measures in three directions. The spatial frequency Ω is defined as the angular frequency ω divided by the speed V . The PSD with respect to ω has the relation with the $\Phi(\Omega)$:

$$S(\omega) = \frac{\Phi(\Omega)}{V} = \frac{\Phi(\omega/V)}{V}. \quad (5.48)$$

For the altitude h (in feet) less than 1000 feet, the L and σ are defined as follows:

$$\begin{aligned}L_w &= h, & L_u = L_v &= \frac{h}{(0.177 + 0.000823h)^{1.2}}, \\ \sigma_w &= 0.1V_{W20}, & \sigma_u = \sigma_v &= \frac{\sigma_w}{(0.177 + 0.000823h)^{0.4}},\end{aligned}\quad (5.49)$$

where V_{W20} denotes the wind speed measured at $h = 20$ feet. A unit variance band-limited white noise signal is required to generate a turbulence process with the predefined PSD characteristics. During the simulation, the random samples are generated from the white noise generator at each time point, subsequently passed through the approximated transfer functions of Equation (5.47). The output signal of the transfer function holds the predefined PSD characteristics. The approximated transfer functions of the von Karman model are shown as below [128]:

$$\begin{aligned}H_u(s) &= \frac{\sigma_u \sqrt{\frac{2L_u}{\pi V}} \left(1 + 0.25 \frac{L_u}{V} s\right)}{1 + 1.357 \frac{L_u}{V} s + 0.1987 \left(\frac{L_u}{V}\right)^2 s^2}, & (\text{Longitudinal}) \\ H_v(s) &= \frac{\sigma_v \sqrt{\frac{L_v}{\pi V}} \left(1 + 2.7478 \frac{L_v}{V} s\right) + 0.3398 \left(\frac{L_v}{V}\right)^2 s^2}{1 + 2.9958 \frac{L_v}{V} s + 0.19754 \left(\frac{L_v}{V}\right)^2 s^2 + 0.1539 \left(\frac{L_v}{V}\right)^3 s^3}, & (\text{Lateral}) \\ H_w(s) &= \frac{\sigma_w \sqrt{\frac{L_w}{\pi V}} \left(1 + 2.7478 \frac{L_w}{V} s\right) + 0.3398 \left(\frac{L_w}{V}\right)^2 s^2}{1 + 2.9958 \frac{L_w}{V} s + 0.19754 \left(\frac{L_w}{V}\right)^2 s^2 + 0.1539 \left(\frac{L_w}{V}\right)^3 s^3}. & (\text{Vertical})\end{aligned}\quad (5.50)$$

5.6.2 Modeling of Turbulence Using KL Expansion

For a given flight condition, for example, $h = 600$ feet, $V_{W20} = 15$ knots, and $V = 140$ knots, the PSDs of the von Karman model can be computed using Equation (5.47). Figure 5.2 shows the PSDs of the longitudinal, lateral, and vertical turbulence. Based on the white noise generator and the transfer function in Equation (5.50), 2000 turbulence series with a length of 256 seconds in 16 hertz are generated for each direction. Figure 5.5 shows one example of the generated turbulence in three directions. The KL expansion method is implemented based on the generated 2000 turbulence series according to the procedure in Algorithm 12.

The sorted eigenvalues λ_k of the autocorrelation matrix $R(\tau)$ are plotted along with the index k in Figure 5.2. Compared with the PSD, the sorted λ_k has a similar shape with the PSD. As discussed in Section 5.4, λ_k in the KL expansion is equivalent to $P_k T$, which is the discrete form of the $P(\omega)$. The index k corresponds to a certain angular frequency ω_k . Furthermore, the original PSD of the von Karman turbulence model decreases with an increase of ω . Thereby, the decreasing PSD shows the same feature with the sorted λ_k . This result validates the KL-based turbulence model coincides with the traditional PSD-based turbulence model.

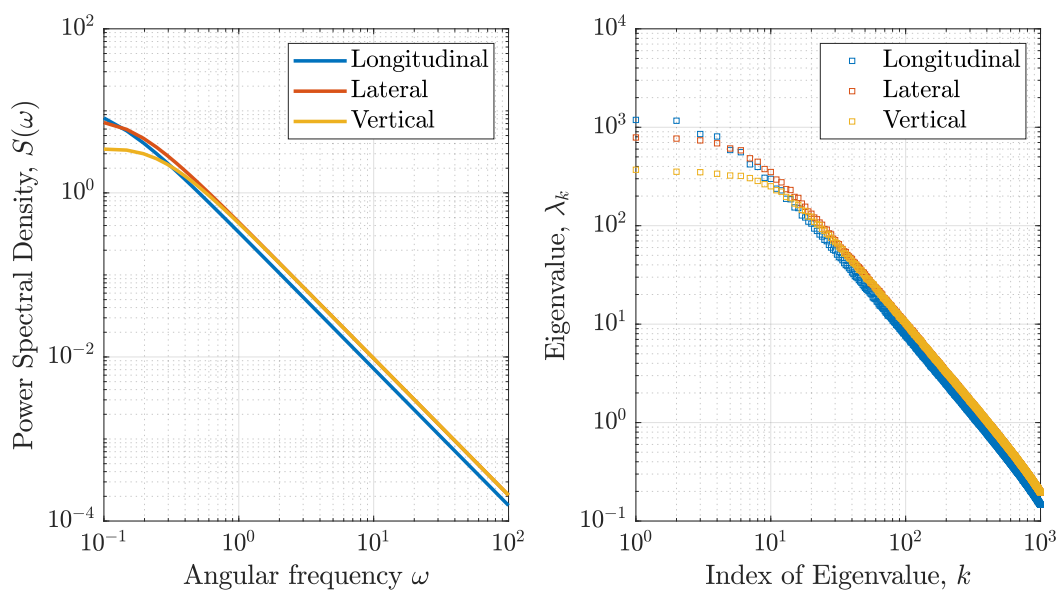


Figure 5.2: Comparison of the PSD and the eigenvalue in KL expansion for turbulence data in a certain flight condition, source: Figure 1 of [51]

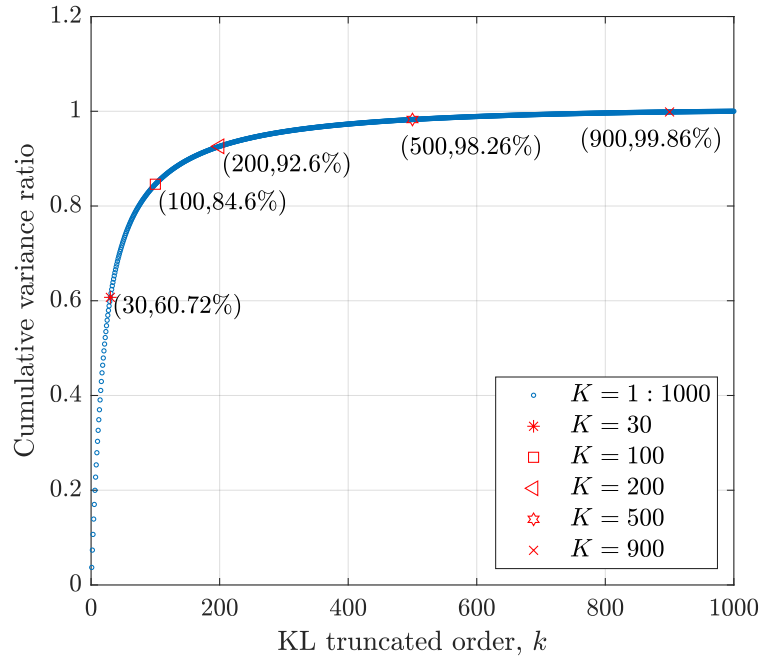


Figure 5.3: Cumulative variance ratio in KL expansion of the longitudinal turbulence data, modified from Figure 3a of [51]

As mentioned, the original turbulence series data used in the KL expansion has 256 seconds in 16 hertz. Therefore, there are $N = 256 \times 16 = 4096$ time points in a process. In order to quantify the effect of the truncated order $K (K \leq N)$ in KL expansion, the cumulative variance ratios η for K from 1 to 1000 are calculated using Equation (5.37). The η for the longitudinal turbulence is shown in Figure 5.3. It indicates that the KL-based turbulence model with $K = 100$ can achieve 84.6%. 92.6%, 98.26%, and 99.86% of variance are achieved for $K = 200$, $K = 500$, and $K = 900$, respectively. 2000 longitudinal turbulence series are regenerated from the KL-based turbulence model for $K = 30, 100, 200, 500, 900$ separately. The PSDs of the original turbulence data and the KL-based regenerated turbulence data are computed and shown in Figure 5.4. The range of frequency is plotted from 0 to 8 hertz (corresponding to the ω of $2\pi \times 8 = 50.2655 \text{ rad/s}$), since the sampling rate is 16 hertz. 3 hertz is the considered frequency in flight dynamics. The longitudinal von Karman PSD function in Equation (5.47) is marked using the black dash line, denoted by ‘theoretical.’ The ‘raw data’ denotes the PSD of the original turbulence data simulated from the von Karman turbulence model. $K = 30, 100, 200, 500, 900$ shows the PSD of the KL-based regenerated turbulence data for the corresponding truncated order. It shows that the high K will improve the accuracy of the obtained PSD, which is the same as indicated by the cumulative variance ratio. Furthermore, the truncated KL expansion of the turbulence data neglects the relatively high-frequency signal. Therefore, the truncated order K can be selected according the considered frequency and the achieved PSD.

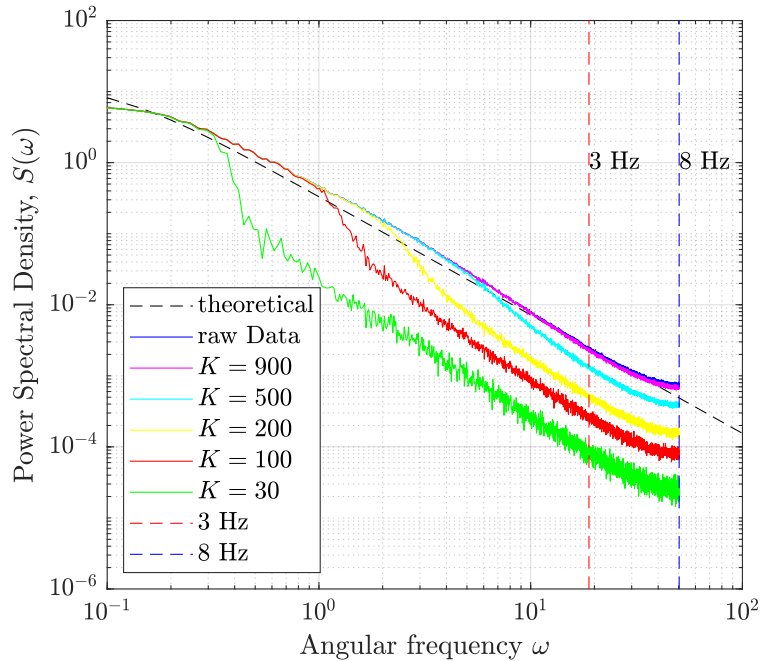


Figure 5.4: PSDs of the original turbulence data and the KL-based regenerated turbulence data, modified from Figure 3b of [51]

The KL-based turbulence data is further compared with original turbulence data in the time domain. Three truncation orders $K = 30$, $K = 200$, and $K = 900$ are applied to reconstruct the turbulence data. The KL reconstruction results of the turbulence series in the longitudinal direction are shown in Figure 5.5. Results indicate that the KL reconstructed turbulence captures the dominant trend in the original turbulence simulated from the PSD. The reconstructed turbulence using $K = 900$ has a high accuracy than that using $K = 30$ and $K = 200$. The matching accuracy of the KL reconstruction can be improved by increasing the truncated order. Since the high order KL term is related to the high-frequency signal in the turbulence case, higher frequencies are neglected in the KL-based turbulence with the low truncated order. As shown in Figure 5.2, the theoretical PSD of the original turbulence is continuous without the dominant signal in specific frequencies. Therefore, turbulence requires a high number of discrete frequency signals (corresponding to the KL terms) to capture the high-frequency signal. Nevertheless, the KL-based turbulence model constructed from the turbulence data is consistent with the conventional von Karman model by using enough terms. This approach allows us to regenerate the turbulence process for a given frequency range.

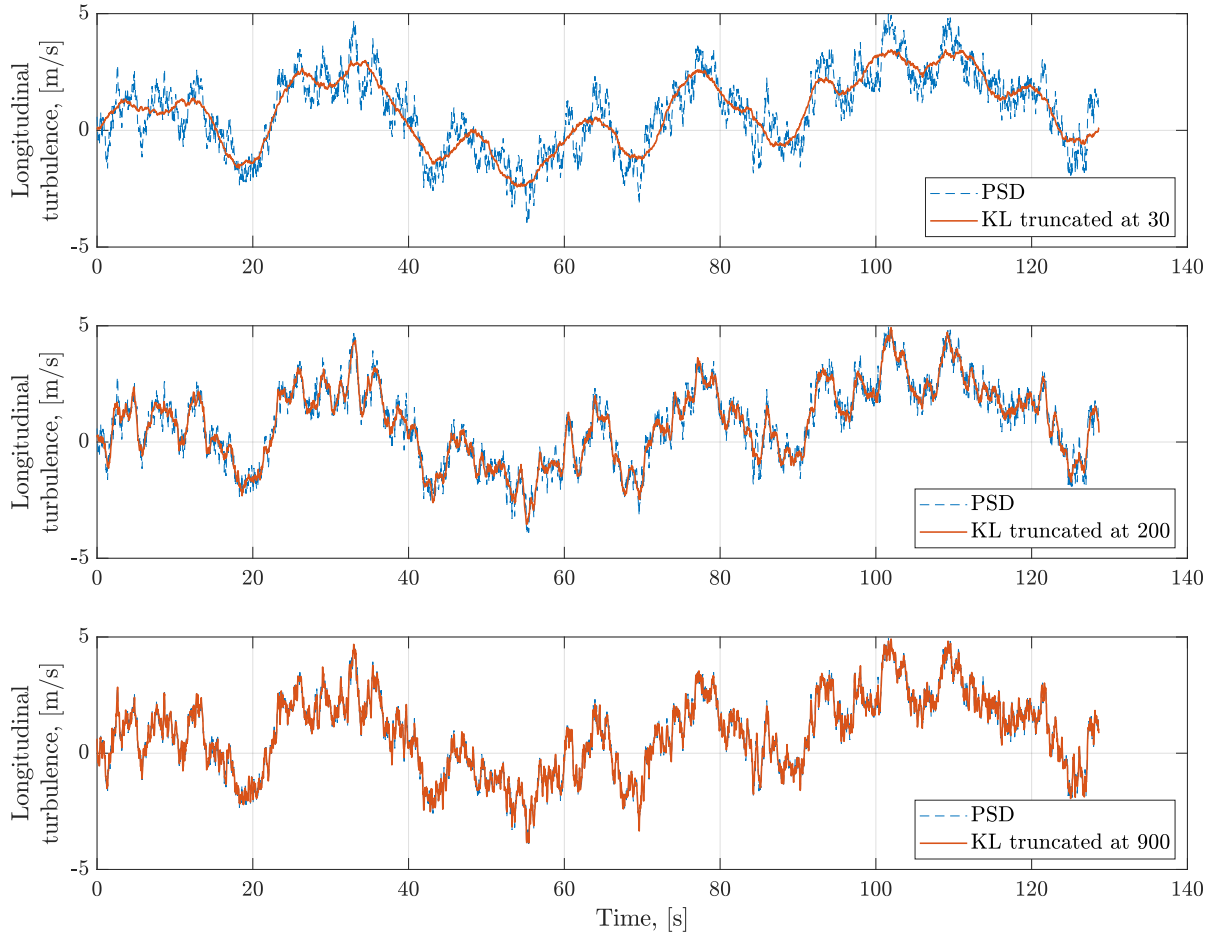


Figure 5.5: Comparison between the generated turbulence based on the PSD and the reconstructed turbulence using the KL expansion with different truncated orders.

5.7 KL-based Wind Series Regeneration from the Operational Flight Data

Based on the encountered wind series of daily flights, the proposed KL expansion method is applied to reconstruct and regenerate the wind series in this section. The encountered wind series are extracted from 849 operational flights of a B747-8F aircraft landing on the same runway. By implementing the KL expansion approach, the headwind regeneration results are presented first, followed by the regeneration of horizontal wind shear series.

5.7.1 Headwind Regeneration

Headwind (or tailwind) is one of the contributing factors for flight safety, especially during the final approach to landing. The negative headwind speed means tailwind in this thesis. The operational flight data during the final approach phase is preprocessed using the Rauch-Tung-Striebel smoother to improve the recorded flight data quality [125]. The

headwind time series from 1000 feet to 50 feet above ground level (AGL) is computed from the smoothed flight data. Since the length of the headwind time series in the final approach is different in different flights, 849 series of headwind data are resampled from the headwind time series based on the altitude. All headwind series are homogenized and have the same length after resampling shown in Figure 5.6.

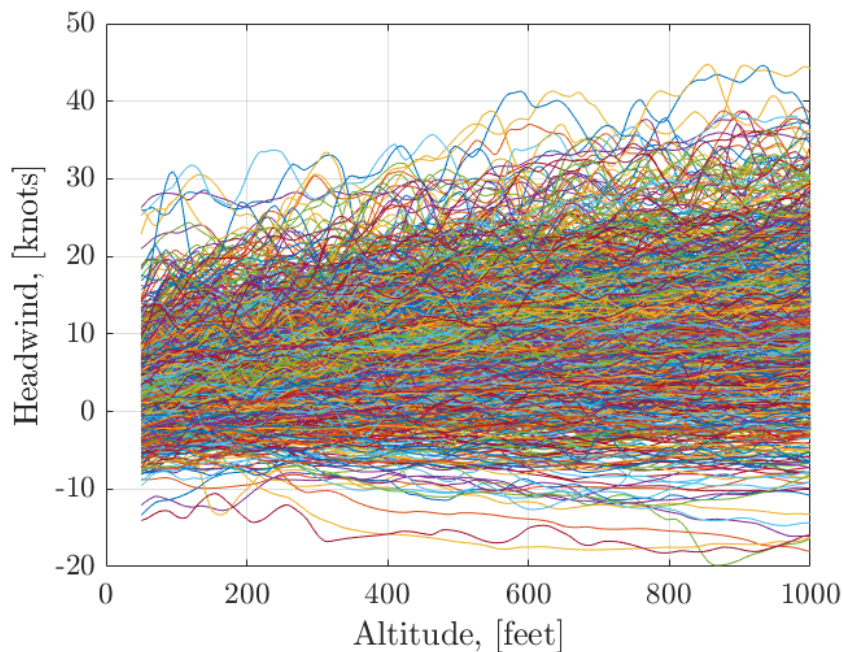


Figure 5.6: *Headwind in the final approach phase from 849 flights, source: Figure 4a of [51].*

The KL expansion approach is applied for the 849 headwind series. The eigenvalues λ_k and eigenfunctions ϕ_k of the autocorrelation matrix are computed. The cumulative variance ratio for the truncated order K from 1 to 30 is shown in Figure 5.7. It shows that the constructed KL expansions can obtain 99.9% variance of the process using 20 terms. $K = 20$ is chosen in the KL reconstruction and regeneration. Figure 5.8 shows the eigenfunctions ϕ_k and highlights the first four eigenfunctions. The shapes of the eigenfunctions are close to the harmonic.

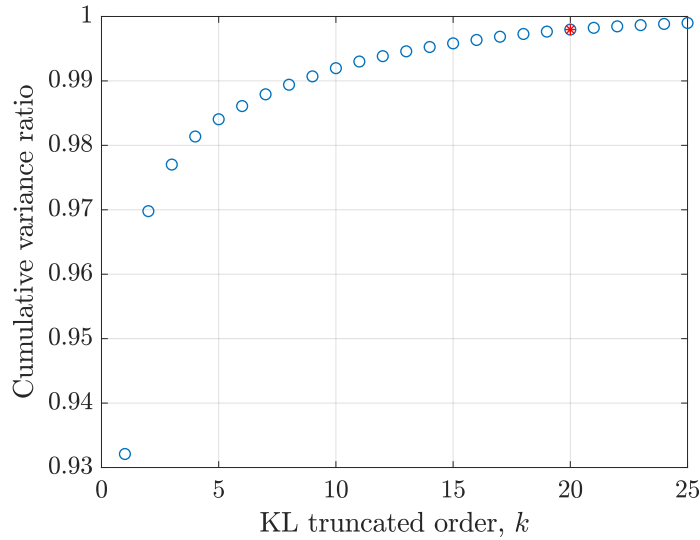


Figure 5.7: Cumulative variance ratio in KL expansion of the headwind, source: Figure 4b of [51].

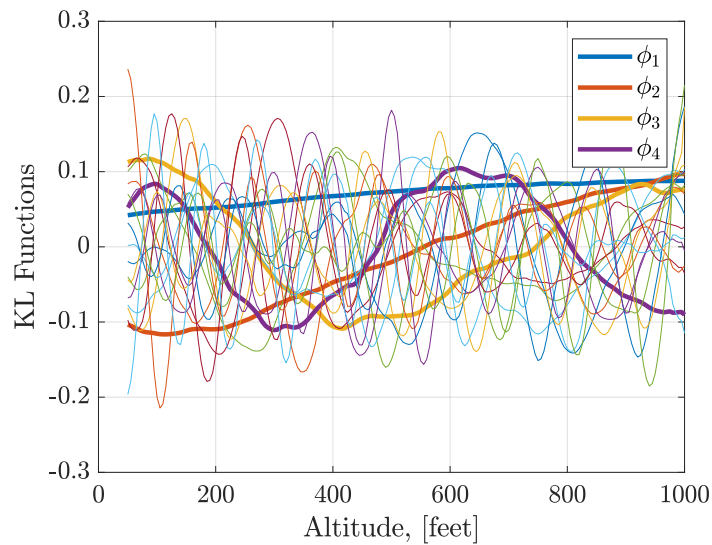


Figure 5.8: Eigenfunctions in KL expansion of the headwind, source: Figure 4c of [51].

By applying the Algorithm 12, the KL coefficients ξ_k are computed and used for reconstruction. Three arbitrary reconstructed wind series of 849 series are compared with the original headwind series in Figure 5.9. Even though only 20 KL functions are applied in the reconstruction, the reconstructed data and the raw data have a good match. Therefore, the constructed KL expansions with $K = 20$ allows us to reconstruct the headwind series with high accuracy.

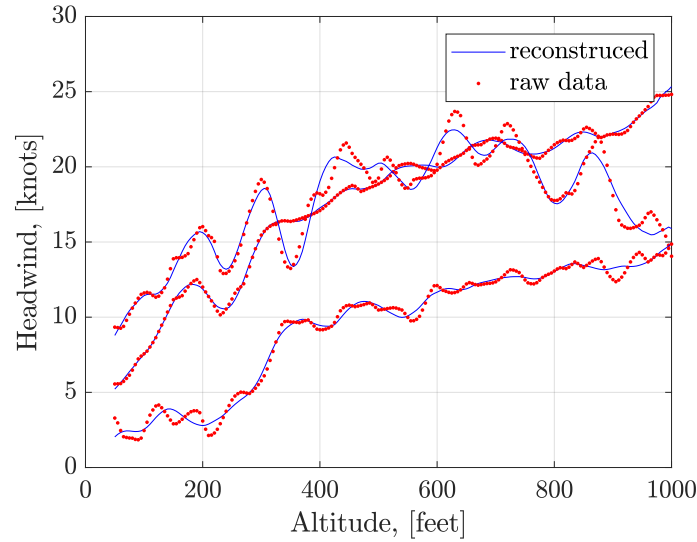


Figure 5.9: Reconstruction of three arbitrary headwind series, source: Figure 4d of [51].

Marginal Distributions of ξ_k

The computed ξ_k is only used to reconstruct the original process. In order to regenerate the new series, the new realizations of the ξ_k are required. The statistical characteristics of the computed ξ_k are analyzed. The box plot of the obtained ξ_k is shown in Figure 5.10. Figure 5.10 reveals that the obtained coefficients ξ_k are distributed close to zero mean and

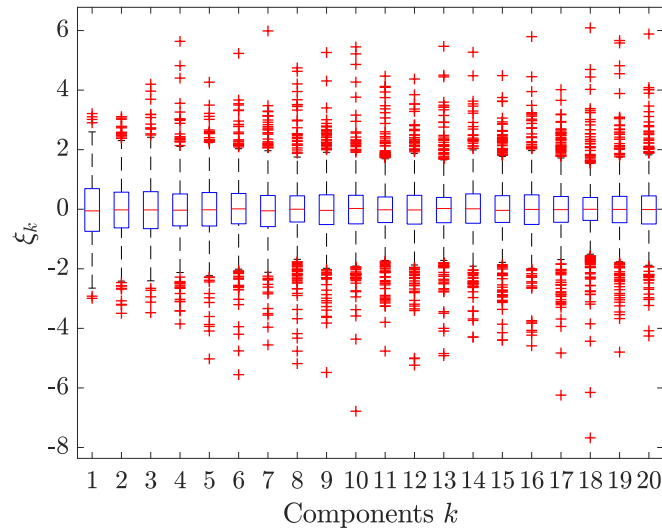


Figure 5.10: Box plot of ξ_k in KL expansion of the headwind, source: Figure 5a of [51].

unit variance. It satisfies the properties mentioned in Equation (5.23). Figure 5.10 also shows that all coefficients do not follow the same distribution. Compared to the standard Gaussian distribution, the empirical CDFs of ξ_k are presented in Figure 5.11. It clearly shows that deviations between the Gaussian and the empirical CDFs exist.

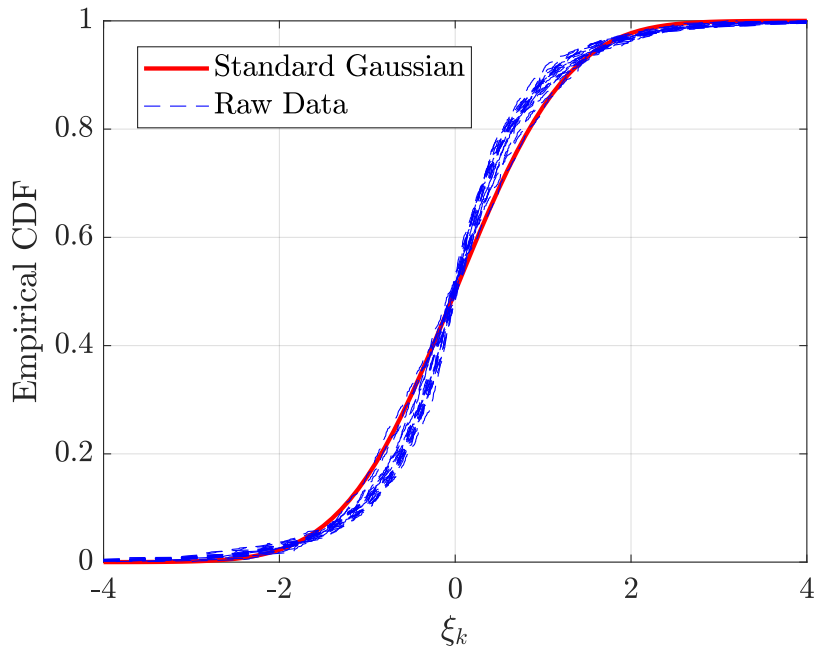


Figure 5.11: Empirical marginal CDFs of ξ_k in KL expansion of the headwind, source: Figure 5b of [51].

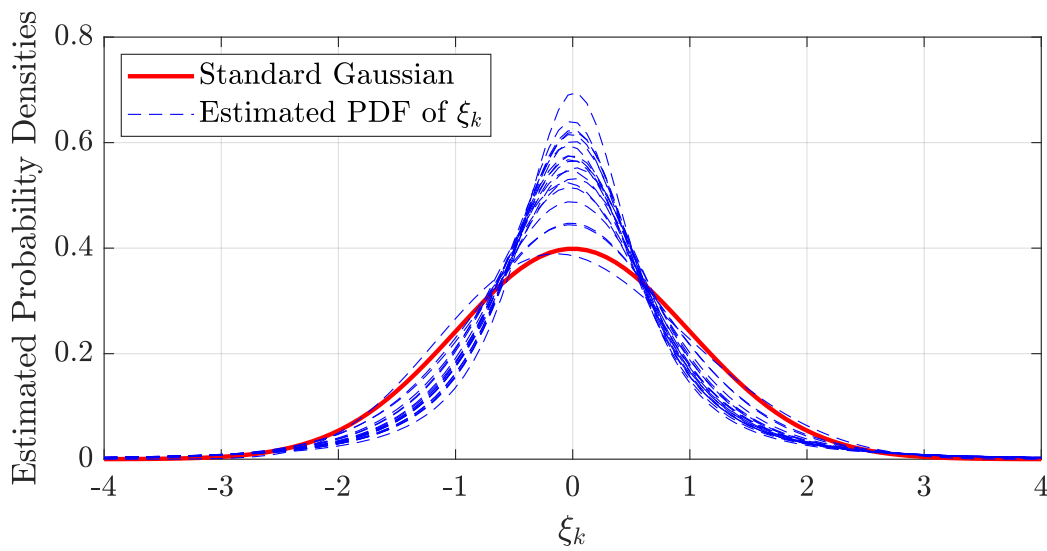


Figure 5.12: Estimated marginal PDFs of ξ_k in KL expansion of the headwind, source: Figure 5c of [51].

Furthermore, marginal PDFs of ξ_k are estimated using the maximum likelihood method from the parametric distribution families, such as: ‘Gaussian’, ‘Student-t’, ‘Generalized Extreme Value’, ‘t Location-Scale’, and so on. The estimated PDFs are shown in Figure 5.12, and the estimated distribution parameters are listed in Table 5.1.

Table 5.1: *Estimated marginal distributions of ξ_k in KL expansion of the headwind, source: Table A1 of [51].*

ξ_k	Distribution	Parameters
ξ_1	Generalized Extreme Value	$[k, \sigma, \mu] = [-0.2220, 0.9704, -0.3804]$
ξ_2	Logistic	$[\mu, \sigma] = [-0.0118, 0.5587]$
ξ_3	t Location-Scale	$[\mu, \sigma, \nu] = [-0.0222, 0.8706, 8.1359]$
ξ_4	t Location-Scale	$[\mu, \sigma, \nu] = [-0.0263, 0.7279, 3.9941]$
ξ_5	t Location-Scale	$[\mu, \sigma, \nu] = [-0.0158, 0.7763, 4.8160]$
ξ_6	t Location-Scale	$[\mu, \sigma, \nu] = [0.0141, 0.7007, 3.6527]$
ξ_7	t Location-Scale	$[\mu, \sigma, \nu] = [-0.0501, 0.7124, 3.7186]$
ξ_8	t Location-Scale	$[\mu, \sigma, \nu] = [0.0060, 0.6046, 2.7169]$
ξ_9	t Location-Scale	$[\mu, \sigma, \nu] = [-0.0234, 0.6777, 3.3137]$
ξ_{10}	t Location-Scale	$[\mu, \sigma, \nu] = [0.0051, 0.6529, 3.2108]$
ξ_{11}	t Location-Scale	$[\mu, \sigma, \nu] = [-0.0087, 0.5859, 2.4526]$
ξ_{12}	t Location-Scale	$[\mu, \sigma, \nu] = [-0.0262, 0.6376, 2.9137]$
ξ_{13}	t Location-Scale	$[\mu, \sigma, \nu] = [0.0075, 0.5823, 2.4866]$
ξ_{14}	t Location-Scale	$[\mu, \sigma, \nu] = [0.0231, 0.6695, 3.2488]$
ξ_{15}	t Location-Scale	$[\mu, \sigma, \nu] = [-0.0345, 0.6110, 2.6175]$
ξ_{16}	t Location-Scale	$[\mu, \sigma, \nu] = [-0.0016, 0.6387, 2.9972]$
ξ_{17}	t Location-Scale	$[\mu, \sigma, \nu] = [-0.0095, 0.5603, 2.3219]$
ξ_{18}	t Location-Scale	$[\mu, \sigma, \nu] = [0.0147, 0.5131, 2.1901]$
ξ_{19}	t Location-Scale	$[\mu, \sigma, \nu] = [-0.0003, 0.5795, 2.4952]$
ξ_{20}	t Location-Scale	$[\mu, \sigma, \nu] = [-0.0211, 0.6468, 2.9694]$

Figure 5.12 shows the deviations between the estimated PDFs and the standard Gaussian PDF are apparent. Most estimated distributions of ξ_k belong to the ‘t Location-Scale’ distribution family, which is usually used to model the tail part than the Gaussian distribution. Thereby, new realizations of ξ_k are simulated from the estimated marginal distributions. Subsequently, these samples are applied to generate new wind series.

Dependence among ξ_k

The estimated marginal distributions of ξ_k allow us to generate the new samples independently. Dependence among ξ_k is analyzed to improve the accuracy of the constructed stochastic wind model. A vine copula approach is applied to integrate the high-dimensional dependence structure into the KL expansion. As mentioned in Section 3.2.2, there are three scales in the study of the copula. Let ξ_k , ξ_k^U , and ξ_k^Z denote the coefficients in the X -scale, U -scale, and Z -scale, respectively. Based on the estimated marginal distributions, all realizations of ξ_k are first transformed into the uniform copula space: U -scale. The obtained copula data ξ_k^U allows us to separate the marginal behaviors from the underlying dependence structure. The empirical normalized contour plots of the ξ_k^U are shown in Figure 5.13, which is generated in R using the ‘VineCopula’ package [87]. It illustrates the pairwise dependence between two ξ_k .

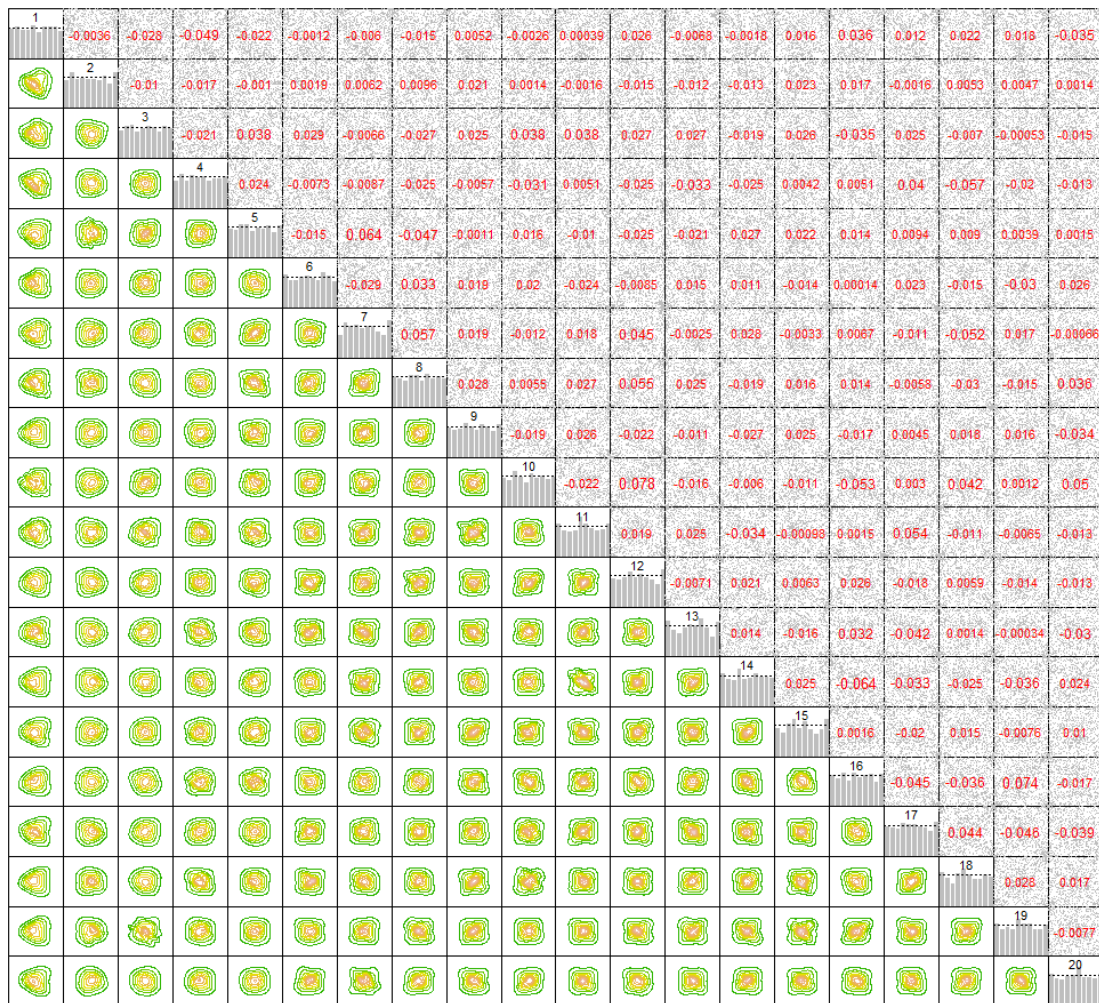
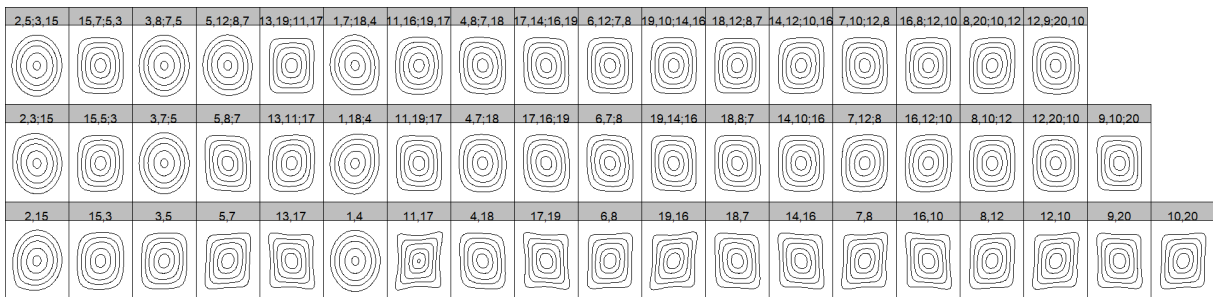


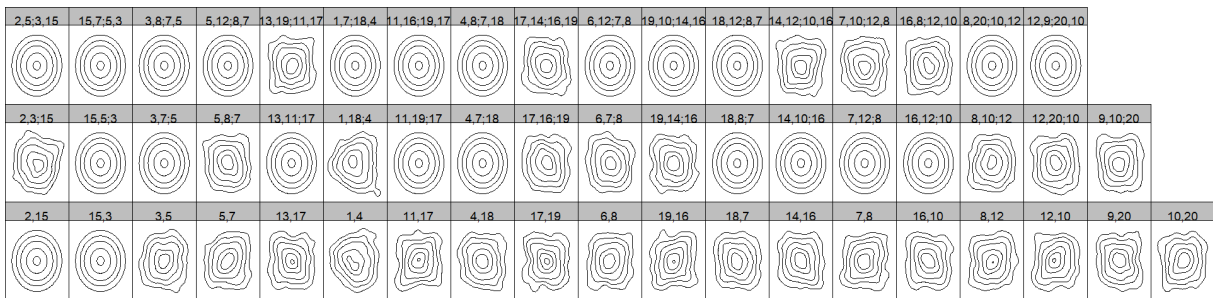
Figure 5.13: Empirical normalized pairwise contour plots in the headwind modeling, source: Figure A1 of [51].

In Figure 5.13, the diagonal plots shows the marginal histograms of ξ_k^U . The number above the histograms denotes the order k of the related ξ_k . The uniform shaped histograms indicate that the estimated marginal distributions are appropriate. The pairwise samples plots of ξ_k^U are shown in the upper triangle. The red number denotes the correlation measure: Kendall's τ between two ξ_k . The normalized copula data ξ_k^Z are obtained from ξ_k^U via the inverse standard Gaussian CDF transformation. The normalized empirical contour plots of ξ_k^Z are shown in the lower triangle. It indicates that most dependence between two ξ_k is weak. However, some pairs of dependence exist, like ξ_{10} and ξ_{16} .

The high-dimensional dependence among ξ_k is modeled using the vine copula. As mentioned, the KL expansion of the headwind is truncated at the order of 20. Therefore, 20 dependent random variables ξ_k are applied for reconstruction and regeneration. To capture the dependence among ξ_k in the vine approach, $\frac{20 \cdot (20-1)}{2} = 190$ conditional and unconditional bivariate copulas are required. The bivariate copula densities are identified from the parametric bivariate copula families [28], and the unknown parameters are estimated using the maximum likelihood method. According to the number of conditioning variables (from 0 to 18), there are 19 types of conditional bivariate copulas in total. The first three types of contour plots of the identified bivariate copulas are shown in Figure 5.14(a).



(a) Contour plots using the parametric bivariate copula



(b) Contour plots using the nonparametric bivariate copula.

Figure 5.14: Normalized contour plots of bivariate copulas in the vine copula, source: Figure A2 of [51].

In Figure 5.14, labels above the contour plot denotes the corresponding random variables ξ_k . For example, the label '2,15' in the left bottom plot indicates $c_{2,15}$, which means the unconditional bivariate copula density between ξ_2 and ξ_5 . The label '2,5;3,15' denotes $c_{2,5;3,15}$, which means the conditional bivariate copula density between ξ_2 and ξ_5 for the given ξ_3 and ξ_{15} . The normalized contour plot of the two independent Gaussian distributed variables is concentric circles. Thereby, the shape of the concentric circles implies that the two variables are independent. Otherwise, the two variables are dependent. Several dependence structures are obvious, especially for the unconditional bivariate copula in the last row of Figure 5.14(a). Furthermore, the nonparametric bivariate copula is also implemented in order to capture the dependence better. Instead of the predefined bivariate copula function, the empirical function for two variables is applied. Figure 5.14(b) shows the contour plots of nonparametric bivariate copulas. Both of the parametric and the nonparametric contour plots in Figure 5.14 are generated in R using the 'rvinecopulib' package [92].

New samples of ξ_k are generated from the constructed vine copula. An example of samples plot between ξ_{10} and ξ_{16} is presented to illustrate the difference between vine copula sampling and independent sampling. In Figure 5.15, samples of ξ_{10} and ξ_{16} in Z scale are used to remove the effect of the marginal distributions. The dependence between ξ_{10} and ξ_{16} is apparent in the raw data. Compared to the independent samples, samples generated via the vine copula sampling capture the same dependence behavior with the raw data. Therefore, the vine copula sampling allows us to generate more realistic samples than independent sampling.

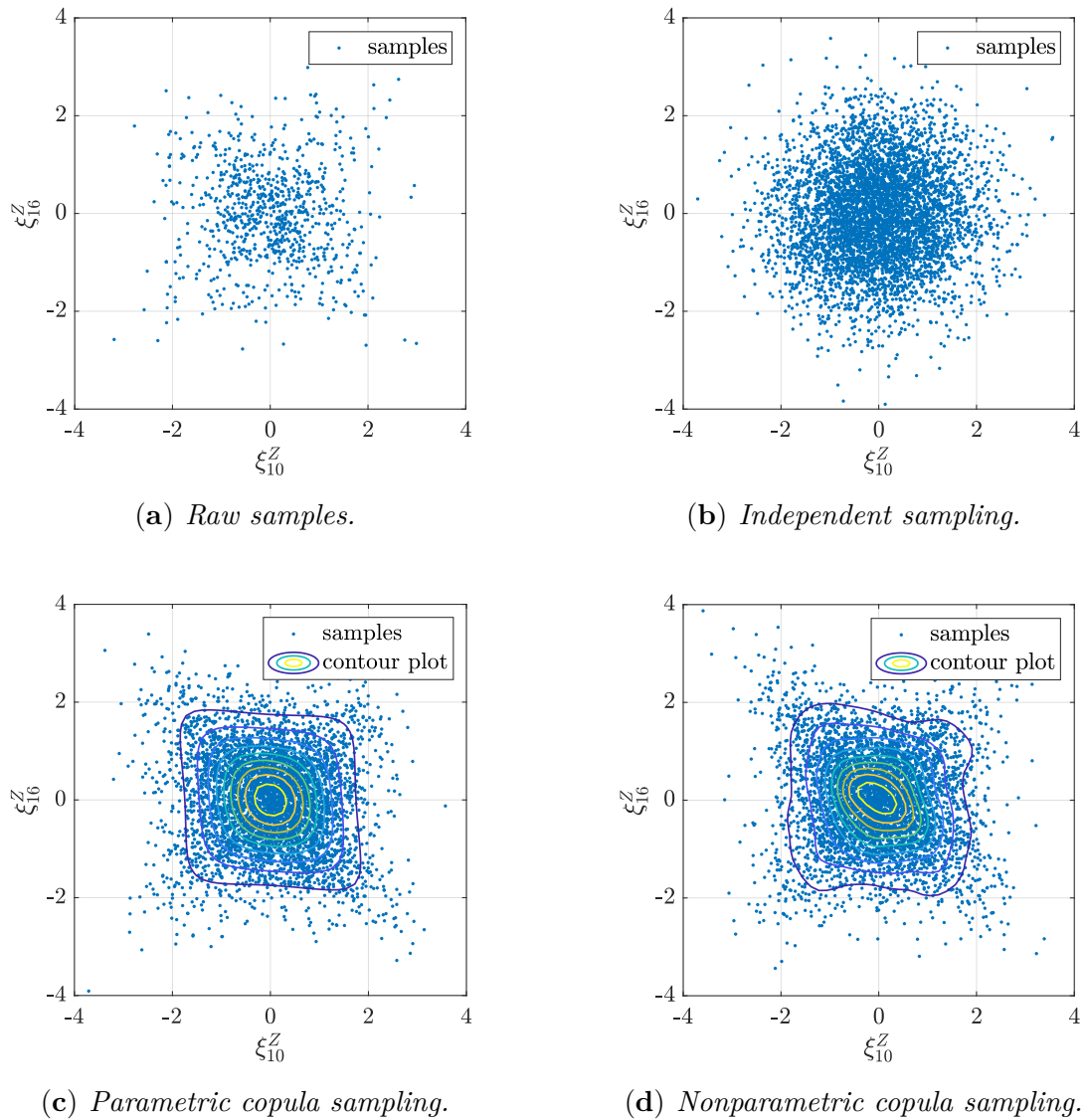


Figure 5.15: Comparison of samples plot between ξ_{10} and ξ_{16} in Z scale, source: Figure 6 of [51].

Independent sampling, parametric vine copula sampling, and nonparametric vine copula sampling are applied to generate 5000 samples of ξ_k , respectively. The corresponding 5000 wind series are constructed using the generated samples. The distribution of the headwind speed at a certain altitude is analyzed to validate the regeneration results. Figure 5.16 and Figure 5.17 shows the empirical CDFs and the kernel-fitted PDFs of the headwind speed at four altitudes, respectively. It shows the regenerated headwind has the same statistics with the raw data. Furthermore, the headwind distributions at 50 feet show that the vine copula sampling presents a better match with the raw data than the independent sampling.

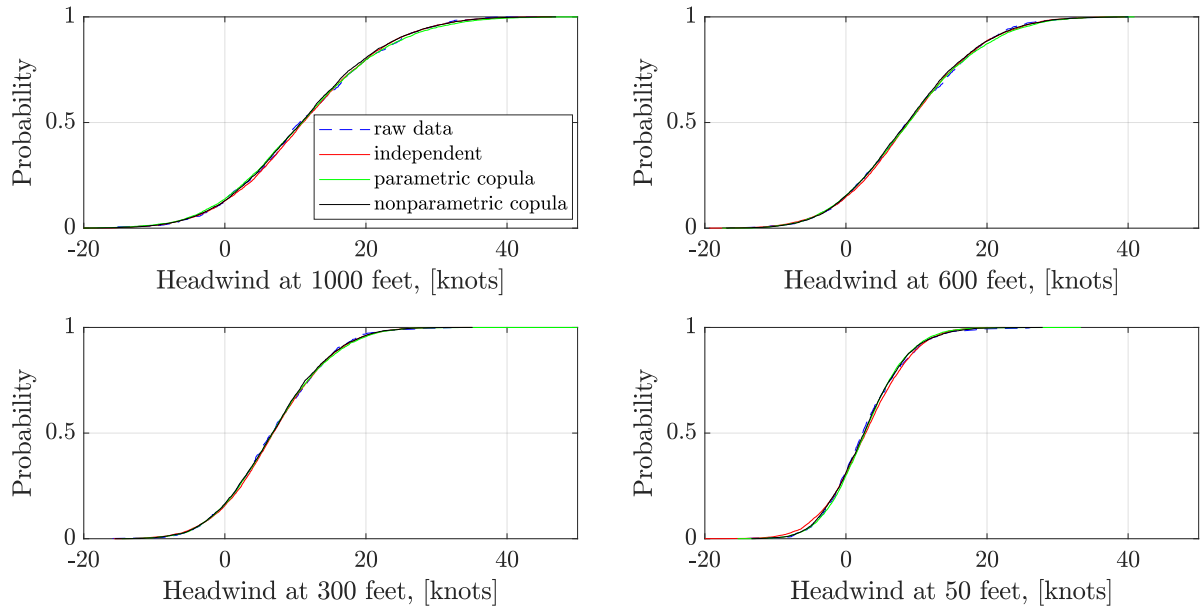


Figure 5.16: Empirical CDFs of headwind at a certain altitude, source: Figure 7 of [51].

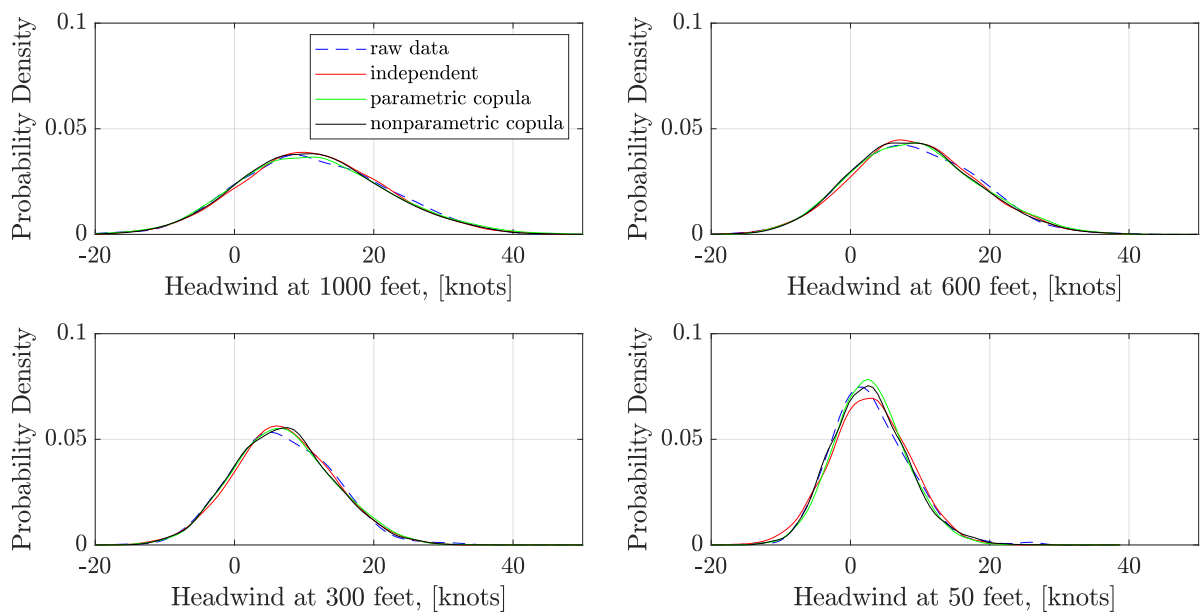


Figure 5.17: Estimated PDFs of headwind at a certain altitude, source: Figure 8 of [51].

The first four order statistical moments of the observed and regenerated headwind at different altitudes are calculated and illustrated in Figure 5.18. It indicates that the mean and standard deviation of the regenerated headwind match the raw data well. The regenerated headwind using independent sampling shows the skewness is close to 0, and the kurtosis is close to 3 for all altitudes. It is similar to the properties of Gaussian distribution. Results show that the regenerated headwind series can not match the skewness and kurtosis of the raw data. Nevertheless, there is no significant difference in skewness and kurtosis in terms of numerical value. Compared with the independent sampling, the statistical moments obtained from the parametric and nonparametric copula are closer to the raw data, particularly at the low altitude. The regenerated headwind series can be further applied in simulations for the quantitative assessment.

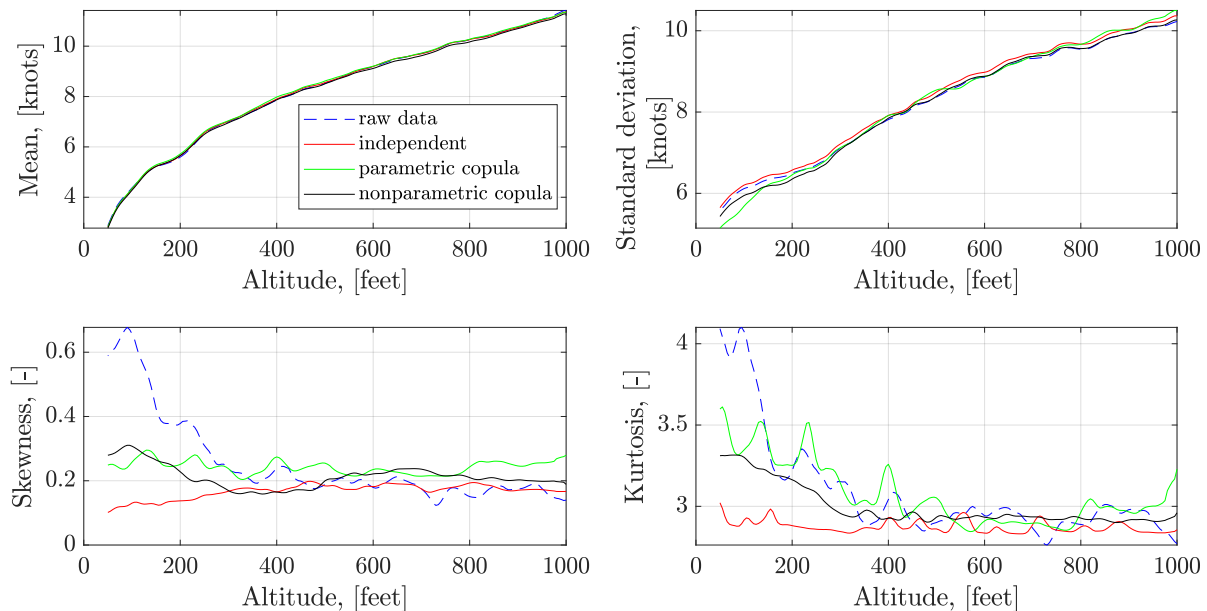


Figure 5.18: First four order statistical moments of headwind along with the altitude, source: Figure 9 of [51].

5.7.2 Windshear Regeneration

The low-level wind shear is a severe hazard for flight safety. To quantify the occurrence probability and the severity of the wind shear from operational flights, the significant increments of the headwind V_W with the variation of the altitude h , also called headwind ramps, are identified from the encountered headwind series. The extreme value of headwind ramps ΔV_W is calculated by a moving window with the length of a height deviation ΔH as below:

$$\Delta V_W = \max V_W(h : h + \Delta H) - V_W(h). \quad (5.51)$$

An example of the headwind during the final approach in one flight is demonstrated in Figure 5.19. The threshold of ΔV_W and the length of ΔH are predefined. In Figure 5.19, the headwind ramps threshold is set to 5 knots, and the length of the moving window is set to 100 feet. Two headwind ramps with $\Delta V_W \geq 5$ knots are extracted from this flight. $V_W(h : h + 2\Delta H)$ is considered as a headwind ramp. Therefore, the length of a headwind ramp is the double length of the moving window.

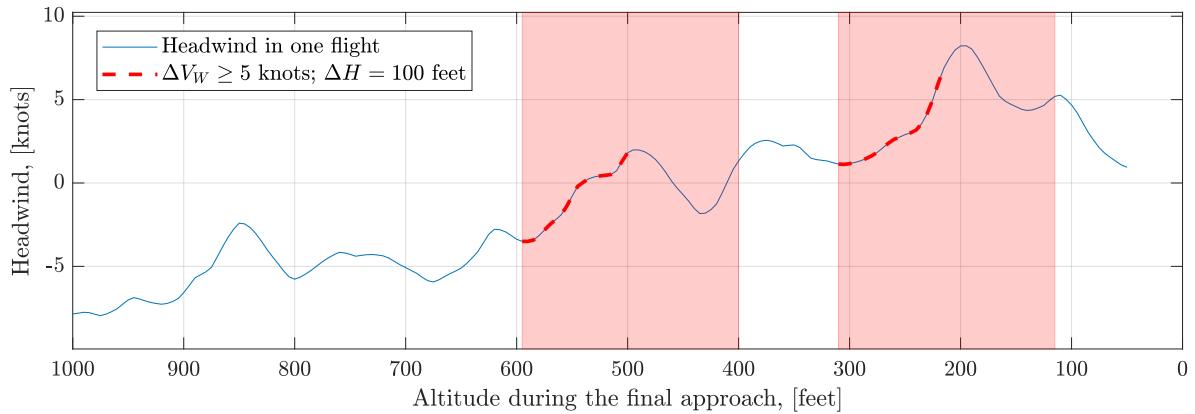


Figure 5.19: Detection of headwind ramps in one flight, source: Figure 10 of [51].

The detection algorithm is implemented in the headwind from 849 flights. The threshold of ΔV_W is set to 5 knots, and the length of ΔH is set to 50 feet, 100 feet, and 200 feet, respectively. Figure 5.20 shows the mean and standard deviation of the extracted headwind ramps series and the corresponding number of extracted series. It indicates the statistical characteristics of the headwind ramps depend on the length of the moving window.

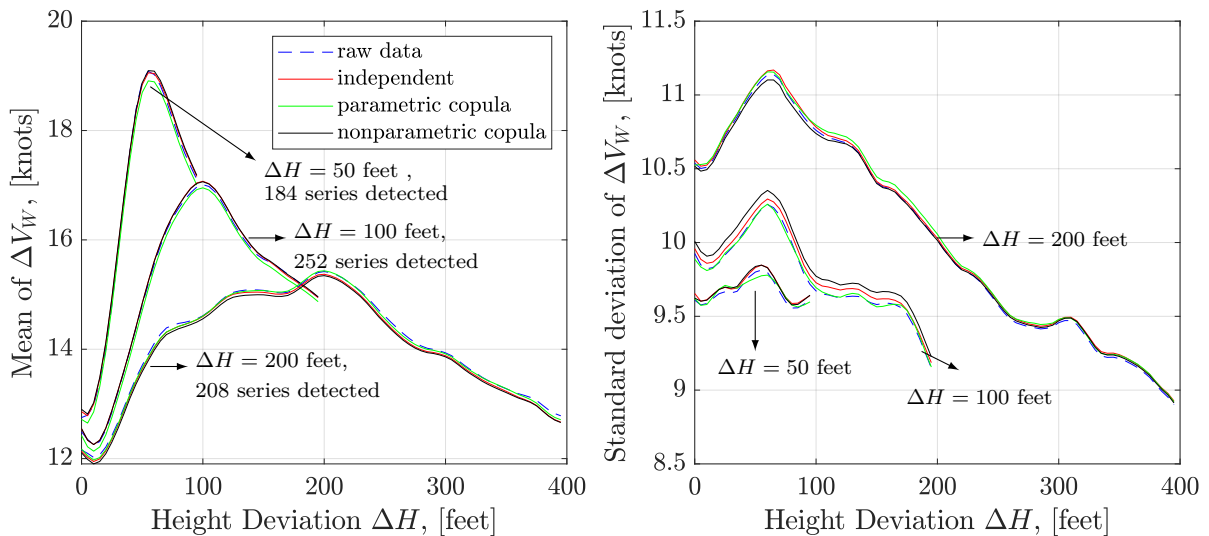
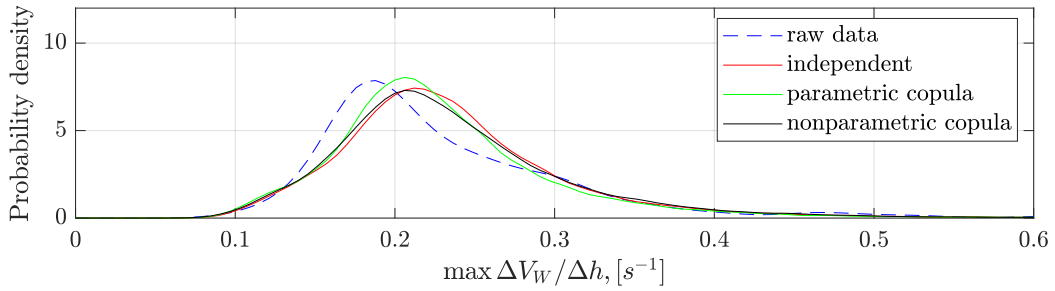


Figure 5.20: Mean and standard deviation of the extracted and regenerated headwind ramps, source: Figure 11 of [51].

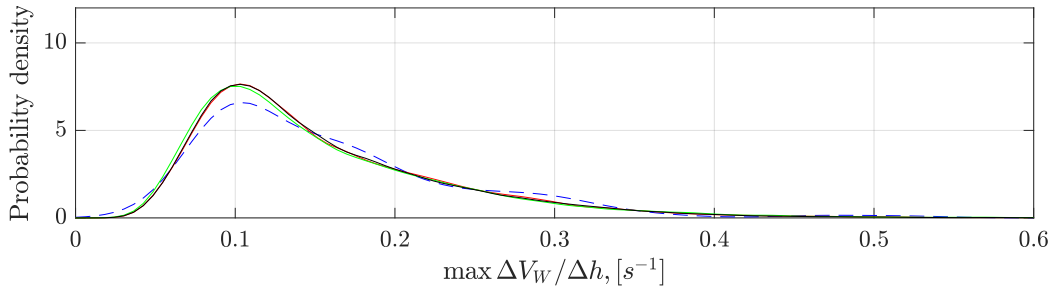
Based on the extracted headwind ramps, the KL-based stochastic process construction method allows us to regenerate new series and enrich the database of the headwind ramps. The truncated order K is set to 10 in the KL expansion of headwind ramps. The independent sampling, parametric and nonparametric copula sampling are applied separately to generate 5000 new series for the three lengths of ΔH . Figure 5.20 shows the mean and the standard deviation of the regenerated series and the extracted raw data match very well. Therefore, the KL-based construction model allows us to generate the new headwind ramps with similar statistical characteristics with the raw data. Results show that the standard deviation obtained using the parametric copula sampling has a better match with the raw data, especially in the case of the $\Delta H = 100$ feet. Furthermore, the wind shear intensity is an important measure to assess the severity of wind shear. A metric of $\max \Delta V_W / \Delta h$ is defined to quantify the wind shear intensity using the maximum change rate of headwind ramps with the altitude as follows:

$$\frac{\max \Delta V_W}{\Delta h} = \frac{\max V_W(h) - \min V_W(h)}{h_{V_W, \max} - h_{V_W, \min}}, \quad h \in [0, 2\Delta H]. \quad (5.52)$$

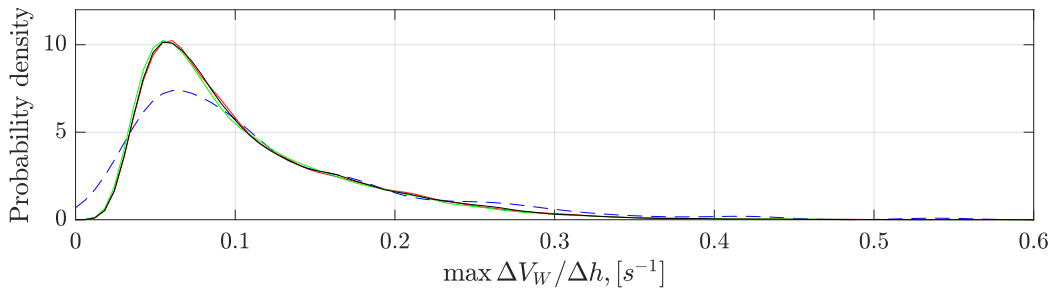
The raw and generated headwind ramps are used to calculate the metric $\max \Delta V_W / \Delta h$, respectively. The distributions of the metric are estimated using the kernel density function. The estimated PDFs for three different cases of ΔH are illustrated in Figure 5.21. For the case of $\Delta H = 100$ feet, estimated PDFs of the raw data and the generated series match very well. However, there are small deviations for the case of $\Delta H = 50$ feet and $\Delta H = 200$ feet. Since the number of the extracted headwind ramps is only about 200, increasing the raw data size might improve the KL-based construction accuracy. Furthermore, the difference in PDFs for the three cases of ΔH is noticeable, which can also be captured by the generated series. The KL-based regeneration of headwind ramps can be applied for the quantitative analysis of the wind shear intensity.



(a) The change rate of headwind for $\Delta V_W \geq 5$ knots, $\Delta H = 50$ feet.



(b) The change rate of headwind for $\Delta V_W \geq 5$ knots, $\Delta H = 100$ feet.



(c) The change rate of headwind for $\Delta V_W \geq 5$ knots, $\Delta H = 200$ feet.

Figure 5.21: The estimated PDFs of the change rate of headwind along with the altitude, source: Figure 12 of [51].

5.8 Summary

A KL-based stochastic wind model is constructed based on the operational flight data in this chapter. For the stationary process like turbulence, the KL expansion method is equivalent to the Fourier series expansion used in the PSD approach. For the non-stationary process, the truncated KL expansion allows us to generate new series with similar statistical characteristics as the raw data set. Furthermore, the vine copula dependence structure is integrated into KL expansion to improve the accuracy of the KL-based wind model. Additionally, the KL-based wind model allows the stochastic series as the model input in the Predictive Analysis framework for the quantitative analysis.

Chapter 6

Low Energy State Analysis

6.1 Motivation

By the analysis of Aircraft LOC-I accidents, Christina et al [139] shows that many LOC-I accidents or incidents involve degradation in situational awareness, especially pertaining to the aircraft energy state, followed by delayed or inappropriate pilot actions [140]. To enhance flight safety, researchers defined many energy-based metrics as safety performance indicators. Energy state prediction methods based on flight simulator data were proposed in [141, 142]. Energy-based metrics are identified for general aviation in [143]. The Joint Safety Analysis Team (JSAT) identified a series of occurrences of energy state awareness [144]. FAA Avionics Systems Harmonization Working Group (ASHWG) reviewed the current low speed awareness, low speed alerting, and low energy alerting systems in Boeing, Airbus, Embraer, and Bombardier aircraft to improve transport airplane standards for low airspeed protection. The European Operators Flight Data Monitoring (EOFDM) also takes the inadequate aircraft energy as a precursor of LOC-I accidents [145].

Although a temporary and small loss of energy will not immediately cause LOC-I accidents, this event could subsequently result in serious consequences if the flight crew is not aware of that, especially in low altitude situations such as during taking-off, approach and landing because there might not be sufficient time and altitude to recover the aircraft. The inadequate management of aircraft's total energy is a factor in unstable approach (UA) [146, 147]. Excess of energy can result in a runway excursion. Low energy might lead to LOC-I, land short, and hard landing. This chapter mainly analyzes the low energy situation during the final approach. The operational safety techniques during the approach and landing are introduced firstly. According to the stabilized approach criteria, the low energy bound is defined. Detection and predictive analysis for low energy using operational flight data is implemented in this chapter.

6.2 Operational Safety Techniques During the Approach

The UA is the largest contributing factor in approach-and-landing accidents. Only in 3% of unstable approaches, the pilot clears the situation by executing a go-around [148]. For the unstable approaches, too low aircraft energy states make the safe go-around execution less likely, and this situation can be exacerbated by a failure to understand how to manage aircraft pitch attitude [148]. Some operational flight techniques for a stable approach are shown in this section. As one type of UA, a low energy situation is more critical, which bears a high risk of a crash.

6.2.1 Stabilization Gate

The stabilization gate is an altitude at which the aircraft must be fully configured for landing and maintain the predefined trajectory with the appropriate speed, sink rate, thrust, attitude. During the approach to landing, rigorous respect of a stabilization gate provides a good basis for a safe landing [149]. A stabilized aircraft at the stabilization height reduces the pilot's workload and allows the pilot to prepare well for a safe landing flare. However, there is not a uniform definition of UA criteria in the airline industry. According to the report published by Flight Safety Foundation [149], the minimum stabilization height is 1000 feet above the runway elevation in instrument meteorological conditions (IMC), or 500 feet above runway elevation in visual meteorological conditions (VMC). Operators should define and provide their flight crew with a clear definition of their stabilization criteria and stabilization height based on their operational guidance, local regulations, and experience. The airline-specific stable approach criteria are obtained from the Standard Operating Procedures (SOPs) of the company, published in their Flight Crew Operating Manual (FCOM) or Operational Manual (OM). As one example, the instrument illustration of the stabilization criteria from the A320 FCOM is shown in Figure 6.1.

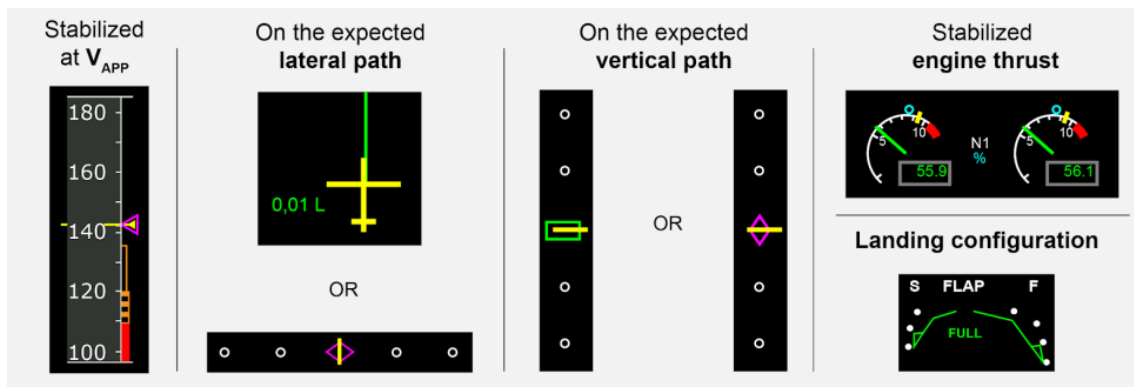


Figure 6.1: FCOM stable approach criteria for an A320 aircraft, source: [150]

Pilots should monitor these instruments carefully during the final approach. The approach is labeled as an UA if one of the criteria is not satisfied. The pilot should initiate a go-around unless only a small correction is required to recover the aircraft to the stabilized approach criteria. According to the B747-8F OM Part B of the airline, the stable approach criteria for IMC is summarized in Table 6.1.

Table 6.1: *Stable approach criteria for B747-8F, source: [148, 151]*

Parameters	Limits
Airspeed	$-5kts \leq \Delta V_{app} = V_{cas} - V_{app} \leq 15kts$
Vertical deviation	$-1 \text{ dot} \leq \delta_{GS,dot} \leq 1 \text{ dot}$
Lateral deviation	$-1 \text{ dot} \leq \delta_{LLZ,dot} \leq 1 \text{ dot}$
Sink rate	$\dot{h} \geq -1000 \text{ fpm}$
Configuration	Gear down; Landing Flap setting
Thrust	Appropriate power
Attitude	Small changes in heading/pitch/bank

The speed information that pilots get from cockpit instruments is the indicated airspeed. V_{cas} denotes the corrected indicator airspeed by considering the position error of the pitot and static port. V_{app} is called approach speed target. $\delta_{GS,dot}$ and $\delta_{LLZ,dot}$ are the Glideslope (GS) deviation and the Localizer (LLZ) deviation in dot, respectively. \dot{h} is the sink rate of the aircraft. The detailed description of V_{app} , $\delta_{GS,dot}$, and $\delta_{LLZ,dot}$ are shown in this section.

Approach speed target V_{app}

During the final approach phase, the pilot aims to descend and to maintain the approach speed target V_{app} [152]. Based on the operational manual of the Boeing aircraft [151], V_{app} is the sum of the reference speed and correction term as below:

$$V_{app} = V_{ref} + \Delta_{corr}, \quad (6.1)$$

where V_{ref} is the reference speed, which depends on the landing weight and the flap configuration. By the definition of V_{ref} in Lufthansa OM-Part B for B747 [153], V_{ref} equals to 1.23 times 1-g stall speed $V_{s,1g}$. $V_{s,1g}$ can be calculated via the equation that lift equals weight.

$$V_{s,1g} = \sqrt{\frac{2mg}{\rho_0 S_{ref} C_{L,max}}}, \quad (6.2)$$

where $C_{L,max}$ denotes the maximum lift coefficients in the corresponding configuration. The correction term in Equation (6.1) is calculated based on the airport wind information as follows:

$$\Delta_{corr} \text{ (knots)} = \max(5 \text{ knots}, \min(20 \text{ knots}, \frac{1}{2} \bar{V}_{hws} + V_{gust})), \quad (6.3)$$

where \bar{V}_{hws} denotes the airport steady headwind component, and V_{gust} is the gust intensity. Flight crews obtain the wind information from the ATC, i.e., Meteorological Airport observation Report (METAR), Automatic Terminal Information Service (ATIS), or tower winds. According to the current operation, the METAR wind is a 10-minute-average wind. The ATIS or tower average wind is a 2-minute average wind. The ATIS or tower gust is the wind peak value during the last 10-minute period. The ATIS message is updated only if the wind direction changes by more than 30-degree or if the wind velocity changes by more than 5-knot over a 5-minute time period [154]. Since the wind varies along with time, there is a deviation between the headwind from the METAR information and the encountered headwind at touchdown estimated from the operational flight data. This deviation, denoted by $\Delta V_{hws,TD}$, is shown in Figure 6.2. The inaccurate headwind information will influence the setting of V_{app} .

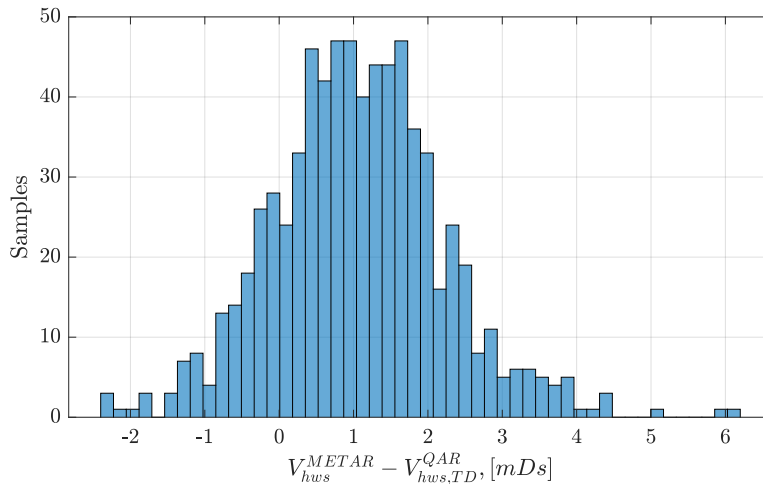


Figure 6.2: The deviation of headwind $\Delta V_{hws,TD}$ between the METAR information and the encountered wind at touchdown calculated using the operational flight data

In order to prevent an aircraft stall, a stick shaker device is triggered for too low airspeed in Boeing aircraft. There is a speed margin between the stick shaker speed V_{ss} and the 1g stall speed $V_{s,1g}$. V_{ss} also depends on the aircraft mass and configurations, and it is obtained from the OM-B [151].

ILS deviation $\delta_{GS,DDM}$ and $\delta_{LLZ,DDM}$

The GS deviation $\delta_{GS,DDM}$ and the LLZ deviation $\delta_{LLZ,DDM}$ are measured as Difference in Depth of Modulation (DDM). Their conversion to dots must not exceed the ± 1 dot shown in the flight instruments during the final approach phase [151]. The geometry of the GS path and the LLZ course is shown in Figure 6.3.

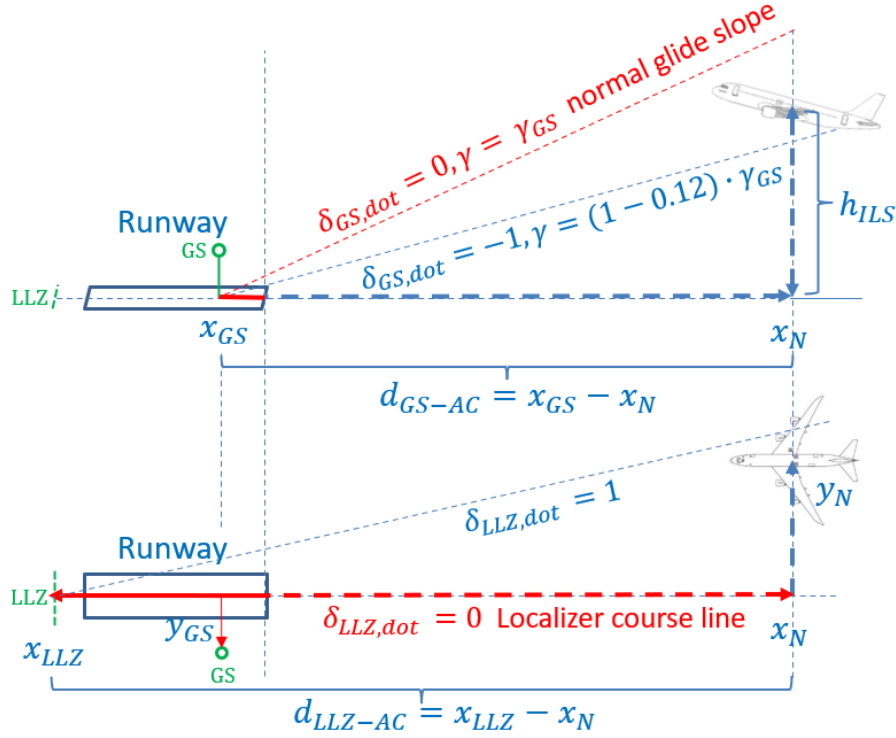


Figure 6.3: The geometry of the glide slope and localizer course, adapted from [71]

According to the ILS specifications in ICAO Annex 10 [155], the ILS deviation measurements can be computed based on the relative position of aircraft and LLZ/GS antennas. The aircraft height above the runway elevation h_{ILS} can be computed based on the measurement $\delta_{GS,DDM}$ via [71]:

$$h_{ILS} = \sqrt{(x_{GS} - x_N)^2 + y_{GS}^2} \cdot \tan \left(\gamma_{GS} + 0.12 \cdot \gamma_{GS} \cdot \frac{\delta_{GS,DDM}}{0.0875 DDM} \right), \quad (6.4)$$

where γ_{GS} is the glide slope angle, and $\gamma_{GS} = -3^\circ$ in most runways. x_{GS} and y_{GS} denote the position of the GS antenna with respect to the runway threshold. x_N is the distance along with the runway from the aircraft current position to the runway threshold. The GS deviation in dot $\delta_{GS,dot}$ is defined as follows:

$$\delta_{GS,dot} = \frac{\delta_{GS,DDM}}{0.0875 DDM}. \quad (6.5)$$

Likewise, the aircraft lateral deviation to the runway center line y_N can be calculated based on the $\delta_{LLZ,DDM}$ via

$$y_N = \frac{x_{LLZ} - x_N}{x_{LLZ}} \cdot \frac{\delta_{LLZ,DDM}}{0.00145 DDM}, \quad (6.6)$$

where x_{LLZ} denotes the position of the LLZ antenna. The LLZ deviation in dot $\delta_{LLZ,dot}$ is calculated as below:

$$\delta_{LLZ,dot} = \frac{\delta_{LLZ,DDM}}{0.00145 DDM}. \quad (6.7)$$

6.2.2 Energy Management

A potential UA might be detected before the stabilized gate, such as the use of the energy circle in aircraft: A320/A330/A340/A350/A380 [150]. By monitoring the high energy situation of the aircraft, pilots can take the time to bleed the aircraft energy properly and recover the situation in cooperation with ATC. Early energy management allows pilots to avoid the situation for last-minute corrections and to avoid some potential unstable approaches, especially the situation with high energy. The total energy of aircraft contains the potential energy and the kinetic energy, and it is computed as below:

$$E = mg_0h + \frac{1}{2}mV_A^2, \quad (6.8)$$

where m is the aircraft mass, g_0 is the gravitational acceleration, h is the aircraft altitude above ground, and V_A denotes the true airspeed. To reduce the influence of the aircraft mass and to make it easily comparable for different flights, the normalized energy, also called the specific energy, is usually used and defined as below:

$$E_s = h + \frac{V_A^2}{2g_0}. \quad (6.9)$$

According to the UA criteria in Table 6.1, the energy boundary during the final approach can be defined based on the airspeed and the GS deviation limits as below:

$$\begin{aligned} \text{Upper bound} : \delta_{GS,dot} &= 1 & \text{and } V_{cas} &= V_{app} + 15kts, \\ \text{Lower bound} : \delta_{GS,dot} &= -1 & \text{and } V_{cas} &= V_{app} - 5kts. \end{aligned} \quad (6.10)$$

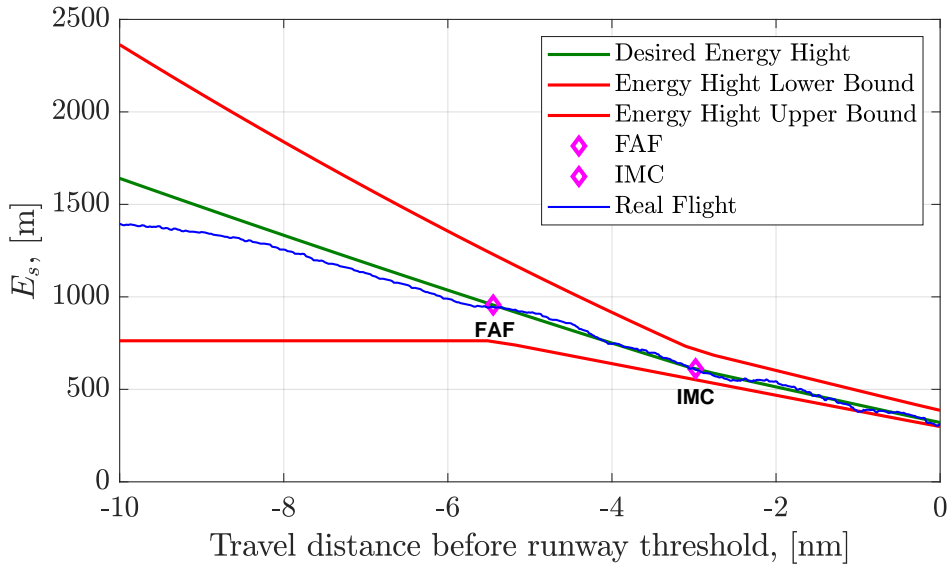
Based on the typical schedule for deceleration on the 3-degree glide slope [146, 156], the aircraft normally decelerate at the speed of 10 knots per nautical mile (nm), which is used as the desired deceleration rate before the stabilized gate in this thesis. Furthermore, the maximum deceleration rate is limited by 20 kt/nm for the cases that all deceleration devices are applied, like spoilers. The lower bound of the speed maintains the same as that in the final approach phase. The speed boundary during the approach is shown in Table 6.2. D denotes the traveled distance of the aircraft to the runway threshold. D_{IMC} means the traveled distance from the IMC gate to the runway threshold. Only $D \leq 10 \text{ nm}$ is considered.

Table 6.2: *Speed boundary during the approach*

	From $D = 10 \text{ nm}$ to IMC gate	From IMC gate to runway threshold
Upper bound	$V_{app} + 15kts + (D - D_{IMC}) \cdot 20kts/nm$	$V_{app} + 15kts$
Desired speed	$V_{app} + (D - D_{IMC}) \cdot 10kts/nm$	V_{app}
Lower bound	$V_{app} - 5kts$	$V_{app} - 5kts$

Table 6.3: *Altitude boundary during the approach*

	From $D = 10 \text{ nm}$ to FAF	From FAF to runway threshold
Upper bound	$\delta_{GS,\dot{dot}} = 1$	$\delta_{GS,\dot{dot}} = 1$
Desired path	$\delta_{GS,\dot{dot}} = 0$	$\delta_{GS,\dot{dot}} = 0$
Lower bound	$h_{ILS} \text{ at FAF}$	$\delta_{GS,\dot{dot}} = -1$

**Figure 6.4:** *Aircraft energy boundary during the approach.*

The boundary of altitude before the stabilized gate is also defined based on the glideslope deviation shown in Table 6.3. The only difference is that the lower bound of the altitude is modified before the Final Approach Fix (FAF) point. According to the speed boundary and altitude boundary, the energy height upper and lower bounds ($E_{s,UB}$, $E_{s,LB}$), and the desired energy height $E_{s,des}$ are calculated and illustrated in Figure 6.4, denoted by the red line and green line, separately. The blue line denotes the energy height of one flight calculated using operational flight data.

During the approach, pilots try to decrease the aircraft's altitude and speed. The energy of aircraft decreases continually. In order to avoid a high energy situation before the stabilization gate, the safety technique of the energy circle is used. The calculation principle of the energy circle depends on the aircraft's descent flight profile shown in Figure 6.5. The energy circle indicates the required distance to descend the aircraft from the current position at the current speed down to the arrival runway at the approach target speed. Flight configurations, speed limits, and wind speed are taken into account during the calculation, and three different flight segments are concerned. If the airport is inside the energy circle, the pilots should increase the energy bleed rate by extending the speed

brakes and/or request more tracking distance from the ATC. This energy circle allows pilots to recover the energy in an early phase, which avoids the high energy situation when crossing the stabilization gate efficiently.

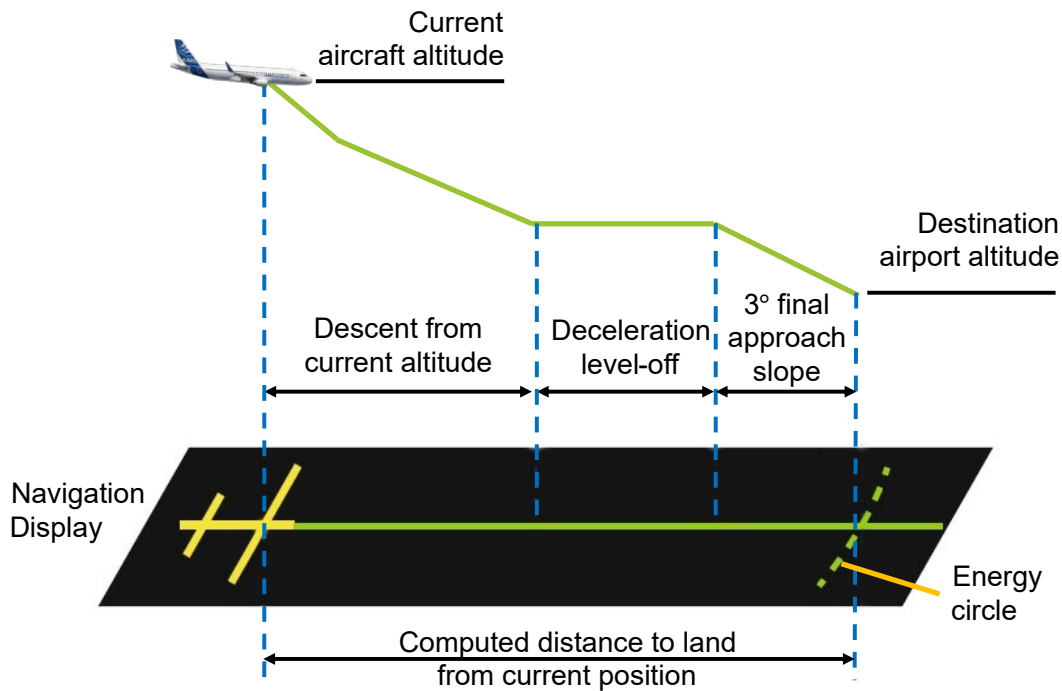


Figure 6.5: Energy circle calculation, source: [150].

6.2.3 Late Destabilization

Only stabilized at the IMC gate is not sufficient to ensure a safe landing. Figure 6.6 shows some flight parameters of a B747-8F flight from 2000 feet above runway elevation to the runway threshold. The vertical dash line in each subfigure of Figure 6.6 denotes the time point that the aircraft flies over 1000 feet above the ground. Due to the sudden decrease of the V_{hws} at about 130s, V_{cas} also decrease quickly. The time history plot of ΔV_{app} illustrates that the airspeed is lower than the lower bound during the final approach. The pilot advanced the thrust level to prevent the loss of speed for recovery.

During the final approach phase, some external conditions like turbulence or wind shear might lead to late destabilization, which is difficult to predict in advance. Therefore, pilots should monitor and keep the critical flight parameters within the limits during the final approach to avoid destabilization. Especially, pilots should avoid low speed or low energy situations, since the low energy at the low altitude has a high risk of incidents or accidents. The following sections will focus on analyzing the low energy situation during the final approach.

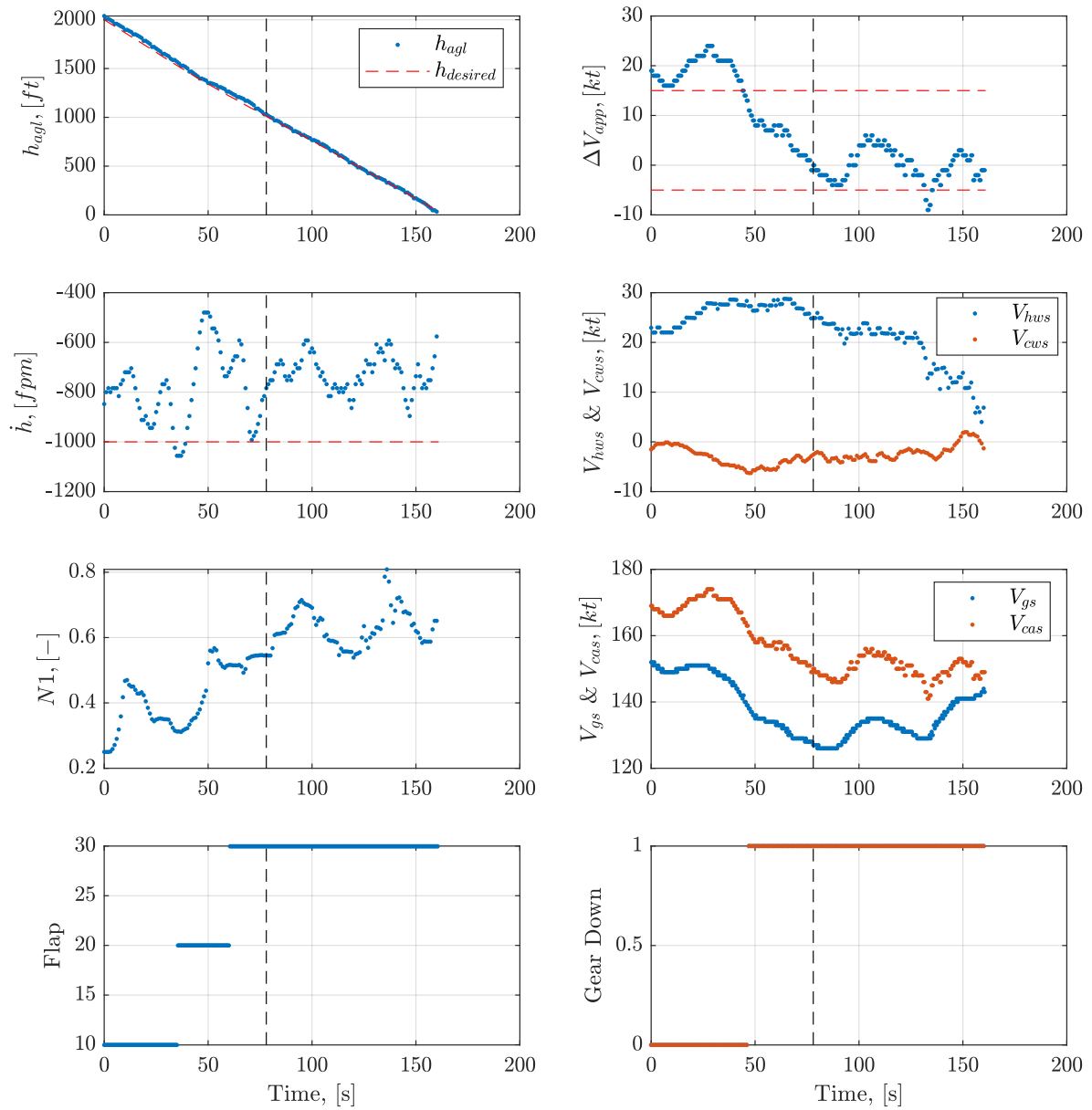


Figure 6.6: Flight parameters time history from 2000 feet above ground to the runway threshold.

6.3 Detection of Low Energy Event During the Final Approach

6.3.1 Low Energy Margin

According to the UA criteria, the deviation between the actual energy height and the UA-based lower energy height bound is computed as below:

$$\Delta E_s = E_s - E_{s,LB} = \left(h + \frac{V_A^2}{2g_0} \right) - \left(h_{LB} + \frac{(V_{app} - 5 \text{ kts})^2}{2g_0} \cdot \frac{\rho_0}{\rho} \right). \quad (6.11)$$

Since the V_{app} refers to the value in terms of the indicated airspeed, the corrected ratio ρ_0/ρ is added to convert the indicated airspeed to the true airspeed in Equation (6.11). Due to the gusty wind, a short exceedance of the UA-based lower energy bound might occur and could be compromised by the pilots and the aircraft. The low energy metric based on the UA criteria will lead to a lot of false detection. Therefore, a more critical bound is used in the section. Instead of the $(V_{app} - 5 \text{ kts})$, the stick shaker speed V_{ss} is used as the lower bound of the V_{cas} . Therefore, the adjusted energy height deviation is shown as follows:

$$\Delta E_s^* = E_s - E_{s,LB}^* = \left(h + \frac{V_A^2}{2g_0} \right) - \left(h_{LB} + \frac{V_{ss}^2}{2g_0} \cdot \frac{\rho_0}{\rho} \right). \quad (6.12)$$

The energy margin for the low energy event is defined as below:

$$EM = \min(\Delta E_s^*). \quad \text{during the final approach.} \quad (6.13)$$

If $EM < 0$, a low energy event occurs. The algorithm to calculate the EM is applied in 849 operational flights, The histogram of the EM is shown in Figure 6.7.

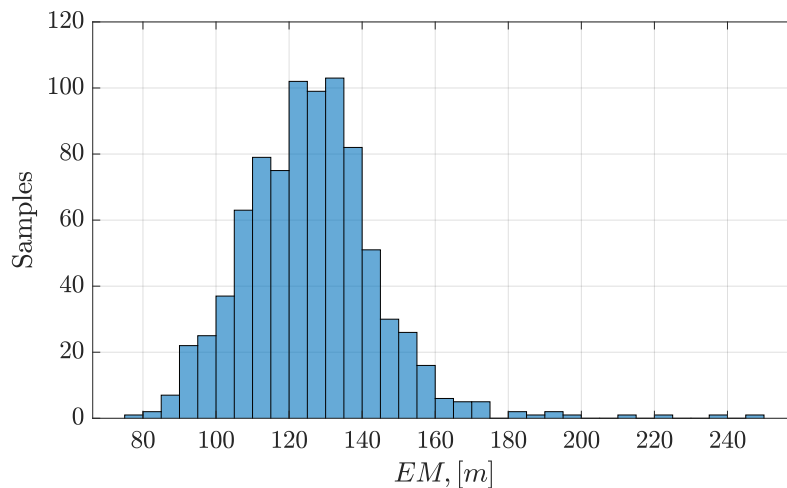


Figure 6.7: Histogram of the energy margin based on 849 flights.

For normal operations, there are no low energy events. To reproduce and predict the low energy situation, the physical model of energy state during the final approach is built based on the aircraft's motion in Section 6.4.2.

6.3.2 Time Margin

Besides the current energy state, the energy bleed rate also has a big influence on the future energy state. The energy bleed rate can be derived by differentiating Equation (6.9) with respect to the time:

$$\dot{E}_s = \dot{h} + \frac{V_A \dot{V}_A}{g}. \quad (6.14)$$

Since the lower energy bound also decrease along with the -1 dot glide path, the rate of the lower energy bound can be computed using the aircraft ground speed V_{gs} :

$$\dot{E}_{s,LB} = \dot{h}_{LB} = V_{gs} \tan(\gamma_{-1dot}). \quad (6.15)$$

Therefore, the rate of the energy height deviation is computed as below:

$$\Delta \dot{E}_s = \dot{h} + \frac{V_A \dot{V}_A}{g} - V_{gs} \cdot \tan(\gamma_{-1dot}). \quad (6.16)$$

In order to compute the $\Delta \dot{E}_s$, the unrecorded \dot{V}_A should be computed from the flight data. As shown in Figure 6.8, V_A has a relationship with the kinematic speed V_K and the wind speed V_W .

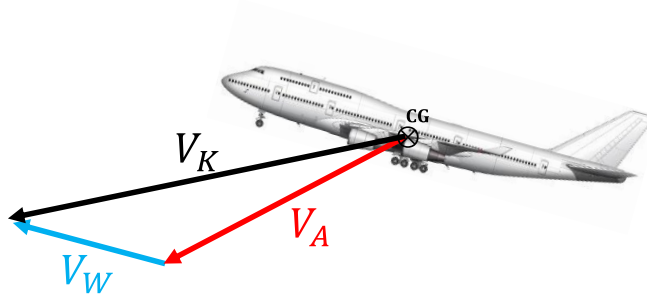


Figure 6.8: Kinematic speed V_K , aerodynamic speed V_A , and wind speed V_W

The relation of three kinds of speeds in aerodynamic frame A is written as below. The nomenclature used in this section is referred to Appendix A.

$$(\vec{V}_A^G)_A^E = (\vec{V}_K^G)_A^E - (\vec{V}_W^G)_A^E. \quad (6.17)$$

In order to compute the derivative of V_A , Equation (6.17) is differentiated with respect to the A frame:

$$(\dot{\vec{V}}_A^G)_A^{EA} = (\dot{\vec{V}}_K^G)_A^{EA} - (\dot{\vec{V}}_W^G)_A^{EA}. \quad (6.18)$$

Based on the translation equations, the derivative of V_K with respect to the A frame can be further derived as below:

$$\begin{aligned} (\dot{\vec{V}}_K^G)^{EA} &= (\dot{\vec{V}}_K^G)^{EO} - (\vec{\omega}^{OA})_A \times (\vec{V}_K^G)^E \\ &= M_{AB}(\dot{\vec{V}}_K^G)^{EO} - (\vec{\omega}^{OA})_A \times (\vec{V}_K^G)^E. \end{aligned} \quad (6.19)$$

The derivative of V_W with respect to the A frame can be further computed as below:

$$\begin{aligned} (\dot{\vec{V}}_W^G)^{EA} &= (\dot{\vec{V}}_W^G)^{EO} - (\vec{\omega}^{OA})_A \times (\vec{V}_W^G)^E \\ &= M_{AO}(\dot{\vec{V}}_W^G)^{EO} - (\vec{\omega}^{OA})_A \times (\vec{V}_W^G)^E. \end{aligned} \quad (6.20)$$

Therefore, the derivative of V_K can be derived as follows:

$$\begin{aligned} (\dot{\vec{V}}_A^G)^{EA} &= (\dot{\vec{V}}_K^G)^{EA} - (\dot{\vec{V}}_W^G)^{EA} \\ &= M_{AB}(\dot{\vec{V}}_K^G)^{EO} - (\vec{\omega}^{OA})_A \times (\vec{V}_K^G)^E - \left(M_{AO}(\dot{\vec{V}}_W^G)^{EO} - (\vec{\omega}^{OA})_A \times (\vec{V}_W^G)^E \right) \\ &= M_{AB}(\dot{\vec{V}}_K^G)^{EO} - M_{AO}(\dot{\vec{V}}_W^G)^{EO} - (\vec{\omega}^{OA})_A \times \left((\vec{V}_K^G)^E - (\vec{V}_W^G)^E \right) \\ &= M_{AB}(\dot{\vec{V}}_K^G)^{EO} - M_{AO}(\dot{\vec{V}}_W^G)^{EO} - (\vec{\omega}^{OA})_A \times (\vec{V}_A^G)^E, \end{aligned} \quad (6.21)$$

where $(\dot{\vec{V}}_K^G)^{EO}$ can be computed using the acceleration rates measured in the IMU as below:

$$(\dot{\vec{V}}_K^G)^{EO} = (\vec{a}_K^G)^{II}_{B,IMU} - M_{BO} \begin{bmatrix} 0 \\ 0 \\ g_0 \end{bmatrix}_O. \quad (6.22)$$

As mentioned in Section 2.3.5, the wind speed is estimated using the RTS smoother in the flight path reconstruction. The third order Gauss-Markov process mentioned in Equation (5.3) is used as the wind model. Thence, $(\dot{\vec{V}}_W^G)^{EO}$ is estimated during the flight path reconstruction. Furthermore, $(\vec{V}_A^G)^E$ and $(\dot{\vec{V}}_A^G)^{EA}$ are written into three components in A frame:

$$(\vec{V}_A^G)^E = \begin{bmatrix} V_A^G \\ 0 \\ 0 \end{bmatrix}_A, \quad (\dot{\vec{V}}_A^G)^{EA} = \begin{bmatrix} \dot{V}_A^G \\ 0 \\ 0 \end{bmatrix}_A. \quad (6.23)$$

Therefore, the \dot{V}_A can be computed using Equation (6.21-6.23). $\Delta \dot{E}_s$ can be computed subsequently. The predicted energy height deviation after t seconds can be calculated as below:

$$\Delta E_{s,pred}^* = \Delta E_{s,0}^* + \int_0^t \Delta \dot{E}_s dt, \quad (6.24)$$

where $\Delta E_{s,0}^*$ denotes the current energy margin. If $\Delta E_{s,pred}^* \leq 0$, the low energy event occurs. Assuming $\Delta \dot{E}_s$ keeps constant, time to the lower energy bound can be computed as below:

$$\Delta t = \begin{cases} \frac{-\Delta E_{s,0}^*}{\Delta \dot{E}_s}, & \text{if } \frac{-\Delta E_{s,0}^*}{\Delta \dot{E}_s} > 0; \\ NaN, & \text{if } \frac{-\Delta E_{s,0}^*}{\Delta \dot{E}_s} \leq 0; \end{cases}, \quad (6.25)$$

where Δt is the time to the lower energy bound, and it describes how long the low energy will occur if the aircraft maintains the current energy bleed rate. Time margin TM is the minimum Δt for one flight. It is defined as below:

$$TM = \min(\Delta t). \quad (6.26)$$

The time margin and energy margin for a flight are computed and demonstrated in Figure 6.9. The time point related to the time margin is a little ahead of time of the energy margin.

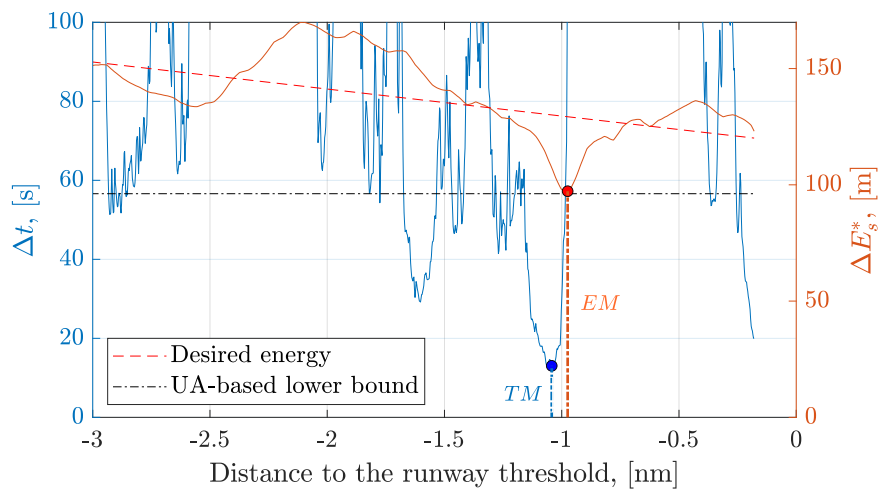


Figure 6.9: Time margin and energy margin in one flight during the final approach.

The algorithm to calculate the time margin is implemented in 849 operational flights. The histogram of the time margin is shown in Figure 6.10.

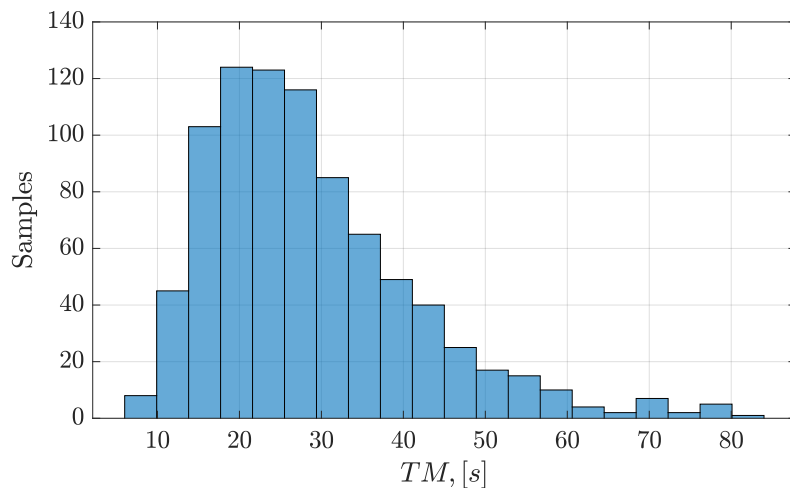


Figure 6.10: Histogram of the time margin based on 849 flights.

6.4 Predictive Analysis of Low Energy Event

The probability of a low energy event is not zero, even there are no low energy events detected from the flight operational data. To cope with the small probability incident in operations, the Predictive Analysis framework is applied to predict the occurrence probability of low energy incidents and to identify key contributing factors.

6.4.1 Incident Metric for Low Energy Event

As shown in Table 6.3, the energy height deviation is reduced along the glide path, since the height margin is reduced. The energy factor E_{factor} , proposed in [156], is used to normalize the energy height deviation. E_{factor} is defined as follows:

$$E_{factor} = \begin{cases} (E_s - E_{s,des})/(E_{s,UB} - E_{s,des}), & \text{when } E_s \geq E_{s,des} \\ (E_s - E_{s,des})/(E_{s,des} - E_{s,LB}), & \text{when } E_s < E_{s,des} \end{cases}, \quad (6.27)$$

where E_{factor} is inside the range of $[-1, 1]$ for normal operations. $E_{factor} = 0$ indicates that the aircraft has the ideal energy. $E_{s,LB}$ in Equation (6.27) is defined based on the UA criteria in [156]. To avoid the false detection, $E_{s,LB}^*$ is used for the adjusted energy deviation in Section 6.3.1. According to the SOPs, the pilot should execute a go-around immediately if the aircraft airspeed is lower than the airspeed lower bound ($V_{app} - 5 \text{ kts}$). Even pilots might tolerate a small and short violation of the low airspeed bound in practice, the V_{ss} is quite difficult to approach during normal operations.

Considering the implementation of UA criteria in practice [157], a persistent violation of UA criteria for several seconds is preferred to be used to distinguish the stable or unstable approach instead of an instantaneous violation. By taking the persistent violation into account, a new incident metric IM for low energy events is proposed and defined as below:

$$IM = \min_{t^*} \left(\int_{t^*}^{t^* + \Delta\tau} (E_{factor}(t) + 1) dt \right), \quad t^* \in [0, T - \Delta\tau] \quad (6.28)$$

where $IM < 0$ denotes that the low energy event occurs. $t^* = 0$ denotes the start time point at the 1000 feet above the ground during the final approach. $t^* = T$ denotes the time point at the 100 feet above the ground. $\Delta\tau$ is the width of the rolling window $[t^*, t^* + \Delta\tau]$, which denotes a certain continuous segment of the flight trajectory. The minimum of the integral of $E_{factor} + 1$ inside the rolling window is used as the low energy incident metric. This metric takes the duration below the UA-based low energy bound into account. For a case that the aircraft flies along the desired trajectory with the target speed, E_{factor} is constant and equals 0. Therefore, $IM = \Delta\tau$ for the desired flight trajectory.

6.4.2 Modeling of Low Energy

The changing of energy is influenced by aircraft dynamics, external forces, and disturbance. In order to predict the E_{factor} during the final approach, a physical model for low energy events is built in this section. The proposed physical-based incident model can be used to identify the contributing factors and map them to the low energy incident metric.

1. External Forces

Aircraft motion is caused by external forces, which mainly include the gravity (\vec{F}_G^G), the aerodynamic forces (\vec{F}_A^G), and the propulsion (\vec{F}_P^G). The total forces \vec{F}_T^G are sum of them shown as below:

$$\vec{F}_T^G = \sum(\vec{F}^G) = (\vec{F}_A^G) + (\vec{F}_G^G) + (\vec{F}_P^G). \quad (6.29)$$

Those forces are explained in detail in the following sections.

① **Aerodynamic Forces** Aerodynamic forces are produced by air flow around the aircraft, denoted in the aerodynamic frame (A frame):

$$(\vec{F}_A^G)_A = \begin{bmatrix} -D \\ Q \\ -L \end{bmatrix}_A = \frac{1}{2}\rho V_A^2 S_{ref} \begin{bmatrix} -C_D \\ C_Q \\ -C_L \end{bmatrix}_A, \quad (6.30)$$

where C_L , C_D , and C_Q denote the lift coefficient, drag coefficient, and side force coefficient, respectively. In this chapter, only the longitudinal motion of the aircraft is considered. Therefore, the side force is assumed to be zero, and $C_Q = 0$. The C_D is computed based on the quadratic lift-drag polar:

$$C_D = C_{D0} + k \cdot C_L^2, \quad (6.31)$$

where k denote the lift-drag polar profile coefficient, which depends on the aircraft configuration. In order to compute the motion of the aircraft, the aerodynamic forces given in the A frame can be transferred into the B frame via:

$$(\vec{F}_A^G)_B = \mathbf{M}_{BA}(\vec{F}_A^G)_A, \quad (6.32)$$

where \mathbf{M}_{BA} is the rotation matrix from the A frame to the B frame as follows:

$$\mathbf{M}_{BA} = \begin{bmatrix} \cos\alpha_A \cos\beta_A & -\cos\alpha_A \sin\beta_A & -\sin\alpha_A \\ \sin\beta_A & \cos\beta_A & 0 \\ \sin\alpha_A \cos\beta_A & -\sin\alpha_A \sin\beta_A & \cos\alpha_A \end{bmatrix}, \quad (6.33)$$

where α_A and β_A are the aerodynamic angle of attack and the aerodynamic angle of side slip.

② Gravity A simplified gravitation model assumes a constant gravitational acceleration. The gravity is given in the North-East-Down (NED) frame (O frame) shown as below:

$$(\vec{\mathbf{F}}_G^G)_O = m \cdot g_0 \cdot \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}_O, \quad (6.34)$$

where $g_0 = 9.80665 \text{ m/s}^2$, which is the gravitational acceleration. The gravity can be transferred into the B frame via

$$(\vec{\mathbf{F}}_G^G)_B = \mathbf{M}_{BO}(\vec{\mathbf{F}}_G^G)_O, \quad (6.35)$$

with the rotation matrix from the O frame to the B frame:

$$\mathbf{M}_{BO} = \begin{bmatrix} \cos\Psi\cos\Theta & \sin\Psi\cos\Theta & -\sin\Theta \\ \cos\Psi\sin\Theta\sin\Phi - \sin\Psi\cos\Phi & \sin\Psi\sin\Theta\sin\Phi + \cos\Psi\cos\Phi & \cos\Theta\sin\Phi \\ \cos\Psi\sin\Theta\cos\Phi + \sin\Psi\sin\Phi & \sin\Psi\sin\Theta\cos\Phi - \cos\Psi\sin\Phi & \cos\Theta\cos\Phi \end{bmatrix}, \quad (6.36)$$

where (Ψ, Θ, Φ) are the rotation angles between the B frame and the O frame. Similarly, The gravity can be transferred into the A frame via

$$(\vec{\mathbf{F}}_G^G)_A = \mathbf{M}_{AO}(\vec{\mathbf{F}}_G^G)_O, \quad (6.37)$$

with the rotation matrix from the O frame to the A frame:

$$\mathbf{M}_{AO} = \begin{bmatrix} \cos\chi_A\cos\gamma_A & \sin\chi_A\cos\gamma_A & -\sin\gamma_A \\ \cos\chi_A\sin\gamma_A\sin\mu_A - \sin\chi_A\cos\mu_A & \sin\chi_A\sin\gamma_A\sin\mu_A + \cos\chi_A\cos\mu_A & \cos\gamma_A\sin\mu_A \\ \cos\chi_A\sin\gamma_A\cos\mu_A + \sin\chi_A\sin\mu_A & \sin\chi_A\sin\gamma_A\cos\mu_A - \cos\chi_A\sin\mu_A & \cos\gamma_A\cos\mu_A \end{bmatrix}, \quad (6.38)$$

where $(\chi_A, \gamma_A, \mu_A)$ are the rotation angles between the O frame and the A frame.

③ Propulsion The propulsion force is usually given in the B frame, denoted by three components X_P , Y_P , and Z_P . By assuming that the propulsion force points directly to the x-axis of the B frame, the simplified propulsion force in the B frame is given as follows:

$$(\vec{\mathbf{F}}_P^G)_B = \begin{bmatrix} X_P \\ Y_P \\ Z_P \end{bmatrix}_B = \begin{bmatrix} T \\ 0 \\ 0 \end{bmatrix}_B, \quad (6.39)$$

where T is the thrust. The propulsion forces can be transferred into the A frame via

$$(\vec{\mathbf{F}}_P^G)_A = \mathbf{M}_{AB}(\vec{\mathbf{F}}_P^G)_B, \quad (6.40)$$

with the rotation matrix from the B frame to the A frame:

$$\mathbf{M}_{AB} = \mathbf{M}_{BA}^T. \quad (6.41)$$

2. Wind Speed in Runway Frame

The local navigation frame, runway frame N , is commonly used for the analysis in the final approach phase. The runway threshold THR is the origin point. The x_N -axis is aligned with the runway centerline, pointing from the runway threshold to the runway ends. The y_N -axis is pointing to the right perpendicular to the x_N in the horizontal plane. The z_N -axis is pointing downwards. The three wind components: tailwind V_{tws} , crosswind V_{cws} , and downwash V_{dws} , are also shown in Figure 6.11.

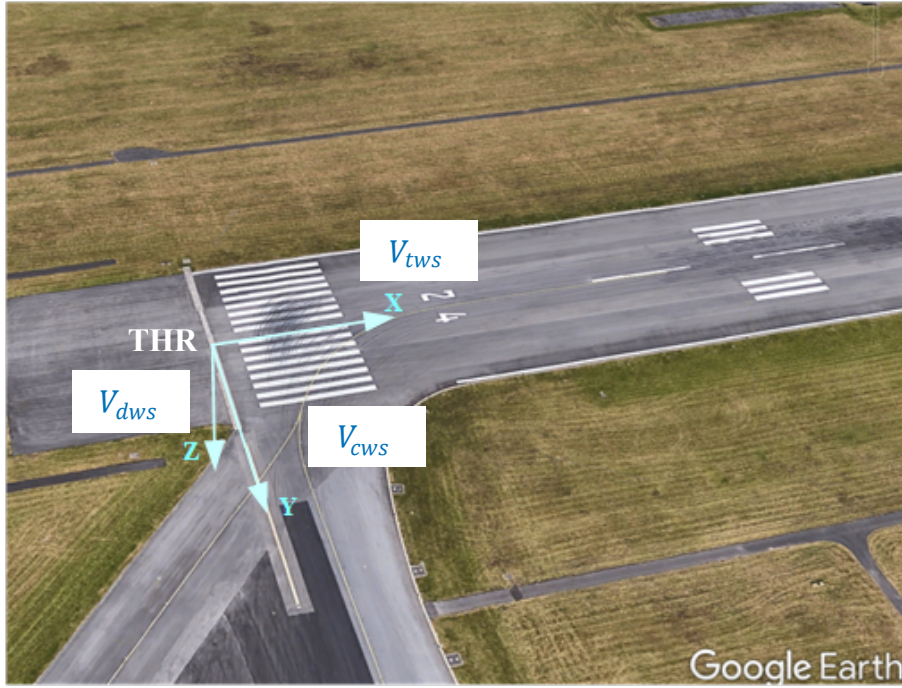


Figure 6.11: The runway frame N with three wind components.

The runway frame is also the inertial frame. The difference between the N frame and the O frame is the runway bearing angle χ_{rwy} and the original point. For example, χ_{rwy} is 240 degrees for the runway 24 at Luxembourg airport in Figure 6.11. The derivative of V_W in N frame is shown as below:

$$(\dot{\vec{V}}_W^G)^{EO} = (\dot{\vec{V}}_W^G)^{EN} = \mathbf{M}_{ON}(\dot{\vec{V}}_W^G)^{EN} = \mathbf{M}_{ON} \begin{bmatrix} \dot{V}_{tws} \\ \dot{V}_{cws} \\ \dot{V}_{dws} \end{bmatrix}_N^{EN}, \quad (6.42)$$

where M_{ON} is shown as below:

$$\mathbf{M}_{ON} = \begin{bmatrix} \cos\chi_{rwy} & -\sin\chi_{rwy} & 0 \\ \sin\chi_{rwy} & \cos\chi_{rwy} & 0 \\ 0 & 0 & -1 \end{bmatrix}. \quad (6.43)$$

3. Equations of Energy Rate

The external aircraft forces and wind disturbance have been modeled in the previous sections. The energy rate is derived in this section based on the aircraft dynamics. The linear translation equations of the motion for the point mass are shown as follows:

$$\sum(\vec{F}^G)_O = m \cdot (\dot{\vec{V}}_K^G)_{O}^{II}. \quad (6.44)$$

Assuming Earth is flat and ignoring its rotation:

$$(\vec{\omega}^{EO})_O = 0, \quad (\vec{\omega}^{IE})_O = 0. \quad (6.45)$$

Equation (6.44) can be further derived as below:

$$(\dot{\vec{V}}_K^G)_{O}^{II} = (\dot{\vec{V}}_K^G)_{O}^{EO} = \frac{1}{m} \cdot \sum(\vec{F}^G)_O. \quad (6.46)$$

This is also valid in A Frame:

$$(\dot{\vec{V}}_K^G)_{A}^{EO} = \frac{1}{m} \cdot \sum(\vec{F}^G)_A. \quad (6.47)$$

By using the linear translation equation:

$$(\dot{\vec{V}}_K^G)_{A}^{EO} = (\dot{\vec{V}}_K^G)_{A}^{EA} + (\vec{\omega}^{OA})_A \times (\vec{V}_K^G)_A^E, \quad (6.48)$$

Equation (6.47) is derived as below:

$$(\dot{\vec{V}}_K^G)_{A}^{EA} = \frac{1}{m} \cdot \sum(\vec{F}^G)_A - (\vec{\omega}^{OA})_A \times (\vec{V}_A^G)_A^E. \quad (6.49)$$

The total external forces $\sum(\vec{F}^G)$ in A frame are calculated as follows:

$$\begin{aligned} \sum(\vec{F}^G)_A &= (\vec{F}_A^G) + \mathbf{M}_{AO}(\vec{F}_O^G) + \mathbf{M}_{AB}(\vec{F}_P^G) \\ &= \begin{bmatrix} -D \\ Q \\ -L \end{bmatrix}_A + \mathbf{M}_{AO}(\chi_A, \gamma_A, \mu_A) \begin{bmatrix} 0 \\ 0 \\ mg_0 \end{bmatrix}_O + \mathbf{M}_{AB}(\alpha_A, \beta_A) \begin{bmatrix} X_P \\ Y_P \\ Z_P \end{bmatrix}_B. \end{aligned} \quad (6.50)$$

By integrating the external forces, Equation (6.18) can be further derived as below:

$$\begin{aligned} (\dot{\vec{V}}_A^G)_A^{EA} &= (\dot{\vec{V}}_K^G)_A^{EA} - (\dot{\vec{V}}_W^G)_A^{EA} \\ &= \frac{1}{m} \cdot \sum(\vec{F}^G)_A - \left((\dot{\vec{V}}_W^G)_A^{EO} - (\vec{\omega}^{OA})_A \times (\vec{V}_A^G)_A^E \right) \\ &= \frac{1}{m} \cdot \sum(\vec{F}^G)_A - \left(\mathbf{M}_{AO}(\dot{\vec{V}}_W^G)_O^{EO} - (\vec{\omega}^{OA})_A \times (\vec{V}_A^G)_A^E \right), \end{aligned} \quad (6.51)$$

which is rewritten in terms of its three components:

$$\begin{aligned} (\dot{\vec{V}}_A^G)_A^E &= \begin{bmatrix} \dot{V}_A^G \\ 0 \\ 0 \end{bmatrix}_A^{EA} = \frac{1}{m} \cdot \left(\begin{bmatrix} -D \\ Q \\ -L \end{bmatrix}_A + \mathbf{M}_{AO} \begin{bmatrix} 0 \\ 0 \\ mg_0 \end{bmatrix}_O + \mathbf{M}_{AB} \begin{bmatrix} X_P \\ Y_P \\ Z_P \end{bmatrix}_B \right) \\ &\quad - \mathbf{M}_{AO} \mathbf{M}_{ON} \begin{bmatrix} \dot{V}_{tws} \\ \dot{V}_{cws} \\ \dot{V}_{dws} \end{bmatrix}_N^{EN} - (\vec{\omega}^{OA})_A \times \begin{bmatrix} V_A^G \\ 0 \\ 0 \end{bmatrix}_A^E. \end{aligned} \quad (6.52)$$

Since only the aircraft's motion in the vertical plane is concerned in the study of low energy events during the final approach in this thesis, assuming there is no sideslip and bank angle. Therefore, $\mu_A \approx 0$, and $\beta_A \approx 0$, and $\chi_A \approx \chi_K \approx \chi_{rwy}$. Equation (6.52) can be further simplified as below:

$$\begin{bmatrix} \dot{V}_A^G \\ 0 \\ 0 \end{bmatrix}_A^{EA} = \begin{bmatrix} \frac{-D+X_P \cdot \cos\alpha_A + Z_P \cdot \sin\alpha_A}{m} - g_0 \sin\gamma_A - \dot{V}_{tws} \cdot \cos\gamma_A + \dot{V}_{dws} \cdot \sin\gamma_A \\ \frac{Q+Y_P}{m} + \dot{\chi}_A \cos\gamma_A V_A^G - \dot{V}_{cws} \\ \frac{-L-X_P \cdot \sin\alpha_A + Z_P \cdot \cos\alpha_A + \dot{V}_{tws} \cdot \sin\gamma_A + \dot{V}_{dws} \cdot \cos\gamma_A - \dot{\gamma}_A V_A^G}{m} + g_0 \cos\gamma_A \end{bmatrix}. \quad (6.53)$$

Assuming the propulsion forces are aligned to the x -axis in B frame. $Y_P = Z_P = 0$, and $X_P = T$. Since $\beta_A \approx 0$, the aerodynamic side force $Q \approx 0$. The first row of Equation (6.53) is derived as follows:

$$\dot{V}_A^G = \frac{T \cdot \cos\alpha_A - D}{m} - g_0 \sin\gamma_A - \dot{V}_{tws} \cdot \cos\gamma_A + \dot{V}_{dws} \cdot \sin\gamma_A. \quad (6.54)$$

Relying on the Taylor expansion of triangular functions for small α_A and γ_A , and neglecting second-order terms, the equation becomes:

$$\dot{V}_A^G = \frac{T - D}{m} - g_0 \cdot \gamma_A - \dot{V}_{tws}. \quad (6.55)$$

Since $\gamma_A \approx (\dot{h} + V_{dws})/V_A$,

$$\begin{aligned} \dot{V}_A^G &\approx \frac{T - D}{m} - g_0 \cdot \frac{\dot{h} + V_{dws}}{V_A} - \dot{V}_{tws} \\ &= g_0 \left(\frac{T - D}{mg_0} - \frac{\dot{h}}{V_A} - \left(\frac{\dot{V}_{tws}}{g_0} + \frac{V_{dws}}{V_A} \right) \right). \end{aligned} \quad (6.56)$$

By substituting \dot{V}_A in Equation (6.14) using Equation (6.56), the energy bleed rate is:

$$\begin{aligned} \dot{E}_s &= \dot{h} + V_A \left(\frac{T - D}{mg_0} - \frac{\dot{h}}{V_A} - \left(\frac{\dot{V}_{tws}}{g_0} + \frac{V_{dws}}{V_A} \right) \right) \\ &= V_A \left(\frac{T - D}{mg_0} - \left(\frac{\dot{V}_{tws}}{g_0} + \frac{V_{dws}}{V_A} \right) \right). \end{aligned} \quad (6.57)$$

Due to that $\dot{E}_{s,LB} = \dot{h}_{LB} = V_{gs} \tan(\gamma_{-1dot})$, the changing rate of the energy deviation during the final approach is:

$$\Delta \dot{E}_s = V_A \left(\frac{T - D}{mg_0} - \left(\frac{\dot{V}_{tws}}{g_0} + \frac{V_{dws}}{V_A} \right) \right) - V_{gs} \cdot \tan(\gamma_{-1dot}), \quad (6.58)$$

where the wind-related term in Equation (6.58) contains the tailwind gradient and the downwash as below:

$$F = \frac{\dot{V}_{tws}}{g_0} + \frac{V_{dws}}{V_A}. \quad (6.59)$$

The wind-related term is also called the wind shear index F factor, which is used to evaluate the severity of the wind shear [158]. According to the FAA regulations, the average F factor among the 1-km trajectory, denoted by F_{avg} , is used as its hazard metric for the wind shear detection systems on jet transports. The wind shear hazardous is considered for $F_{avg} > 0.1$, and the alert threshold is 0.13 to avoid nuisance alert [158, 159]. The calculation of F_{avg} is applied in the final approach phase of 849 flights. The maximum value of F_{avg} in each flight is extracted. The histogram of the maximum F_{avg} is shown in Figure 6.12.

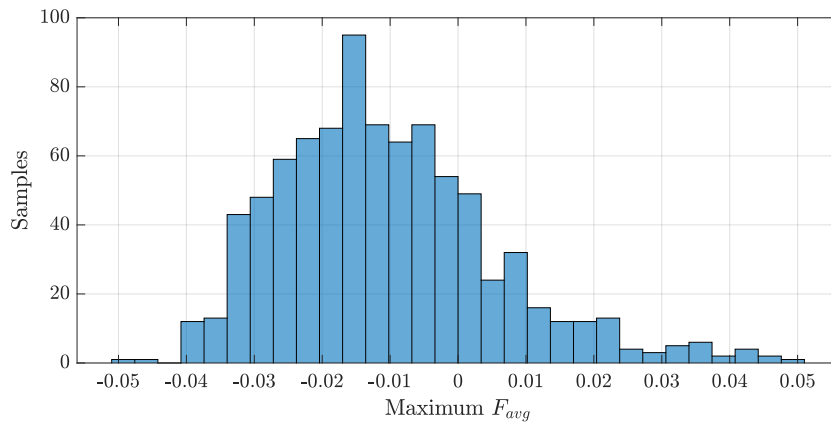


Figure 6.12: Histogram of the maximum F_{avg} factor.

Besides the aircraft dynamics and the external environment, the thrust force and flight trajectory still need to be controlled. The speed tracking autothrottle (A/T) system and the autopilot (AP) system for glide slope tracking are used and described below.

4. Speed Tracking Autothrottle System

According to the SOPs of the stable approach mentioned in Section 6.2.1, the pilot should maintain the airspeed to the selected approach speed target value V_{app} . The velocity is mainly held via the thrust [160]. The structure of the speed tracking autothrottle system during the final approach is shown in Figure 6.13.

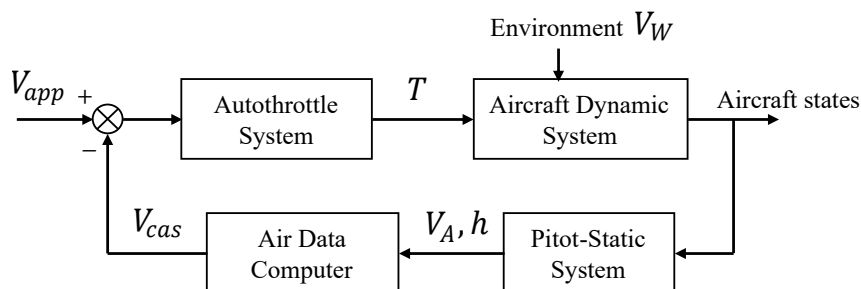


Figure 6.13: Speed tracking autothrottle system.

The thrust force T produced by the propulsion system depends on several parameters, such as throttle angle level (TLA), fan speed $N1$, true airspeed V_A , and environmental factors, like the static temperature T_s and the static pressure p_s . For the autothrottle system of Boeing aircraft, the flight mode ‘SPD’ is engaged during the final approach phase. ‘SPD’ mode means that the autothrottle system maintains speed command, which can be set using the Mode Control Panel (MCP) IAS/MACH selector or the Flight Management Computer (FMC). V_{app} is used as a speed command. The simplified autothrottle system using a proportional-integral (PI) controller is shown in Figure 6.14.

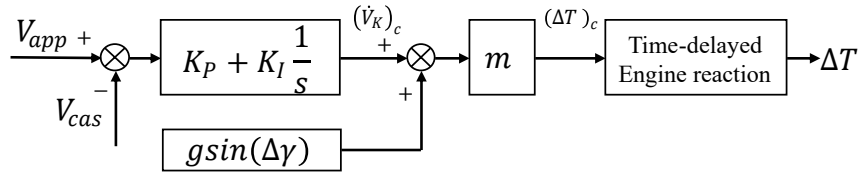


Figure 6.14: *Simplified autothrottle system block diagram*

In Figure 6.14, m denotes the mass of the aircraft. $(\dot{V}_K)_c$ means the desired kinematic speed gradient. $(\Delta T)_c$ denotes the desired extra thrust. The time delay of engine reaction is simplified using a PT1 system with the time constant T_d . The trimmed thrust can be computed based on the equilibrium equations of forces in flight path direction and its perpendicular direction as below:

$$\begin{aligned} L &= mg \cdot \cos(\gamma_{GS}); \\ T_{trimmed} &= D + mg \cdot \sin(\gamma_{GS}). \end{aligned} \quad (6.60)$$

Three unknown parameters K_P , K_I , and T_d need to be estimated in the simplified autothrottle system. According to the detection of the flight mode ‘SPD’ from the QAR data, the autothrottle system is engaged in 89 flights of 849 B748 flights during the final approach. By utilizing those flights data, the equation error method (EEM) is applied to minimize the mean square error (MSE) of the recorded time series and the simulated series of the model output, for example, V_{cas} in this case. Since the content in one flight might not enough, several flights could be used in estimation together, which is suggested in [77]. The estimated results are shown in Table 6.4. The comparison of the raw V_{cas} and the simulated V_{cas} using the estimated parameters are illustrated using three flights in Figure 6.15 .

Table 6.4: *Estimated parameters in autothrottle system*

Parameters	Value
K_P	0.1075
K_I	0.0064
T_d	2.3493

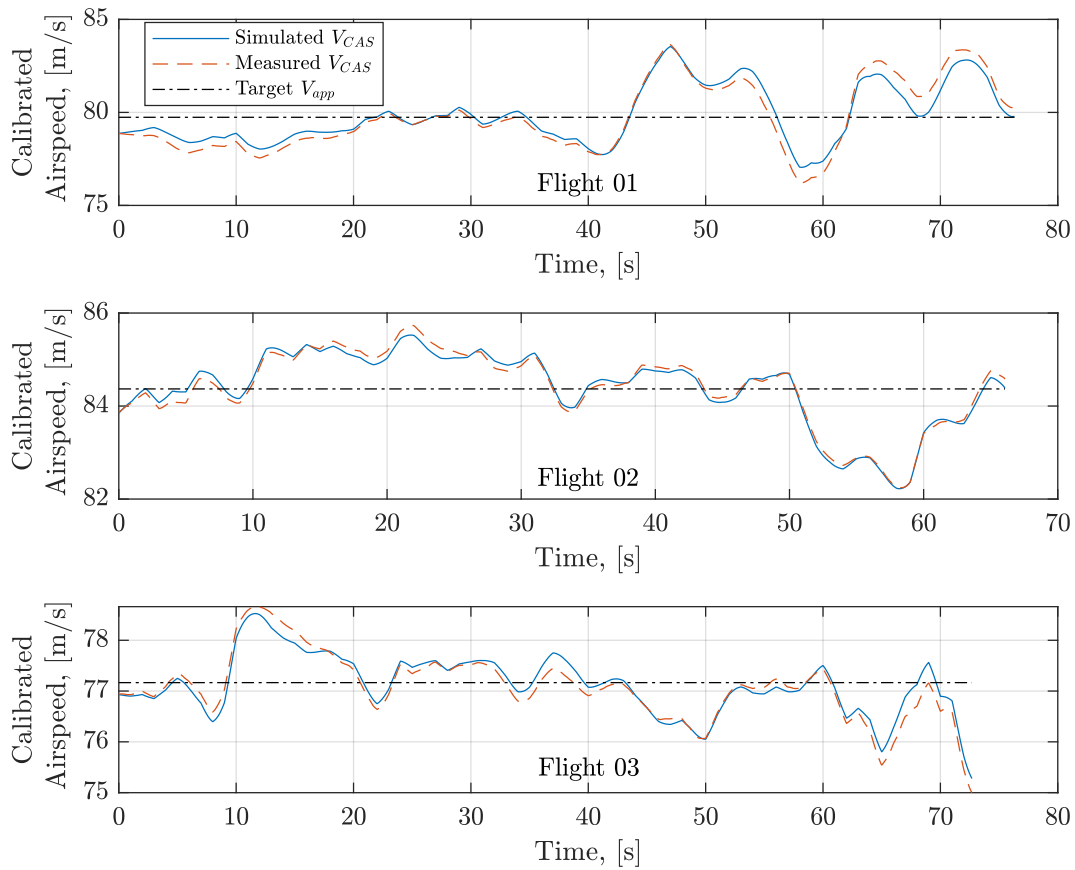


Figure 6.15: Comparison between the raw calibrated airspeed and the simulated calibrated airspeed using the estimated parameters in Table 6.4.

5. Glideslope Tracking Autopilot System

During the final approach phase, pilots try to track the glideslope with a constant flight path angle of $\gamma \approx -3^\circ$, depending on the runway. The control structure for the glideslope tracking is shown in Figure 6.16.

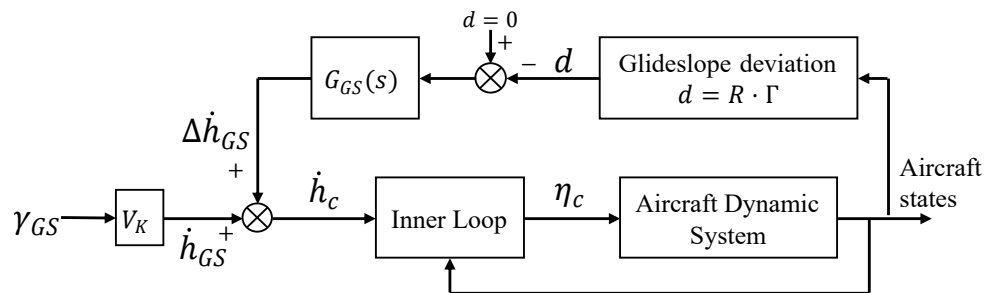


Figure 6.16: Glideslope tracking structure

In the structure of the glideslope tracking system, d is the vertical deviation between the current position of the aircraft and the glideslope. It can be computed using the distance R from the aircraft position to the runway threshold and the glideslope deviation angle Γ . According to Equation (6.4), Γ can be computed based on the aircraft position as below:

$$\Gamma = 0.12 \cdot \gamma_{GS} \cdot \frac{\delta_{GS,DDM}}{0.0875} = \text{atan}\left(\frac{h_{ILS}}{\sqrt{(x_N - x_{GS})^2 + (y_{GS})^2}}\right) - \gamma_{GS}. \quad (6.61)$$

In Figure 6.16, $G_{GS}(s)$ is used to eliminate the aircraft vertical deviation to the glideslope by using a PI controller with $K_{P,gs}$ and $K_{I,gs}$. In addition, the inner loop and the aircraft dynamics are simplified shown in Figure 6.17. The altitude controller is utilized for the control of the flight path. $(T_{\theta 2}s + 1)/(T_{\theta 2}^*s + 1)$ is the lead-lag-filter to convert the flight path rate $\dot{\gamma}_c$ to the pitch rate command q_c [161]. $T_{\theta 2} = 1/Z_\alpha$ and $T_{\theta 2}^* \ll T_{\theta 2}$. The transfer function $G_{qq_c}(s)$ is very fast and is assumed to be 1. In summary, the parameters $K_{P,gs}$, $K_{I,gs}$, K_γ , and $T_{\theta 2}^*$ are required.

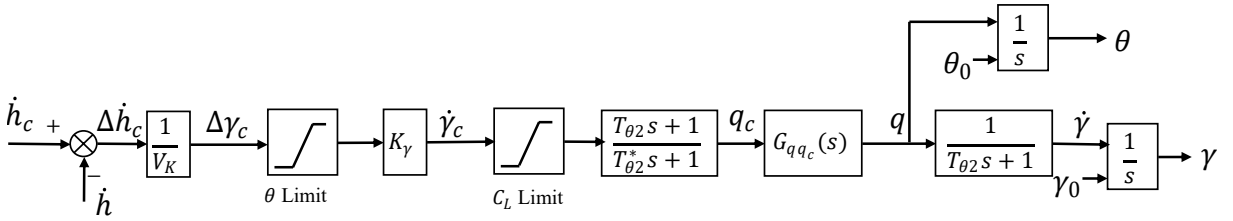


Figure 6.17: Simplified inner loop and aircraft dynamics

The same approach in the estimation of the autothrottle system is applied in the estimation of unknown parameters in the glideslope tracking system. 89 flights of B748 with engaged AP in the final approach phase are used. In practice, the flight path trajectory is almost located along the glideslope. There is no considerable variation in the flight path angle and the vertical deviation to the glideslope. Based on the flight data and the engineering experience, the value of unknown parameters used in this thesis is shown in Table 6.5.

Table 6.5: Parameters in the glideslope tracking autopilot system.

Parameters	Value
$K_{P,gs}$	1
$K_{I,gs}$	0.01
K_γ	0.5
$T_{\theta 2}$	2
$T_{\theta 2}^*$	0.5

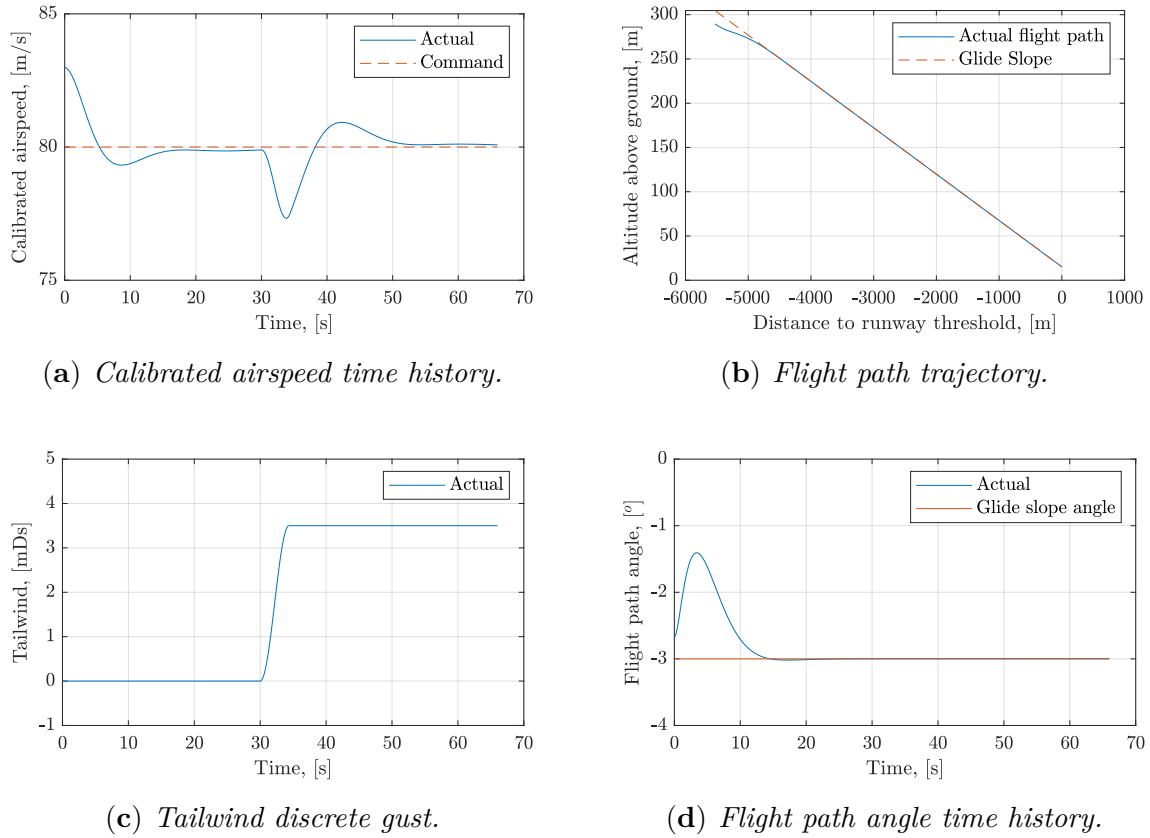


Figure 6.18: Time histories of aircraft states during the final approach with the initial deviation in the speed and the altitude, and the gust encountered at 30 seconds.

To validate the speed tracking autothrottle system and the glideslope tracking autopilot system, an example of simulation starting from 1000 feet above the runway elevation to the runway threshold is given. The initial approach speed deviation ΔV_{app} is 3 m/s, and the initial altitude deviation Δh is 50 feet. Besides, the aircraft encounters a tailwind gust at 30 seconds. The time series of the airspeed and the vertical deviation is shown in Figure 6.18. Simulation results show that the aircraft can recover to the glideslope and maintain the approach speed target.

6.4.3 Identify Contributing Factors

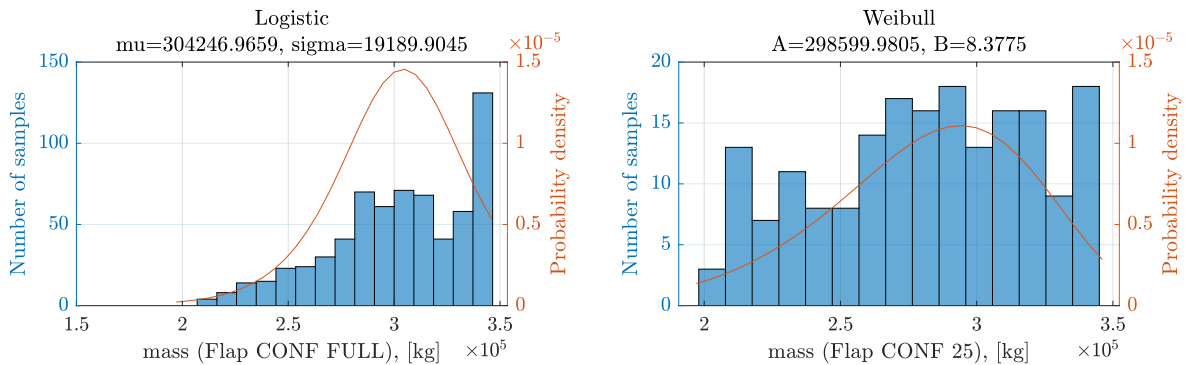
During daily operations, the aircraft should stabilize at 1000 feet above the runway elevation. Otherwise, a go-around should be initiated. Since the go-around procedure is out of the scope of this thesis, this chapter focuses on the low energy event during the final approach due to late destabilization mentioned in Section 6.2.3. The predefined scenario starts from 1000 feet above the runway elevation with no initial deviation at the speed and altitude. Therefore, $\Delta V_{app} = 0$ and $\delta_{GS, dot} = 0$ at the start point : 1000 feet above the

runway elevation. All parameters involved in the modeling of low energy events during the final approach are summarized as contributing factors in Table 6.6. The ‘OPS’ stands for ‘operations’, and the ‘ENV’ stands for ‘environments’.

Table 6.6: *Contributing factors to low energy events during the final approach*

Contributing Factors	Symbols	Type	-	Time Frame
Mass	m	OPS	Continues	During the final approach
Flap	δ_F	OPS	Discrete	During the final approach
Air density	ρ_{TD}	ENV	Continues	At the touch down point
Downwash	V_{dws}	ENV	Continues	During the final approach
Tailwind gradient	\dot{V}_{tws}	ENV	Continues	During the final approach
Headwind	$V_{hws,TD}$	ENV	Continues	At the touch down point
Headwind deviation	$\Delta V_{hws,TD}$	ENV	Continues	At the touch down point
Gust intensity	V_{gust}	ENV	Continues	During the final approach

Since the aircraft mass does not change too much during the final approach and landing, the mass is assumed to be constant during the final approach and the average mass during the final approach as the measurement. Two types of flap configurations ‘CONF 25’ and ‘CONF FULL’ are used for the Boeing 748F landing according to the OM-B. 659 flights with ‘CONF FULL’ and 187 flights with ‘CONF 25’ of the Boeing 748F landing at the same runway are used for analysis. The histogram and distribution of mass are shown in Figure 6.19. There is a significant difference in mass distribution between the two flap settings.



(a) *Landing mass with Flap ‘CONF FULL’.*

(b) *Landing mass with Flap ‘CONF 25’.*

Figure 6.19: *Distributions of the landing mass during the final approach*

Figure 6.19 shows that there is a large deviation between the fitted distribution and the histogram. Instead of all parametric distribution families, the non-parametric kernel distribution is used, and the distribution is truncated in the boundary of the mass during operations. Results in Figure 6.20 show that the estimated kernel distribution has a good match with the flight data.

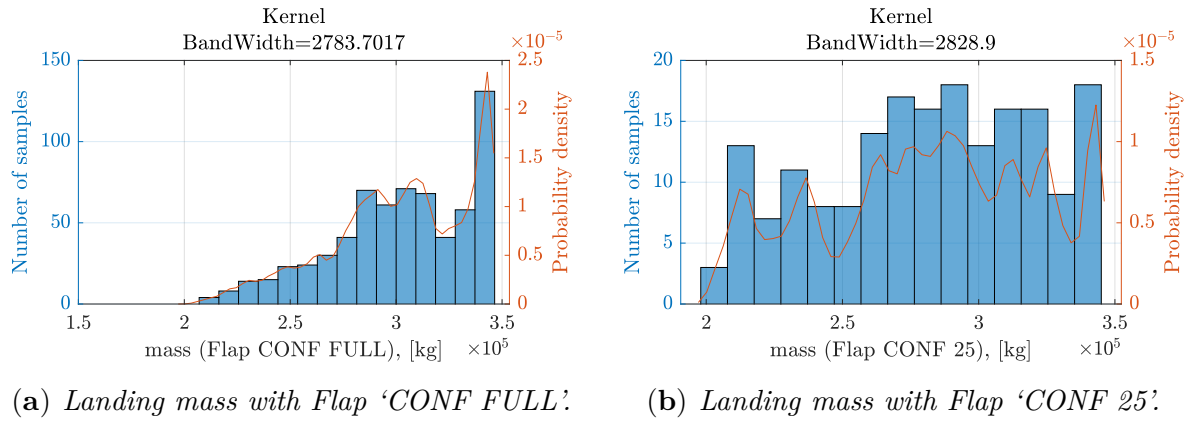


Figure 6.20: Kernel distributions of the landing mass during the final approach

The air density at the touchdown is used as a contributing factor. The air density along the final approach trajectory can be computed based on the altitude as below:

$$\rho = \rho_R \left(1 - \frac{n-1}{n} \cdot \frac{h}{H_0} \right)^{\frac{1}{n-1}}. \quad (6.62)$$

The polytropic exponent $n = 1.235$, and the geopotential height $H_0 = 8432 \text{ m}$. ρ_R is the measurement of the air density at the touchdown ρ_{TD} . h is the height deviation between the aircraft and the runway threshold. The histogram and the estimated distribution of the air density at the touchdown are shown in Figure 6.21. Two distributions of air density with different flap settings are similar according to the flight measurements.

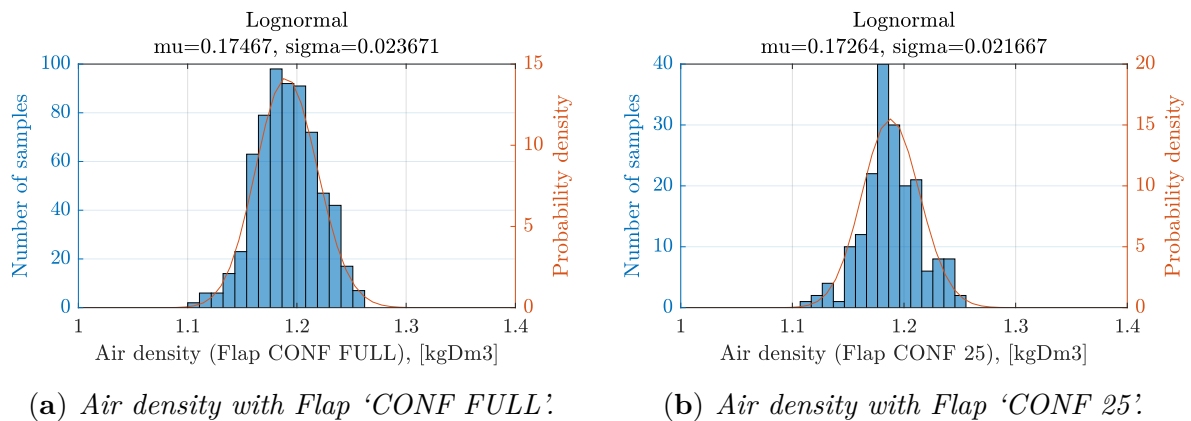
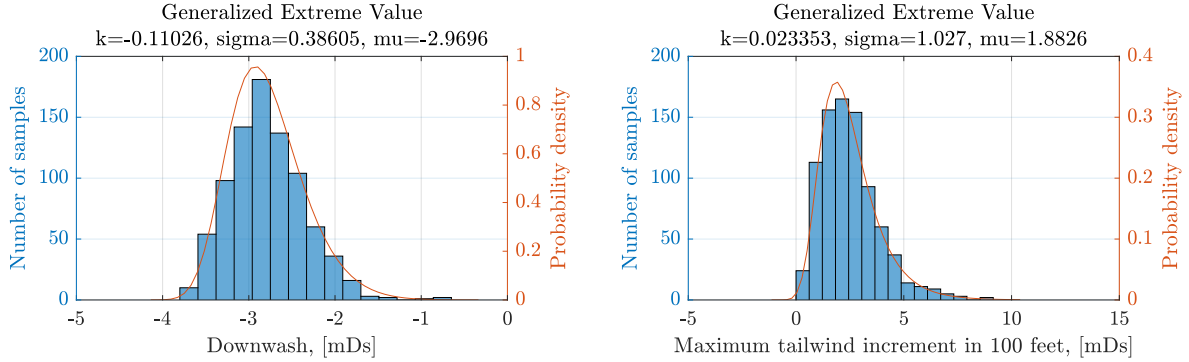


Figure 6.21: Distributions of the air density at the touchdown point

According to the low energy model built in the previous section, the downwash and tailwind gradient play a big role in the energy bleed rate. The downwash is assumed to be constant during the final approach, the mean value of the downwash series during the final approach is used as the measurement. The obtained distribution of the downwash is shown in Figure 6.22(a). Since most measurements of the downwash are negative, the upwash prevails along with the final approach trajectory in this local area.



(a) Downwash during the final approach.

(b) Maximum tailwind increment in 100 feet.

Figure 6.22: Distributions of the downwash and the maximum tailwind increment in 100 feet during the final approach phase

The encountered tailwind gradient depends on the meteorology and the aircraft trajectory. It can be expressed in three terms:

$$\dot{V}_{tws} = \frac{\partial V_{tws}}{\partial x} \dot{x} + \frac{\partial V_{tws}}{\partial z} \dot{z} + \frac{\partial V_{tws}}{\partial t}, \quad (6.63)$$

where x denotes the direction along with the runway, and z denotes the altitude. The first term denotes horizontal wind shear, which is the primary contributor to aircraft performance loss. The second term is the vertical wind shear, and it predominates during the aircraft climbs or descends. The third term is the local change rate of the tailwind with the time, which is usually secondary. During the normal final approach, the estimated wind is along the flight trajectory. The horizontal shear, vertical shear, and the local change of tailwind can not be distinguished directly from the recorded wind data in the QAR. Assume that the wind field during the final approach is frozen. Therefore, only the vertical shear is considered during the simulation. The simplified expression of the tailwind derivative is as follows:

$$\dot{V}_{tws} = \frac{\partial V_{tws}}{\partial h} \dot{h}, \quad (6.64)$$

where $\partial V_{tws}/\partial h$ is a metric for the wind shear. The wind shear intensity criteria recommended by the fifth Air Navigation Conference [162] are shown in Table 6.7. Measurements of the maximum tailwind increment in 100 feet along the final approach trajectory are extracted. The corresponding histogram and the estimated distribution are shown in Figure 6.22(b). However, the estimated distribution of the tailwind increment will not

be used directly for sampling in simulation. Instead, the tailwind series generator is built using the KL expansion method proposed in Chapter 5. By generating the random samples of the KL coefficients from their corresponding distributions, the tailwind series are constructed along with the altitude during the final approach phase. The probability of the maximum tailwind increment in 100 feet (denoted by $\max \Delta V_{tws}$) exceeding a certain value is estimated using the subset simulation and the KL-based tailwind model, which is shown in Table 6.7. The occurrence probability of the severe wind shear decreases sharply compared to the strong wind shear.

Table 6.7: *Wind shear intensity criteria [162]*

Intensity	ΔV_{tws} [m/s] in 100 feet	$\partial V_{tws}/\partial h$ [s ⁻¹]	Estimated probability using the KL-based wind model
Light	0 ~ 2.0	0 ~ 0.07	$Pr(\max \Delta V_{tws} \leq 2.0) = 0.3804$
Moderate	2.1 ~ 4.0	0.08 ~ 0.13	$Pr(\max \Delta V_{tws} > 2.0) = 0.6196$
Strong	4.1 ~ 6.0	0.14 ~ 0.20	$Pr(\max \Delta V_{tws} > 4.0) = 0.0359$
Severe	> 6.0	> 0.20	$Pr(\max \Delta V_{tws} > 6.0) = 6.1389 \cdot 10^{-4}$

According to Equation (6.3), another two environmental factors, the headwind at touchdown $V_{hws,TD}$ and the gust intensity during the final approach V_{gust} , influence the calculation of V_{app} . The corresponding distributions of their measurements from 846 flights are shown in Figure 6.23.

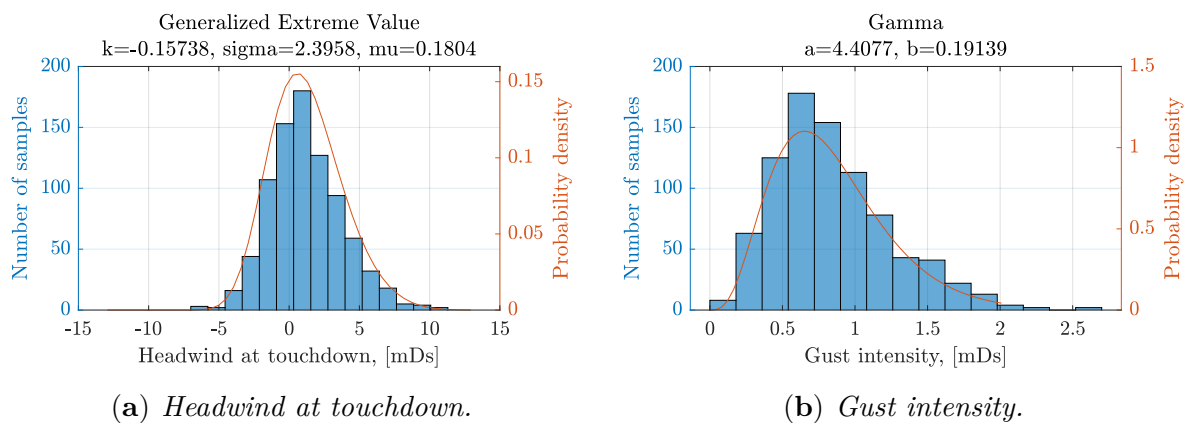


Figure 6.23: *Distributions of the headwind and the gust intensity*

The $\Delta V_{hws,TD}$ is the headwind deviation between the METAR information and the encountered wind, and its distribution is shown in Figure 6.2. Instead of sampling directly from the estimated distributions of $V_{hws,TD}$ and V_{gust} in Predictive Analysis, the KL-based tailwind model is used. These contributing factors related to the tailwind can be calculated based on the generated tailwind series. 5000 wind series are generated using the

KL-based tailwind model, and 5000 samples of $V_{hws,TD}$ and V_{gust} are obtained from the generated wind series. The PDFs of the two wind-related factors are estimated based on the 5000 samples, which are compared with the estimated PDFs based on the direct measurements. Results in Figure 6.24 show that the estimated PDFs of $V_{hws,TD}$ and V_{gust} using the raw data and the simulated data match very well.

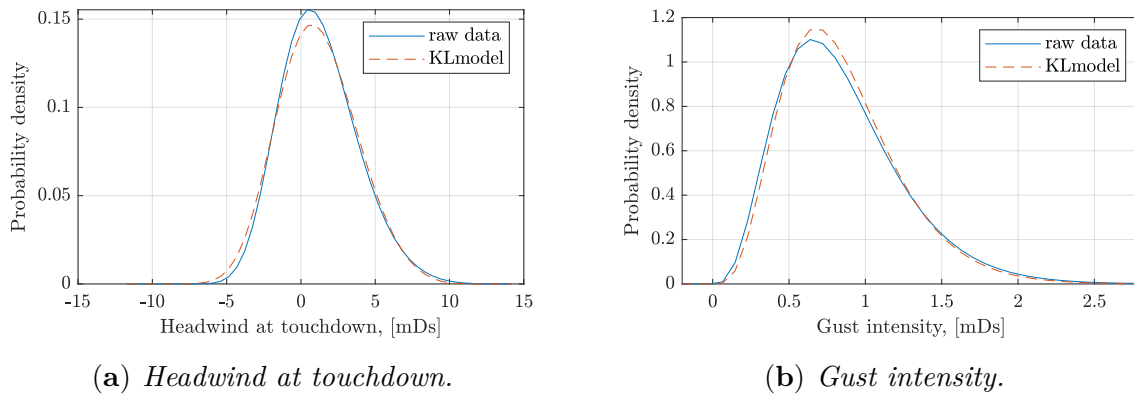


Figure 6.24: Estimated PDFs of two wind-related factors using the raw data and the simulated data from the KL-based tailwind model

6.4.4 Occurrence Probability of Low Energy Event

By propagating uncertainties in all contributing factors through the low energy incident model, the occurrence probability of the low energy event due to destabilization during the final approach can be quantified using the Predictive Analysis. The pure Monte Carlo simulation is executed using 10000 samples of the model inputs, randomly generated from the distributions of four contributing factors ($mass, \rho_{TD}, V_{dws}, \Delta V_{hws,TD}$) and 20 coefficients ξ_k in KL wind model. For the incident metric IM , the width of the time window $\Delta\tau$ in Equation (6.28) is set to 6 seconds. The histogram and the estimated distribution of IM are shown in Figure 6.25. There are no samples located in the failure domain $IM < 0$, which does not mean the probability of failure is zero.

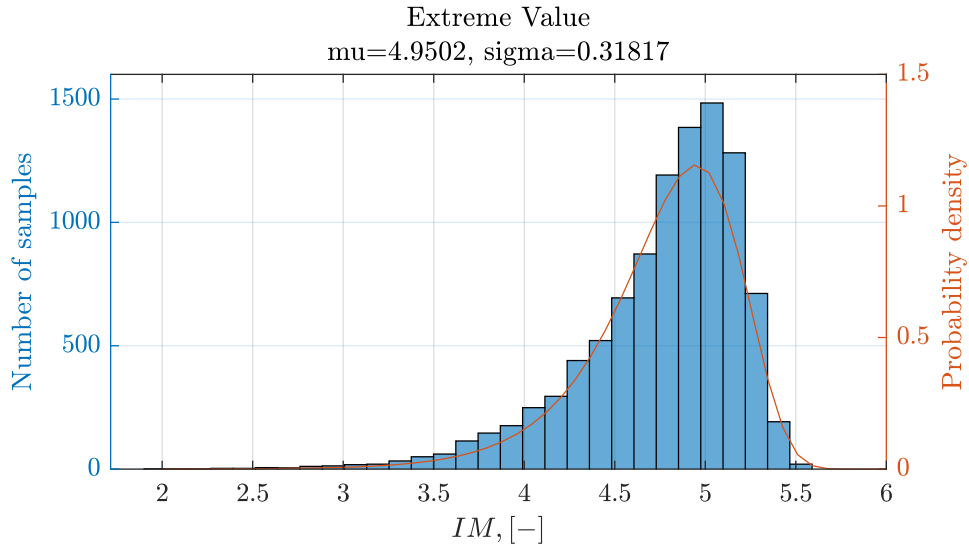


Figure 6.25: Histogram of IM based on the MCS with 10000 samples.

To estimate the rare event probability, the subset simulation is applied. The number of samples in each subset N is set to 2000. The conditional probability p_0 is set to 0.1. The adaptive conditional MCMC sampling method is used to generate new samples in each subset level. According to the samples generated in different subset levels, the cumulative CDF of the incident metric is calculated and denoted by the blue line shown in Figure 6.26. The estimated probability of $IM < 0$ is $2.81 \cdot 10^{-7}$, marked by the blue circle. The blue error bar and the green error bar denote the small and larger covariance boundaries calculated using Equation (3.87) and Equation (3.86) in Section 3.3.3. The black circles denote the intermediate threshold in each subset level. The histograms of the incident metric samples in each subset level are shown in Figure 6.27(a). The corresponding duration of the aircraft below the energy bound during the flight is also calculated and shown in Figure 6.27(b). The duration below the low energy bound in final subset level is less than the $\Delta\tau = 6s$, since there are segments above the low energy bound in the integral interval.

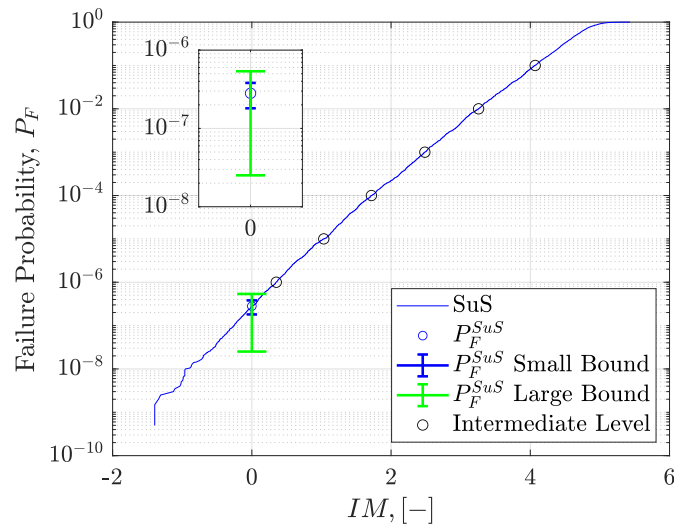
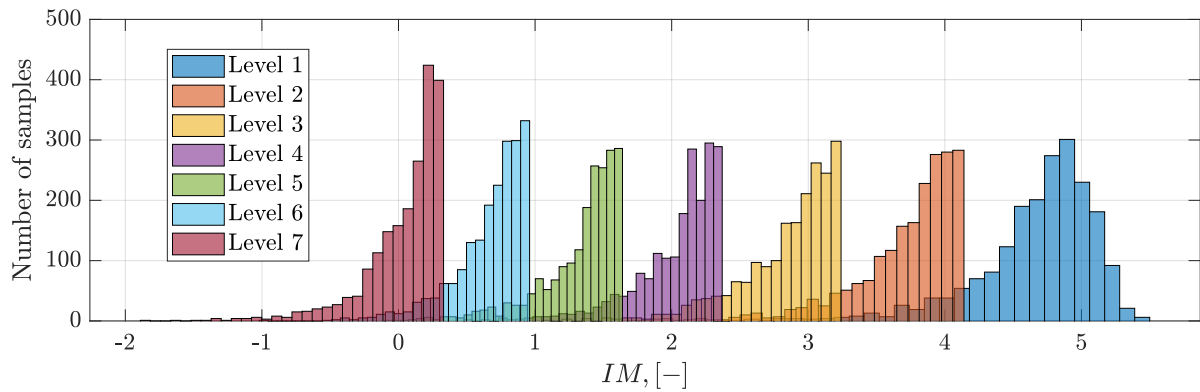
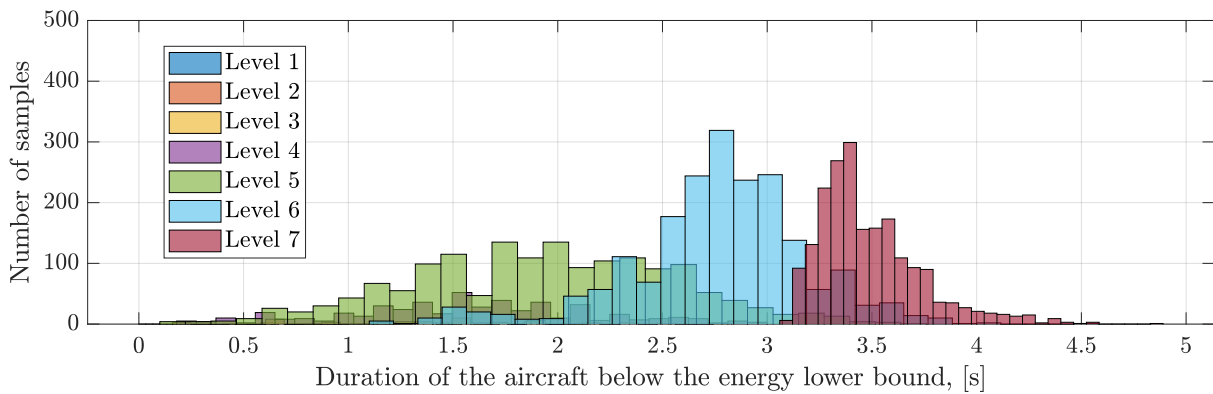


Figure 6.26: Cumulative CDF of the incident metric based on the samples generated in the subset simulation.



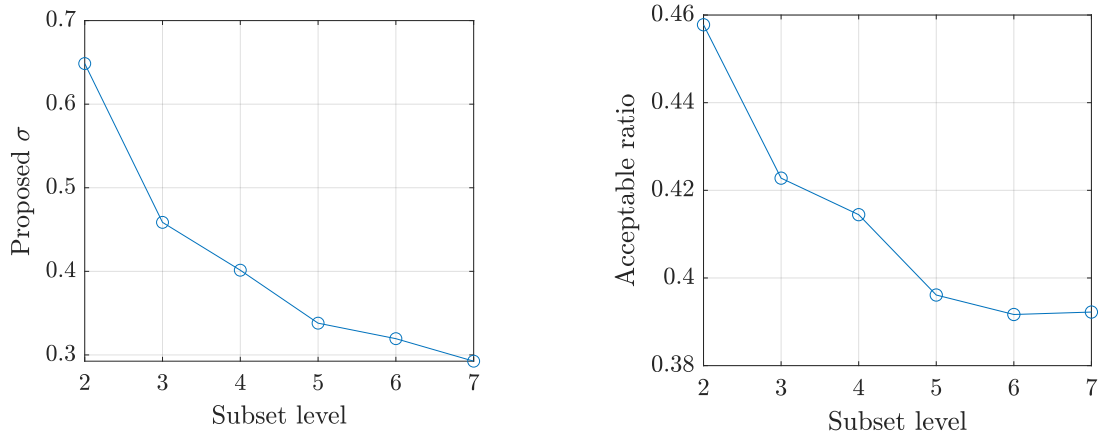
(a) Histograms of the incident metric in each subset level.



(b) Histograms of the duration that the aircraft below the low energy bound in each subset level.

Figure 6.27: Histograms of the model outputs in each subset level.

In the subset simulation, the σ of the proposal distribution and the acceptable ratio of generated samples in each subset level is shown in Figure 6.28(a) and Figure 6.28(b). By adjusting the σ using the adaptive conditional sampling method, the acceptable ratios in each subset are close to the optimal value 0.44, and do not decrease too much with the increase of the subset level.



(a) Proposed σ in each subset level using the adaptive conditional sampling.

(b) Acceptable ratios of generated samples in each subset level.

Figure 6.28: Proposed σ and acceptable ratios in each subset level.

6.4.5 Sensitivity Analysis of Contributing Factors

1. Sensitivity Analysis Based on MCS Samples

Since there is no direct physical meaning for the KL coefficients, the corresponding critical wind-related factors $V_{hws,TD}$, V_{gust} , and $\max \Delta V_{tws}$ in 100 feet, are calculated based on the wind series generated using the KL coefficients. Based on the MC samples, different kinds of sensitivity indices are calculated and shown in Figure 6.29 and Figure 6.30.

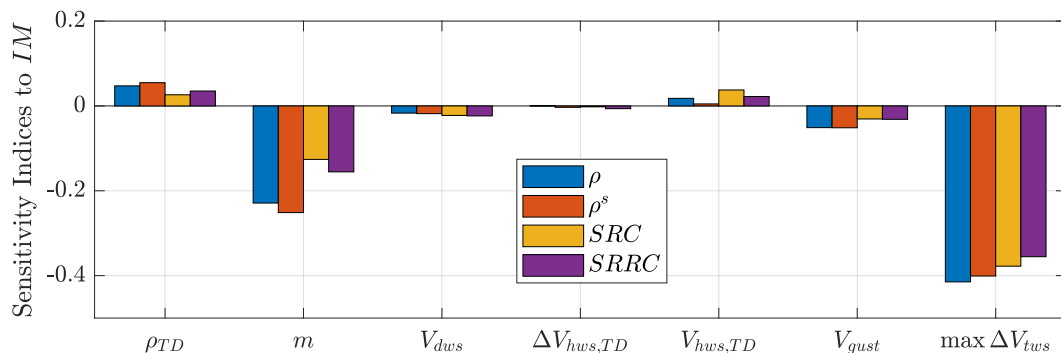


Figure 6.29: Correlation-based and regression-based sensitivity indices to IM.

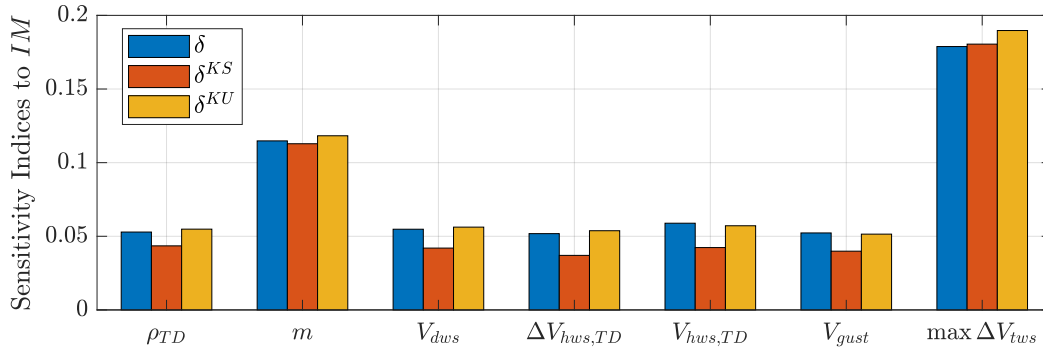


Figure 6.30: Moment-independent sensitivity indices to IM.

The results of correlation-based and regression-based sensitivity indices are similar. Both of them can indicate the positive and negative effects of the contributing factors to the incident metric. The moment-independent sensitivity indices also extract the same two most important factors. However, there is no significant difference detected among the remaining factors by using the moment-independent sensitivity indices. The highest effect factor is the $\max \Delta V_{tws}$ in 100 feet. The higher $\max \Delta V_{tws}$ is, the lower the incident metric is. The second important factor, landing mass m , has a negative effect on the incident metric. It can be explained based on Equation (6.58). A larger landing mass will lead to a small thrust-to-mass ratio, which weakens the ability to resist the wind shear. Compared to m and $\max \Delta V_{tws}$, another five factors ρ_{TD} , V_{dws} , V_{gust} , $V_{hws,TD}$, and $\Delta V_{hws,TD}$ have small influence. The ρ_{TD} has a positive effect on the incident metric. One possible reason could be that a larger ρ_{TD} leads to a smaller V_{tas} under the same approach target speed V_{app} . With the same wind condition, the V_{gs} is also smaller. Therefore, the aircraft flies slower and will encounter a smaller \dot{V}_{tws} under the same $\partial V_{tws}/\partial h$. According to Equation (6.58), V_{dws} has a negative effect on the energy bleed rate. Therefore, a larger V_{dws} will lead to a smaller IM . It is validated in Figure 6.29. Factors $V_{hws,TD}$, V_{gust} and $\Delta V_{hws,TD}$ will be used to correct the V_{app} based on Equation (6.3). Results shows that $V_{hws,TD}$ has a positive effect on the IM , while V_{gust} has a negative effect on the IM . The influence of $\Delta V_{hws,TD}$ is not significant compared to other factors.

2. Sensitivity Analysis Based on Subset Samples

To quantify the effect of the contributing factors on the failure probability, samples generated from the subset simulation are used. The \mathcal{L}_1 -modified reliability sensitivity indices are calculated using Equation (3.137). The sensitivity indices of contributing factors to each intermediate fail probability are shown in Figure 6.31. The horizontal axis denotes the incident metric. The vertical axis denotes the sensitivity indices of contributing factors to the incident probability. Each broken line with markers denotes one contributing factor. The horizontal value of the marker indicates the intermediate thresholds for one

subset level, while the vertical value denotes the sensitivity indices. The corresponding subset level increases along with the incident metric decreases from the right to the left in the horizontal axis of Figure 6.31.

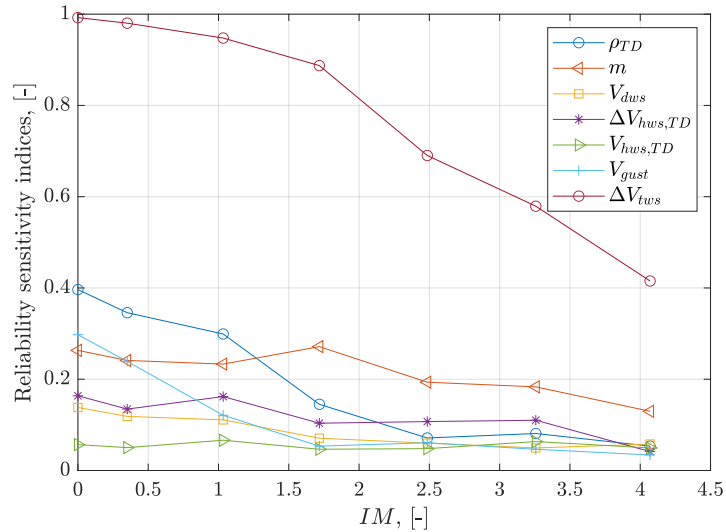


Figure 6.31: \mathcal{L}_1 -modified reliability sensitivity indices in each subset level.

Similar to the sensitivity indices for IM , the max ΔV_{twS} in 100 feet plays a significant role on the occurrence probability, following by the m and the ρ_{TD} . The samples plots and histograms of m and max ΔV_{twS} are shown in Figure 6.32 and Figure 6.33.

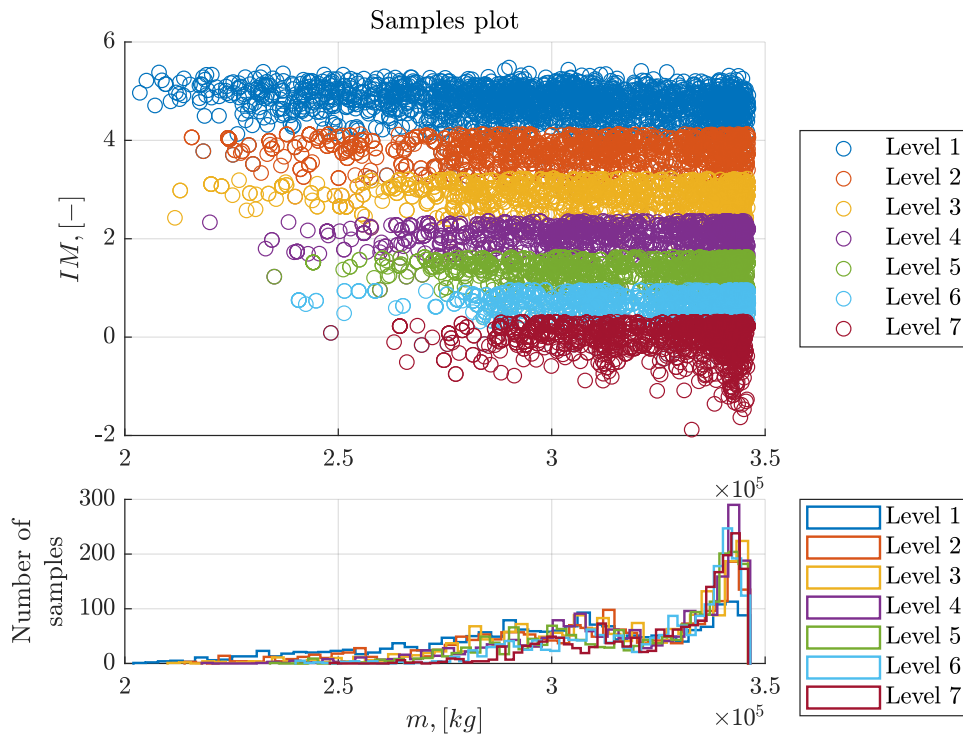


Figure 6.32: Samples plots and histograms of the landing mass in subset simulation.

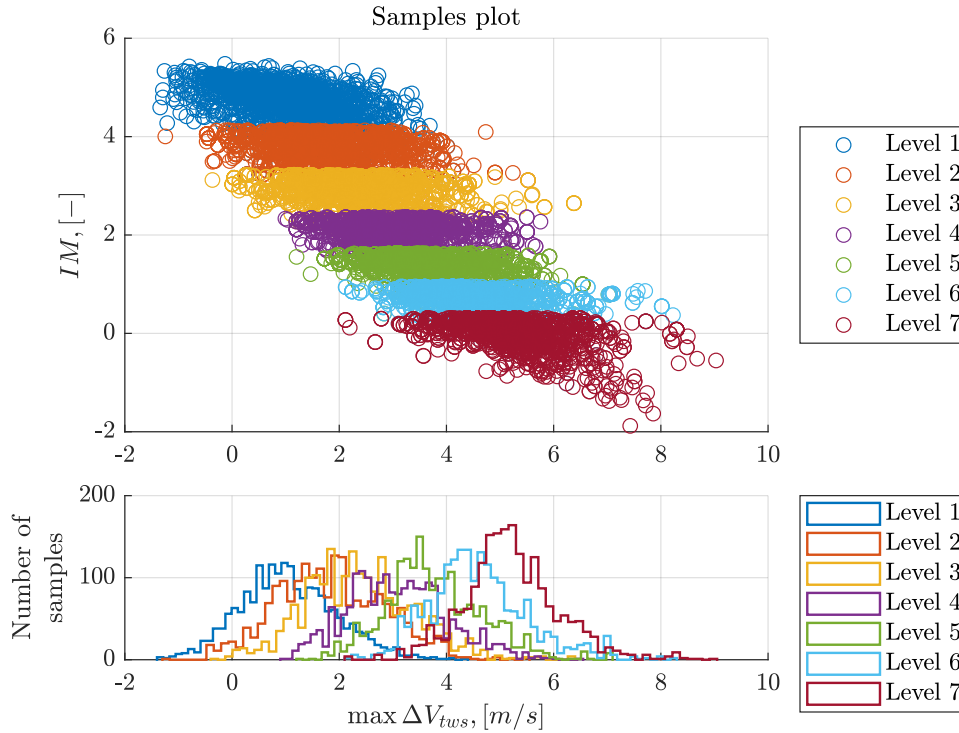
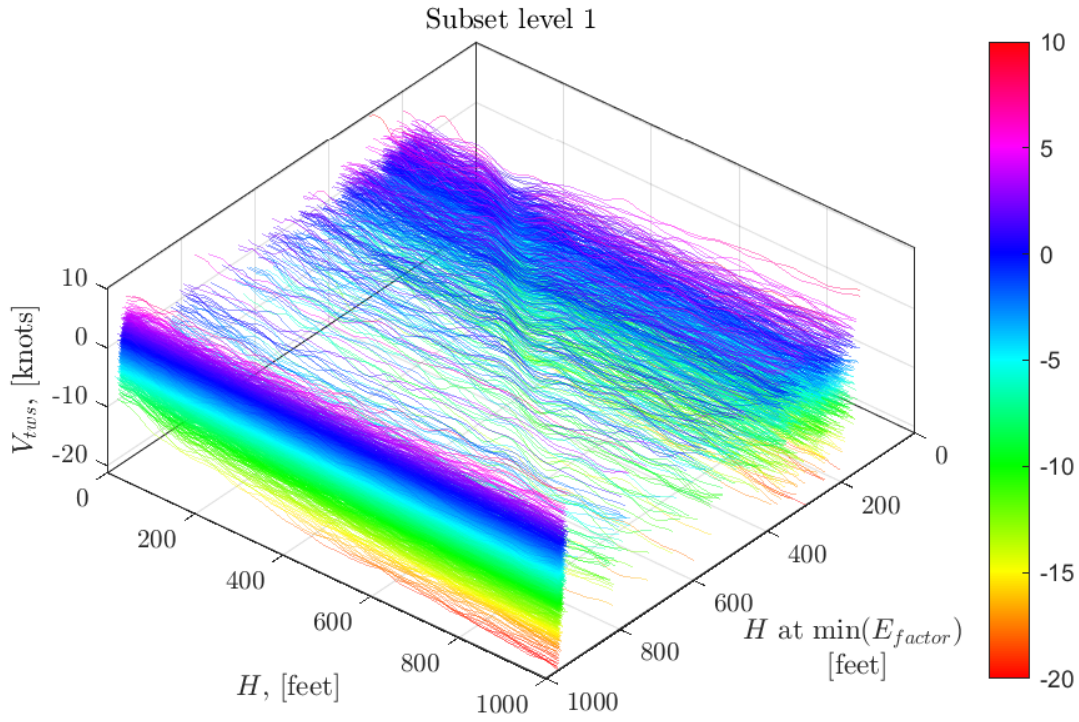


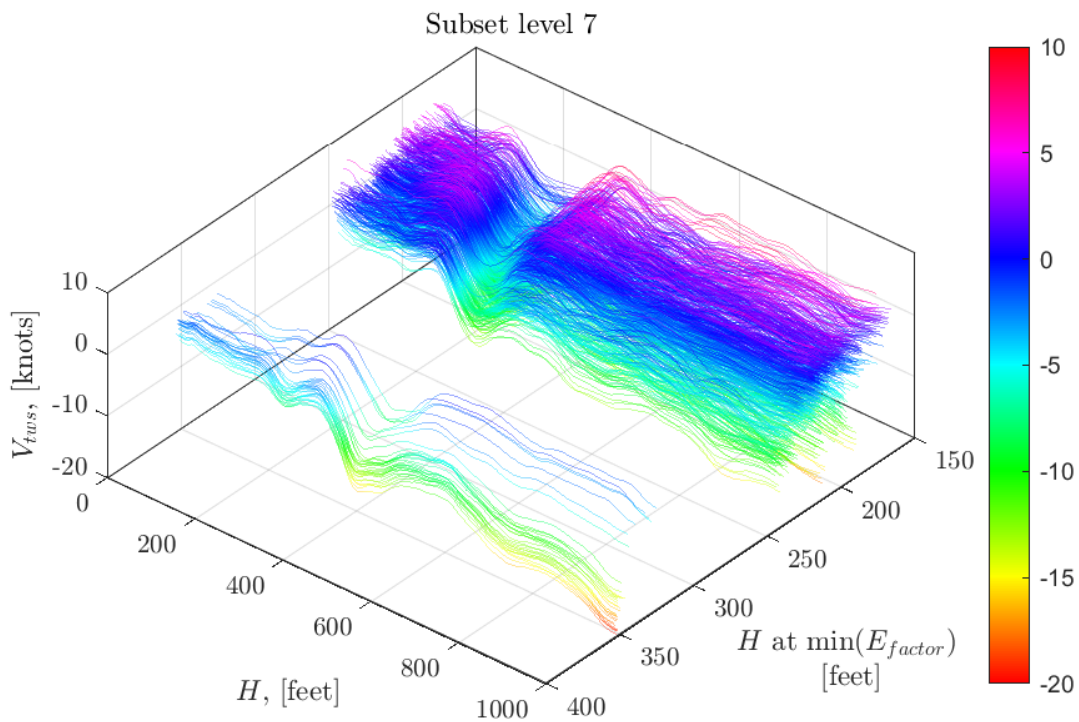
Figure 6.33: Samples plots and histograms of the maximum tailwind increment in 100 feet in subset simulation.

The $\max \Delta V_{tws}$ is extracted from 100 feet interval during the final approach. To obtain the overview of the tailwind series, the tailwind series along with the altitude is plotted using the samples from the subset simulation. The tailwind series in subset level 1 and level 7 (final subset level here) are shown in Figure 6.34. The tailwind series are plotted in the xz plane. The y axis denotes the altitude that the minimum of E_{factor} is located. In other words, for the wind series drawn in the slice of the xz plane with a low y , the minimum of E_{factor} occurs at a low altitude above ground. The corresponding E_{factor} series are shown in Figure 6.35. Similar to Figure 6.34, the E_{factor} is drawn in xz plane, and y axis denotes the altitude that the E_{factor} is minimum. The shadow xy plane indicates the energy factor lower bound that $E_{factor} = -1$.

Samples at subset level 1 are based on the direct Monte Carlo sampling. There are no obvious features for the tailwind series in Figure 6.34(a). Figure 6.35(a) shows that the altitude at the minimum of E_{factor} is spread over the entire final approach phase. Compared to these figures using subset level 1, what can be clearly seen in Figure 6.34(b) and Figure 6.35(b) is that the decrease of E_{factor} occurs along with the increase of V_{tws} . The steep increase of V_{tws} will lead to a drop in aircraft energy. In addition, Figure 6.35(b) shows most of E_{factor} series in the final subset level are at low altitude. This might be caused by the less height margin along with the decrease of altitude during the final approach phase.

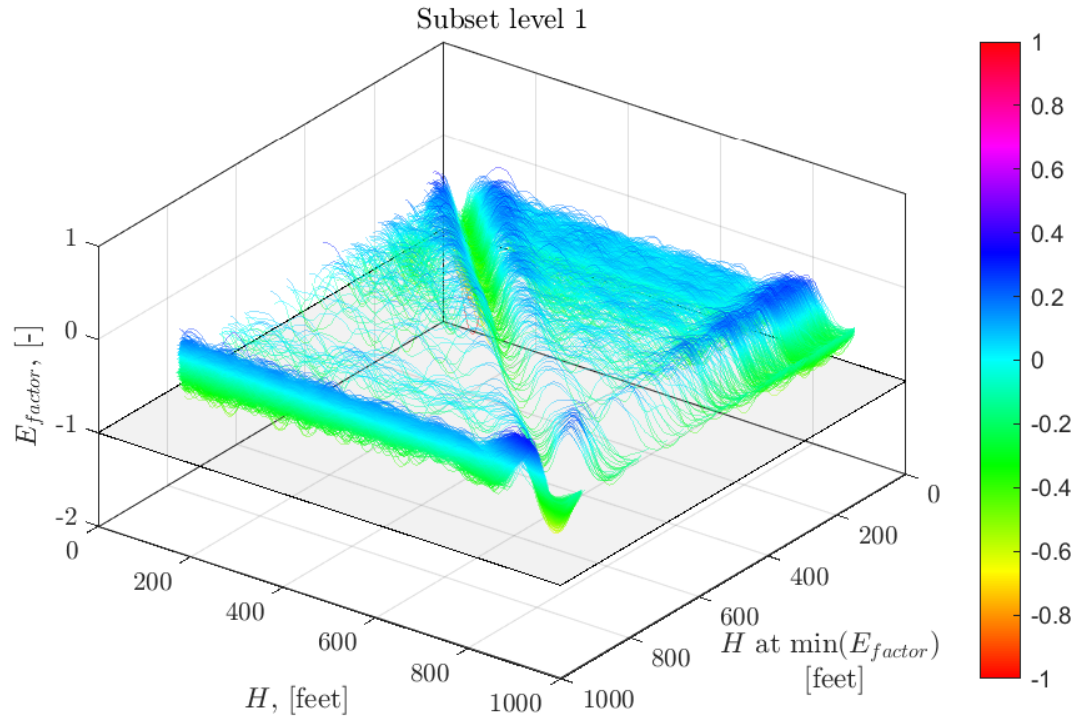
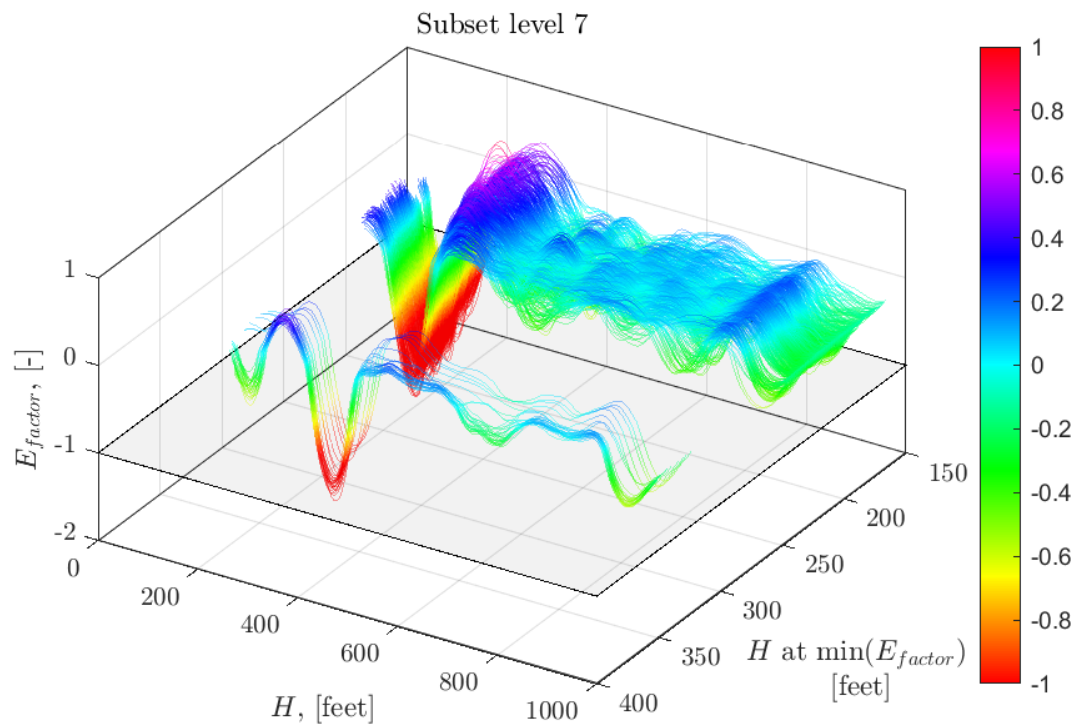


(a) V_{tws} in subset level 1.



(b) V_{tws} in subset level 7.

Figure 6.34: Tailwind series along with the altitude in different subset level.

(a) E_{factor} in subset level 1.(b) E_{factor} in subset level 7.**Figure 6.35:** Energy factor series along with the altitude in different subset level.

3. Comparison of incident probabilities for different situations

To assess quantitatively the direct effect of the contributing factors on the incident probabilities, one contributing factor can be fixed as a specific constant value when the subset simulation is executed. By taking several fixed values inside the interval of the contributing factor, the corresponding incident probabilities can be compared. The landing mass m is taken as an example. Four different values of m : 20t, 25t, 30t, and 34.6t are used in simulations, respectively. The incident probability conditioned at a specific m is estimated using the subset simulation. The flap setting is also taken into account. Two landing flap configurations ‘CONF FULL’ and ‘CONF 25’ are applied. The estimated results are shown in Figure 6.36. Besides, two baseline scenarios are also drawn. ‘F30’ and ‘F25’ stand for the 30-degree flap (‘CONF FULL’) and the 25-degree flap (‘CONF 25’). The ‘baseF30’ denotes the baseline scenario that the landing mass is distributed according to the flight data using the flap ‘CONF FULL’. Likewise, the ‘baseF25’ denotes the scenario using the ‘CONF 25’ for landing. The ‘m1’, ‘m2’, ‘m3’, and ‘m4’ denote the four different landing mass settings: 20t, 25t, 30t, and 35t, respectively.

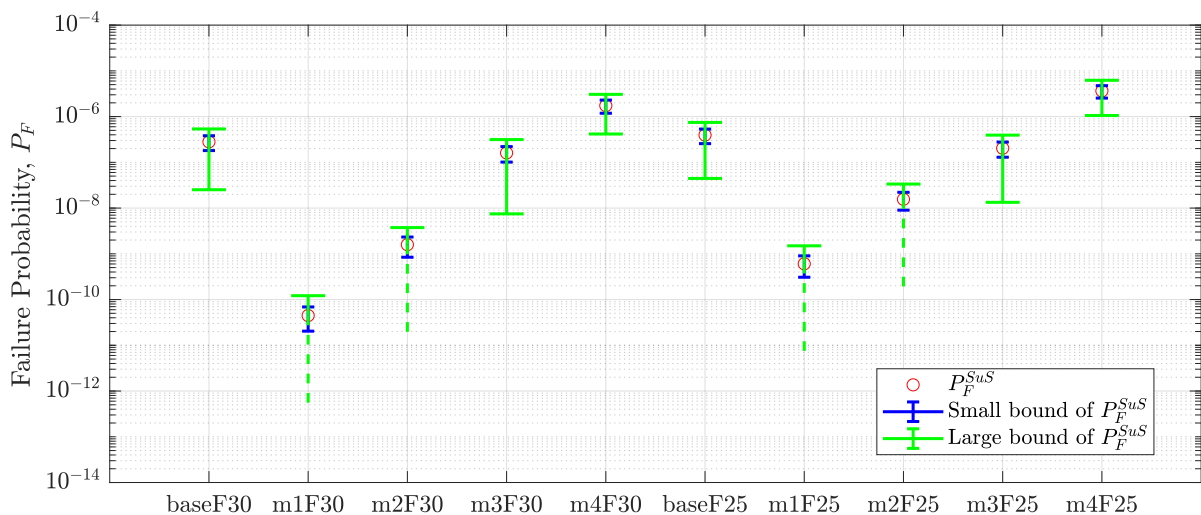


Figure 6.36: Comparison of incident probabilities for different mass and flap configuration.

In the cases of ‘m1F30’, ‘m2F30’, ‘m1F25’, and ‘m2F25’, the lower bound based on the estimated large covariance of the incident probability is less than 0, which are denoted by the dash green line instead of the solid green line. The obvious feature obtained in Figure 6.36 is that the larger landing mass will lead to a higher incident probability of low energy events. This conclusion is in compliance with the results in Figure 6.32. For the same mass value in subset simulation, the incident probability with two different configurations is almost in the same order of magnitude. Furthermore, the combination of ‘CONF 25’ and the largest landing mass will cause the highest incident probability.

The influence of the interval width $\Delta\tau$ in Equation (6.28) on the incident probability is also analyzed. The larger the $\Delta\tau$ is, the more critical the low energy situation is. The value of $\Delta\tau$ is set as 2, 4, 6, and 8 seconds, respectively.

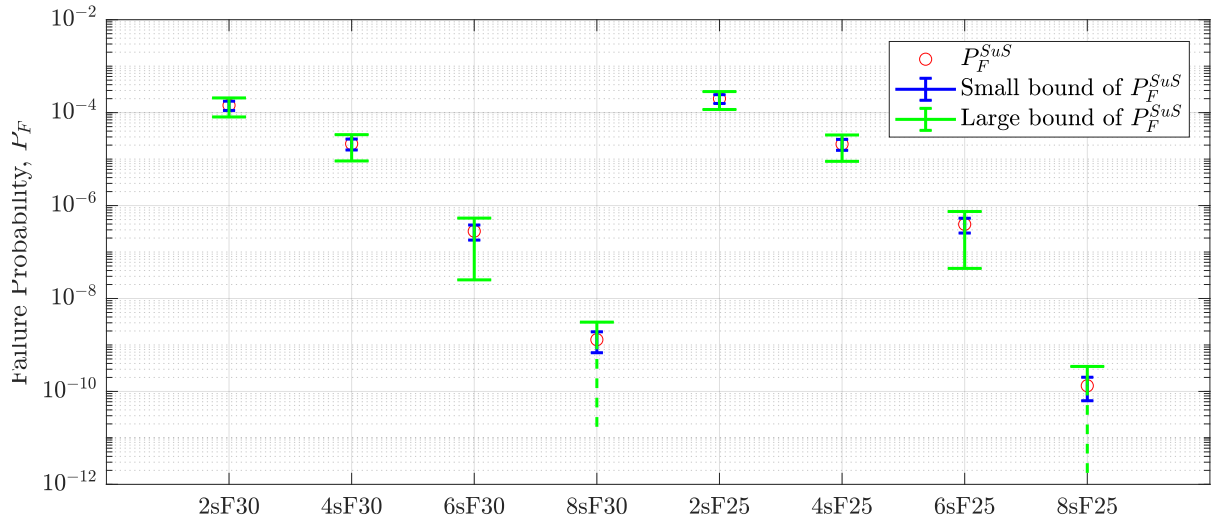


Figure 6.37: Comparison of incident probabilities for different duration setting.

As shown in Figure 6.37, the incident probability drops more steeply than the trend of the log curve, along with the linear increase of the $\Delta\tau$. For the scenarios of $\Delta\tau = 2s, 4s,$ or $6s$, the estimated incident probabilities are close for two different flap configurations. In contrast, the incident probability in the case of ‘8sF25’ is obviously less than the scenario of ‘8sF30’. As we discussed in Section 6.4.5, the sensitivity analysis on the incident probability shows that a sudden and step increase of the tailwind plays a significant role in the critical situation. ‘CONF 25’ can be better used in handling critical windy conditions. Thereby, the ‘CONF 25’ configuration causes a lower incident probability of low energy in the critical situation than the ‘CONF FULL’ configuration.

6.5 Summary

In this chapter, aircraft energy boundary is analyzed according to the stabilization gate and the stable approach criteria. By taking the stick shaker speed and the -1 dot glideslope path as the lower bound, the energy margin and time margin are proposed as safety metrics of low energy events during the final approach phase. The detection algorithm is applied to a large set of operational flight data to analyze the low energy event. No low energy events are detected, but the distributions of the energy margin and time margin are obtained.

By considering the implementation of the stable approach criteria in practice, a persistent violation of low energy bound is taken into account in a new metric, which is used for predictive analysis. A physical model for low energy events is built, and the corresponding

contributing factors are extracted. A KL-based stochastic wind model is integrated into the predictive analysis algorithm. By propagating the uncertainties in contributing factors obtained from the operational flight data of the B748F, the occurrence probability of low energy event for the interval $\Delta\tau = 6s$ is $2.81 \cdot 10^{-7}$ with the estimated covariance boundary $[1.8077 \cdot 10^{-7}, 3.8123 \cdot 10^{-7}]$. Along with the increase of $\Delta\tau$, the occurrence probability declines steeply.

During the post-processing, the wind-related contributing factors are also calculated based on the wind series constructed using the KL-based wind model and samples. Sensitivity analysis is implemented based on the MCS samples and the subset simulation samples. The step increase of tailwind plays a significant role in the incidents of low energy. Furthermore, a larger landing mass will lead to a high occurrence probability. A lower air density will also cause a high incident probability. There is no significant difference in occurrence probabilities of low energy for two flap configurations, except in the more critical situation that $\Delta\tau = 8s$. The aircraft has a better performance to deal with the critical windy condition using ‘CONF 25’ than ‘CONF FULL’.

Chapter 7

Conclusion and Outlook

7.1 Conclusion

The developed model-based Predictive Analysis (PA) framework provides a quantitative statement of incident probability by combining the operational flight data and the physical models using statistical methods. In this dissertation, advanced uncertainty quantification methods are applied to enhance the quality of the PA algorithms in contributing factors modeling, model calibration, and sensitivity analysis. In addition to that, the aircraft's low energy event during the final approach phase is modeled and analyzed using the enhanced PA algorithms.

At the beginning of the dissertation, the overview of handling the operational flight data is given. The flight data recording architecture shows how the flight data are transferred and recorded by the data acquisition unit (DAU) and buses. As one type of recorded data, Quick Access Recorder (QAR), is the main source of flight data in this thesis. The decoding algorithm allows us to convert the recorded QAR binary data stream into decimal data, which are the engineering values of flight parameters. Afterward, several flight data preprocessing steps are summarized, containing QAR file splitting by flight, departure/arrival airport and runway detection, time points detection, measurements calculation, and flight path reconstruction. First, QAR file splitting is used to split the flight data recorded in a QAR file into several individual flights, which is convenient for later data management and analysis. For each specific flight, the departure/arrival airport and runway are detected according to the world airport database, combined with the flight parameters, such as the aircraft's latitude, longitude, and heading. It is followed by the time points detection and the measurement calculation, which provide the critical data for flight safety analysis. The accuracy of these time points and measurements directly influences the flight safety analysis results. The applied Rauch–Tung–Striebel (RTS) smoother in flight path reconstruction improves the quality of flight variables and measurements. Flight Data Monitoring (FDM) events detection and statistical analysis

of critical measurements are the main approaches for flight data analysis. Beyond this, the physical-based incident model is integrated into the flight data analysis in the PA framework. An example of the runway overrun incident model explained how the corresponding incident metric and contributing factors are defined. All algorithms for flight data analysis mentioned above are developed at the Flight Safety working group at the Institute of Flight System Dynamics (FSD).

Uncertainty Quantification (UQ) methods provide theoretical foundations and mathematical tools in the incident probability estimation of the PA framework. The four main aspects of UQ methods involved in the PA framework are discussed in detail in Chapter 3, including input uncertainty quantification, rare event probability estimation, sensitivity analysis, and surrogate model. First, input uncertainty quantification methods are provided to quantify the uncertainties of contributing factors. The high dimensional dependence among contributing factors is captured using the vine copula approach. To generate samples of contributing factors statistically, sampling methods such as inverse CDF, rejection sampling, MCMC sampling, and copula sampling, are described. In the estimation of small probability in aviation, the subset simulation (SuS) method is applied. To obtain more accurate results of the SuS, several MCMC sampling methods used in SuS are discussed. The Component-wise Metropolis-Hastings MCMC algorithm (CWMH-MCMC) and the conditional sampling (called the limiting algorithm in [34, 102]) are used in previous studies. A more advanced adaptive conditional sampling method is provided to obtain an optimal acceptable ratio of generated samples, subsequently to obtain a better accurate result. The adaptive conditional sampling is also recommended in the reliability sensitivity analysis. Furthermore, the global sensitivity analysis is proposed as the post-processing step to enhance the quality of the PA results. Especially, the \mathcal{L}_1 -based reliability sensitivity index with intermediate thresholds is quite useful in practice to identify the key drivers for the incident. Chapter 3 also introduced the fundamentals of the polynomial chaos expansion (PCE) surrogate model, which underlies the method developed for the ‘Calibrate’ step of PA in Chapter 4.

In the ‘Calibrate’ step, the tuned distribution parameters of contributing factors are integrated into the cost function as a penalty term to obtain the overall minimum changes in all contributing factors and incident metric, instead of only matching the recorded and simulated incident metric distributions. By using the PCE surrogate model, a new calibration framework with a frozen sample strategy is proposed in this dissertation. The nonlinear dependence structure is also considered using the vine copula approach. This new calibration approach greatly improves efficiency because it is not required to rerun the Monte Carlo simulation (MCS) during each iteration of optimization. During the calibration procedure, the sensitivities of contributing factors to the incident metric are computed based on the PCE-based Sobol indices, which quantify the importance of each

contributing factor and is applied to reduce the dimension of contributing factors in the PCE surrogate model. Calibration results for a first-order model and runway overrun model validate the new calibration approach.

To consider the time series contributing factor in the PA framework, an efficient approach based on Karhunen–Loève (KL) expansion is proposed in this dissertation for producing time series with statistical characteristics matching measurements. This approach allows us to appropriately replicate the actual statistical characteristic of the atmospheric motion (wind, gusts, and turbulences) encountered in flights. Compared to the spectral representation (SR) methods using the power spectral density (PSD), the proposed approach is successfully verified against the well-established von Karman turbulence model. Beyond the wind series reconstruction in this dissertation, this approach is also implemented for sensor noise analysis and regeneration. Results show that the regenerated noise series using the KL expansion method match the PSD of the measurements very well. Furthermore, this statistical modeling approach for producing time series is implemented in simulations to generate noise signals instead of white noise.

High dimensional dependence among the KL coefficients is also analyzed by utilizing vine copula to have a better description of the statistical characteristics of the KL coefficients. Headwind and the horizontal wind shear ramps, encountered by the aircraft during the final approach, are reconstructed and regenerated using the KL expansion method with vine copula dependence.

To extend the use of the PA framework in the incident probability assessment, low energy situations during the final approach phase due to late destabilization are discussed in detail in Chapter 6. Energy margin is calculated based on the stable approach criteria. By considering the energy rate, the time margin (time to the low energy bound) is also calculated and applied to a larger set of operational flight data. In the implementation of the PA framework on the low energy event, a new incident metric is first proposed via integrating normalized energy margin, called E_{factor} , in a certain interval, in which the duration time of the low energy bound violation is taken into account. A physical-based low energy incident model is built according to the aircraft's motion, environment factors, and autothrottle and autopilot systems. The KL-based stochastic wind model is applied to generated wind series during the final approach trajectory and integrated into the SuS. Sensitivity analysis is implemented based on the samples from the MCS and the SuS. \mathcal{L}_1 -modified reliability sensitivity indices work well with the SuS results. As it is difficult to explain the physical meanings of KL coefficients directly, the wind-related factors are calculated from the wind series generated using the KL model and KL coefficients. These calculated wind-related factors are considered as contributing factors in the sensitivity analysis instead of KL coefficients. Developed sensitivity analysis algorithms are used to identify the key drivers of the incident metric uncertainty and incident probability. Results show the steep increase of tailwind plays a significant role in low energy incidents.

In addition, the larger landing mass and the lower air density will lead to a higher incident probability of low energy events. Furthermore, incident probabilities in different scenarios are also compared to analyze the effect of different factors.

7.2 Outlook

For future activities, the UQ methods can be further studied and integrated into the PA framework to obtain more reliable results efficiently. In this dissertation, the PCE surrogate model is only used in the ‘Calibrate’ step. However, the PCE surrogate model can be also used to improve the efficiency in uncertainties propagation from contributing factors to the incident metric for the incident probability estimation.

For the stochastic wind series modeling, the KL expansion method with vine copula dependence is used to construct time series of one variable, wind speed in a direction in this dissertation. This method can be also applied to construct multiple dependent time series together. The dependence among multiple time series could be described by modeling the dependence structure among their corresponding KL expansion coefficients using the vine copula approach. For example, tailwind, crosswind, and downwash could be constructed together using the operational flight data. It allows us to consider the dependence among wind speeds in three directions. Therefore, a more realistic three-dimensional atmospheric motion along the flight path could be replicated. The same approach could be also applied to model and replicate the noise series with the dependence of other series or measurements.

During the modeling of low energy events, the autothrottle and autopilot systems are applied. In practice, manual flying by pilots will lead to larger deviations at speed and altitude than auto flying. The inappropriate manual inputs or incorrect decision-making actions may cause a high risk of low energy during the final approach phase. However, modeling of pilot behavior in low energy events is outside the scope of this dissertation. In the future, the pilot modeling might be integrated into the low energy model. The pilot model proposed by researchers, such as the quasi-linear pilot model by McRuer in [163, 164], an optimal control model of human response in [165, 166], or a data-driven pilot behavior model in [167], might be useful to describe the pilot behavior combined with the operational flight data. Anyway, modeling the continuous behavior of pilots is still a challenging part, especially using the low-frequency operational data without enough excitations.

The PA framework has been applied in several incidents. Runway overrun and veer-off incidents, low energy incidents during the final approach, and Maximum Mach Operating exceedances incidents are developed by the Flight Safety working group at FSD. Other accident types, such as Controlled Flight into Terrain (CFIT), can be also assessed to extend the application of PA in safety assessment.

Appendix A

Nomenclature Principles and Coordinate Frames

The material presented here is mainly adapted from the lecture Flight System Dynamics 1 at the Institute of Flight System Dynamics, Technische Universität München [73].

A.1 Nomenclature Principles

Notations used to describe variables of the aircraft motion in this thesis refer to the following nomenclatures. Scalars, such as the components of a vector, are denoted by regular letters. The index may address three components (x, y, z) . For example, a_x denotes the acceleration in x -axis. The lower index of the scalar also describes its type, such as K (*Kinematic*), A (*Aerodynamic*), and W (*Wind*). For example, V_K and V_A denote the kinematic speed and the aerodynamic speed, respectively. The index can also be used to indicate the meaning of the scalar, like V_{app} , which is the approach target speed. Angles are predominantly denoted by Greek letters, i.e. the aerodynamic angle of attack α_A , and the kinematic flight path angle γ_K .

Generally, a vector is a collection of different scalar quantities, and it has two types: row and column vectors. All vectors are printed in bold type fonts. For the vectors in Euclidean space (three-dimensional orthonormal space), they are characterized by the arrow symbol. Matrices are used to map the input vector space and output vector space. Matrices are denoted by bold and capital letters, e.g. \mathbf{T} .

The vectors used to describe the variables of aircraft motion, such as position, velocity, are introduced in detail in the following examples. The vector of position is denoted by

$$(\vec{\mathbf{r}}^{GP})_B = \begin{bmatrix} x^{GP} \\ y^{GP} \\ z^{GP} \end{bmatrix}_B, \quad (\text{A.1})$$

which means the position of the point P relative to the point G in B frame. The subscript B is called the notation frame, which is used to specify the components of the vector. If the point P is ignored in the superscript, $(\vec{r}^G)_B$ is the position of the point G relative to the center of the earth O . Velocity is defined as the rate of change of a position vector over time, such as

$$(\vec{V}_K^{GP})_B^I = (\dot{\vec{r}}^{GP})_B^I = \left(\frac{d}{dt}\right)^I (\vec{r}^{GP})_B = \begin{bmatrix} u_K^{GP} \\ v_K^{GP} \\ w_K^{GP} \end{bmatrix}_B^I. \quad (\text{A.2})$$

Velocity always needs to be defined with respect to which frame (called reference frame) the rate of change of the vector is considered. In many cases, the reference frame is different from the notation frame, for example, the reference frame I and the notation frame B in Equation (A.2). The rate of change of velocity over time is called acceleration. Similar to velocity, the acceleration vector also needs to be defined with respect to which frame that the rate of change over time is considered. As shown in Equation (A.3), the $(\vec{a}_K^{GP})_B^{IE}$ is obtained by differentiating the velocity $(\vec{V}_K^{GP})_B^I$ with respect to the E -frame.

$$(\vec{a}_K^{GP})_B^{IE} = \left(\frac{d}{dt}\right)^E (\vec{V}_K^{GP})_B^I = \left(\frac{d}{dt}\right)^E \left[\left(\frac{d}{dt}\right)^I (\vec{r}^{GP})_B \right] = \begin{bmatrix} \dot{u}_K^{GP} \\ \dot{v}_K^{GP} \\ \dot{w}_K^{GP} \end{bmatrix}_B^{IE}. \quad (\text{A.3})$$

Angular rate is used to describe the rotation motion, denoted by the letter ω as below:

$$(\vec{\omega}_K^{IB})_B = \begin{bmatrix} \omega_{K,x}^{IB} \\ \omega_{K,y}^{IB} \\ \omega_{K,z}^{IB} \end{bmatrix}_B = \begin{bmatrix} p_K \\ q_K \\ r_K \end{bmatrix}_B, \quad (\text{A.4})$$

which is the rotation rate of B -frame with respect to I -frame. The subscript K inside the bracket indicates the type of angular rate. When the angular rate is the rotation of B -frame with respect to another frame specified in B -frame, the simplified notation p, q, r can be used to denote the three components of the angular rate. Similar to the acceleration, the angular acceleration can be obtained by the rate of change of an angular rate over time with respect to a frame as below:

$$(\dot{\vec{\omega}}_K^{IB})_B^O = \left(\frac{d}{dt}\right)^O (\vec{\omega}_K^{IB})_B = \begin{bmatrix} \dot{\omega}_{K,x}^{IB} \\ \dot{\omega}_{K,y}^{IB} \\ \dot{\omega}_{K,z}^{IB} \end{bmatrix}_B^O. \quad (\text{A.5})$$

Besides the variables of aircraft motion, the nomenclature of forces and moments are defined as below:

$$(\vec{F}_A^R)_B = \begin{bmatrix} X_A^R \\ Y_A^R \\ Z_A^R \end{bmatrix}_B, \quad (\vec{M}_A^R)_B = \begin{bmatrix} L_A^R \\ M_A^R \\ N_A^R \end{bmatrix}_B. \quad (\text{A.6})$$

The forces denoted by \vec{F} are referenced to the point R in the superscript, called the reference point. The lower index indicates the physical source of the force. Here, \vec{F}_A means the aerodynamic forces. The subscript outside the bracket refers to the notation frame, in which the coordinate components are specified. Similar definition principles are used to denote moments \vec{M} .

A.2 Coordinate Frames and Transformation

The used coordinate frames in this thesis are shown in Table A.1-A.7.

Table A.1: *Earth-Centered Inertia (ECI) Frame I*

Property	Description
Index	I
Role	Euclidean Frame (Inertial Axis System – Newton’s Law may be used)
Origin	Center of the Earth
x -axis	In the equatorial plane, in the direction of the vernal equinox
y -axis	In the equatorial plane, forming a right hand system with x and z -axis
z -axis	Rotation axis of the Earth

Table A.2: *Earth-Centered-Earth-Fixed Frame E*

Property	Description
Index	E
Role	Navigation frame for specification of positions
Origin	Center of the Earth
x -axis	In the equatorial plane in the direction of the Greenwich Meridian
y -axis	In the equatorial plane, forming a right hand system with x and z -axis
z -axis	Rotation axis of the Earth (collinear with the ECI ’s z -axis)

Table A.3: *North-East-Down (NED) Frame O*

Property	Description
Index	O
Role	Attitude-/Orientation Frame – used to specify the attitude of the plane
Origin	Reference point of the aircraft (R)
x -axis	Parallel to the local geoid surface, pointing to the geographic north pole
y -axis	Parallel to the local geoid surface, pointing east to form a right hand system with x -axis and z -axis
z -axis	Pointing downwards, perpendicular to the local geoid surface

Table A.4: *Body-fixed Frame B*

Property	Description
Index	B
Role	Notation Frame – used for forces, moments, ...
Origin	Reference point of the aircraft (R)
x -axis	Pointing towards the aircraft nose in the xz -symmetry plane
y -axis	Pointing to the right (starboard) wing to form an orthogonal right hand system
z -axis	Pointing downwards in the symmetry plane of the aircraft, perpendicular to the x - and z -axis

Table A.5: *Aerodynamic Frame A*

Property	Description
Index	A
Role	Notation Frame for aerodynamic flow
Origin	Reference point of the aircraft (R)
x -axis	Aligned with the aerodynamic velocity, pointing into the direction of the aerodynamic velocity
y -axis	Pointing to the right perpendicular to the x - and z -axis
z -axis	Pointing downwards in the symmetry plane of the aircraft, perpendicular to the xy -plane

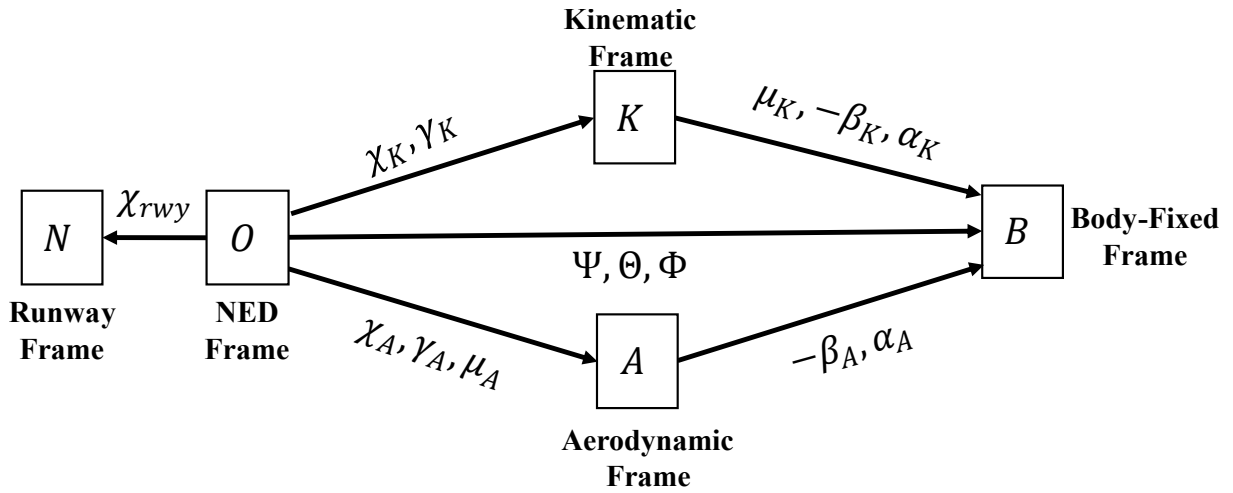
Table A.6: *Kinematic Frame K*

Property	Description
Index	K
Role	Flight-Path Axis Frame
Origin	Reference point of the aircraft (R)
x -axis	Aligned with the kinematic velocity, pointing into the direction of the kinematic velocity
y -axis	Pointing to the right perpendicular to the x - and z -axis
z -axis	Pointing downwards parallel to the projection of the local normal to the WGS-84 ellipsoid into a plane perpendicular to the x -axis (i.e. to the kinematic velocity)

Table A.7: *Navigation Frame N*

Property	Description
Index	N
Role	Navigation Frame, derived from NED-Frame to specify the position (i.e. the runway threshold)
Origin	Point on the earth surface
x -axis	Parallel to the local geoid surface, pointing to a direction that deviates with alignment angle χ_N from north direction (or pointing to the runway center line, also called runway frame)
y -axis	Parallel to the local geoid surface, to form a right hand system with the x - and z -axis
z -axis	Pointing downwards, perpendicular to the local geoid surface

The angles between different frames are summarized in Figure A.1. The transformation matrix between two frames can be constructed based on the angles.


Figure A.1: *Overview of rotation angles between frames*

Appendix B

Distribution Families

B.1 One-dimensional Probability Distributions

The following probability distribution families are used as distribution candidates in this thesis to quantify and model uncertainties of contributing factors.

Table B.1: *One-dimensional continuous distribution*

Distribution	Param	PDF	Ranges of x
Uniform $U(a, b)$	$a \in \mathbf{R}$ $b \in \mathbf{R}$	$f(x) = \frac{1}{b-a}$	$[a, b]$
Gaussian (Normal) $N(\mu, \sigma)$	$\mu \in \mathbf{R}$ $\sigma \geq 0$	$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$	$(-\infty, +\infty)$
Logistic $Log(\mu, \sigma)$	$\mu \in \mathbf{R}$ $\sigma \geq 0$	$f(x) = \frac{e^{-\frac{x-\mu}{\sigma}}}{\sigma \left(1 + e^{-\frac{x-\mu}{\sigma}}\right)^2}$	$(-\infty, +\infty)$
Lognormal $Ln(\mu, \sigma)$	$\mu \in \mathbf{R}$ $\sigma \geq 0$	$f(x) = \frac{1}{x\sigma\sqrt{2\pi}} e^{-\frac{(\ln x - \mu)^2}{2\sigma^2}}$	$x > 0$
Log-Logistic $Ll(\mu, \sigma)$	$\mu > 0$ $\sigma > 0$	$f(x) = \frac{e^{-\frac{\ln x - \mu}{\sigma}}}{x\sigma \left(1 + e^{-\frac{\ln x - \mu}{\sigma}}\right)^2}$	$(0, +\infty)$
Weibull $Wb(a, b)$	$a > 0$ $b > 0$	$f(x) = \frac{b}{a} \left(\frac{x}{a}\right)^{b-1} e^{-(x/a)^b}$	$x \geq 0$
Exponential $Exp(\mu)$	$\mu > 0$	$f(x) = \frac{1}{\mu} e^{-\frac{x}{\mu}}$	$[0, +\infty)$
Gamma $Ga(a, b)$	$a > 0$ $b > 0$	$f(x) = \frac{x^{a-1} e^{-\frac{x}{b}}}{b^a \Gamma(a)}$	$(0, +\infty)$
Beta $Be(a, b)$	$a > 0$ $b > 0$	$f(x) = \frac{1}{B(a, b)} x^{a-1} (1-x)^{b-1}$, where $B(a, b) = \frac{\Gamma(a)\Gamma(b)}{\Gamma(a+b)}$	$(0, 1)$

Generalized Extreme Value $GEV(k, \sigma, \mu)$	$k \in \mathbf{R}$ $\sigma \geq 0$ $\mu \in \mathbf{R}$	$f(x) = \frac{1}{\sigma} t(x)^{k+1} e^{-t(x)}$, where $t(x) = \left(1 + k \frac{x-\mu}{\sigma}\right)^{-\frac{1}{k}}$, $k \neq 0$ $t(x) = e^{-\frac{(x-\mu)}{\sigma}}$, $k = 0$	$(-\infty, -\frac{\sigma}{k} + \mu)$, $k > 0$ $(-\frac{\sigma}{k} + \mu, +\infty)$, $k < 0$ $(-\infty, +\infty)$, $k = 0$
t Location-Scale $tLoc(\mu, \sigma, \nu)$	$\mu \in \mathbf{R}$ $\sigma > 0$ $\nu > 0$	$f(x) = \frac{\Gamma(\frac{\nu+1}{2})}{\sigma \sqrt{\nu\pi} \Gamma(\frac{\nu}{2})} \cdot \left(\frac{\nu + (\frac{x-\mu}{\sigma})^2}{\nu}\right)^{-\frac{\nu+1}{2}}$	$(-\infty, +\infty)$
Student's t $Stu(\nu)$	$\nu > 0$	$f(x) = \frac{\Gamma(\frac{\nu+1}{2})}{\sqrt{\nu\pi} \Gamma(\frac{\nu}{2})} \cdot \left(\frac{\nu+x^2}{\nu}\right)^{-\frac{\nu+1}{2}}$	$(-\infty, +\infty)$
Gumbel $Gb(a, b)$	$a \in \mathbf{R}$ $b > 0$	$f(x) = \frac{1}{b} e^{-\frac{x-a}{b}} - e^{-\frac{x-a}{b}}$	$(-\infty, +\infty)$
Rayleigh $Ra(b)$	$b > 0$	$f(x) = \frac{x}{b^2} e^{-x^2/(2b^2)}$	$[0, +\infty)$
Nakagami $Na(\mu, \omega)$	$\mu > 0$ $\omega > 0$	$f(x) = 2 \left(\frac{\mu}{\omega}\right)^\mu \frac{1}{\Gamma(\mu)} x^{(2\mu-1)} e^{-\frac{\mu x^2}{\omega}}$	$(0, +\infty)$
Kernal KS	-	$f_h(x) = \frac{1}{nh} \sum_{i=1}^n K\left(\frac{x-x_i}{h}\right)$	$(-\infty, +\infty)$

B.2 Bivariate Copula Families

The following table shows bivariate copula families with their corresponding parameter ranges. ‘No.’ denotes the corresponding label of the specific copula family. The ‘par’ and ‘par2’ denote two parameters used in the copula distribution functions.

Table B.2: *Bivariate copula families available in VineCopula R package, source: [87]*

Copula family	No.	par	par2
Gaussian	1	$(-1, 1)$	-
Student t	2	$(-1, 1)$	$(2, +\infty)$
(Survival) Clayton	3, 13	$(0, +\infty)$	-
Rotated Clayton (90° and 270°)	23, 33	$(-\infty, 0)$	-
(Survival) Gumbel	4, 14	$[1, +\infty)$	-
Rotated Gumbel (90° and 270°)	24, 34	$(-\infty, -1]$	-
Frank	5	$(-\infty, 0) \cup (0, +\infty)$	-

(Survival) Joe	6, 16	$(1, +\infty)$	-
Rotated Joe (90° and 270°)	26, 36	$(-\infty, -1)$	-
(Survival) Clayton-Gumbel (BB1)	7, 17	$(0, +\infty)$	$[1, +\infty)$
Rotated Clayton-Gumbel (90° and 270°)	27, 37	$(-\infty, 0)$	$(-\infty, -1]$
(Survival) Joe-Gumbel (BB6)	8, 18	$[1, +\infty)$	$[1, +\infty)$
Rotated Joe-Gumbel (90° and 270°)	28, 38	$(-\infty, -1]$	$(-\infty, -1]$
(Survival) Joe-Clayton (BB7)	9, 19	$[1, +\infty)$	$(0, +\infty)$
Rotated Joe-Clayton (90° and 270°)	29, 39	$(-\infty, -1]$	$(-\infty, 0)$
(Survival) Joe-Frank (BB8)	10, 20	$[1, +\infty)$	$(0, 1]$
Rotated Joe-Frank (90° and 270°)	30, 40	$(-\infty, -1]$	$[-1, 0)$
(Survival) Tawn type 1	104, 114	$[1, +\infty)$	$[0, 1]$
Rotated Tawn type 1 (90° and 270°)	124, 134	$(-\infty, -1]$	$[0, 1]$
(Survival) Tawn type 2	204, 214	$[1, +\infty)$	$[0, 1]$
Rotated Tawn type 2 (90° and 270°)	224, 234	$(-\infty, -1]$	$[0, 1]$

As shown in Table B.2, several bivariate copula families are provided in the VineCopula R package. It includes the elliptical copulas (Gaussian and Student-t), Archimedean copulas (Clayton, Gumbel, Frank, Joe, BB1, BB6, BB7, and BB8), and extreme-value copulas. The corresponding bivariate copula distribution functions are shown in Table B.3. To extend the range of dependence, the rotated copulas are also included in Table B.2. The rotated copula density is obtained from the counterclockwise rotation of the copula density by 90°, 180°, or 270° as below:

$$\begin{aligned}
 90^\circ : c_{90}(u_1, u_2) &= c(1 - u_2, u_1) \\
 180^\circ : c_{180}(u_1, u_2) &= c(1 - u_1, 1 - u_2) \\
 270^\circ : c_{270}(u_1, u_2) &= c(u_2, 1 - u_1).
 \end{aligned}
 \tag{B.1}$$

In addition, the survival copulas mentioned in Table B.2 are defined as $\bar{C}(u_1, u_2) = Pr(U_1 > u_1, U_2 > u_2)$. Compared to the definition of cumulative distribution function that $C(u_1, u_2) = Pr(U_1 < u_1, U_2 < u_2)$, the survival copula density is calculated as below and it is same as the copula rotated by 180°.

$$\bar{c}(u_1, u_2) = c(1 - u_1, 1 - u_2) = c_{180}(u_1, u_2).
 \tag{B.2}$$

Table B.3: *Bivariate copula distribution function, source: [27, 28]. (a) $\Phi_{2;\rho}$ denotes two-dimensional Gaussian distribution function with zero mean and correlation ρ . Φ denotes univariate standard Gaussian distribution function. (b) $T_{2;\nu,\rho}$ denotes two-dimensional student t distribution function with ν degrees of freedom and correlation ρ . T_ν denotes univariate student t distribution function with ν degrees of freedom.*

Copula	$C(u_1, u_2)$	parameter range
Independent	uv	-
Gaussian	$\Phi_{2;\rho}(\Phi^{-1}(u_1), \Phi^{-1}(u_2))$ (a)	$\rho \in (-1, 1)$
Student t	$T_{2;\nu,\rho}(T_\nu^{-1}(u_1), T_\nu^{-1}(u_2))$ (b)	$\rho \in (-1, 1), \nu > 1$
Clayton	$(u_1^{-\delta} + u_2^{-\delta} - 1)^{-\frac{1}{\delta}}$	$\delta \in (0, +\infty)$
Gumbel	$\exp(-((-\ln u_1)^\delta + (-\ln u_2)^\delta)^{\frac{1}{\delta}})$	$\delta \in [1, +\infty)$
Frank	$-\frac{1}{\delta} \ln \left(\frac{(1-e^{-\delta}) - (1-e^{-\delta u_1})(1-e^{-\delta u_2})}{1-e^{-\delta}} \right)$	$\delta \in (-\infty, 0) \cup (0, +\infty)$
Joe	$1 - [(1 - u_1)^\delta + (1 - u_2)^\delta - (1 - u_1)^\delta(1 - u_2)^\delta]^{\frac{1}{\delta}}$	$\delta \in (1, +\infty)$
BB1	$\eta(\eta^{-1}(u_1) + \eta^{-1}(u_2))$ where $\eta(s) = (1 + s^{\frac{1}{\delta}})^{-\frac{1}{\theta}}$, and $\eta^{-1}(t) = (t^{-\theta} - 1)^\delta$	$\theta \in (0, +\infty)$ $\delta \in [1, \infty)$
BB6	$\eta(\eta^{-1}(u_1) + \eta^{-1}(u_2))$ where $\eta(s) = 1 - [1 - \exp\{-s^{\frac{1}{\delta}}\}]^{\frac{1}{\theta}}$, and $\eta^{-1}(t) = (-\ln[1 - (1 - t)^\theta])^\delta$	$\theta \in [1, +\infty)$ $\delta \in [1, +\infty)$
BB7	$\eta(\eta^{-1}(u_1) + \eta^{-1}(u_2))$ where $\eta(s) = 1 - [1 - (1 + s)^{-\frac{1}{\delta}}]^{\frac{1}{\theta}}$, and $\eta^{-1}(t) = (1 - (1 - t)^\theta)^{-\delta} - 1$	$\theta \in [1, +\infty)$ $\delta \in (0, +\infty)$
BB8	$\frac{1}{\delta}[1 - (1 - \frac{xy}{z})^{\frac{1}{\theta}}]$ where $x = 1 - (1 - \delta u_1)^\theta$, $y = 1 - (1 - \delta u_2)^\theta$, and $z = 1 - (1 - \delta)^\theta$	$\theta \in [1, +\infty)$ $\delta \in (0, 1]$

Tawn type 1	$\exp\{[\ln(u_1) + \ln(u_2)]A\left(\frac{\ln(u_2)}{\ln(u_1 u_2)}\right)\} \text{ where}$ $A(t) = (1 - \psi_1)(1 - t) + (1 - \psi_2)t$ $+[\psi_1(1 - t)^\theta + (\psi_2 t)^\theta]^{\frac{1}{\theta}}$	$\theta \in [1, +\infty)$ $\psi_1 = 1$ $\psi_2 \in [0, 1]$
Tawn type 2	$\exp\{[\ln(u_1) + \ln(u_2)]A\left(\frac{\ln(u_2)}{\ln(u_1 u_2)}\right)\} \text{ where}$ $A(t) = (1 - \psi_1)(1 - t) + (1 - \psi_2)t$ $+[\psi_1(1 - t)^\theta + (\psi_2 t)^\theta]^{\frac{1}{\theta}}$	$\theta \in [1, +\infty)$ $\psi_1 \in [0, 1]$ $\psi_2 = 1$

Appendix C

Efficiency of MCMC estimator

The efficiency of Monte Carlo simulation (MCS) and Markov chain Monte Carlo (MCMC) is derived via an example of estimating a quantity of interest. Without loss of generality, the input \mathbf{X} follows the known distribution $f_{\mathbf{X}}$, and $\mathbf{X} \in \mathbb{R}^n$. The function of random variables is denoted by $Y = g(\mathbf{X})$. $Y \in \mathbb{R}$ is the model output. The mean of the output is taken as the quantity of interest in this example as below:

$$Q = E[g(\mathbf{X})]. \quad (\text{C.1})$$

MCS estimator

Then estimation procedure of Q using the MCS is as follows:

1. Generate i.i.d. samples $\{\mathbf{x}_i, i = 1, \dots, N\}$ from $f_{\mathbf{X}}$ using direct sampling method like the inverse CDF transformation;
2. Evaluate the model $\{y_i = g(\mathbf{x}_i), i = 1, \dots, N\}$;
3. Estimate Q via:

$$\hat{Q} = \frac{1}{N} \sum_{i=1}^N y_i. \quad (\text{C.2})$$

The output realization y_i follows f_Y and is also i.i.d since \mathbf{x}_i is i.i.d.. \hat{Q} is a function of random outcomes of $Y = g(\mathbf{X})$. Therefore, the \hat{Q} is a random variable. The mean of the estimator \hat{Q} is derived as below:

$$E[\hat{Q}] = \frac{1}{N} \sum_{i=1}^N E[y_i] = \frac{NE[Y]}{N} = E[Y] = Q. \quad (\text{C.3})$$

The MCS estimator of Q is unbiased. The variance of the \hat{Q} is computed as follow:

$$Var[\hat{Q}] = \frac{1}{N^2} \sum_{i=1}^N Var[y_i] = \frac{NVar[Y]}{N^2} = \frac{Var[Y]}{N}. \quad (\text{C.4})$$

The relative error of the estimator, called coefficient of variance (c.o.v.), is obtained via:

$$\delta_{\hat{Q}} = \frac{\sqrt{\text{Var}[\hat{Q}]}}{E[\hat{Q}]} = \sqrt{\frac{\text{Var}[Y]}{NQ}}, \quad (\text{C.5})$$

when Q is not zero mean. The c.o.v. will decrease along with the increase of number of sample N .

MCMC estimator

In the case that the CDF of $f_{\mathbf{X}}$ is not analytically invertible, and the distribution is only known point-wise, MCMC algorithm can be used to generated the samples $\{x_i, i = 1, \dots, N\}$, which are states of a Markov Chain with stationary distribution $f_{\mathbf{X}}$. The output $\{y_i = g(\mathbf{x}_i), i = 1, \dots, N\}$ are states of a random process with the distribution f_Y . As the current state in Markov chain depends on the previous state, the samples $\{x_i, i = 1, \dots, N\}$ is not i.i.d.. Therefore, the y_i is also not i.i.d.. The correlation of two different states y_i and y_{i+k} is denoted by the autocorrelation coefficients $\rho_Y(k)$, which is computed as below:

$$\rho_Y(k) = \frac{\text{Cov}[y_i, y_{i+k}]}{\text{Var}(y_i)}, \quad (\text{C.6})$$

where $\text{Cov}[y_i, y_{i+k}]$ is the covariance between two states. k , called lag, counts the number of states between two states in a chain shown in Figure C.1. The $\rho_Y(k)$ between two arbitrary states only depends on the lag k . For a chain with N states, there are $N - k$ pairs of states with the same correlation $\rho_Y(k)$.

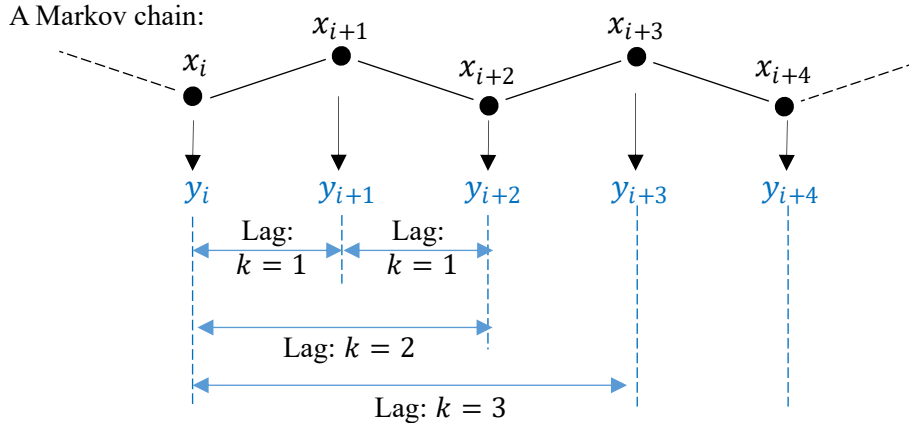


Figure C.1: States in a Markov chain.

The MCMC estimator of \hat{Q} is as below:

$$\hat{Q} = \frac{1}{N} \sum_{i=1}^N y_i. \quad (\text{C.7})$$

The mean of the MCMC estimator is derived:

$$E[\hat{Q}] = \frac{1}{N} \sum_{i=1}^N E[y_i] = \frac{NE[Y]}{N} = E[Y] = Q. \quad (\text{C.8})$$

Therefore, the variance of the MCMC estimator is derived:

$$\begin{aligned}
 Var[\hat{Q}] &= Var\left[\frac{1}{N}\sum_{i=1}^N y_i\right] = \frac{1}{N^2}Var\left[\sum_{i=1}^N y_i\right] \\
 &= \frac{1}{N^2}\left(\sum_{i=1}^N Var[y_i] + \sum_{i=1}^N \sum_{j=1, j \neq i}^N Cov[y_i, y_j]\right) \\
 &= \frac{1}{N^2}\left(\sum_{i=1}^N Var[y_i] + \sum_{i=1}^N \sum_{j=1, j \neq i}^N \rho_Y(y_i, y_j)\sqrt{Var[y_i]}\sqrt{Var[y_j]}\right) \quad (C.9) \\
 &= \frac{1}{N^2}\left(NVar[Y] + 2\sum_{k=1}^{N-1}(N-k)Var[Y]\rho_Y(k)\right) \\
 &= \frac{Var[Y]}{N}\left(1 + 2\sum_{k=1}^{N-1}\left(1 - \frac{k}{N}\right)\rho_Y(k)\right) = \frac{Var[Y]}{N}(1 + \lambda),
 \end{aligned}$$

where

$$\lambda = 2\sum_{k=1}^{N-1}\left(1 - \frac{k}{N}\right)\rho_Y(k). \quad (C.10)$$

The relative error of the MCMC estimator is calculated as below:

$$\delta_{\hat{Q}}^{MCMC} = \frac{\sqrt{Var[\hat{Q}]}}{E[\hat{Q}]} = \sqrt{\frac{Var[Y]}{NQ}}(1 + \lambda), \quad (C.11)$$

Due to the autocorrelation of the Markov chain samples, denoted by the positive λ , the relative error of MCMC estimator is larger than the MCS estimator with the same number of samples. The efficiency of the MCMC estimator, denoted by eff , can be measured via:

$$eff = \frac{1}{1 + \lambda} = \frac{1}{1 + 2\sum_{k=1}^{N-1}\left(1 - \frac{k}{N}\right)\rho_Y(k)}. \quad (C.12)$$

As the $\rho_Y(k)$ decrease, the λ will decrease and eff will increase. Therefore, the efficiency of MCMC estimator will increase with the decrease of the autocorrelation coefficients.

Bibliography

- [1] International Civil Aviation Organization (ICAO), *Operation of aircraft: Annex 6 to the Convention on International Civil Aviation: International standards and recommended practices*. International Civil Aviation Organization, Montréal, Québec, Canada, 9th ed., 2010. ISBN:978-92-9231-536-8.
- [2] International Civil Aviation Organization (ICAO), *Annex 19 Safety Management*. International Civil Aviation Organization, Montréal, QC, Canada, 1st ed., 2013.
- [3] International Civil Aviation Organization (ICAO), *Safety Management Manual: Doc 9859*. International Civil Aviation Organization, Montréal, Québec, Canada, 3rd ed., 2013.
- [4] European Commission (EC), *Flightpath 2050 Europe's Vision for Aviation*. Luxembourg: Office for Official Publications of the European Communities, 2011. ISBN:978-92-79-19724-6.
- [5] J. Raps, "Sicherheit muss jeden tag neu produziert werden: Risk management bei lufthansa," September 24, 2009.
- [6] International Air Transport Association (IATA), *IATA Safety Report 2020*. International Air Transport Association, Montréal, Québec, Canada, 57th ed., April 2021. ISBN:978-92-9264-412-3.
- [7] Airplane State Awareness Joint Safety Analysis Team (JSAT), *Airplane State Awareness: Final Report Analysis and Results*, June 17, 2014.
- [8] C. M. Belcastro, J. V. Foster, G. H. Shah, I. M. Gregory, D. E. Cox, D. A. Crider, L. Groff, R. L. Newman, and D. H. Klyde, "Aircraft Loss of Control Problem Analysis and Research Toward a Holistic Solution," *Journal of Guidance, Control, and Dynamics*, vol. 40, no. 4, pp. 733–775, 2017. DOI: 10.2514/1.G002815.
- [9] L. Drees, *Predictive Analysis: Quantifying Operational Airline Risks*. PhD thesis, Technische Universität München, München, 2016.

- [10] European Operators Flight Data Monitoring (EOFDM) WGB, *Guidance for the Implementation of FDM Precursors*. European Aviation Safety Agency, Cologne, Germany, revision 2 ed., 2019.
- [11] International Civil Aviation Organization (ICAO), *Manual on Flight Data Analysis Programmes (FDAP): Doc10000*. International Civil Aviation Organization, Montréal, Québec, Canada, 1st ed., 2014.
- [12] Civil Aviation Authority (CAA), *CAP 739 Flight data monitoring*. Civil Aviation Authority, Crawley, England, 2nd ed., 2003. ISBN:978-0-11792-840-4.
- [13] Federal Aviation Administration (FAA), *Advisory Circular No: 120-82 Flight Operational Quality Assurance*. U.S. Department of Transportation, Federal Aviation Administration, Washington, D.C., December 2004.
- [14] Teledyne Controls, *AirFase Flight Data Monitoring Tool*, pp. 6–9. Teledyne Product Brochure, 2019. Available online: <https://www.skybrary.aero/bookshelf/books/2825.pdf> (accessed on 05 September 2020).
- [15] Safran S.A., *Cassiopee Analysis Ground Station (AGS) Data Management Software*, pp. 1–3. Safran Product Brochure, 2019. Available online: <https://www.safran-electronics-defense.com> (accessed on 05 September 2020).
- [16] A. Bunn, D. Mastad, *et al.*, “Aerobytes FDM.” Available online: <https://www.aerobytes.co.uk/> (accessed on 05 September 2020).
- [17] Austin Digital, Inc., “Austin Digital’s Event Measurement System.” Available online: <https://www.swiss49.com/docs/EMS%20Version%204%20Overview.pdf/> (accessed on 17 September 2021).
- [18] European Aviation Safety Agency, “Data-driven research addressing aviation safety intelligence,” 2016. Available online: <https://innaxis.aero/full-blog/safeclouds/> (accessed on 05 September 2020).
- [19] C. Halford and M. Harper, “Asias: Aviation safety information analysis and sharing,” in *2008 IEEE/AIAA 27th Digital Avionics Systems Conference*, 2008.
- [20] D. Jesse and R. Quevedo, “flight data connect: World leading flight data analysis service.,” 2017.
- [21] J. Schimmel, “SAGA : Safety analysis in general aviation leveraging digital flight data towards the development of a flight performance database tool to improve general aviation safety,” in *7th International Conference on Research in Air Transportation (ICRAT), June 20-24, 2016 - Drexel University, Philadelphia, PA*, 2016.

- [22] J. L. Ackley, T. G. Puranik, and D. Mavris, “A supervised learning approach for safety event precursor identification in commercial aviation,” in *AIAA Aviation 2020 Forum*, p. 2880, 2020.
- [23] H. Haverdings and P. W. Chan, “Quick access recorder data analysis software for windshear and turbulence studies,” *Journal of Aircraft*, vol. 47, no. 4, pp. 1443–1447, 2010. DOI: 10.2514/1.46954.
- [24] L. Drees, J. Sembiring, L. Höhdorf, C. Wang, and F. Holzapfel, *IATA Safety Report 2013, Section 9: Predictive Analysis*, pp. 81–87. Montreal, Geneva: International Air Transport Association, April, 2014. ISBN: 978-92-9252-349-7.
- [25] D. Bigoni, *Uncertainty quantification with applications to engineering problems*. PhD thesis, Technical University of Denmark, Copenhagen, Denmark, 2015.
- [26] S. Mahadevan and S. Sarkar, *Uncertainty Analysis Methods*. Vanderbilt University, 2009. Available online: https://www-pub.iaea.org/MTCD/Publications/PDF/TE-1701_add-CD/PDF/USA%20Attachment%2012.pdf (accessed on 05 September 2020).
- [27] H. Joe and D. Kurowicka, *Dependence modeling: Vine copula handbook*. World Scientific, 2011. ISBN: 978-9-814-29987-9, DOI: 10.1142/7699.
- [28] C. Czado, *Analyzing Dependent Data with Vine Copulas: A Practical Guide With R*. Springer International Publishing: USA, 2019. ISBN: 978-3-030-13784-7, DOI: 10.1007/978-3-030-13785-4.
- [29] K. W. Breitung, *Asymptotic approximations for probability integrals*. Springer, 2006.
- [30] R. Y. Rubinstein and D. P. Kroese, *Simulation and the Monte Carlo method*, vol. 10. John Wiley & Sons, 2016.
- [31] R. Srinivasan, *Importance sampling: Applications in communications and detection*. Springer Science & Business Media, 2013.
- [32] S.-K. Au and Y. Wang, *Engineering risk assessment with subset simulation*. John Wiley & Sons, 2014.
- [33] I. Papaioannou, W. Betz, K. Zwirgmaier, and D. Straub, “MCMC algorithms for subset simulation,” *Probabilistic Engineering Mechanics*, vol. 41, pp. 89–103, 2015.
- [34] L. Höhdorf, C. Wang, P. Koppitz, S. Weiß, S. Olmos, C. Czado, and F. Holzapfel, “Integration of vine copula dependence structures into subset simulation for accident probability quantifications,” in *ICAS 31st Congress of the International Council of the Aeronautical Science*, 2018.

- [35] T. Baumüller, *Development of Generic Calibration Algorithms for Contributing Factors of Physics-Based Incident Models*. Master thesis, Technische Universität München, Garching, March 21, 2016.
- [36] M. C. Kennedy and A. O’Hagan, “Bayesian calibration of computer models,” *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 63, no. 3, pp. 425–464, 2001.
- [37] E. Borgonovo *et al.*, *Sensitivity analysis: An Introduction for the Management Scientist*, vol. 251. Springer International Publishing AG, 2017.
- [38] A. Saltelli, M. Ratto, T. Andres, F. Campolongo, J. Cariboni, D. Gatelli, M. Saisana, and S. Tarantola, *Global sensitivity analysis: The primer*. John Wiley & Sons, 2008.
- [39] Q. Liu and T. Homma, “A new computational method of a moment-independent uncertainty importance measure,” *Reliability Engineering & System Safety*, vol. 94, no. 7, pp. 1205–1211, 2009.
- [40] S. K. Au, “Reliability-based design sensitivity by efficient simulation,” *Computers & structures*, vol. 83, no. 14, 2005. DOI: 10.1016/j.compstruc.2004.11.015.
- [41] D. Xiu, *Numerical methods for stochastic computations: A spectral method approach*. Princeton university press, 2010.
- [42] S. Marelli and B. Sudret, “UQLab user manual—polynomial chaos expansions - report UQLab-V1.3-104,” *Chair of Risk, Safety & Uncertainty Quantification, ETH Zürich*, 2019.
- [43] L. Drees and F. Holzapfel, “Determining and quantifying hazard chains and their contribution to incident probabilities in flight operation,” in *AIAA Modeling and Simulation Technologies Conference, Minneapolis, Minnesota, USA*, August 13-16, 2012. DOI: 10.2514/6.2012-4855.
- [44] L. Drees, C. Wang, and F. Holzapfel, “Using subset simulation to quantify stakeholder contribution to runway overrun,” in *PSAM12 Probabilistic Safety Assessment and Management*, 2014.
- [45] K. Zwirgmaier, L. Drees, F. Holzapfel, and D. Straub, “Reliability analysis for runway overrun using subset simulation,” in *Proceedings of the European Safety and Reliability Conference*, 2014.
- [46] C. Wang, *Model-Based Quantification of Accident Probabilities from Operational Flight Data*. PhD thesis, Technische Universität München, München, 2019.

- [47] T. Puranik, H. Jimenez, and D. Mavris, “Energy-based metrics for safety analysis of general aviation operations,” *Journal of Aircraft*, vol. 54, no. 6, pp. 2285–2297, 2017.
- [48] T. G. Puranik, *A methodology for quantitative data-driven safety assessment for general aviation*. PhD thesis, Georgia Institute of Technology, 2018.
- [49] Z. Lu, S. Zhang, R. Dai, and others., “Research on abnormal energy risk criteria of large civil airplanes in approach and landing (in Chinese),” *Acta Aeronautica et Astronautica Sinica*, vol. 40, 2020. DOI: 10.7527/S1000-6893.2020.24132.
- [50] X. Wang, X. Fang, L. Beller, and F. Holzapfel, “Calibration of contributing factors for model-based predictive analysis algorithm using polynomial chaos expansion methods,” in *Proceedings of the 30th European Safety and Reliability Conference and the 15th Probabilistic Safety Assessment and Management Conference*, 2020.
- [51] X. Wang, L. Beller, C. Czado, and F. Holzapfel, “Modeling of stochastic wind based on operational flight data using karhunen–loève expansion method,” *Sensors*, vol. 20, no. 16, p. 4634, 2020.
- [52] X. Wang, J. Sembiring, P. Koppitz, L. Höhndorf, C. Wang, and F. Holzapfel, “Modeling of the aircraft’s low energy state during the final approach phase using operational flight data,” in *AIAA Scitech 2019 Forum*, 2019.
- [53] D. R. Grossi, “Aviation recorder overview,” in *International Symposium On Transportation Recorders, Arlington, Virginia*, 1999.
- [54] J.R. Sturgeon, *Technical Prospects of the Use of Digital Flight Recorders for Technical Prospects of the Use of Digital Flight Recorders for Operational Research and Accident Prevention: RAE Technical Report 69201*, 1969.
- [55] J. Sear, *The ARL ‘black box’ flight recorder - invention and memory*. Bachelor thesis, University of Melbourne, 2001.
- [56] Boeing, “Flight Data Recorder Rule Change.” Available online: http://www.boeing.com/commercial/aeromagazine/aero_02/textonly/s01txt.html (accessed on October 12, 2014).
- [57] N. A. H. Campbell, “The evolution of flight data analysis,” in *ASASI 2007: Proceedings of Australian Society of Air Safety Investigators conference, Civic Square, Australia*, 2007.
- [58] European Organization for Civil Aviation Equipment (EUROCAE), *Minimum Operational Performance Specification for Chrash Protected Airborne Recorder Systems (ED-112)*. European Organization for Civil Aviation Equipment, Cologne, Germany, September 2013.

- [59] Federal Aviation Administration (FAA), *Flight data recorder systems (TSO-C124b)*. U.S. Department of Transportation, Federal Aviation Administration, Washington, D.C., April 2007.
- [60] Federal Aviation Administration (FAA), *Cockpit voice recorder equipment (TSO-C123c)*. U.S. Department of Transportation, Federal Aviation Administration, Washington, D.C., December 2013.
- [61] “Black box flight recorders.” Available online: <https://www.atsb.gov.au/publications/2014/black-box-flight-recorders/> (accessed on January 10, 2021).
- [62] P. Dubois, “Flight data analysis,” in *Lima FDA Seminar.*, September 2014.
- [63] Aeronautical Radio Incorporated., *Flight Data Acquisition and Recording System. (ARINC Characteristic 717-15)*. 2551 Riva Road, Annapolis, MA, USA., June 1, 2011.
- [64] Safran Electronics & Defense, “Flight data interface & management Unit: ED48,” 2017.
- [65] N. Mohr, *Flight Data Decoding used for Generating En-Route Information based on Binary Quick Access Recorder Data*. Master thesis, Technische Universität München, Garching, July 01, 2016.
- [66] Aeronautical Radio Incorporated., *Enhanced Airborne Flight Recorder. (ARINC Characteristic 767-1)*. 2551 Riva Road, Annapolis, MA, USA., May 29, 2009.
- [67] C. Wang, N. Mohr, and F. Holzzapfel, “Decoding of binary flight data in ARINC 717 format,” in *1st International Conference in Aerospace For Young Scientists (ICAYS)*, 2016.
- [68] M. Tooley, *Aircraft Digital Electronic And Computer Systems*. Oxford, UK: Elsevier, 1st ed., 2007.
- [69] Bureau d’Enquêtes et d’Analyses, *Flight Data Recorder Read-Out: Technical and Regulatory Aspects*. Bureau d’Enquêtes et d’Analyses pour la Sécurité de l’Aviation Civile, 2015.
- [70] “Airport Database.” Available online: <https://airportdatabase.net/> (accessed on Dec 17, 2020).
- [71] J. Siegel, *Reconstruction and Safety Assessment of Aircraft Landings Based on Operational Quick Access Recorder Data*. Master thesis, Technische Universität München, Garching, April 25, 2017.

-
- [72] P. Koppitz, J. Siegel, N. Romanow, L. Höhndorf, and F. Holzapfel, “Touchdown point detection for operational flight data using quality measures and a model based approach,” in *2018 AIAA Atmospheric Flight Mechanics Conference*, 2018. DOI: 10.2514/6.2018-1018.
- [73] F. Holzapfel, “Lecture Notes in Flight System Dynamics 2.” Technical University of Munich, February 2015.
- [74] J. Mickel, *Bewertung operationeller Flugrisiken*. Johann Wolfgang GoetheUniversität Frankfurt am Main, 2011.
- [75] R. Jategaonkar, *Flight vehicle system identification: a time domain methodology*. NewYork: AIAA Progress in Aeronautics and Astronautics, 2nd ed., 2015.
- [76] J. Siegel, *Quick Access Recorder Flight Data Analysis Using Extended Kalman Filter-Based Methods*. Bachelor thesis, Technische Universität München, Garching, 2014.
- [77] J. Sembiring, *A Heuristic Parameter Estimation Approach for FDM Data*. PhD thesis, Technische Universität München, München, 2020.
- [78] P. D. Groves, *Principles of GNSS, inertial, and multisensor integrated navigation systems*. The GNSS technology and applications series, Boston, Mass. and London: Artech House, 2008. ISBN: 978-1-58053-255-6.
- [79] H. E. Rauch, F. Tung, and C. T. Striebel, “Maximum likelihood estimates of linear dynamic systems,” *AIAA Journal*, vol. 3, no. 8, pp. 1445–1450, 1965. DOI: 10.2514/3.3166.
- [80] L. Höhndorf, J. Siegel, J. Sembiring, P. Koppitz, and F. Holzapfel, “Reconstruction of aircraft trajectories during landing using a rauch-tung-striebel smoother, instrument landing system deviation information, and taxiway locations,” in *AIAA Atmospheric Flight Mechanics Conference*, 2016. DOI: 10.2514/6.2016-3705.
- [81] N. Bähr, *Controlled Flight Into Terrain Analyses In Flight Data Monitoring*. Semester thesis, Technische Universität München, Garching, November 08, 2017.
- [82] L. Höhndorf, *Statistical Dependence Analyzes of Flight Data for Safety Management*. PhD thesis, Technische Universität München, München, 2019.
- [83] T. Jones, J. W. Rustenburg, D. A. Skinn, D. O. Tipps, and T. DeFiore, *Statistical Data for the Boeing 747-400 Aircraft in Commercial Operations*. Department of Transportation Federal Aviation Administration, 2005.
- [84] Avions de Transport Régional (ATR), *Flight data monitoring on ATR aircraft*. Avions de Transport Régional, 2016.

- [85] Bronshtein, H. Mühlig, G. Musiol, and K. A. Semendyayev, *Handbook of Mathematics*. Berlin Heidelberg: Springer Berlin Heidelberg and Imprint: Springer, 6th ed., 2015. ISBN: 978-3-662-46221-8.
- [86] A. Sklar, “Fonctions de repartition an dimensions et leursmarges,” in *Publications de l’Institut Statistique de l’Université de Paris*, vol. 8, pp. 229–231, 1959.
- [87] N. Thomas, “Statistical Inference of Vine Copulas: Package ‘VineCopula,’” 2019. Available online: <https://cran.r-project.org/web/packages/VineCopula/VineCopula.pdf> (accessed on 05 July 2020).
- [88] M. Rosenblatt, “Remarks on a multivariate transformation,” *The annals of mathematical statistics*, vol. 23, no. 3, pp. 470–472, 1952.
- [89] K. Aas, C. Czado, A. Frigessi, and H. Bakken, “Pair-copula constructions of multiple dependence,” *Insurance: Mathematics and economics*, vol. 44, no. 2, pp. 182–198, 2009. DOI: 10.1016/j.insmatheco.2007.02.001.
- [90] T. Bedford and R. M. Cooke, “Probability density decomposition for conditionally dependent random variables modeled by vines,” *Annals of Mathematics and Artificial Intelligence*, vol. 32, no. 1, pp. 245–268, 2001. DOI: 10.1023/A:1016725902970.
- [91] J. Dissmann, E. C. Brechmann, C. Czado, and D. Kurowicka, “Selecting and estimating regular vine copulae and application to financial returns,” *Computational Statistics & Data Analysis*, vol. 59, pp. 52–69, 2013. DOI: 10.1016/j.csda.2012.08.010.
- [92] N. Thomas, “High Performance Algorithms for Vine Copula Modeling,” 2019. Available online: <https://cran.r-project.org/web/packages/rvinecopulib/rvinecopulib.pdf> (accessed on 05 July 2020).
- [93] W. K. Hastings, “Monte Carlo sampling methods using Markov chains and their applications,” *Biometrika*, vol. 57, no. 1, pp. 97–109, 1970. DOI: 10.2307/2334940.
- [94] G. O. Roberts and J. S. Rosenthal, “Optimal scaling for various metropolis-hastings algorithms,” *Statistical Science*, vol. 16, no. 4, pp. 351–367, Nov., 2001.
- [95] G. O. Roberts, A. Gelman, W. R. Gilks, *et al.*, “Weak convergence and optimal scaling of random walk metropolis algorithms,” *The annals of applied probability*, vol. 7, no. 1, pp. 110–120, 1997.
- [96] A. Gelman, G. O. Roberts, W. R. Gilks, *et al.*, “Efficient metropolis jumping rules,” *Bayesian statistics*, vol. 5, no. 599-608, p. 42, 1996.
- [97] S.-K. Au and J. L. Beck, “Important sampling in high dimensions,” *Structural Safety*, vol. 25, no. 2, pp. 139–163, 2003. DOI: 10.1016/S0167-4730(02)00047-4.

- [98] L. Höhdorf, C. Wang, P. Koppitz, S. Weiß, S. Olmos, C. Czado, and F. Holzapfel, “Integration of vine copula dependence structures into subset simulation for accident probability quantifications,” in *31st Congress of the International Council of the Aeronautical Science*, 2018.
- [99] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, “Equation of state calculations by fast computing machines,” *The Journal of Chemical Physics*, vol. 21, no. 6, pp. 1087–1092, 1953. DOI: 10.1063/1.1699114.
- [100] S.-K. Au and J. L. Beck, “Estimation of small failure probabilities in high dimensions by subset simulation,” *Probabilistic Engineering Mechanics*, vol. 16, no. 4, pp. 263–277, 2001. DOI: 10.1016/S0266-8920(01)00019-4.
- [101] L. S. Katafygiotis and K. M. Zuev, “Estimation of small failure probabilities in high dimensions by adaptive linked importance sampling,” in *Sampling, COMPDYN 2007, Rethymno Crete Greece*, 2007.
- [102] S.-K. Au and E. Patelli, “Rare event simulation in finite-infinite dimensional space,” *Reliability Engineering & System Safety*, vol. 148, pp. 67–77, 2016. DOI: 10.1016/j.ress.2015.11.012.
- [103] P. Koppitz, C. Wang, L. Höhdorf, J. Sembiring, X. Wang, and F. Holzapfel, “From raw operational flight data to incident probabilities using subset simulation and a complex thrust model,” in *AIAA Scitech 2019 Forum*, p. 2233, 2019.
- [104] N. A. Weiss, P. T. Holmes, and M. Hardy, *A course in probability*. Boston: Pearson Addison Wesley, 2006. ISBN: 9780201774719.
- [105] A. Saltelli and I. M. Sobol’, “Sensitivity analysis for nonlinear mathematical models: Numerical experience,” *Matematicheskoe Modelirovanie*, vol. 7, no. 11, pp. 16–28, 1995.
- [106] A. Saltelli, P. Annoni, I. Azzini, F. Campolongo, M. Ratto, and S. Tarantola, “Variance based sensitivity analysis of model output. design and estimator for the total sensitivity index,” *Computer Physics Communications*, vol. 181, no. 2, pp. 259–270, 2010.
- [107] A. Janon, T. Klein, A. Lagnoux, M. Nodet, and C. Prieur, “Asymptotic normality and efficiency of two sobol index estimators,” *ESAIM: Probability and Statistics*, vol. 18, pp. 342–364, 2014.
- [108] T. Homma and A. Saltelli, “Importance measures in global sensitivity analysis of nonlinear models,” *Reliability Engineering & System Safety*, vol. 52, no. 1, pp. 1–17, 1996.

- [109] L. Li, I. Papaioannou, and D. Straub, “Global reliability sensitivity estimation based on failure samples,” *Structural Safety*, vol. 81, p. 101871, 2019.
- [110] S. Xiao, S. Oladyshkin, and W. Nowak, “Reliability sensitivity analysis with subset simulation: Application to a carbon dioxide storage problem,” in *IOP Conference Series: Materials Science and Engineering*, vol. 615, p. 012051, 2019. DOI: 10.1088/1757-899X/615/1/012051.
- [111] L. F. Shampine, “Vectorized adaptive quadrature in matlab,” *Journal of Computational and Applied Mathematics*, vol. 211, no. 2, pp. 131–140, 2008. DOI: 10.1016/j.cam.2006.11.021.
- [112] H.-J. Bungartz and M. Griebel, “Sparse grids,” *Acta numerica*, vol. 13, no. 1, pp. 147–269, 2004. DOI: 10.1017/S0962492904000182.
- [113] G. Blatman and B. Sudret, “Adaptive sparse polynomial chaos expansion based on least angle regression,” *Journal of Computational Physics*, vol. 230, no. 6, pp. 2345–2367, 2011.
- [114] B. Sudret, “Lecture notes: Polynomial chaos expansions,” in *Modeling and Numerical Methods for Uncertainty Quantification 2019, Porquerolles, France*, September, 2019.
- [115] B. Sudret, “Global sensitivity analysis using polynomial chaos expansions,” *Reliability Engineering & System Safety*, vol. 93, no. 7, pp. 964–979, 2008.
- [116] T. Baumüller, *Development of Generic Calibration Algorithms for Contributing Factors of Physics-Based Incident Models*. Master thesis, Technische Universität München, 2016.
- [117] P. E. Gill, W. Murray, and M. H. Wright, *Practical optimization*. London , United Kingdom: Academic Press, 1981.
- [118] F. M. Hoblit, *Gust loads on aircraft: Concepts and applications*. AIAA education series, American Institute of Aeronautics and Astronautics: Reston, VA, USA, 1988. ISBN: 978-0-930403-45-4, DOI:10.2514/4.861888.
- [119] W. Frost, D. W. Camp, and S. T. Wang, *Wind shear modeling for aircraft hazard definition*. Huntsville, AL, United States: National Aeronautics and Space Administration Marshall Space Flight Center, 1978.
- [120] F. B. Tatom, S. R. Smith, G. H. Fichtl, and C. W. Campbell, “Simulation of atmospheric turbulent gusts and gust gradients,” *Journal of Aircraft*, vol. 19, no. 4, pp. 264–271, 1982. DOI: 10.2514/3.57388.

-
- [121] European Aviation Safety Agency (EASA), *Easy Access Rules for All Weather Operations (CS-AWO)*. European Aviation Safety Agency, 2018.
- [122] B. Boashash, *Time frequency signal analysis and processing*. Oxford: Elsevier, 2003. ISBN: 978-0-08-044335-5.
- [123] H. G. Dehling, T. Gottschalk, and A. C. Hoffmann, *Stochastic modelling in process technology*. Elsevier Science, 2007. ISBN: 9780444520265.
- [124] M. Sri-Jayantha and R. F. Stengel, "Determination of nonlinear aerodynamic coefficients using the estimation-before-modeling method," *Journal of Aircraft*, vol. 25, no. 9, pp. 796–804, 1988. DOI: 10.2514/3.45662.
- [125] L. Höhndorf, J. Siegel, J. Sembiring, P. Koppitz, and F. Holzapfel, "Reconstruction of Aircraft States During Landing Based on Quick Access Recorder Data," *AIAA Journal of Guidance, Control, and Dynamics*, vol. 40, pp. 1–6, 2017. DOI: 10.2514/1.G002637.
- [126] G. E. P. Box, G. M. Jenkins, and G. C. Reinsel, *Time series analysis: forecasting and control*, vol. 734. John Wiley & Sons, 2011.
- [127] D. W. Ricker, *Echo signal processing*, vol. 725. USA: Springer, 2012. ISBN: 978-1-4615-0312-5, DOI: 10.1007/978-1-4615-0312-5.
- [128] D. Moorhouse and R. Woodcock, *U.S. military specification MIL-F-8785C*. U.S. Department of Defense, 1980.
- [129] F. Poirion and I. Zentner, "Stochastic model construction of observed random phenomena," *Probabilistic Engineering Mechanics*, vol. 36, pp. 63–71, 2014. DOI: 10.1016/j.pro bengmech.2014.03.005.
- [130] F. Poirion and M. Benoit, "Stochastic model construction of correlated nonlinear wave fields," in *Proceedings of the ASME 2015 34th International Conference on Ocean, Offshore and Arctic Engineering, St. John's, Newfoundland, Canada*, May 31–June 5, 2015. DOI: 10.1115/OMAE2015-41125.
- [131] E. W. Weisstein. "Eigenfunction." From MathWorld—A Wolfram Web Resource. <https://mathworld.wolfram.com/Eigenfunction.html> (accessed on November 25, 2021).
- [132] F. Poirion and I. Zentner, "Non-Gaussian non-stationary models for natural hazard modeling," *Applied Mathematical Modelling*, vol. 37, no. 8, pp. 5938–5950, 2013. DOI: 10.1016/j.apm.2012.11.021.
- [133] L. Wang, *Karhunen-Loeve expansions and their applications*. PhD thesis, London School of Economics and Political Science, London , United Kingdom, 2008.

- [134] O. Le Maître and O. M. Knio, *Spectral methods for uncertainty quantification: with applications to computational fluid dynamics*. Springer: Netherlands, 2010. DOI: 10.1007/978-90-481-3520-2.
- [135] J. Mercer, “Functions of positive and negative type, and their connection the theory of integral equations,” *Philosophical Transactions of the Royal Society A*, vol. 209, no. 441-458, pp. 415–446, 1909. DOI: 10.1098/rsta.1909.0016.
- [136] M. Grigoriu, “Evaluation of Karhunen–Loève, spectral, and sampling representations for stochastic processes,” *Journal of engineering mechanics*, vol. 132, no. 2, pp. 179–189, 2006. DOI: 10.1061/(ASCE)0733-9399(2006)132:2(179).
- [137] E. J. Chen, L. Ding, Y. Liu, X. Ma, and M. J. Skibniewski, “On spectral representation method and Karhunen–Loève expansion in modeling construction material properties,” *Archives of Civil and Mechanical Engineering*, vol. 18, no. 3, pp. 768–783, 2018. DOI: 10.1016/j.acme.2017.12.008.
- [138] T. Cacoullos, “Estimation of a multivariate density,” *Annals of the Institute of Statistical Mathematics*, vol. 18, p. 179–189, 1966. DOI: 10.1007/BF02869528.
- [139] C. Belcastro and J. Foster, “Aircraft Loss-of-Control Accident Analysis,” in *AIAA Guidance, Navigation, and Control Conference*, American Institute of Aeronautics and Astronautics, 2010. DOI: 10.2514/6.2010-8004.
- [140] N. Gandhi, N. Richards, A. Bateman, C. Bolstad, and A. Costello, “Pilot-in-the-loop demonstration of an energy monitor and crew alerting system,” in *AIAA Guidance, Navigation, and Control Conference*, p. 4819, 2012. DOI: 10.2514/6.2012-4819.
- [141] M. U. De Haag and P. Duan, “A multiple hypothesis predictive alerting (MHPA) method for improved aircraft state awareness,” in *Digital Avionics Systems Conference (DASC)*, pp. 6A4–1, IEEE, 2015.
- [142] K. H. Shish, J. Kaneshige, D. M. Acosta, S. Schuet, T. Lombaerts, L. Martin, and A. N. Madavan, “Trajectory prediction and alerting for aircraft mode and energy state awareness,” in *AIAA Infotech @ Aerospace*, pp. 1113–1132, American Institute of Aeronautics and Astronautics, 2015.
- [143] T. Puranik, H. Jimenez, and D. Mavris, “Energy-based metrics for safety analysis of general aviation operations,” *Journal of Aircraft*, vol. 54, no. 6, pp. 2285–2297, 2017.
- [144] Commercial Aviation Safety Team (CAST), *Airplane State Awareness Joint Safety Implementation Team Final Report*. Commercial Aviation Safety Team, Washington, DC, 2014.

- [145] European Operators Flight Data Monitoring (EOFDM), *Review of Accident Precursors for Loss of Control In-flight*. European Aviation Safety Agency, Cologne, Germany, 2017.
- [146] Flight Safety Foundation (FSF), *FSF ALAR Briefing Note 4.2: Energy Management*. Flight Safety Foundation, Alexandria, USA, 2000.
- [147] J. R. Merkt, “Flight energy management training: Promoting safety and efficiency,” *Journal of Aviation Technology and Engineering*, vol. 3, no. 1, pp. 24–36, 2013.
- [148] T. Blajev and W. Curtis, *Go-Around Decision-Making and Execution Project*. Flight Safety Foundation, Alexandria, USA, 2017.
- [149] Flight Safety Foundation (FSF), *FSF ALAR Briefing Note 7.1: Stabilized Approach*. Flight Safety Foundation, Alexandria, USA, 2009.
- [150] D. DE-Winter, C. Hildebrandt, and V. Sibelle, “Prevention of unstable approaches,” *Airbus Safety First*, October, 2020.
- [151] Cargolux Airlines International S.A., *747 Operations Manual Part B*. Sandweiler, Luxembourg, October 2019.
- [152] Flight Safety Foundation (FSF), *FSF ALAR Briefing Note 8.2: The Final Approach Speed*. Flight Safety Foundation, Alexandria, 2000.
- [153] Deutsche Lufthansa AG, *Operations Manual Part A - General: Flight Operations Procedures*. Frankfurt, Germany, August 2019.
- [154] Airbus Industrie, *Flight Operations Support - getting to grips with Approach-and-Landing Accidents Reduction: A Flight Operations View*. Airbus Training and Flight Operations Support and Line Assistance, Blagnac Cedex, France, October 2000.
- [155] International Civil Aviation Organization (ICAO), *Annex 10 to the Convention on International Civil Aviation*. International Civil Aviation Organization, Montréal, Québec, Canada, 6th ed., 2006.
- [156] J. Sembiring, C. Liu, P. Koppitz, and F. Holzapfel, “Energy management for unstable approach detection,” in *2018 IEEE International Conference on Aerospace Electronics and Remote Sensing Technology (ICARES)*, pp. 1–6, 2018.
- [157] D. Walter, *Development of an Unstable Approach Detection and Prediction Model*. Semester thesis, Technische Universität München, Garching, September 2020.
- [158] F. H. Proctor, D. A. Hinton, and R. L. Bowles, “A windshear hazard index,” in *9th Conference on Aviation, Range and Aerospace Meteorology*, pp. 482–487, 11-15 September, 2000.

- [159] D. Hinton, “Flight test evaluation of a data link and aircraft integration of TDWR wind-shear information. airborne windshear detection and warning systems,” in *Fifth and Final Combined Manufacturers’ and Technologists’ Conference, NASA CP-10139, DOT/FAA/RD-94/14-1*, pp. 23 – 48, 1994.
- [160] A. Lambregts, “Integrated system design for flight and propulsion control using total energy principles,” in *Aircraft Design, Systems and Technology Meeting, Reston, Virginia*, American Institute of Aeronautics and Astronautics, October 17, 1983. DOI: 10.2514/6.1983-2561.
- [161] F. Holzapfel, “Lecture Notes in Flight Control 2.” Technical University of Munich, February 2015.
- [162] International Civil Aviation Organization (ICAO), *Doc9817 Manual on Low-level Wind Shear*. International Civil Aviation Organization, Montréal, Québec, Canada, 1st ed., 2005.
- [163] D. McRuer and E. Krendel, *Mathematical Models of Human Pilot Behavior*, 1974.
- [164] D. McRuer and H. R. Jex, “A review of quasi-linear pilot models,” *IEEE Transactions on Human Factors in Electronics*, vol. HFE-8, no. 3, pp. 231–249, 1967. DOI: 10.1109/THFE.1967.234304.
- [165] D.L. Kleinman, S. Baron, and W.H. Levison, “An optimal control model of human response part i: Theory and validation,” *Automatica*, vol. 6, no. 3, pp. 357–369, 1970. DOI: 10.1016/0005-1098(70)90051-8.
- [166] S. Baron, D. L. Kleinman, and W. H. Levison, “An optimal control model of human response part ii: Prediction of human performance in a complex task,” *Automatica*, vol. 6, no. 3, pp. 371–383, 1970. DOI: 10.1016/0005-1098(70)90052-X.
- [167] F. M. S. Turetta, H. V. H. Ayala, L. G. Trabasso, L. S. Coelho, and J. Alfredson, “Data-driven pilot behavior modeling applied to an aircraft offset landing task,” in *Advances in Human Aspects of Transportation* (N. A. Stanton, ed.), vol. 597 of *Advances in Intelligent Systems and Computing*, (Cham), pp. 117–127, Springer International Publishing, 2018. DOI: 10.1007/978-3-319-60441-1_12.